# Deploying to webMethods Integration Cloud

Version 5.6.0

July 2019

WEBMETHODS

**Document ID: IL-IUG-560-20190731**

# Table of Contents

# Deploying to webMethods Integration Cloud Documentation

This document explains how to deploy user-created packages and configuration assets that reside within on-premises runtimes or repositories to webMethods Integration Cloud.

## Document Conventions

| Convention | Description |
|---|---|
| **Bold** | Identifies elements on a screen. |
| Narrowfont | Identifies service names and locations in the format *folder.subfolder.service* , APIs, Java classes, methods, properties. |
| *Italic* | Identifies:<br>Variables for which you must supply values specific to your own situation or environment.<br>New terms the first time they occur in the text.<br>References to other documentation sources. |
| `Monospace font` | Identifies:<br>Text you must type in.<br>Messages displayed by the system.<br>Program code.<br>Service names and locations in the format *folder.subfolder.service* APIs, Java classes, methods, properties. |
| { } | Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols. |
| \| | Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the \| symbol. |
| [ ] | Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols. |

| Convention | Description |
| --- | --- |
| ... | Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...). |

# Online Information and Support

### Software AG Documentation Website

You can find documentation on the Software AG Documentation website at "http://documentation.softwareag.com". The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

### Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to "empower@softwareag.com" with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at "https://empower.softwareag.com/".

You can find product information on the Software AG Empower Product Support website at "https://empower.softwareag.com".

To submit feature/enhancement requests, get information about product availability, and download products, go to "Products".

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the "Knowledge Center".

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at "https://empower.softwareag.com/public_directory.asp" and give us a call.

### Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at "http://techcommunity.softwareag.com". You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

# Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

# 1 Deploying to Integration Cloud

# Deploying to Integration Cloud

Cloud deployment is the process of deploying user-created packages and configurations that reside within on-premise runtimes or repositories to webMethods Integration Cloud. Using Software AG Designer you can seamlessly deploy your on-premise Integration Server packages and configuration assets to solutions present on Integration Cloud.

**Note:** In the cloud deployment context, configuration assets are limited to Integration Server and Universal Messaging configurations.

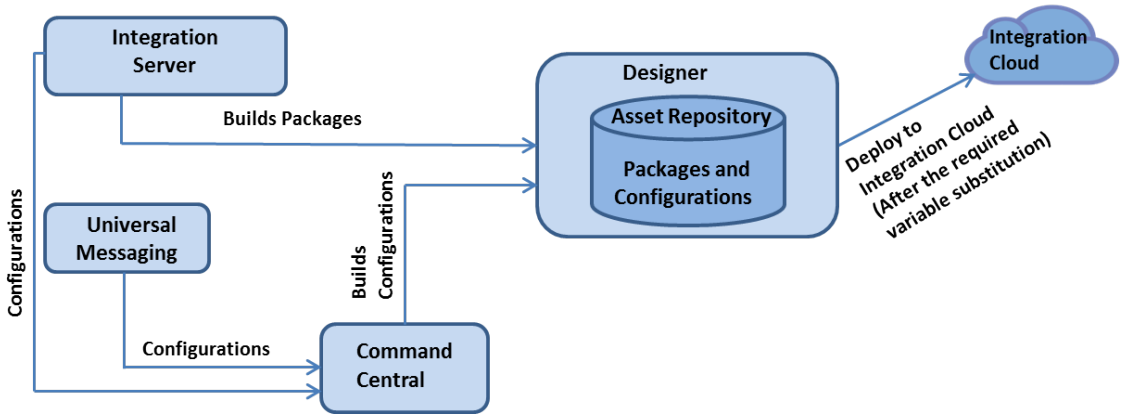
Software AG Designer allows you to deploy the Integration Server packages or configuration assets that you have created, verified, and tested on on-premise Integration Server or Universal Messaging to Integration Cloud. When you initiate the deployment from Designer, Integration Server packages and configuration assets are built from Integration Server and Command Central respectively, and are published to an asset repository present in Designer.

After performing variable substitutions to make the on-premise configuration data compatible for cloud deployment, you can publish the packages and configurations from the asset repository to a remote repository provisioned for the tenant on Integration Cloud.

Platform Manager plug-ins for Integration Server and Universal Messaging continuously monitor the remote repository on Integration Cloud. When a fresh deployment is detected, Platform Manager plug-ins deploy the Integration Server packages and configuration assets to respective runtime instances on Integration Cloud.

**Note:** Deploying assets to Integration Cloud using Designer eliminates the previous need to install or invoke webMethods Deployer on cloud for the purpose of cloud deployment.

The following figure provides a high-level basic overview of the process involved in deploying on-premise Integration Server packages and configuration assets to Integration Cloud.


Integration Server
Builds Packages
Designer
Asset Repository
Packages and Configurations
Integration Cloud
Deploy to Integration Cloud (After the required variable substitution)
Configurations
Universal Messaging
Builds Configurations
Configurations
Command Central

# 2 Get started with Cloud Deployment

# Get started with Cloud Deployment

The following table briefly describes the high-level tasks to get started with Cloud Deployment.

| Tasks | References |
| --- | --- |
| What is Cloud Deployment? | See the *Deploying to Integration Cloud* section.<br>**Note:** See this "video" for information on how to perform Cloud Deployment. |
| License and installation information. | ■ Install the relevant products as mentioned in the "Installation Information" on page 20 section.<br>■ Create a tenant in webMethods Integration Cloud.<br>■ Download the latest version of Software AG Designer to use the new features available for Cloud Deployment. |
| Assets and configurations that can be deployed to Integration Cloud. | See "Deployable Assets and Configurations" on page 22. |
| Create a solution in Integration Cloud. | See the *Cloud Deployment* section in this document for information on how to create a solution. |
| Connect to Integration Cloud. | See "Connecting to Integration Cloud for Cloud Deployment" on page 30. |
| View and publish assets and configurations to the solution created in Integration Cloud using any of the Cloud Deployment approaches. | See "Cloud Deployment Approaches" on page 28. |
| View and test the published assets and configurations in Integration Cloud. | See the *Cloud Deployment* section in this document. |

| Tasks | References |
| --- | --- |
| Promote the assets and configurations to different stages. | See the *Cloud Deployment* section in this document. |
| View and run services in Integration Cloud. | See the *Cloud Deployment* section in this document. |
| View the solution landscape, configure webMethods Integration Server service access settings, administer, and restart webMethods Integration Server. | See the *Cloud Deployment* section in this document. |
| Monitor the health and availability of the deployed solutions and run-time instances, alerts, and alert statuses. | See the *Cloud Deployment* section in this document. |
| Best Practices and General Information. | See "Best Practices, General Information, and Dependencies" on page 66. |

# 3 Installation information

# Installation Information

Install Software AG Command Central using the Command Central bootstrapper or provision a Command Central container from the Docker Store. For instructions, see the Command Central documentation.

Install the following products into a different directory or on a different machine, using the Software AG Installer or Command Central:

- webMethods Asset Build Environment
- Service Development in Software AG Designer with Local Version Control System Integration feature and webMethods Unit Test Framework.
- webMethods Integration Server with webMethods Unit Test Framework.
- Software AG Universal Messaging. All child nodes except Enablement for Cloud Foundry and webMethods Broker to UM Migration Utility.

If you want to use an external RDBMS for webMethods Integration Server, install webMethods Database Component Configurator on any machine on the same network as your database server, create the database components for webMethods Integration Server, and then connect webMethods Integration Server to the database components.

For detailed instructions, see the following documents:

- *Installing Software AG Products* document.
- If you are using Software AG Installer to install, see the *Using Software AG Installer* document.
- If you are using Command Central to install, see the *Software AG Command Central Help*.

# 4 Deployable assets and configurations

# Deployable Assets and Configurations

The following table provides information on the deployable assets and configurations for webMethods Integration Server.

| Configuration/Asset Type | Asset/Configuration |
|---|---|
| ACLs | Configuration |
| Groups | Configuration |
| Consumer Web service endpoint alias | Configuration |
| Extended Settings * (See "Best Practices, General Information, and Dependencies" on page 66) | Configuration |
| Global Variables | Configuration |
| JMS aliases | Configuration |
| JNDI aliases | Configuration |
| Keystores | Configuration |
| Provider Web service endpoint alias | Configuration |
| Resources Settings * | Configuration |
| SMTP settings * | Configuration |
| Truststores | Configuration |
| Users | Configuration |
| webMethods messaging | Configuration |
| Flow Service | Asset |

| Configuration/Asset Type | Asset/Configuration |
|---|---|
| Java Service | Asset |
| Map Service | Asset |
| OData Service | Asset |
| Adapter Service | Asset |
| .NET Service | Asset |
| C Service | Asset |
| Adapter Notification | Asset |
| Document Type | Asset |
| Flat File Dictionary | Asset |
| Flat File Schema | Asset |
| JMS Trigger | Asset |
| REST API Descriptor | Asset |
| REST Resource | Asset |
| Schema | Asset |
| Specification | Asset |
| Web Service Descriptor | Asset |
| webMethods Messaging Trigger | Asset |
| XML Document Type | Asset |
| XSLT Service | Asset |

*** - Restart Required**

The following table provides information on the deployable assets and configurations for webMethods CloudStreams.

| Configuration/Asset Type | Asset/Configuration |
|---|---|
| CloudStreams Connector Services | Asset |
| CloudStreams Connection | Asset |
| CloudStreams Streaming Listener Service | Asset |
| CloudStreams OAuth Tokens* | Configuration |
| CloudStreams Database* | Configuration |
| CloudStreams Large Data* | Configuration |
| CloudStreams Streaming Subscribers* | Configuration |
| CloudStreams Streaming Providers* | Configuration |

*** - Restart Required**

The following table provides information on the deployable configurations for Software AG Universal Messaging.

| Configuration Type | Asset/Configuration |
|---|---|
| Channels | Transient |
| | Simple |
| | Reliable |
| | Persistent |
| | Mixed |
| | Off-Heap |
| | Paged |

| Configuration Type | Asset/Configuration |
|---|---|
| Cluster | UM Cluster |
| JNDI Connection Factories | Connection Factory>Shared Durable Type |
| | Connection Factory> Non Shared Durable Type |
| | TopicConnection Factory>Shared Durable Type |
| | TopicConnection Factory>Non Shared Durable Type |
| | Queue Connection Factory |
| | XA Connection Factory |
| JNDI Destinations | Topic |
| | Queue |
| Memory * | JAVA_MIN_MEM JAVA_MAX_MEM |
| Queues | Transient |
| | Simple |
| | Reliable |
| | Persistent |
| | Mixed |
| | Off-Heap |
| | Paged |
| Users | All Custom Users |
| Zones | UM Zone |

*** - Restart Required**

# 5 Cloud Deployment approaches

# Cloud Deployment Approaches

Choose the following approaches to publish user-created packages and configurations to a solution created in webMethods Integration Cloud. See the *Cloud Deployment* section in the *webMethods Integration Cloud Help* for information on how to create a solution in webMethods Integration Cloud.

### Using Software AG Designer

- "Package Navigator" on page 34: Use the **Package Navigator** view in Software AG Designer to deploy webMethods Integration Server packages and configuration assets to webMethods Integration Cloud.
- "Asset Repository" on page 40: Use the **Asset Repository** view in the Service Development perspective of Software AG Designer to deploy webMethods Integration Server packages and configuration assets to webMethods Integration Cloud.
- "Landscape Navigator" on page 46: Use the **Landscape Navigator** view in the Service Development perspective of Software AG Designer to deploy customized configuration templates to webMethods Integration Cloud. "Export the configurations" on page 52 into a Software AG Designer configuration project and then build and deploy this project to webMethods Integration Cloud. For continuous integration purposes, these projects can be stored in a Version Control System (VCS) location, built through webMethods Asset Build Environment, and then deployed to webMethods Integration Cloud.

### Using Continuous Integration and Continuous Deployment (CI/CD)

- "Continuous Integration and Continuous Deployment" on page 58

# 6 Connecting to Integration Cloud for Cloud Deployment

# Connecting to Integration Cloud for Cloud Deployment

Using Designer you can deploy your on-premises Integration Server packages and configuration assets to Integration Cloud. Before deploying the packages and configuration assets, you must configure a connection to Integration Cloud.

Before you configure a connection to Integration Cloud, ensure that the following criteria are met:

- A valid URL exists to connect to Integration Cloud.
- A valid user account is created on Integration Cloud.

## Configuring a Connection to Integration Cloud

You need to provide specific information in Designer for initiating a connection to Integration Cloud. Designer saves this information in a connection configuration. You can add, edit, and update connection configurations in Designer.

### Adding a Connection Configuration for Integration Cloud

Using Designer you can add connections to different instances of Integration Cloud.

**To add a connection configuration for Integration Cloud**

1. In Designer, select **Window > Preferences**.
2. In the preferences navigation tree, select **Software AG > Integration Cloud**.
3. Click **Add**.
4. In the **Add connection configuration** dialog box, enter the following information:

| Field | Description |
|---|---|
| **Name** | The name to use for the Integration Cloud connection configuration.<br>**Note:** The name cannot contain control characters, special characters, and characters outside of the basic ASCII character set, such as multi-byte characters. |
| **URL** | URL of the Integration Cloud host to which Designer is to connect. For example, `https://<sub-domain>.<domain-name>.<domain-suffix>`, for example, https://mysubdomain.webmethodscloud.com. |

| Field | Description |
| --- | --- |
| **User** | The user name for an account on Integration Cloud. |
| **Password** | The password for the specified **User**. |
| **Save password (in the Eclipse secure storage)** | Indicates whether the password for the specified user account should be saved in Eclipse secure storage. Integration Cloud uses this password from the Eclipse secure storage whenever user authorization is required. If you want to save the password in Eclipse secure storage, select this check box.<br><br>If you decide not to save the password in Eclipse secure storage, you must specify your password each time your user authorization is required for connecting to Integration Cloud. |

5. To verify whether Integration Cloud can be accessed by using the specified information, click **Test**.
6. To store the connection configuration details, click **OK**.

   A connection configuration is added to the Connections page with the specified details. The first connection configuration that you create is automatically marked as default. This default configuration is indicated with a check mark on the Connections page. Designer always uses the default connection configuration for Integration Cloud. If there are multiple connections configured, you can select the connection that you want to make it as default.

## Editing a Connection Configuration for Integration Cloud

You can edit a connection configuration for Integration Cloud if there are any changes in the configuration values.

**To edit a connection configuration for Integration Cloud**

1. In Designer, select **Window > Preferences**.
2. In the preferences navigation tree, select **Software AG > Integration Cloud**.
3. Click **Edit**.
4. Enter new values in the connection configuration fields you want to change.
5. In the **Edit Connection Configuration** dialog box, click **OK**.
6. In the **Connections** page, click **OK**.

## Removing a Connection Configuration for Integration Cloud

You can remove connection configurations for Integration Cloud one at a time from the **Connections** page.

**To remove a connection configuration for Integration Cloud**

1. In Designer, select **Window > Preferences**
2. In the preferences navigation tree, select **Software AG > Integration Cloud**.
3. Select the Integration Cloud connection configuration you want to remove.
4. Click **Remove**.

   Designer prompts you to confirm that you want to remove the selected connection configuration.
5. Click **OK**.

# 7 Deploying packages and configuration assets to Integration Cloud

# Deploying Packages and Configuration Assets to Integration Cloud

Using Designer you can deploy Integration Server packages and configuration assets to Integration Cloud.

Before deploying packages and configuration assets to Integration Cloud, ensure the following:

- Designer is configured to connect to Integration Cloud using the Integration Cloud preference page. For more information on configuring a connection to Integration Cloud, see Connecting to Integration Cloud for Cloud Deployment chapter.
- Designer is configured to connect to Command Central. Use the Landscape Navigator view to connect to Command Central. For more information on Landscape Navigator view and connecting to Command Central, see "Using Landscape Navigator to View Runtime Configurations" on page 46.
- Command Central is connected to Integration Server and Universal Messaging instances.

**Note:** If you plan to deploy only Integration Server packages and not the Universal Messaging and Integration Server configurations, you do not need to connect to Command Central.

### To deploy assets to Integration Cloud

**Note:** The following procedure explains how to deploy packages and configuration assets only to Integration Cloud. If you want to deploy, refresh,or remove the packages and configuration assets from Integration Cloud, use the Asset Repository view in Designer.

1. In Package Navigator view, select the Integration Server package that you want to deploy to Integration Cloud.
2. Right-click the package and click **Deploy to Cloud**.
3. In the Publish Assets to Cloud dialog box, select the package to deploy. You can select multiple packages.

   If you want to publish the configuration assets along with the selected Integration Server package, select **Include runtime configurations**, click **Next**, and go to step 4. Otherwise, click **Next** and go to step 5.

   Designer fetches the configuration data from Command Central.
4. In the Confirm selected configurations dialog box, select the configuration that you want to deploy and click **Next**.

5. In the Integration Server package variable substitution dialog box, select the package and edit the value of the property that you want to modify before deploying to cloud. Click **Next**.
6. In the Select Cloud Solution dialog box, select the solution on Integration Cloud to which you want to deploy the assets and click **Finish**.

   Designer deploys the assets and configurations to a remote repository provisioned for the tenant on Integration Cloud.

## Checking the Deployment Status

After you deploy the assets to the Integration Cloud, you can check the deployment status.

**To check the deployment status**

1. In the Service Development perspective of Designer, select **Window > Show View > Other**.
2. In the Show View dialog box, select **Software AG Service Development >Deployment Status**.
3. Click **Open**.

   Designer displays the **Deployment Status** view with the current deployment status of the assets.

   **Note:**
   - To reload the deployment status, click .
   - To remove a specific deployment status, select the asset under the `Asset Name` and click .

# 8 Deploying Adapter for JDBC assets

# Deploying Adapter for JDBC Assets

To deploy Adapter for JDBC assets such as driver jars and TypeMapping configurations to Integration Cloud, perform the following steps:

1. Create the below folder structure inside any package which needs to be deployed apart from the core product package:
   - *<package_Name>*\adapters\WmJDBCAdapter\config
2. Copy the database specific driver jar files from on-premises installation, `<webM_Home>`\Integration Server`\instances\<instance_Name>\packages\WmJDBCAdapter\code\jars` to `<package_Name>\code\jars\static`. Also, copy the Adapter for JDBC configurations (TypeMapping.xml files) from `<webM_Home>`\Integration Server`\instances\<instance_Name>\packages\WmJDBCAdapter\config` to `<package_Name>\adapters\WmJDBCAdapter\config` folder.
3. To enable the assets (Adapter Connections, Polling Notifications) on cloud, ensure to enable the **State After Deployment** option during deployment from Software AG Designer. The state of the assets (Adapter Connections, Polling Notifications) should be disabled on-premises before deploying to Integration Cloud. To deploy Adapter for JDBC to Integration Cloud, see "Deploying Packages and Configuration Assets to Integration Cloud" on page 34.

**Note:**
- To connect to an on-premises database using VPN, you must variable substitute the serverName to IP address of the server as the hostname of the server cannot be resolved.
- Segregate the driver jars and solutions such as Adapter for JDBC's services, notifications into separate test packages.
- Set the proper package dependency between the jar package and the Adapter for JDBC's solutions package.

### Limitations

- You can use only Type 4 JDBC drivers for Integration Cloud deployment.
- Kerberos authentication is not supported.
- Data type configuration file editing is not supported.

# 9 Asset Repository view

# Asset Repository

Asset Repository is a binary repository that provides components for managing user assets . Asset Repository enables the process of repository-based deployment, in which runtime servers pull assets from a repository. You can build Integration Server packages and configuration assets and add to the repository making the assets available for deployment to Integration Cloud. Asset Repository is installed along with Designer using the Software AG Installer.

The Asset Repository view in Designer allows you to build, deploy, refresh, remove, or retract the assets.

## Opening the Asset Repository View

**To open the Asset Repository view**

1. In the Service Development perspective of Designer, select **Window > Show View > Other**.
2. In the **Show View** dialog box, select **Software AG Service Development > Asset Repository**.
3. Click **OK**.

## Building Packages and Assets in Asset Repository View

If the Asset Repository view does not already display the list of Integration Server packages and configuration assets, you can build them.

**To build packages and configuration assets**

1. In Asset Repository view, click the View Menu button ▽.
2. Select **Build**.

   The Asset Repository view displays all the newly created Integration Server packages and configuration assets.

## Deploying Packages and Assets from Asset Repository View

You can deploy Integration Server package or configuration assets using the Asset Repository view.

**To deploy packages and configuration assets from Asset Repository View**

1. In Asset Repository view, select the Integration Server package or configuration assets that you want to deploy to Integration Cloud.
2. Right-click the package and click **Deploy to Cloud**.

3. In the **Publish Assets to Cloud** dialog box, select the package to deploy.

   If you want to publish the configuration assets along with the selected Integration Server package, select **Include runtime configurations**, click **Next**, and go to step 4. Otherwise, click **Next** and go to step 5.

4. In the **Confirm selected configurations** dialog box, select the configuration that you want to deploy and click **Next**.

   Designer fetches the configuration data from Software AG Command Central.

5. In the **Integration Server package variable substitution** dialog box, select the package and edit the value of the property that you want modify before deploying to cloud. Click **Next**.

6. In the **Select Cloud Solution** dialog box, select the solution on Integration Cloud to which you want to deploy the assets and click **Finish**.

   Designer deploys the assets and configurations to a remote repository provisioned for the tenant on Integration Cloud.

## Retracting Packages and Assets

Using Designer you can retract Integration Server packages, configuration assets, or both from Integration Cloud.

**To retract packages and configuration assets**

1. In Asset Repository view, select the Integration Server package or configuration assets that you want to retract.

2. Right-click and select **Retract**.

   Designer retracts the packages and configuration assets fromIntegration Cloud.

## Deleting Packages and Assets

You can delete Integration Server packages or configuration assets from the Asset Repository view and from Integration Cloud. You can do this to remove packages and configuration assets with incorrect variable substitution or remove old versions of them from Integration Cloud.

**To delete packages and configuration assets**

1. In Asset Repository view, select the Integration Server package or configuration assets that you want to delete.

2. Right-click and select **Remove**.

   Designer removes the Integration Server packages or configuration assets from Asset Repository view and Integration Cloud.

# 10 Deploying Command Central assets to Integration Cloud

# About Deploying Command Central Assets to Integration Cloud

Command Central can export configuration properties for Software AG run-time components as YAML templates. Configuration properties are exported as Command Central composite assets using Software AG Designer, stored in the Landscape Asset Repository, and deployed to product instances, running on webMethods Integration Cloud.

To create Command Central assets and deploy them to Integration Cloud, in Designer:

1. Connect to Command Central.
2. Select and export the configurations to be deployed.
3. Edit the YAML configurations.
4. Deploy the assets to Integration Cloud.

# 11 Using Landscape Navigator View to view runtime configurations

# Using Landscape Navigator to View Runtime Configurations

Using Landscape Navigator view you can connect to Command Central and view list of registered installations and runtime instances. For more information about connecting to Command Central see the *Creating Command Central Definitions* section.

## What Does the Landscape Navigator View Contain?

You can use Landscape Navigator view and Configuration Browser views to display the list of installations that are registered in Command Central. The available run-time instances are listed in the view for each installation and these available run-time instances are available for deploy to Integration Cloud.

Landscape Navigator view provides an option to connect to a local or remote Command Central server. To view the contents of installations that are registered in Command Central, you must configure a connection to Command Central and then establish a connection.

You can find Landscape Navigator and Configuration Browser views under the **Windows > Show View > Other > Software AG Command Central**.

## Creating Command Central Definitions

Command Central definitions create the connection between Designer and Command Central. The definition for a server contains the connection information needed to establish a connection with the Command Central server.

**To create a Command Central definition**

1. On the Landscape Navigator toolbar, click .
2. In the New Command Central Connection dialog box, enter the details for connecting to the Command Central server as described in following table:

| Field | Description |
|---|---|
| **Connection name** | The connection name. |
| **Command Central Location** | Choose one of the following options to specify the location:<br>■ **URL:** The URL for Command Central connection. For example,<br>https://<Command_Central_host>:<Command_Central_port>/cce. |

| Field | Description |
| --- | --- |
| | ■ **Installation Directory:** The local Software AG product installation directory where Command Central is installed. |

3. Under **Authentication**, enter the details as described in the following table:

| Field | Description |
| --- | --- |
| **Username** | Username to use for authentication on the Command Central. |
| **Password** | The password to use for authentication on the Command Central. |
| **Connection timeout (s):** | The maximum seconds Designer waits for response from Command Central. |

4. Click **Save** password.

   Designer stores the password in Eclipse secure storage location.

5. Click **Test Connection** to ping the Command Central server and to verify that the connection is working.

6. Click **Finish**.

   Designer refreshes the Landscape Navigator view and displays the new connection.

## Editing Command Central Definition

If there are changes in the associated Command Central server that require you to update the server definition, edit the Command Central connection properties. For example, to add or edit the username and password, you must edit the Command Central Connection page to reflect that change.

**To edit a Command Central definition**

1. Open the Landscape Navigator view.
2. Select the connection you want to edit.
3. Right-click the connection and select **Edit Command Central Connection**.
4. Edit the connection properties as needed.

## Deleting a Command Central Definition

**To delete a Command Central definition**

1. Open the Landscape Navigator view.
2. Select the connection that you want to delete.
3. Right-click the connection and select **Delete Command Central Connection**.
4. Click **Yes** to confirm the deletion from the view.

# What Does the Configuration Browser View Contain?

Designer offers a detailed view of run-time specific configuration in Configuration Browser view. Designer retrieves information about a specific run-time component selected in Landscape Navigator.

The following table describes the runtime component information that is displayed in the Configuration Browser view.

| Name | Description |
|---|---|
| Name | Display name of the configuration instance. |
| Id | Id of the runtime instance. |
| Product | Product to which the runtime instance belongs. |
| Category | Configuration category to which the runtime instance belongs. |
| Deployable | Whether or not the asset is deployable to Integration Cloud. |
| Description | Description of the configuration instance. |

# Browsing Runtime Instances and Configurations

After you connect to Command Central, you can use the Landscape Navigator view to browse the registered installations, for available runtimes and corresponding configurations.

When you expand an installation node, you can view the list of the runtime instances in that installation. You can also view the name and the runtime status as ONLINE, UNKNOWN, FAILED, OFFLINE, and so on. Upon expanding the runtime instances,

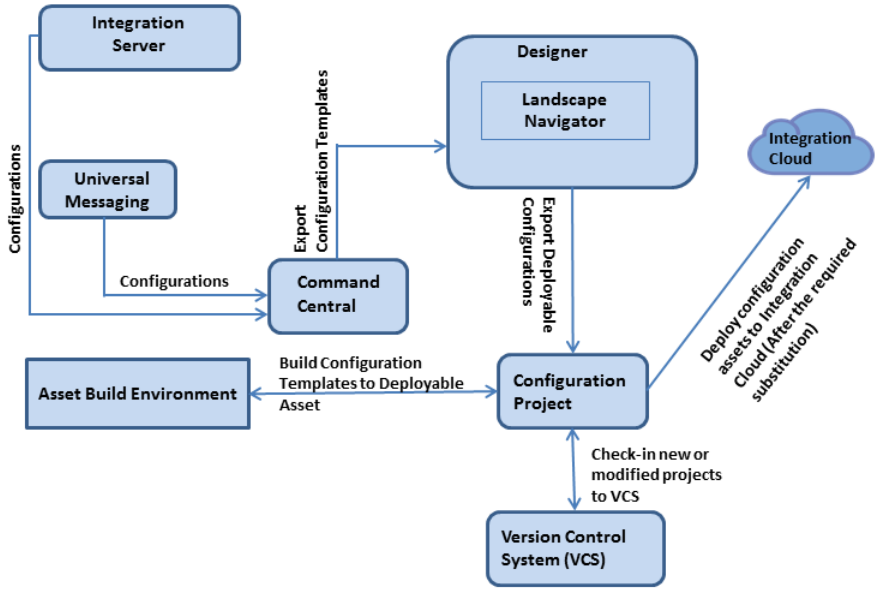
the list of configurations are displayed. You can view the configuration data for each configuration instance in Properties view, under Data tab.

# 12 Deploying customized configurations to Integration Cloud

# Deploying Customized Configuration to Integration Cloud

Using Designer you can deploy customized configuration templates to Integration Cloud. You can use Landscape Navigator view to export the configuration into a Designer configuration project. Designer can be used to build and deploy this project to Integration Cloud. For continuous integration purpose, these projects can be stored in Version Control System location and built through Asset Build Environment and deployed to Integration Cloud.

The following figure depicts the processes involved in exporting deployable configurations.




# Exporting Deployable Configuration

You can export one or more runtime configurations to a YAML template file or as a deployable asset. You can export only deployable configurations. Deployable configurations are those configurations that can be deployed to Integration Cloud. Configuration Browser view lets you identify these deployable configurations assets in the list with the  icon.

You can export deployable configuration from a runtime instance as YAML template within a configuration project. For more information about configuration projects, see "Using Configuration Project" on page 54. This YAML template can be manually edited and build as deployable asset. Alternatively, you can export configurations as deployable asset, which can be readily deployed to cloud solutions.

**To export a configuration**

1. On the Landscape Navigator view , select one or more configurations from a runtime to export.
2. Right-click and select **Export Configurations**.
3. Under **Location**, provide the following information for the configuration project:
   - Project: The project name for configuration project. You can assign a project in either of the following ways:
     - Browse and select an existing configuration template project.
     - Specify a new configuration project resource. For more information about creating a new project resource, see "Using Configuration Project" on page 54.
4. In the **Name** field, select a template name.

   The template contains the data for the selected configurations.
5. Under **Format**, select any one of the following values:
   - YAML source: To export configurations to YAML template. You can manually edit and build YAML template as a deployable asset.
   - Deployable Asset : To export configurations that are readily deployable for cloud solutions.
6. In the **Parameterization** field, type the name of a parameter used to identify parameterization.

   An identifier can include lowercase a-z, uppercase A-Z, digits, dollar sign, and underscore. An identifier cannot start with a digit and cannot include any other special characters.
7. Click **Next**.

   By default, Designer displays deployable configurations that are included for export.
8. Select the configuration contents to confirm the set of configurations to export.
9. Click **Finish**.

   The exported YAML template file will open in an editor. You can view the exported contents using the Project Explorer view.

# Using Configuration Project

The configuration project is used for adding, editing, and building configuration templates exported from Integration Server and Universal Messaging runtimes. This project is pre-configured to build with Asset Build Environment which is used to prepare a deployable configuration asset from YAML source.

Before you configure a project, specify the build path entries to be used as the default path. For more information on specifying build path entries, see the Build Path Preferences page in Designer.

**To create a configuration project**

1. In Designer, navigate to **File > New > Project > SoftwareAG**.
2. Under SoftwareAG element, select **Asset Builder** and click **Configuration template**.
3. Click **Next**.

   Designer displays New Configuration Project dialog box.
4. In the **Project name** field, type the new name for the project.
5. Clear the **Use default location** check box, to choose a different location.

   By default Designer uses the Workspace root location. For example, (c: \<username> \workspace).
6. Click **Browse** adjacent to the **Location** field. Navigate to and select the location directory.
7. In the **Project Settings**, edit the default project structure preference to use for the source and output of builds.
8. Under **Working sets**, select check box **Add project to working sets** to specify a working sets.
9. Click **Finish**.

   Designer creates a configuration project.

**Note:** You can install third-party YAML editors such as YEdit for Syntax highlighting support.

# Editing Configuration Template

Designer stores each configuration template as a YAML resource. A configuration template may contain multiple configuration types and corresponding instances from either Integration Server or Universal Messaging runtime. You can add or edit configuration types or instances in a configuration template exported from Landscape

Navigator. Designer uses the default YAML editor, or the Eclipse text editor to load the template.

**To add configuration type or instance to a template**

1. On the Landscape Navigator view , select an existing configuration from a runtime.
2. In the **Properties** view, select **Template** tab.

   The **Template** tab displays the YAML source for the selected configuration.
3. Copy the required configuration data from the YAML source.
4. Append the data in the template.

   Designer internally validates the YAML content for syntactical correctness and produces Deployable Configuration Asset upon save.

**Note:** You can install third-party YAML editors such as YEdit for Syntax highlighting support.

# Deploying a Configuration Project

**To deploy configuration assets to Integration Cloud**

1. In **Project Explorer** view, select the configuration project that you want to deploy to Integration Cloud.
2. Right-click the project and click **Deploy Configurations to Cloud**.
3. In the **Configuration Templates** dialog box, select the templates to deploy. Click **Next**.
4. In the **Configuration Template Variable Substitution** dialog box, select the template and edit the values of the property that you want to modify before deploying to cloud. Click **Next**.
5. In the **Select Cloud Solution** dialog box, select the solution on Integration Cloud to which you want to deploy the assets and click **Finish**.

   Designer deploys the configurations assets to a remote repository provisioned for the tenant on Integration Cloud.

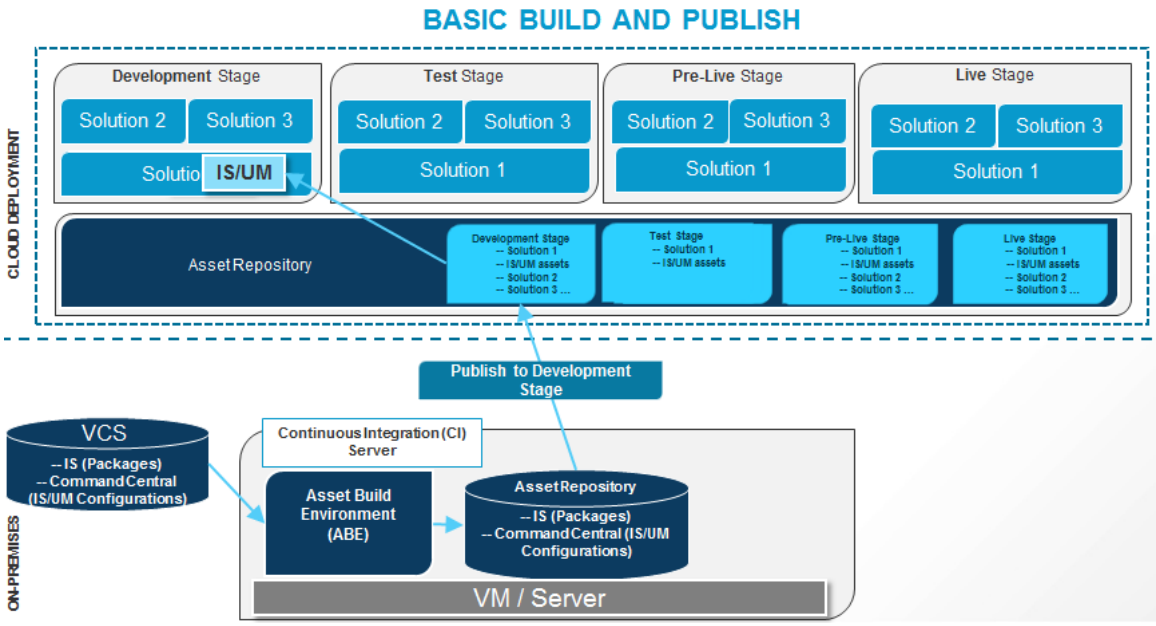
# 13 Continuous Build and Publish

# Continuous Build and Publish

You can *build* user-created assets and configurations using webMethods Asset Build Environment (ABE), *retrieve* those assets and configurations from a VCS by using ABE or an automated tool like Jenkins, and then *deploy* those assets and configurations to the Integration Cloud*Development Stage* by using ABE.

**Note:** See "http://techcommunity.softwareag.com/web/guest/pwiki/-/wiki/Main/CLOUD+DEPLOYMENT+USING+ABE" for information on how to deploy on-premises assets and configurations using webMethods Asset Build Environment.

The following figure depicts the processes involved in the basic build and publish mechanism:




The following high-level steps describe the continuous integration (CI) and continuous deployment (CD) mechanism:

1. Create source packages and store them in a file system or any version control system (VCS). The source directory must be accessible to the build scripts. If the source directory is a file system on a VCS, ensure that the source directory is checked out in the local file system.

2. **Creating and exporting configurations from Software AG Command Central**

   To create Command Central configurations, use the following command: `sagcc exec templates composite generate -i <generate-templatemetadata-file>.xml`

To export the configuration YAML files to a directory that you intend to use as sources for ABE, use the following command: `sagcc get templates composite export <templateAlias> -o <template-file>.yaml -f application/yaml`. Ensure that the exported YAML files are accessible to the build scripts.

3. Install webMethods Asset Build Environment (ABE). ABE installs the build scripts and the build properties file that you use to build the composites and descriptors. See the *Building Composites for Repository-Based Deployment* section in the *webMethods Deployer User's Guide* for more information.
4. Set the properties for the build and run the build script in ABE to create the composites and descriptors. See the *Building Composites for Repository-Based Deployment* section in the *webMethods Deployer User's Guide* for more information.
5. Use ABE to push the composites and descriptors to the Integration Cloud Asset Repository. Apart from providing values to the properties in build.properties, specify the following *additional* properties to upload composites and descriptors to the Integration Cloud Asset Repository.

| Property | Definition |
|---|---|
| localStore | Represents the local folder where the Git repository is cloned or the local Git repository. *localStore* must be the same as the *build.output.dir* . |
| remoteStore | URL of the Integration Cloud Asset Repository. The URL format is:<br>`https://mysubdomain.webmethodscloud.com/integration/rest/internal/wmic-git/<stagename>-<solutionname>-<productaliasname>`<br>Example:<br>`https://mysubdomain.webmethodscloud.com/integration/rest/internal/wmic-git/development-mysolution-myis` |
| message | Required. Message which describes the commits. |
| username | Required. Integration Cloud login user name. |
| password | Required. Integration Cloud login password. |

Run one of the following commands from the <Software AG_directory>\common\AssetBuildEnvironment\bin directory to build and upload the composites and descriptors to the Integration Cloud Asset Repository:

| For this platform... | Run the following command... |
|---|---|
| Windows | `build.bat buildUploadAssets` |
| UNIX | `build.sh buildUploadAssets` |

6. **Promoting assets to the next stage**

   a. Using an automated tool like Jenkins, invoke the following API to promote the assets to the next stage:

   **Method**: POST

   **URL**: https://{subdomain}.webmethodscloud.com/integration/rest/external/v1/cdep/binaryAssets/solutions/{solutionName}/nodes/{nodeName}?action=promote

   **URI parameters**

   - *solutionName* - Name of the solution in Integration Cloud
   - *nodeName* - webMethods Integration Server instance where the assets need to be deployed.

   **Note:** Use *basic* as the HTTP authorization scheme.

   **Request Body**: The request body should contain the following:

   - **fromStageName**: In which stage the assets are available in Integration Cloud
   - **toStageName**: To which stage the assets are to be promoted in Integration Cloud
   - **commitMessage**: An optional commit message
   - **acdlComposite**: List of composites, assets, and their properties, which are available in the previous stage.

   **Note:** If you want to retrieve the request body, invoke the *getcomposites* API as provided in *Step b*. The *getcomposites* API will get the composites and asset details from the previous stage.

   **Sample Request Body**

```
{
"integration": {
   "cicdBean": {
     "fromStageName": "Development",
     "toStageName": "Test",
     "commitMessage": "Promoting assets from Development stage to
     Test stage.",
     "acdlComposite": {
        "IS": [{
           "name": "mySqlAssets",
           "targetNamespace": "http://namespaces.softwareag.com/
           webMethods/IS",
```

```
            "assets": [],
            "properties": [{
               "name": "activatePkgOnInstall",
               "values": [
                       "true"
               ]
            },
            {
               "name": "archivePkgOnInstall",
               "values": [
                       "true"
               ]
            },
            {
               "name": "compilePackage",
               "values": [
                       "true"
               ]
            }
          ]
        },
        {
            "name": "JDBCRegression_deployment",
            "targetNamespace": "http://namespaces.softwareag.com/
            webMethods/IS",
            "assets": [{
                    "name": "deploy.notifications.deleteN",
                    "properties": [{
                          "name": "art.deployment.state",
                          "values": [
                              "disable"
                          ]
                        },
                        {
                          "name": "notificationImmediate",
                          "values": [
                              "false"
                          ]
                        },
                        {
                          "name": "notificationInterval",
                          "values": [
                              "10"
                         ]
                        },
                        {
                          "name": "notificationOverlap",
                          "values": [
                              "false"
                         ]
                        }
                   ]
              }],
              "properties": [{
                   "name": "activatePkgOnInstall",
                   "values": [
                           "true"
                       ]
              }]
          }
        ]
      }
    }
```

```
  }
}
```

b. **getcomposites API**

*getcomposites* retrieves all the composites and its assets which have properties.

**Method**: GET

**URL**: https://{subdomain}.webmethodscloud.com/integration/rest/external/v1/cdep/binaryAssets/solutions/{solutionName}/nodes/{nodeName}?stageName={stageName}

**URI parameters**

- *solutionName*: Name of the solution in Integration Cloud
- *nodeName*: webMethods Integration Server instance where the assets need to be deployed

**Query parameters**

- *stagename*: The stage from where the composites and assets have to be retrieved

*Allowed stage names*: development, test, live, preLive

**Note:** Use *basic* as the HTTP authorization scheme.

**Sample Response**

```
{
 "integration": {
  "message": {
    "code": 0,
    "description": "Success"
 },
 "cicdBean": {
    "fromStageName": "Development",
    "toStageName": "Test",
    "commitMessage": null,
    "acdlComposite": {
      "IS": [{
        "name": "mySqlAssets",
        "targetNamespace": "http://namespaces.softwareag.com/
        webMethods/IS",
        "assets": [],
        "properties": [{
           "name": "activatePkgOnInstall",
           "values": [
                   "true"
           ]
         },
         {
           "name": "archivePkgOnInstall",
           "values": [
                   "true"
           ]
         },
         {
           "name": "compilePackage",
           "values": [
                   "true"
```

```
            ]
          }
        ]
      },
      {
         "name": "JDBCRegression_deployment",
         "targetNamespace": "http://namespaces.softwareag.com/
         webMethods/IS",
         "assets": [{
                   "name": "deploy.notifications.deleteN",
                   "properties": [{
                       "name": "art.deployment.state",
                       "values": [
                                 "disable"
                       ]
                   },
                   {
                       "name": "notificationImmediate",
                       "values": [
                               "false"
                       ]
                   },
                   {
                       "name": "notificationInterval",
                       "values": [
                               "10"
                       ]
                   },
                   {
                       "name": "notificationOverlap",
                       "values": [
                                "false"
                       ]
                   }
                  ]
          }],
          "properties": [{
                    "name": "activatePkgOnInstall",
                    "values": [
                            "true"
                  ]
          }]
         }
       ]
      }
    }
  }
}
```

The response contains the following:

- **fromStageName**: The stage where the assets are available
- **toStageName**: The stage where the assets can be promoted
- **acdlComposite**: The composites, assets, and their properties. You can replace the properties for the assets wherever required, and invoke the Promote Assets API as mentioned in *Step a* to promote the assets to the next stage.

# 14 Best Practices, General Information, and Dependencies

# Best Practices, General Information, and Dependencies

This section contains information on best practices and general notes on Cloud Deployment.

## Best Practices

### Extended Settings deployment

- If you use the **Deploy To Cloud** option in Software AG Designer, you cannot deploy *Extended Settings* parameters of on-premises webMethods Integration Server. For deploying extended settings, use the **Landscape Navigator** to generate *Templates*, which can then be deployed to webMethods Integration Cloud.

## General Information

- For specifying JDBC Jars that can be shared by multiple packages, on-premises users may have placed those files under the custom/jars directory in webMethods Integration Server. For cloud deployment, place those files in the code/jars/static directory of some common package to make the required JDBC classes available when the package is deployed to webMethods Integration Cloud.
- If keystores and truststores are created in the customer's environment, location of the keystores and truststores may be pointing to some file on a local file system. It is recommended to place these files in the *config* directory of some common package. This package can be deployed to webMethods Integration Cloud and keystores and truststores can use the packages/CustomPackage/config/myKeystore.jks path to refer to the keystore location.
- In case of failure to send messages from Software AG Universal Messaging, webMethods Integration Server stores the messages in Client side Queue (CSQ), which is currently an embedded database. As webMethods Integration Server is in a Docker container, any such locally stored messages may be lost.
- Time out errors like *504 Gateway Time-out* may appear, if the processing time is greater than 2 minutes.
- Software AG Designer allows you to deploy assets without the required dependencies as well as delete assets which are referenced by other assets. It is recommended to manually check and manage the dependencies.

## Dependencies

- While deploying packages containing webMethods CloudStreams assets like CloudStreams connector services, CloudStreams connection, and CloudStreams

connector listeners, ensure that you set the dependency on the Provider package that was used to create those assets.

# 15 Cloud Deployment

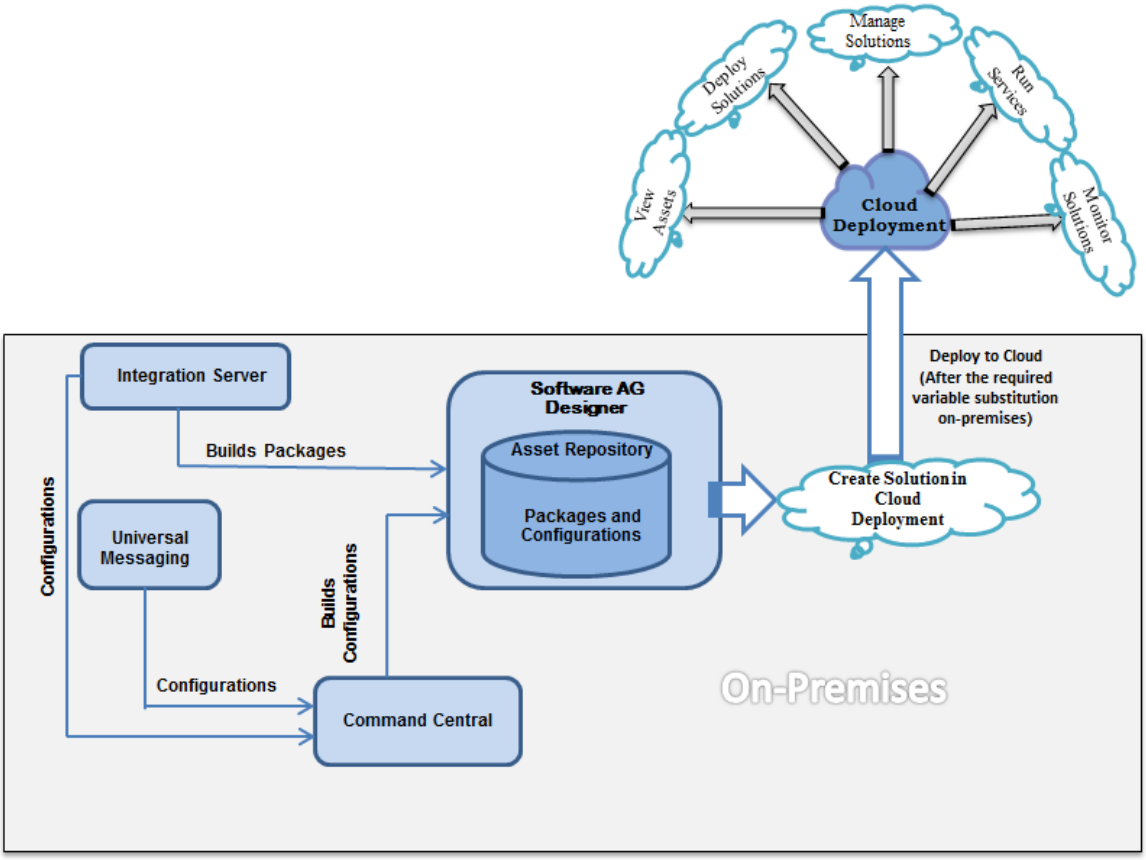
# Overview

Cloud deployment is the process of deploying packages and configurations that reside within on-premises runtimes or repositories to webMethods Integration Cloud. Using Software AG Designer or by employing the Continuous Integration (CI) approach, you can seamlessly deploy your on-premises webMethods Integration Server packages, Adapter packages, webMethods CloudStreams packages, configuration assets, and customized configurations to solutions created on Integration Cloud. In the cloud deployment context, configuration assets are limited to webMethods Integration Server and Universal Messaging configurations.

**Note:** As soon as you register, Cloud Deployment capability is enabled by default for all tenants.

**Note:** See *Deploying to webMethods Integration Cloud* for information on how to deploy packages and configurations that reside within on-premises runtimes or repositories to webMethods Integration Cloud. See the article on "Software AG TECHcommunity" website to get started on Cloud Deployment.

The following figure provides a high-level overview of the process involved in deploying on-premises webMethods Integration Server packages and configuration assets to Integration Cloud.

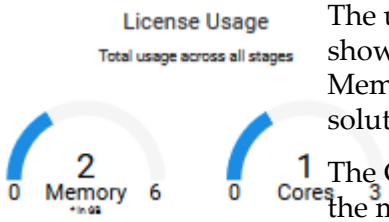



With Software AG Designer, you can deploy the webMethods Integration Server packages or configuration assets that you have created, verified, and tested on on-premises webMethods Integration Server or Universal Messaging to Integration Cloud. When you initiate the deployment from Software AG Designer, webMethods Integration Server packages and configuration assets are built from webMethods Integration Server and Software AG Command Central respectively, and are published to the Asset Repository available in Software AG Designer. After performing variable substitutions to make the on-premises configuration data compatible for cloud deployment, you can publish the packages and configurations to an asset repository provisioned for the tenant on Integration Cloud.

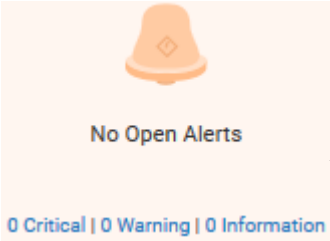
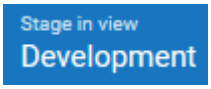
### User Interface elements

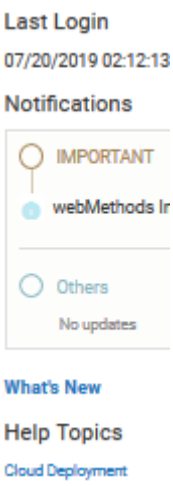
The following table describes the various user interface elements that appear on the **Cloud Deployment** workspace:

| Page Elements | Icon | Description |
|---|---|---|
| webMethods Cloud Deployment | WEBMETHODS Cloud Deployment | Access the home page. |
| Solutions | Solutions | View and create solutions, deploy |

| Page Elements | Icon | Description |
|---|---|---|
| | | solutions to stages, view the Asset Repository, and manage webMethods Integration Server instances. |
| Monitoring | Monitoring | View the overall status of the solutions, landscapes, system and runtime alerts, KPI graphs of the runtimes, service executions of the webMethods Integration Server instances, availability of the run times, alert status of the solutions, and logs. |
| Runtime availability |  | View the overall run-time availability for all the solutions in the last 24 hours. |
| Usage Statistics |  | The usage statistics shows the CPU and Memory usage for all solutions in all stages.<br>The CPU bar shows the maximum CPU that is licensed for the tenant. The colored part of the bar shows how much of the allowed CPU is currently used by the tenant.<br>The Memory bar shows the memory in Gb that is licensed |

| Page Elements | Icon | Description |
|---|---|---|
| | | for the tenant. The colored part of the bar shows how much of the allowed memory is currently used by the tenant. |
| Database |  | Add a database to your cloud deployment subscription. This enables you to configure, store, and monitor your database directly in the cloud instead of using external systems. |
| Solutions section |  | Total number of solutions created. View the *Solution List* page by clicking the Details link and the *Monitoring Dashboard* by clicking the Monitoring link. |
| Service Executions section |  | Number of service executions and their status in the last 24 hours for all the solutions that are currently available in the selected *Stage in view*. To view the Services page under Monitoring, click the service execution link. |

| Page Elements | Icon | Description |
|---|---|---|
| Active Alerts section |  | Number of currently open alerts and their severity. Click on the links to go to the *Alerts* page. |
| Change the stage |  | Click to change and manage stages. |
| Application Launcher |  | Click to access integration Apps. |
| Help |  | Access Help topics, Software AG TECHcommunity, Licensing capabilities, and the About page. The About page displays the version information, Global Support information, Copyright information, Impressum and Privacy Policy, and the release readme. |
| Profile |  | Name of the logged in user, profile information of the logged in user, and Logout option. |

| Page Elements | Icon | Description |
|---|---|---|
| Last Login, Notifications, and Help topics |  | When was the last login, important and other notifications, what is new in this release, and context-sensitive help topics. |

# Solutions

A **Solution** consists of packages bundled together into one coherent service. It is a logical combination of webMethods Integration Server packages, Adapter packages, Services, webMethods CloudStreams packages, and webMethods Integration Server and Universal Messaging configuration assets or configurations.

> **Note:** After deploying the configurations, due to webMethods Integration Server restart, the assets will appear after a short delay.

Integration Cloud Integrations can invoke Cloud Deployment webMethods Integration Server services for the same tenant. Using the predefined *Cloud Deployment Application* available in Integration Cloud, you can select the solution webMethods Integration Server services that you want to invoke from Integration Cloud.

## Creating Solutions

The **Solution List** page displays the solutions created in Integration Cloud.

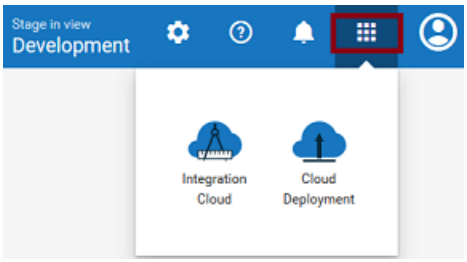
> **Note:** You must have the required permissions under **Settings**  **> Access Profiles > Administrative Permissions > Functional Controls > Solution** to create, update, or delete Solutions.
>
> **Note:** You must first create a solution in Integration Cloud before deploying the on-premises assets and configurations.

All solutions created are initially copied to the Development stage. You can create a new solution only in the Development stage. Further, you cannot modify a solution after it

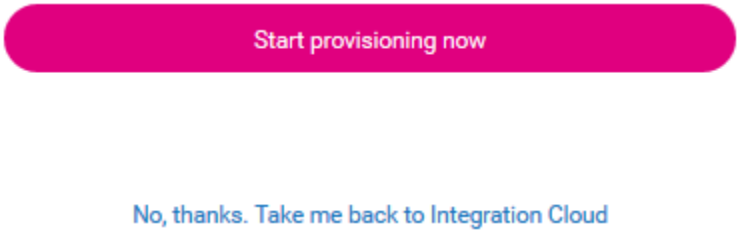
is created. You can configure the solution in subsequent stages but after you configure a solution in a stage, you cannot modify the solution again in that stage. In a stage, you can configure only those solutions that are marked as **Not Configured**.

### To create a solution

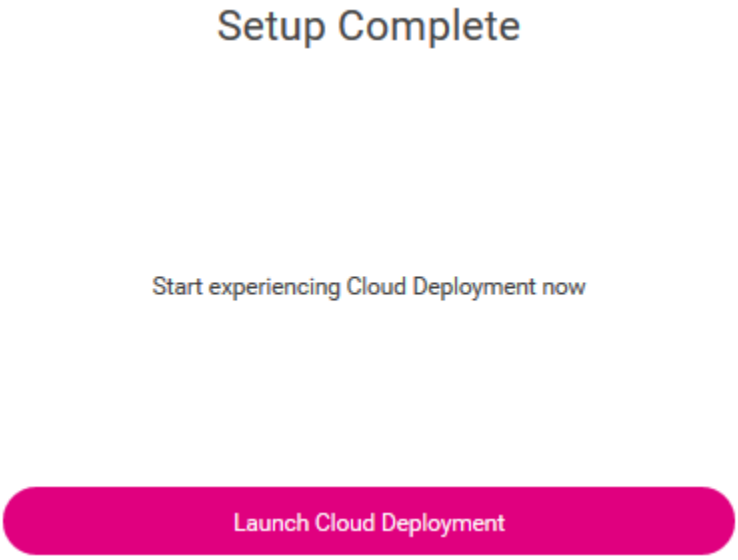
1. Switch to the **Cloud Deployment** perspective.




As soon as you register, Cloud Deployment capability is enabled by default for all tenants. By default, 3 CPU core and 6 GB memory are allocated for all tenants. When you access Cloud Deployment for the first time using the application launcher, you need to start provisioning.




After provisioning, your setup will be complete, and you can launch Cloud Deployment.



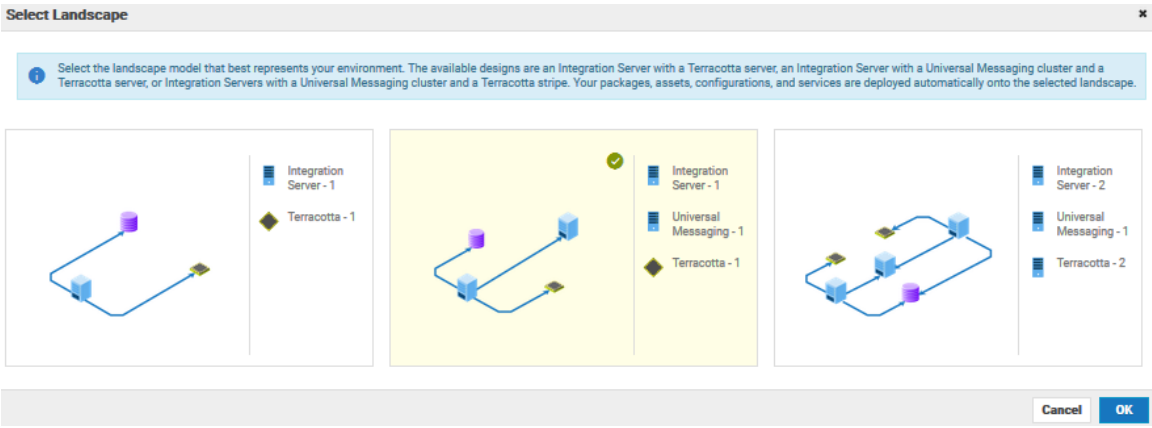
> **Note:** Solutions created using a trial account are deactivated daily. After you log in, you need to reactivate the solutions. All assets will be available after a short delay.

2. After you launch Cloud Deployment, from the **Cloud Deployment** navigation bar, click **Solutions > Solution List**.

   The **Solution List** page appears.

3. From the **Solution List** page, click **Create New Solution** to create a solution. You can create a solution only in the **Development** stage.

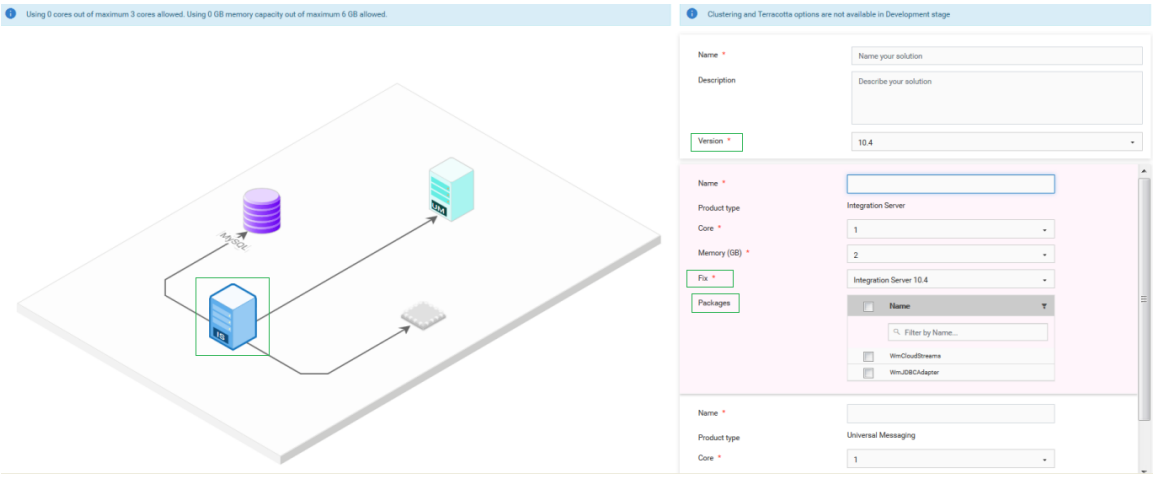
4. Select the landscape model that best represents your environment. The available designs are an webMethods Integration Server with a Terracotta server, an webMethods Integration Server with a cluster and a Terracotta server, or webMethods Integration Servers with a Universal Messaging cluster and a Terracotta stripe. Your packages, assets, configurations, and services are deployed automatically onto the selected landscape.




5. Click **OK**.

6. In the **New Solution** page, fill in the solution **Name**, the solution **Description**, the name of the webMethods Integration Server and Universal Messaging instances, number of CPU **Cores**, and **Memory** characteristics of the hardware to support each service in the solution landscape.

> **Note:** You can also select the webMethods Integration Server and Universal Messaging icons to highlight the sections in the **New Solution** page. Terracotta is available only when webMethods Integration Server runs in a clustered mode. Further, clustering and Terracotta options are not available in the **Development** stage.




The **Version** list box lists all the available major versions. The solution will be created based on the available **Fix** version. webMethods Integration Server packages such as WmCloudStreams and WmJDBCAdapter will appear in the **Packages** group box based on the **Fix** version. Select the **Packages** you want to enable. After you save the solution, you will not be able to change the major **Version**. You can change only the number of CPU Cores, Memory, Fix version if available, and Packages.

You can deploy CloudStreams provider packages, CloudStreams connector services, CloudStreams connection, and CloudStreams connector listeners to a solution in Cloud Deployment only if you have selected WmCloudStreams as the package option while creating the solution.

**Note:** After creating a solution, if you modify the package list options in the **Packages** group box and save the solution, due to webMethods Integration Server restart, the assets will appear after a short delay.

### Updating products to a higher fix version in a solution

You can update any product in a solution to the available higher fix version after you create the solution. The **Update Available** option appears if a higher fix version is available for any of the products in the solution. The latest fix version appears in the **Fix** drop-down list. For example, if you have selected Fix 3 of webMethods Integration Server while creating a solution, and if Fix 4 is now available, you can select Fix 4 and save the solution.

**Note:** While configuring a solution in a higher stage, the fix version of the products in the solution will be taken from the lower stage. If a solution is updated in a lower stage, you will be given an option to update the solution in the higher stage, to the fix versions of the products in the lower stage.

For example, if you configure a solution in a higher stage (Test stage) and then go back to a lower stage (Development stage), and If a higher fix version is available for a product in the Development stage, you can click **Update Available**, and in the **Edit Solution** screen, you can select the higher fix version for the product in the solution.

Now if you go to the higher stage (Test), the **Update Available** option will appear in the Test stage. You can update products in a solution at every stage, to the fix versions of the products in the solution available in the immediate lower stage. Note that after you update or update the products in the solution, the products in the solution will not be accessible for sometime.

**Updating products to a higher version in a solution**

You can update any product in a solution to the available higher version after you create the solution. The **Update Available** option appears if a higher version is available for any of the products in the solution. The latest version appears in the **Version** drop-down list. For example, if you have selected 10.3 of webMethods Integration Server while creating a solution, and if 10.4 is now available, you can select version 10.4 and save the solution. Further, if you update to v10.4, the latest available fix for 10.4 will be automatically selected.

> **Note:** While configuring a solution in a higher stage, the version of the products in the solution will be taken from the lower stage.

If a higher version is available for a product in a stage, you can click **Update Available**, and in the **Edit Solution** screen, you can select the higher version for the product in the solution.

The **Schedule** option appears if you select a higher version available for any of the products in the solution. Click **Schedule** to schedule the update process by specifying the date and time on which you want the update process to execute. Once the solution is scheduled for update, you can click **Cancel Schedule** to cancel the schedule or click **Modify Schedule** to modify the schedule. If the update process is scheduled, the status of the solution on the **Solution List** page displays **Update scheduled**. The status of the solution on the **Solution List** page displays **Update in progress** if the scheduled update process is under way.

> **Note:** After a solution is updated successfully, ensure that you click **Confirm Update** within seven days to complete the update process or click **Rollback** to rollback the solution to the previous version. After seven days, except for the **Confirm Update** and **Rollback** options, the solution page will not be available. If you rollback, the status of the solution on the **Solution List** page displays **Rollback in progress** until the solution is rolled back to the previous version.

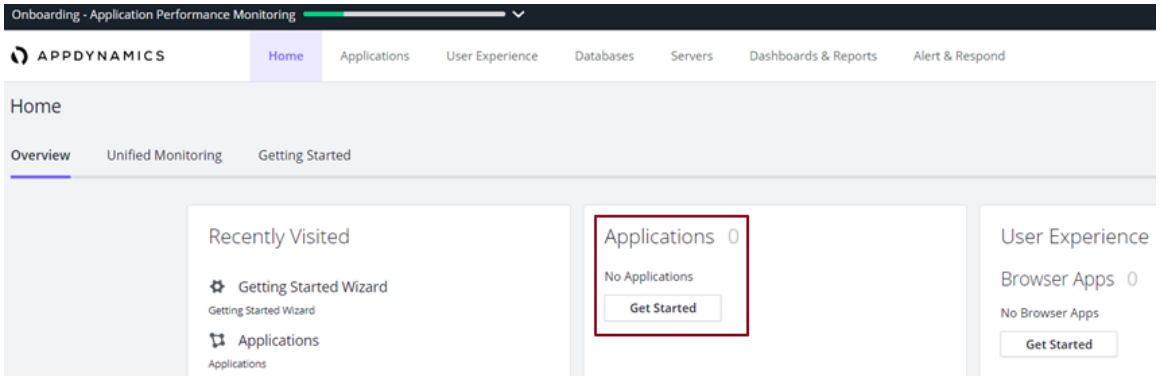
7. The **Tracing** list box appears only if you have the App Dynamics capability and if you are currently using AppDynamics to trace end to end business flows. It allows you to trace logs after you create or update a solution for an webMethods Integration Server runtime. Currently, tracing support is provided only for the webMethods Integration Server runtime. Select **AppDynamics** from the drop-down list to provide the AppDynamics tracing support and upload a valid Controller XML and Config XML file containing the Appdynamics details. You can update the controller file for each webMethods Integration Server runtime. You also have the option to download the controller and config files and then upload the files after modifying them.

**Note:** If you select AppDynamics tracing and provide a valid controller file, the tracing data will appear on the AppDynamics cloud application.

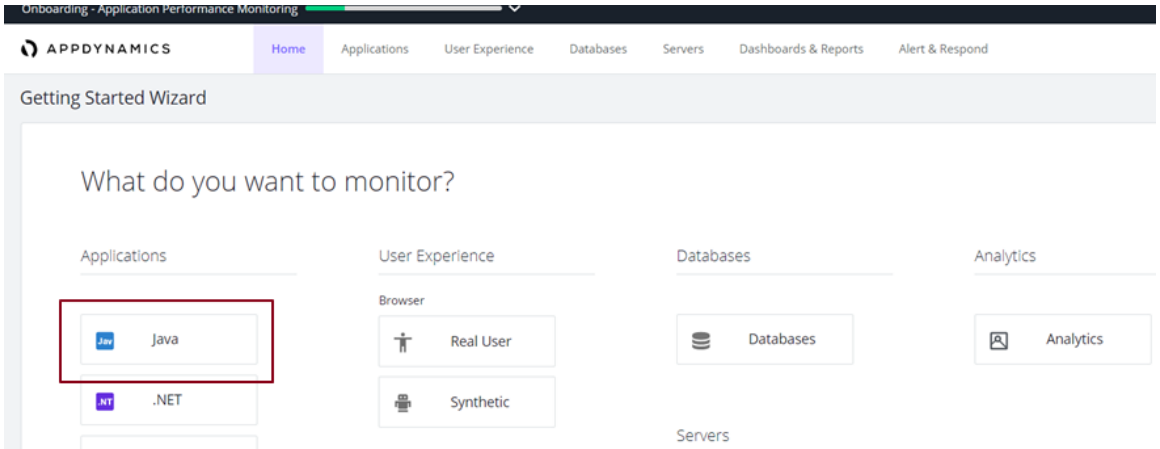
Let us see an example of how the tracing data appears on Appdynamics when you create a solution in Cloud Deployment.

a. Go to https://www.appdynamics.com/ and register in AppDynamics.

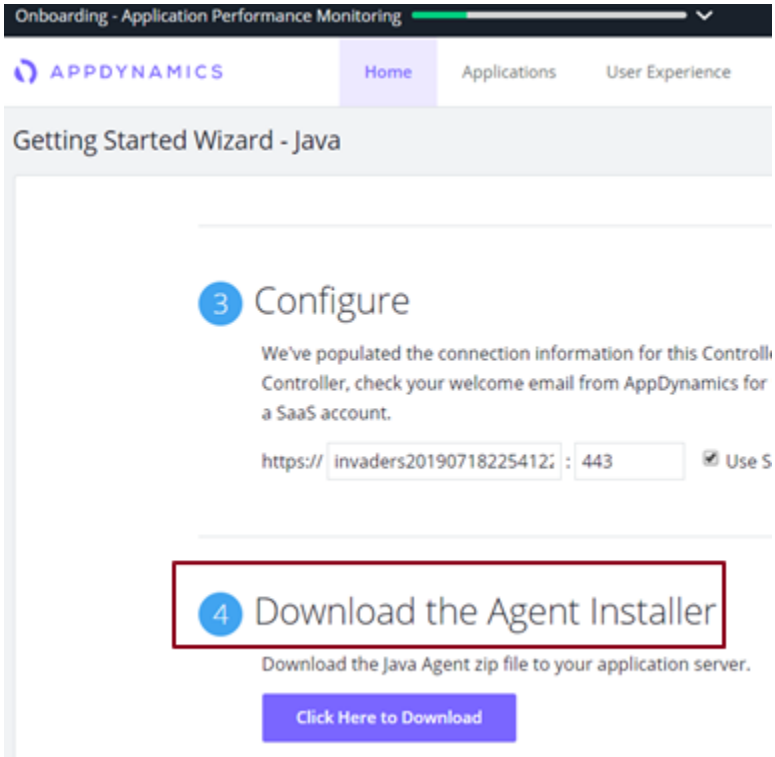
b. On the Overview > Applications panel, click **Get Started**.




c. On the Getting Started wizard, click **Java** under the Applications panel.




d. Download the Java Agent Installer.




e. Go to the ver<xxx> > **conf** folder on your local system and view the controller-info.xml and app-agent-config.xml files.

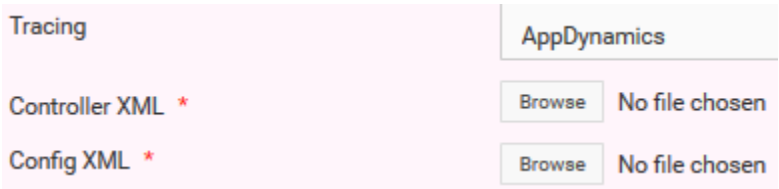
f. Open the controller-info.xml file and provide the following values:

   <application-name>MyApplication</application-name

   <node-name>AppIS</node-name>

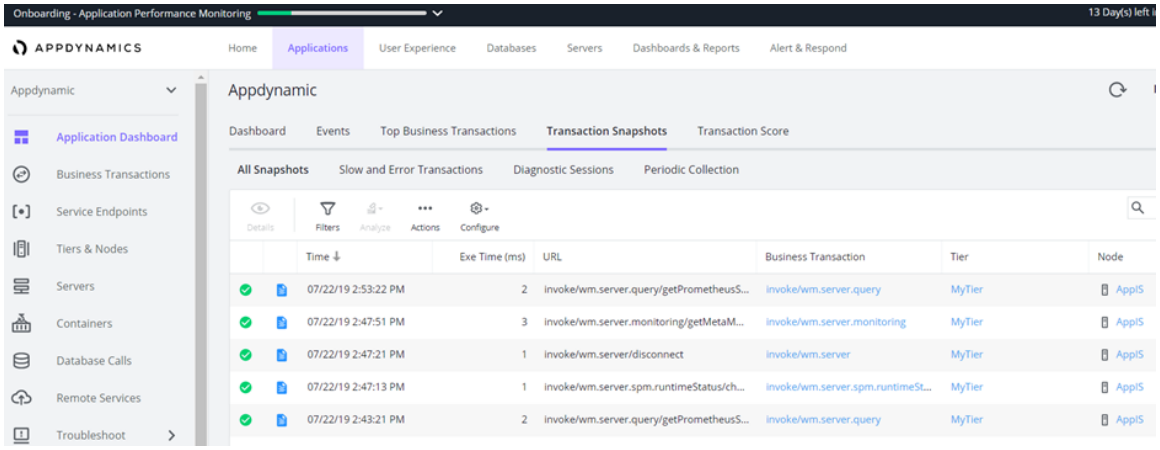
g. Log in to Integration Cloud and switch to Cloud Deployment using the App switcher.

h. Create a new solution in Cloud Deployment and select **AppDynamics** from the drop-down list to provide the AppDynamics tracing support. Upload the controller-info.xml and app-agent-config.xml files, and save the solution.




i. Log in to AppDynamics. Your application, that is, **MyApplication** appears on the Applications panel.

j. Click on MyApplication and view the service execution status and execution logs.



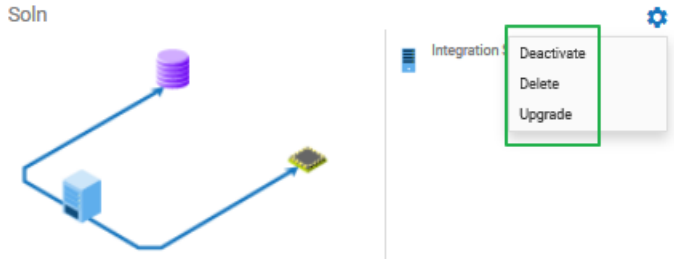

8. Click **Save** to save the solution.

   The new solution is created and appears in the **Solution List** page. You cannot modify the solution after the solution is created. You can configure the solution in subsequent stages but once a solution is configured in a stage, you cannot modify the solution again in that stage.

   **Note:** You can now deploy the solution to the next stage.

   **Deactivate, Activate, and Delete a solution**

   Click the ⚙ icon and select **Deactivate** to deactivate a solution. All packages and assets will be permanently deleted and cannot be recovered. Click the ⚙ icon and select **Activate** to activate an inactive solution.




   Click the ⚙ icon and select **Delete** to permanently delete the solution. If a solution is configured in a subsequent stage, it will be permanently deleted from the current stage and cannot be recovered. Further, if you delete the solution, you will not be able to promote assets from the current stage.

## Exploring Solutions

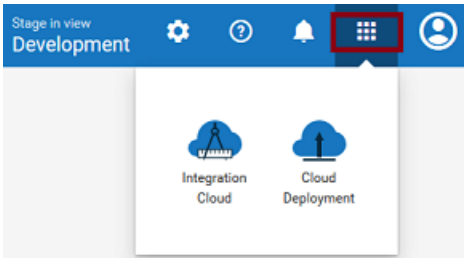
The solution details page allows you to view the packages, assets, configurations and services for different runtimes in the solution, deploy the solution to another stage, view the Asset Repository, and manage the solution, that is, view the landscape, configure

webMethods Integration Server service access settings, administer the webMethods Integration Server, or restart the webMethods Integration Server instances.

**Note:** You can create a new solution only in the Development stage. You cannot modify a solution after you create a solution. You can configure the solution in subsequent stages but after you configure the solution in a stage, you cannot modify it again in that stage.

### To view the Solution Explorer

1. Switch to the **Cloud Deployment** perspective.

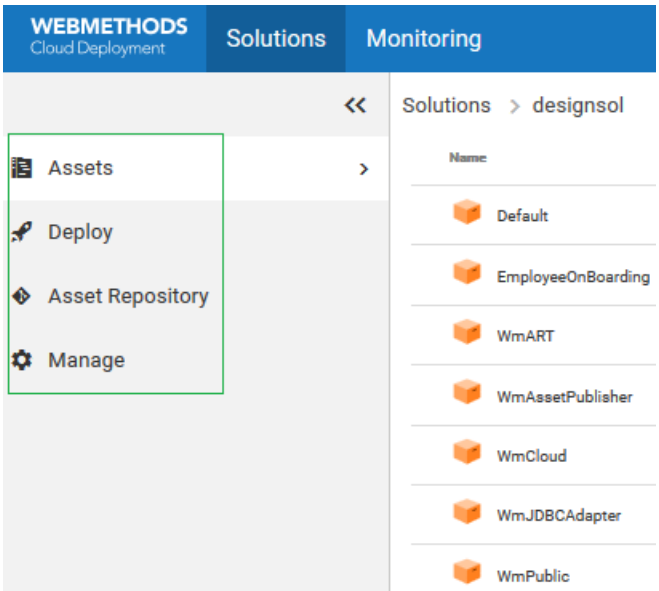



2. From the **Cloud Deployment** navigation bar, click **Solutions > Solution List**.

   The **Solution List** page appears listing all the solutions.

3. Click on an existing solution. The Solution Explorer page appears.

   **Solution Explorer**




The following table provides a high-level overview of the Solution Explorer page:

| Component | Description |
| --- | --- |
| "Assets" on page 84 | View the Packages, Folders, Assets, and Services for webMethods Integration Server, Adapters, webMethods CloudStreams packages, and configurations for the Universal Messaging runtime. |
| "Deploy" on page 88 | Deploy the solution to another stage. |
| "Asset Repository" on page 87 | View the Asset Repository which displays the contents of the on-premises packages published to Integration Cloud. |
| "Manage" on page 94 | View the landscape, configure webMethods Integration Server service access settings, administer the webMethods Integration Server, or restart the webMethods Integration Server instances. |

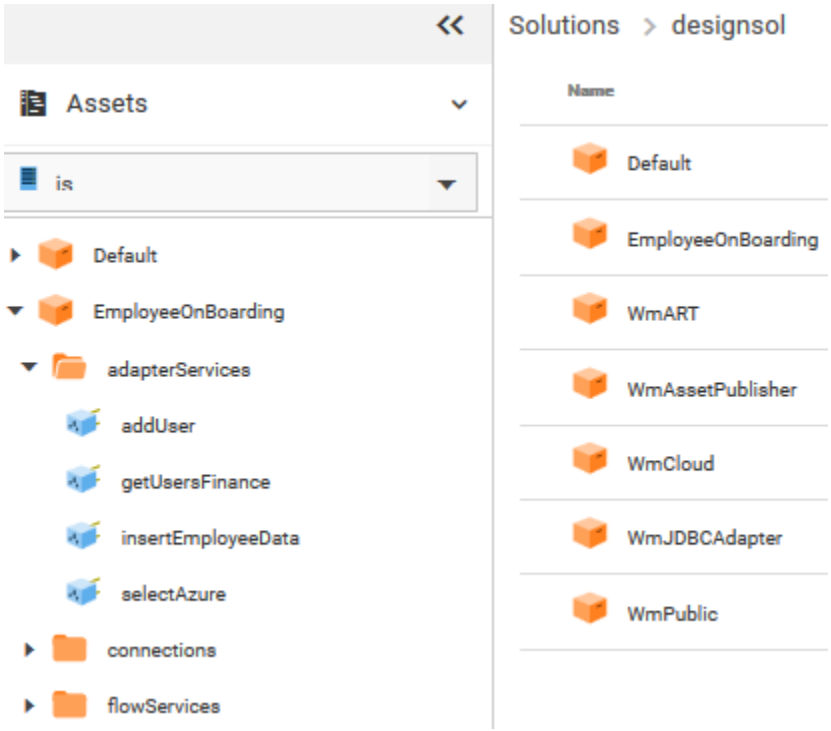
## Assets

A package is a container that is used to bundle services and related elements, such as specifications, webMethods Integration Server document types, webMethods Integration Server schemas, and triggers. When you create a folder, service, webMethods Integration Server document type, or any element, you save it in a package.

**Note:** To view and access webMethods Integration Server packages in Integration Cloud, you must assign any custom user groups created in webMethods Integration Server, which are assigned to Access Profiles in the **Solution Permissions** page, to the following Access Control Lists in webMethods Integration Server:

- Administrators ACL
- Developers ACL
- Replicators ACL

The following figure depicts the package structure in a solution.




Packages are designed to hold all of the components of a logical unit in an integration solution. For example, you might group all the services and files specific to a particular marketplace in a single package. All the components that belong to a package reside in the package's subdirectory.

**Note:** Click the arrow beside a folder or package to view its contents.

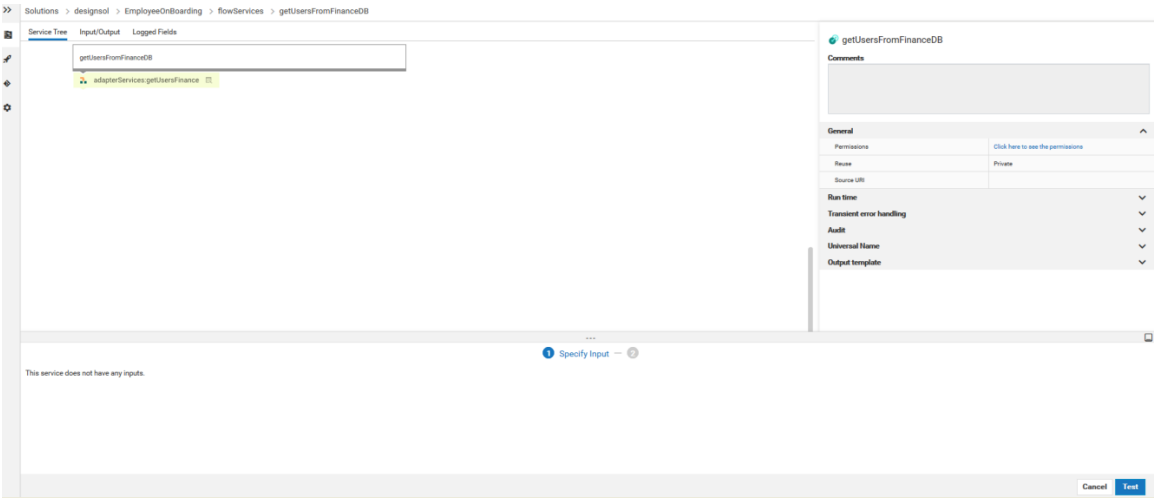
### Downloading assets

You can download user deployed packages and configurations from the **Asset** page. To download assets, point to an asset, click the ⋮ icon, and then click **Download**. The assets will be zipped and downloaded to your local storage space.

**Note:** You cannot download default packages such as packages that come with webMethods Integration Server installation, for example, Default, WmART, WmCloud, WmJDBCAdapter, WmPublic, and so on.

### Services

Services are method-like units of logic that operate on documents. You build services to carry out work such as extracting data from documents, interacting with back-end resources (for example, submitting a query to a database or executing a transaction on a mainframe computer), and publishing documents. Adapters and other add-on packages provide additional services that you use to interact with specific resources or applications. The service editor allows you to view and run the services.




### Service Signature

Input and output parameters are the names and types of fields that the service requires as input and generates as output. These parameters are also collectively referred to as a *signature*. You declare a signature for all types of services: flow services, Java services, and services written in other supported programming languages.

For a flow service, the input side describes the initial contents of the pipeline. In other words, it specifies the variables that this flow service expects to find in the pipeline at run time. The output side identifies the variables produced by the flow service and returned to the pipeline. An webMethods Integration Server document type can also be used to define the input or output parameters for a service.

Click **Test** to run the service after providing the data to pass into the service.

### Service Editor

Use the service editor to view the services. The source code, properties, inputs, and outputs are read only. The editor has the following tabs:

- **Source** tab contains the code or flow for the service.
- **Input/Output** tab contains the input and output signature of the service.
- **Logged Fields** tab indicates the input and output parameters for which the data is logged. You define the data to pass into the service by defining the input parameters on the lower panel of the editor.

### Load pipeline for testing services

The pipeline is the general term used to refer to the data structure in which input and output values are maintained for a service in Software AG Designer. The pipeline holds the input and output for a service. The pipeline starts with the input to the service and collects inputs and outputs from subsequent services. When a service runs, it has access to all data in the pipeline at that point.

When you run a service in Software AG Designer, you can click **Save** and save the pipeline data as an XML document to your local file system. After you deploy the

service, you can click the **Load Data** option in the service editor to select the XML file, and load or update the pipeline data to test the service.

**Note:** Integration Cloud Integrations can now invoke Cloud Deployment webMethods Integration Server services for the same tenant. A new pre-defined Application, *Cloud Deployment*, is added in Integration Cloud. Using this Application, you can select the solution webMethods Integration Server services that you want to call from Integration Cloud.

**Note:** See the *webMethods Service Development Help*, *webMethods Integration Server Administrator's Guide*, and the *webMethods Adapter for JDBC Installation and User's Guide* for detailed descriptions of all the services and document types including Adapter services.

#### Displaying the API details of an executable service

After deploying assets, on the Asset explorer page, click the *API Details* option to view the API details of the service such as the HTTP Method, URL, Input structure, and the parameters that are required to invoke this service from an external system, for example, a REST client. You can copy the required API details to execute the service from the external system.

**Note:** The *API Details* option appears only for executable services. Some examples of executable services are Adapter services, Flow services, Java services, Flat File Schema, and Map services.

**Note:** You will be able to execute a Flat File Schema by using *only* the SoapUI tool. You must specify the following settings in the SoapUI tool:

- Set the following query parameters in the **Request** section:

  skipWhiteSpace = true

  encoding = UTF8

  file = file:file1

- In the **Attachments** section, browse for the source file and update the following column values:

  Name = file1

  ContentID = file

## Asset Repository

The **Asset Repository** page **(Cloud Deployment > Solutions > Asset Repository)** displays the list of all solutions and the user-created assets. You can also view the asset type, version of the assets, and services. Before deploying packages and configurations from Software AG Designer, you must create a solution in Integration Cloud to which you want to deploy the configuration assets. Software AG Designer deploys the assets and configurations to the Asset Repository in Integration Cloud.

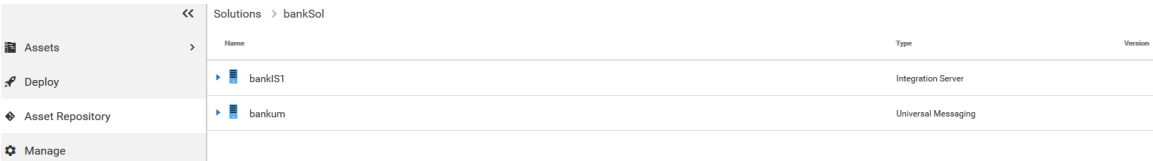
### Asset Repository for all solutions

The following figure depicts the Asset Repository structure for all solutions.


### Asset Repository for a selected solution

The **Asset Repository** page for a solution **(Cloud Deployment > Solutions > Select a Solution > Asset Repository)** displays the assets for the selected solution, including the asset type and version of the assets.

The following figure depicts the Asset Repository structure for a selected solution.




### Downloading packages and assets

You can download user deployed packages and configurations from the **Asset Repository** page. You can either download individual packages or the whole repository for each product. To download assets, point to an asset or product and click the icon. The assets including ACDL files will be zipped and downloaded to your local storage.

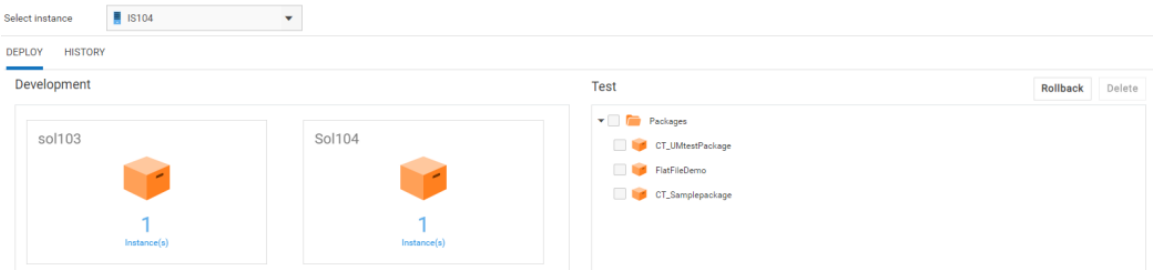
## Deploy

After publishing the assets and configurations that reside within on-premises runtimes or repositories to webMethods Integration Cloud, you can promote them from the previous stage to the current stage. Within a tenant, you can promote assets from a solution to another solution, from a previous stage to the current stage, for the same runtime type. You can promote assets if the source runtime version is lesser than or same as the target runtime version.

If you are in the **Development** stage, click the **Change Stage To View** link to change the stage. Only active and accessible stages appear in the drop-down list for selection in the **Stage to view** field. Select a different stage other than Development and click **Submit**.

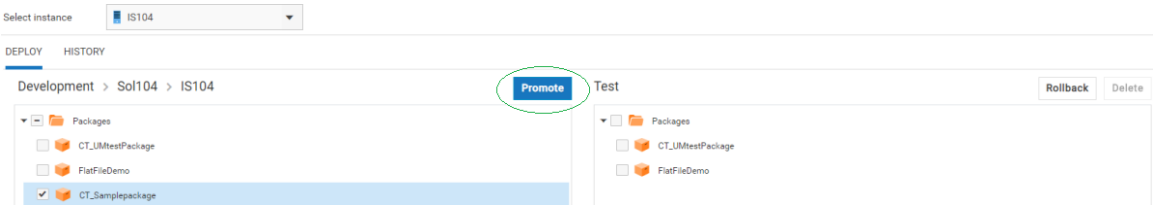
In the next stage, select a solution and then click **Deploy**. Select a runtime instance. All *solutions* of the previous stage will be listed on the left panel. The right panel will list all the assets of the selected solution in the current stage for the selected runtime instance.

**Note:** You will *not* be able to promote assets from a higher version solution to a lower version solution. For example, if you have a v10.4 solution in the source stage and a v10.3 solution in the current stage, you will not be able to promote the v10.4 assets to the current stage.




**Note:** You can access a stage only if your Access Profile is assigned to the stage.

Click on a solution and select the runtime package folder. Then click on the runtime instance. The assets of the solution corresponding to the selected runtime instance will appear. Select the assets and then click **Promote** to promote the assets to the current stage (right panel).




After you click **Promote**, the **Promote Assets** dialog box appears for the selected asset.

Select an asset and change the values for the variable substitution properties, if needed. The variable substitution properties appear for the selected asset, only if the asset has properties. If there are any similar type of assets for which you want the same values, then select the **Show similar assets to apply values** option. Then select the assets in the lower panel. Click **Apply** to apply the property values to the selected assets. The changed values will be applied to all the selected similar assets during promotion.

On the **Promote** dialog box, you can type a message to describe the promotion. The promotion message will appear on the **History** page.

Click **Simulate Promote** to check the consistency of the assets and their dependencies. If there are dependencies, then for a successful promotion, you have to select all the dependent assets. Select **Save and Promote** to save the variable substitution and promote the assets to the next stage.

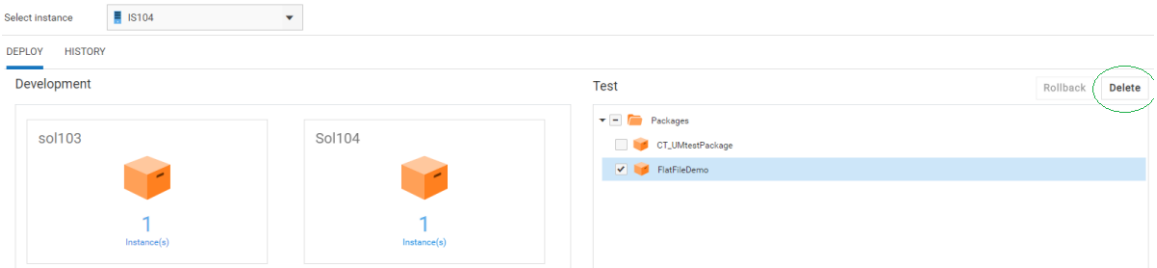
On the **Deploy** page, click **Rollback** to rollback *all* promoted assets to their previous state.

You can type a message to describe the rollback. The rollback message will appear on the **History** page.

### Deleting assets

On the **Deploy** page, you can delete an asset from the current stage (right panel). The asset will be deleted from the asset repository in that stage as well as from the runtime.

> **Note:** Currently you can delete only webMethods Integration Server packages and not webMethods Integration Server and Universal Messaging configurations.




### History

The **History** page shows the **Trace ID**, that is, the tracking ID, which is automatically generated on every successful or unsuccessful promotion, rollback, or deletion, the Deployment, Rollback, or Deletion **Action**, **Date** when the asset was promoted, rolled back, or deleted, the **User** who promoted, rolled back, or deleted the asset, and the commit **Message** for the selected instance. You can click on a **Trace ID** to see the Track History for the specific action.

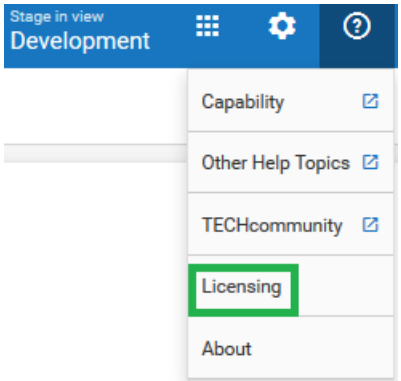
The **Track History** window displays the following details:

- **Timestamp** - The date and time when the log was generated.
- **Product** - The product that was promoted or rolled back.
- **Log Level** - Information on whether the log type is an Error, Info, or Debug.
- **Message** - Log or status message.

> **Note:** Promotion, rollback, or deletion details appear only for the current stage.

## Capability

The **Capability** ( > Licensing) page allows you to view the status of some of the system capabilities, based on your license offering.




You can view the details of the following capabilities in **Integration Cloud**:

| Field | Description |
|---|---|
| **Allowed application count** | Total number of Applications that can be utilized by the tenant. |
| **On-premises connection** | If **Yes**, then on-premises applications can be uploaded from on-premises systems. |
| **Max allowed users** | Maximum number of active users allowed for the tenant. |
| **Allowed number of stages** | Maximum number of staging environments allowed for the tenant. |
| **Integration restart and resume** | Integrations can be restarted and resumed. |
| **Integration import and export** | Integrations can be imported and exported. |
| **Trial account** | If **Yes**, then the account is a trial account. |
| **Trial end date** | The trial period end date. This field appears only if the account is a trial account. |

You can view the details of the following capabilities in **Cloud Deployment**:

| Field | Description |
|---|---|
| **Max allowed cores** | Maximum number of CPU cores allowed across all active solutions and all stages for the tenant. You will not be |

| Field | Description |
|---|---|
| | able to create additional solutions if you exceed this capability. |
| **Max allowed memory** | Maximum memory capacity allowed across all active solutions and all stages for the tenant. You will not be able to create additional solutions if you exceed this capability. |
| **Allowed number of stages** | Maximum number of staging environments allowed for the tenant. |
| **Trial account** | If **Yes**, then the account is a trial account. |
| **Trial end date** | The trial period end date. This field appears only if the account is a trial account. |

### Usage

The **Usage** page allows you to view the current usage of CPU cores and Memory (GB) for all the active solutions in all the stages.

To access this page, from the Cloud Deployment navigation bar, click and select *Licensing* > *Usage*.

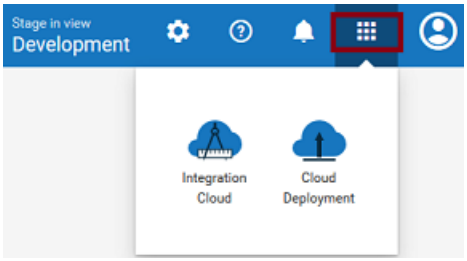
## Copying Solutions

The **Solution List** page allows you to copy solutions in any stage. Copying solutions allows you to have a back up of your solution before you make any changes and deploy your solution to production. This reduces the risk of not having a back up in case you want to revert to the original solution.

**Note:** If you have the required permission under **Settings** **> Access Profiles > Administrative Permissions > Functional Controls > Solution**, you can copy, update, and delete solutions.

**To copy a solution**

1. Switch to the **Cloud Deployment** perspective.



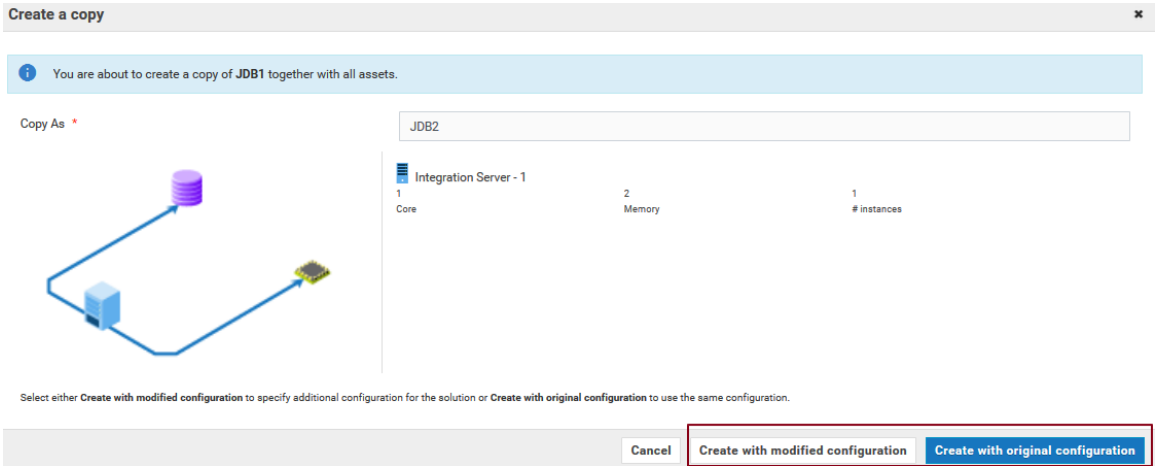

2. After you launch Cloud Deployment, from the **Cloud Deployment** navigation bar, click **Solutions > Solution List**.

   The **Solution List** page appears.

3. On a solution, click ⚙ **> Create a copy**.


4. On the **Create a copy** page, fill in the new solution **Name**.

   You can choose to copy solutions using the same configuration and services in the solution landscape by clicking the **Create with original configuration** option or modify the configuration and services in the solution landscape by clicking the **Create with modified configuration** option.




5. Select **Create with modified configuration** and specify additional configuration of the solution, if necessary. Then click **Configure** to save the configuration details.

   The new solution is created and appears on the **Solution List** page.

**Note:** You can now deploy the new solution to the next stage.

**Deactivate, Activate, and Delete a solution**

Click the icon and select **Deactivate** to deactivate a solution. All packages and assets will be permanently deleted and cannot be recovered. Select **Activate** to activate an inactive solution. Select **Delete** to permanently delete a solution.

## Managing Solutions

The **Manage** options allow you to view the solution landscape, configure webMethods Integration Server service access settings, administer the webMethods Integration Server, or restart the webMethods Integration Server instances.

### Landscape

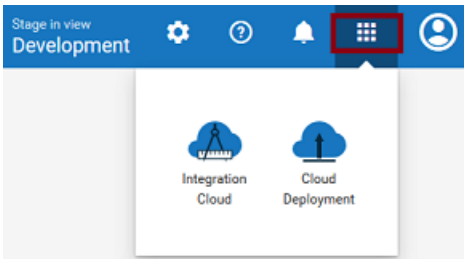
The **Landscape** page displays the landscape configuration for the selected solution.

**Note:** You cannot modify the solution landscape configuration in the Development stage after the solution is created. You can *configure* the solution in subsequent stages but after a solution is configured, the solution cannot be modified again in that stage. Further, clustering and Terracotta options are not available in the Development stage.

**Note:** In a stage, you can configure only those solutions that are marked as **Not Configured**.

**To view the landscape configuration for a solution**

1. Switch to the **Cloud Deployment** perspective.




2. From the **Cloud Deployment** navigation bar, click **Solutions > Solution List > Select a solution > Manage > Landscape**.
3. On the **Landscape** page, you can view the landscape configuration design, landscape solution name and description, and the landscape components. For each landscape component, you can view the landscape component name, product type, whether the landscape component is in a ready state, and the number of CPU cores and memory characteristics of the hardware to support each service in the solution.

4. Terracotta is available only when webMethods Integration Server runs in a clustered mode. Further, clustering and Terracotta options are not available in the **Development** stage.

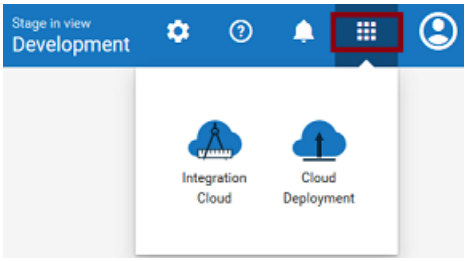
   On the **Landscape** page, select the **Cluster Type** as **Stateless** if the group of webMethods Integration Servers function in a manner similar to a cluster but are not part of a configured cluster. A stateless cluster of webMethods Integration Servers does not use a Terracotta Server Array. Select **Stateful** to add the Terracotta section. The Terracotta icons will be activated.

## Service Access Settings

The **Service Access Settings** page allows you to configure the webMethods Integration Server services to be called externally over HTTPS. The services will be available to consumers based on the **Allow All** and **Deny All** access modes.

### To configure service access settings

1. Switch to the **Cloud Deployment** perspective.




2. From the **Cloud Deployment** navigation bar, click **Solutions > Solution List > Select a solution > Manage > Service Access Settings**.
3. On the **Service Access Settings** page, configure the webMethods Integration Server services to be called externally over HTTPS. Required fields are marked with an asterisk on the screen.

| Field | Description |
|---|---|
| **Base URL** | The base URL is a part of the complete service URL, for example, https://wmic1.saglive.com/integration/clouddeployment/service/development/test/tt54/invoke/pub.* |
| **Solution Alias** | An alias for the solution in the base URL. The alias name for a solution is unique in that particular stage. For example, if the completed service URL is https://wmic1.saglive.com/integration/clouddeployment/service/development/test/ |

| Field | Description |
| --- | --- |
| | tt54/invoke/pub.*, then the Solution Alias is test, the webMethods Integration Server instance is tt54, and the service URL is /invoke/pub.*. |
| **Select Integration Server** | Select the solution webMethods Integration Server instance where the services need to be configured. |
| **Access Mode** | Ensure that the access mode of the services are properly set.<br><br>Select **Deny All** if you want to deny most of the services to be invoked and allow a few. Then click **ADD** to add services to the **Allowed Services** table. In the **Add Service** window, the **Base URL** is a part of the complete service URL. In the **Service URL** field, type the webMethods Integration Server Service URL. The Base URL and the Service URL together forms the complete service URL.<br><br>Select **Allow All** if you want to allow most of the services to be invoked and deny a few. Click **ADD** to add services to the **Denied Services** table. In the **Add Service** window, the **Base URL** is a part of the complete service URL. In the **Service URL** field, type the webMethods Integration Server Service URL. The Base URL and the Service URL together forms the complete service URL.<br><br>**Note:** You can update the service URL by clicking the **Edit** icon beside the service URL in the services table.<br><br>The services will be available to be invoked from a software application, for example, a REST client, after you add the services to the table.<br><br>For example, the Service URL */invoke/pub.math:addInts* has the following components:<br><br>■ Directive - *Invoke* |

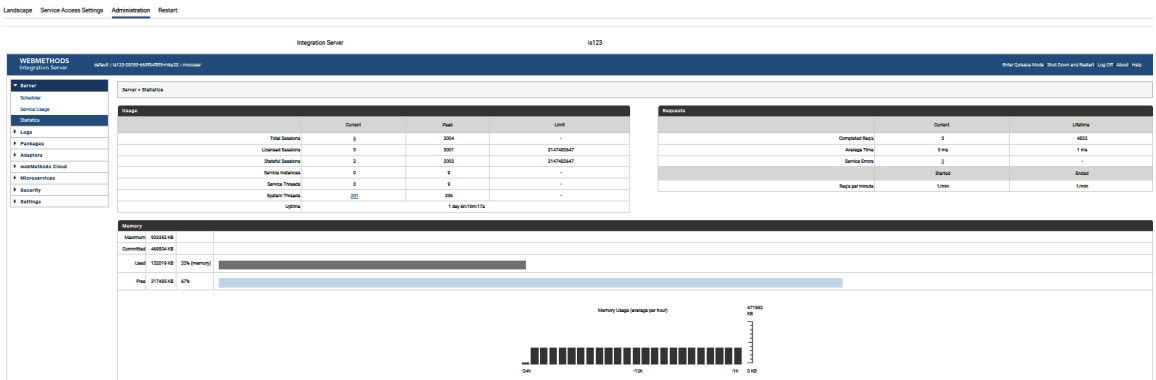
| Field | Description |
| --- | --- |
| | ■ Namespace of the service - *pub.math:addInts*<br><br>**Note:** You can match all services to be allowed or denied by typing * at the end of the Service URL. For example, if you have two services, service url1: /invoke/pub.date:formatDate and service url2: /invoke/pub.date:getCurrentDate /invoke/, then instead of typing two entries, you can provide only one entry, service url: /invoke/pub.date.*. All services matching pub.date will be allowed or denied. |

## Administration

Use this page to manage a solution webMethods Integration Server Administrator instance.

The webMethods Integration Server Administrator is an HTML-based utility you use to administer the webMethods Integration Server. It allows you to monitor server activity, manage user accounts, make performance adjustments, and set operating parameters. You can run the webMethods Integration Server from any browser-equipped workstation on your network. When you click **Administration**, your browser displays the **Statistics** screen.

### The webMethods Integration Server Administrator Screen



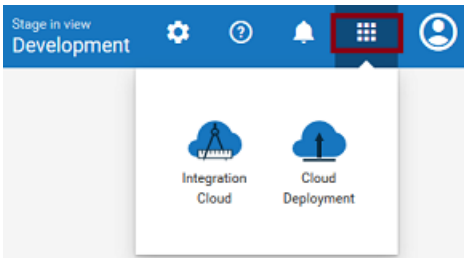

The Title bar displays the name of the host machine where webMethods Integration Server is running, the name of the webMethods Integration Server instance, and the name of the user currently logged into the webMethods Integration Server instance .

The Navigation panel on the left side of the page displays the names of menus from which you can select a task. To start a task, click a subject in the Navigation panel. The server displays a screen that corresponds to the task you select.

**Note:** Click **Help** to view the Help system, which provides information about the features and functionality of webMethods Integration Server.

**To view the Administration page**

1. Switch to the **Cloud Deployment** perspective.




2. From the **Cloud Deployment** navigation bar, click **Solutions > Solution List > Select a solution > Manage > Administration**.

   The webMethods Integration Server Administrator page appears.

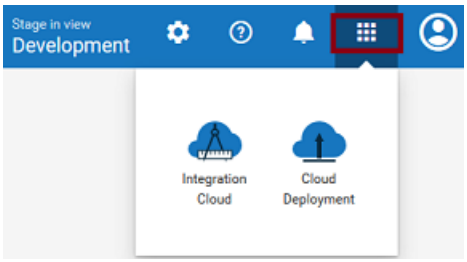
## Restart

Restart the server when you need to stop and reload the server. You should restart the server when:

- You make certain configuration changes. Some configuration changes require the server to be restarted before they take effect.
- If you encounter an operational problem or the server is in an inconsistent state.

**To restart the server**

1. Switch to the **Cloud Deployment** perspective.




2. From the **Cloud Deployment** navigation bar, click **Solutions > Solution List > Select a solution > Manage > Restart**.

   **Note:** webMethods Integration Servers running in a clustered mode cannot be restarted. Further, restarting servers will terminate all active sessions.

3. Click **Restart**.

# Monitoring Solutions

The Monitoring part of **Cloud Deployment** enables you to monitor the health and availability of the solutions and run-time instances, alerts and alert statuses. You receive an email whenever there is a condition that might affect the solution.

The monitoring of a new solution starts automatically 10 minutes after the creation of the solution. The data of the solution is collected and analyzed every 60 seconds.

You can access the following monitoring pages from the left-side navigation menu of the Monitoring main page:


- "Dashboard" on page 100
- "Solutions" on page 101
- "Runtimes" on page 102
- "Services " on page 105
- "Uptime" on page 106
- "Alerts" on page 106
- "Logs" on page 111


You can filter the information on most Monitoring pages based on time. To specify the time-range, select a value in the time-range selector.

The following table describes the options in the time-range selector.

| Option | Description |
|---|---|
| 12h | Displays the information for the last 12 hours. |
| 24h | Default. Displays the information for the last 24 hours. |
| 2d | Displays the information for the last 2 days. |
| 1w | Displays the information for the last week. |
| 2w | Displays the information for the last 2 weeks. |
| 3w | Displays the information for the last 3 weeks. |
| 4w | Displays the information for the last 4 weeks. |

To navigate to the Monitoring main page, log in to Integration Cloud, switch to the **Cloud Deployment** perspective, and select **Monitoring** in the Cloud Deployment navigation bar.

## Dashboard

On the Dashboard page, you can view:

- The health of the solutions
- The number of the alerts that have been raised for all the solutions
- The landscape view of the solutions, and the number of alerts for all run-times that are part of the solutions

The following table provides more information about the panes on the Dashboard page.

| Pane | Description |
|---|---|
| Overall KPI Status | Shows the following information about the health of the solutions for the selected time range:<br>■ Total number of solutions<br>■ The number of healthy solutions<br>■ The number of unhealthy solutions<br>■ The health of the solutions, as a percentage value calculated by the formula (Number of healthy solutions / total number of solutions) * 100<br>A healthy solution is a solution without any open critical alerts.<br>An unhealthy solution is a solution which has at least one open critical alert.<br>To see more information about the KPI status of the solutions, click **More Details**. |
| Alerts | Shows the total number of open and resolved alerts that have been raised for all solutions for the selected time-range, and the number of alerts from each type: critical, warning, or information.<br>To see more details about the alerts, click **More Details**.<br>For more information about the alert types, see "Alert Types" on page 108. |

| Pane | Description |
| --- | --- |
| Landscapes | Displays the topology of the solutions and the number of alerts for each run-time type since the solution has been activated. |

By default, the page displays information for the last 24 hours. To view the information for a different time period, use the time-range selector. For more information about the time-range selector, see "Monitoring Solutions" on page 99.

## Solutions

On the Solutions page, you can check the health of the run-time instances from all the solutions. For each run-time instance, you can view the current data, and the data for the last 24 hours.

The health metrics are grouped into three categories:

- Memory - indicate the memory utilization of a run-time.
- Uptime - indicate the availability of a run-time.
- Failures- indicate failures of the run-time

The following table describes the icons on the Solutions page.

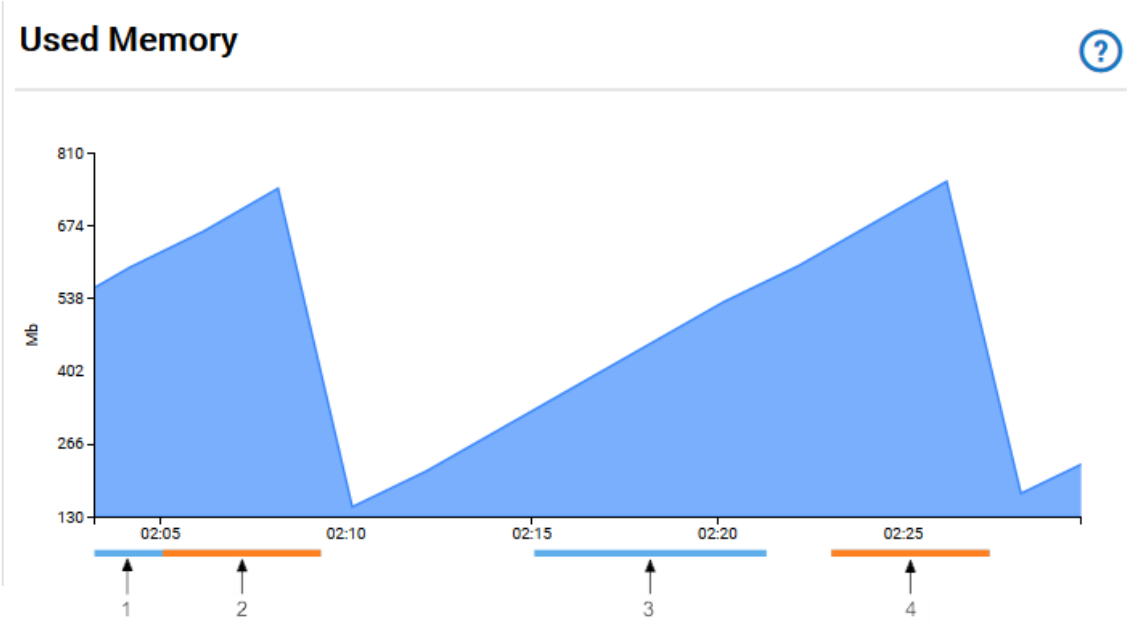
| Icon | Description |
| --- | --- |
|  | Normal health of the run-time instance. |
|  | The health of the run-time instance is deteriorating. Take preventive measures. |
|  | There are critical issues with the health oh the run-time instance. Your urgent attention is needed. |
|  | The run-time instance is not available. |

**Note:** If the solution uses an Integration Server cluster, the number of Integration Server instances is indicated in brackets after the Integration Server instance name.

To view more details about a run-time instance on the Runtime page, click the name of the run-time instance.

## Runtimes

On the Runtimes page, you can view the graphs for monitored KPIs for the selected run-time instances from all the solutions.




The example image shows the graph for the Used Memory KPI. The horizontal lines below the graph represent the severity and duration of the alerts that were raised for the KPI. The information alerts are displayed in blue, the warning alerts are in orange, and the critical alerts are in red.

The following table describes the meaning of the alert lines from the example graph for the Used Memory KPI.

| Time Period | Details |
|---|---|
| 1 | Until 2:05 h, there had been an open information alert. |
| 2 | At 2:05 h, the severity of the information alert was changed to warning. |
| 3 | An information alert existed during that period. |
| 4 | A warning alert existed during that period. |

You can change the value in the **Solutions** drop-down field to load the information about the run-time instances from a specific solution.

You can use the **INTEGRATION SERVER**, **UNIVERSAL MESSAGING**, and **TERRACOTTA** tabs to view the information related to the selected solution and runtime.

By default, the page displays information for the last 24 hours. To view the information for a different time period, use the time-range selector. For more information about the time-range selector, see "Monitoring Solutions" on page 99.

The following table describes the monitored Integration Server KPIs.

| Name | Description |
|---|---|
| Used Memory | The total used memory for the Java VM. |
| Service Threads | The number of active service threads. |
| Sessions | The number of active licensed sessions. |
| Stateful Sessions | The number of the current stateful HTTP sessions. |

The following table describes the monitored Universal Messaging KPIs.

| Name | Description |
|---|---|
| Free Memory | The amount of free memory that the Realm Server has within the Java VM. This indicates the difference between what the Java VM has currently allocated and what the Realm Server has used. |
| Published Events | Total number of events published on this realm from the time it started. |
| Subscribed Events | Total number of events that this realm has sent to clients from the time it started. |

The following table describes the monitored Terracotta KPIs.

| Name | Description |
|---|---|
| Off-Heap Used Memory | Shows the amount of off-heap memory that is currently used. |
| Live Objects | Shows the total number of live objects in the cluster, mirror group, server, or clients. If the trend for the total number of live objects goes up continuously, clients in the cluster will eventually run out of memory |

| Name | Description |
| --- | --- |
| | and applications might fail. Upward trends indicate a problem with application logic, garbage collection, or the tuning of one or more clients. |

## Viewing Adapter KPIs

On the Runtimes page, you can view the KPIs for the adapters that are installed on the Integration Server instances.

1. Navigate to the Runtimes page.
2. Select a solution.
3. On the **INTEGRATION SERVER** tab, select an Integration Server instance.
4. Click **Connectivity KPIs**.
5. On the ADAPTERS tab, select an Adapter.

   The Adapter KPIs are displayed.

   The following table describes the monitored Adapter KPIs.

| Name | Description |
| --- | --- |
| Connections | The number of connection pools in the adapter and how many of them are currently enabled. |
| Notifications | The number of adapter notifications (polling notifications) and how many of them are currently enabled. |

**Note:** You can view Adapter KPIs only for the current time.

## Viewing Connector KPIs

On the Runtimes page, you can view the KPIs for the connectors that are installed on the Integration Server instances.

1. Navigate to the Runtimes page.
2. Select a solution.
3. On the **INTEGRATION SERVER** tab, select an Integration Server instance.
4. Click **Connectivity KPIs**.
5. Click the **CONNECTORS** tab.

6. Select a provider.
7. Select a connector.

   The Connector KPIs are displayed.

   The following table describes the monitored Connector KPIs.

| Name | Description |
|---|---|
| Connections | The number of connection pools in the connector and how many of them are currently enabled. |
| Listeners | The number of connector listeners and how many of them are currently enabled. |

**Note:** You can view Connector KPIs only for the current time.

## Services

On the Services page, you can view the number of successful and failed service executions of the Integration Server instances from the solutions.

The Services page consists of two the Service Executions pane and the History pane.

| Pane | Description |
|---|---|
| Service Executions | Shows the following information about the service executions of the Integration Server instances for the selected time range:<br>■ Total number of service executions<br>■ The number of successful service executions<br>■ The number of failed service executions<br>■ The successful service execution, as a percentage value calculated by the formula (Number of successful service executions / total number of service executions) * 100 |
| History | Shows a chart with the history of successful (green) and failed (red) service executions.<br>Hovering over the green and red bars displays the total number of successful and failed service executions, correspondingly. |

The numbers of service executions on the Services page includes the public and internal services of the Integration Server instance and their child services.

You can change the value in the **Solutions** drop-down field to view the information about a specific solution, or the information for all solutions.

By default, the page displays information for the last 24 hours. To view the information for a different time period, use the time-range selector. For more information about the time-range selector, see "Monitoring Solutions" on page 99.

## Uptime

On the Uptime page, you can view time lines that represent the availability of all run-time instances of the solutions.

The color of the time lines changes based on the status of the run-time instances.

The following table describes the meaning of the different colors.

| Time line color | Indicates that |
|---|---|
| green | the run-time instance was available during the indicated time period. |
| red | the run-time instance was unavailable during the indicated time period. |
| grey | the run-time instance did not exist during the indicated time period. |
| blue | at least one node from the cluster is unavailable. |

**Note:** If the solution uses an Integration Server cluster, the number of Integration Server instances is indicated in brackets after the Integration Server instance name.

By default, the time line displays the availability of the instances during the last 24 hours. To view the information for a different time period, use the time-range selector. For more information about the time-range selector, see "Monitoring Solutions" on page 99.

## Alerts

The alert is a notification that a rule is violated.

On the Alerts page you can:

- View the number of critical, warning, and information alerts for all the solutions for the selected time range
- Filter the alerts by solution, runtime, severity, and status
- Configure the rules by adjusting the alert threshold values
- Configure the summary of the alerts
- Configure the recipient email for the alerts. For more information about configuring the alerts, see "Configuring the Alerts" on page 109.

By default, the Alerts page displays the number of alerts (critical, warning, and information) for all the solutions, and detailed information about the alerts in a tabular format.

**Note:** If the duration of the rule violation is less than the time interval at which the rule is evaluated, the alert does not appear on the Alerts page. For more information about the interval, see "Configuring the Alerts" on page 109.

If you deactivate a solution, the Alerts page will not display the alerts for the solution.

If you activate a solution, the Alerts page will display both the historical alerts for the solutions that had been raised before the deactivation of the solution, and the alerts that were raised after the activation of the solution.

The following table describes the information that is displayed in the table on the Alerts page.

| Column | Description |
|---|---|
| **Solution** | Name of the solution. |
| **Runtime** | Run-time type.<br>■ Integration Server<br>■ Universal Messaging<br>■ Terracotta |
| **Instance** | Name of the run-time instance. |
| **Start Date** | Date and time when the alert was raised. |
| **Resolved On** | Date and time when the alert was resolved. The field is empty if the alert is still active. |
| **Message** | Description of the alert. |

| Column | Description |
|---|---|
| **Status** | Status of the alert.<br>■ The alert is inactive.<br>■ The alert is active. |

**Note:** The Alerts page might not display the alerts for all nodes from a cluster. For example, if you monitor an Integration Server cluster with two Integration Server instances, and both instances have alerts for the same property with different severity, the Alerts page will show the alert of lower severity only, as explained in the following table.

| Integration Server instance | Alert type | Visibility on the Alerts page |
|---|---|---|
| Integration Server instance 1 | Information. Free memory is low. | Yes |
| Integration Server instance 2 | Warning. Free memory is low. | No |

You can view all alerts for all the nodes from the cluster in the email alerts.

By default, the page displays information for the last 24 hours. To view the information for a different time period, use the time-range selector. For more information about the time-range selector, see "Monitoring Solutions" on page 99.

## Alert Types

The following table provides more information about the alert types.

**Note:** Warning alerts and information alerts are not available for KPIs that monitor the availability of a run-time instance.

| Alert Severity | Description | Color Coding |
|---|---|---|
| Critical | A condition exists that is critical for the system performance. | red |
| Warning | A condition exists that might deteriorate the system performance. | orange |

| Alert Severity | Description | Color Coding |
|---|---|---|
| Information | A condition exists that might evolve into a warning or critical alert. | blue |

## Configuring the Alerts

You can change the default threshold values and the recipient email for the system alerts. Threshold values determine when a rule is violated and when the system raises an alert.

**To configure the system alerts**

1. Navigate to the Alerts page.
2. Select the **CONFIGURATION** tab.

   The Configuration page shows information about the alerts for all solutions. The following table describes the columns in the form.

| Column | Description |
|---|---|
| **Name** | Alert Name. |
| **Runtime** | Integration Server, Universal Messaging, or Terracotta. |
| **Action** | The icon activates the configuration view for the alert. |

3. Click the **Edit this rule** icon in the Action column for the alert that you want to configure.

   A form with the configuration details for the alert rule is displayed. The following table describes the fields in the form.

| Field | Description |
|---|---|
| **Threshold** | The KPI's boundary values. When the value of the KPI is outside the range that is specified by these boundary values, the alert is raised.<br>You can configure the threshold values of critical alerts by adjusting the ends of the red line.<br>You can configure the threshold values of warning alerts by adjusting the ends of the orange line.<br>You can configure the threshold values of information alerts by adjusting the ends of the blue line. |

| Field | Description |
|---|---|
| | **Note:** The Threshold field is read-only for KPIs that monitor the availability of a runtime instance. |
| **Runtime** | Integration Server, Universal Messaging, or Terracotta. |
| **Summary** | Summary of the alert. |
| **Interval** | The scrape interval. The scrape interval is the frequency at which the system collects the data. The scrape interval is 60 seconds for all rules. Read-only.<br>**Note:** The alert does not appear immediately when the corresponding rule violation occurs.<br>The time delay from the actual time of the rule violation to the system alert is the following:<br>■ up to 70 seconds for run-time availability rules of critical severity<br>■ up to 420 seconds for run-time availability rules of information severity<br>■ up to 180 seconds for the rest of the rules<br>The system will not send an alert if the rule violation condition is resolved during the corresponding delay period. |
| **Email on alert** | Email of the user who will receive the alerts.<br>**Note:** The email is used for all rules. In case there are alerts, the system sends emails once every 10 minutes.<br>To configure more than one email, use comma-separated values. |

4. In the **Threshold** field, change the default threshold value(s) for the alert.
5. In the **Email on Alert** field, type the email(s) of the user(s) who will receive the email alerts for all rules.

   **Note:** webMethods Integration Cloud stores the email(s) in the local alertManager.yaml file. When you uninstall webMethods Integration Cloud or delete a tenant, the related information is deleted automatically.

6. Click **Apply**.

## Logs

On the Logs page, you can view the logs of the run-time instances in the solutions for a selected time period.

To view the logs for all the instances from a specific solution, select the solution in the solutions drop-down list.

To view the logs for a specific run-time instance in a solution, select the run-time instance in the run-time instance drop-down list.

By default, the page displays the logs for the last 24 hours. To view the logs for a different time period, use the time-range selector. For more information about the time-range selector, see "Monitoring Solutions" on page 99.

You can view logs message statistics and logs details, such as log timestamp and message text. You can change the type of log details that you see by adding or editing filters as described in the Kibana documentation.

# Database

For optimal performance, you can add a MySQL database to your cloud deployment subscription. This enables you to configure, store, and monitor your database directly in the cloud instead of using external systems. The database endpoint can be shared by multiple solutions deployed by the tenant.

## Creating a Database

**To create a database instance in the cloud**

1. On the home page in Cloud Deployment, under **Database**, click **Learn more**, and then click **Start database setup**.
2. On the Database setup details screen, do the following:
   a. In the **Database instance identifier** field, specify a name for the database.

      **Important:** Each Software AG Cloud tenant can create only one database instance.

   b. In the **Create DB master username** and **DB master password** fields, specify the username and password that you will use to access the database.

      **Note:** The master user has full privileges on the database. For more information on database privileges, see the MySQL documentation.

   c. Confirm the password and click **Continue**.

When the database configuration is complete, you are redirected to the home page of Cloud Deployment where the **Database** element shows that the database server is running.

## Connecting a Solution to the Database

After you create the database instance in the cloud, you can connect one or more solutions to it.

**To connect a solution to the database instance in the cloud**

1. Enable the wMJDBCAdapter package for the solution. For more information, see the Solutions chapter in *Cloud Deployment*.
2. In Software AG Designer, deploy to the cloud the on-premises webMethods Integration Server packages that you want to include in the solution.
   a. In the Package Navigator view, right-click any of the packages that you want to include and select **Deploy to Cloud**.
   b. In the Publish Assets to Integration Cloud dialog box, select all packages that you want to include in the solution and click **Next**.
   c. In the Integration Server Packages Variable Substitution dialog box, do the following for each of the packages that have JDBC connections:
      - Enable the **State after Deployment** property.
      - In the **user**, **password**, and **serverName** fields, specify the username, password, and host name of the database instance in the cloud.

        **Note:** You can see the host name of the database in the **DB instance endpoint** field on the Database page in Cloud Deployment.
   d. In the Select the Integration Cloud Solution dialog box, select the solution that you want to connect to the database and click **Finish**.

## Monitoring the Database

After you create the database instance in the cloud, you can monitor its status on the Database page in Cloud Deployment.

On the **Database** dashboard of the Database page, you can view the following elements:

| Element | Description |
|---|---|
| **DB CPU usage** | CPU usage in percentage. |

| Element | Description |
| --- | --- |
| **DB storage size** | The allocated storage space and the storage space used by the database. |
| **DB server status** | Shows if the database server is running. You can start or stop the server at any time by clicking **Start server** or **Stop server**. |

In the **Configuration details** section of the Database page, you can view the following fields:

| Field | Description |
| --- | --- |
| **DB instance** | The size of the database, based on your license offering. |
| **DB instance identifier** | The name of the database. |
| **DB instance endpoint** | The host and port that you can use to access the database. |
| **Allowed IPs** | The list of external IPs that can access the database.<br><br>Software AG recommends using the database only from applications in Software AG Cloud. By default, you can access the database only in the cloud. For administrative purposes you can enable access to the database from external IPs.<br><br>To add external IPs, click **Edit** and type each IP on a new row. The supported IP format is Classless Inter-Domain Routing (CIDR) block format, xxx.xxx.xxx.xxx/yy. If you specify only xxx.xxx.xxx.xxx, mask /32 is automatically added. |

# Cloud Deployment CLI

The Cloud Deployment CLI interacts with webMethods Cloud Deployment and performs tasks such as managing a solution, monitoring the status of all runtimes in a solution, promoting assets from one stage to another, and so on. The CLI supports the following modes:

- **Interactive Mode**: To start the CLI in interactive mode, run the following command:

```
wmcd-cli --mode interactive
```

```
C:\cli>wmcd-cli --profile default --mode interactive

Reading Credentials from C:\Users\arub\AppData\Roaming\wmcd_cli\config.json
Connecting to https://setup.saglive.com
Signing in as setup
No Commands specified

                                                                                                  CLI v1.0.0

Copyright © 2014-2019 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

Connected to webMethods Cloud Deployment...

wmcd_cli~$ Commands ...
```

- **Normal Mode**: To start the CLI in normal mode, run the following command:

```
wmcd-cli <commands> [options]
```

# Install Cloud Deployment CLI

### Overview

You install the Cloud Deployment CLI (wmcd-cli) using the NPM registry. This section describes tasks such as installing the Cloud Deployment CLI and specifying credentials to connect to Cloud Deployment.

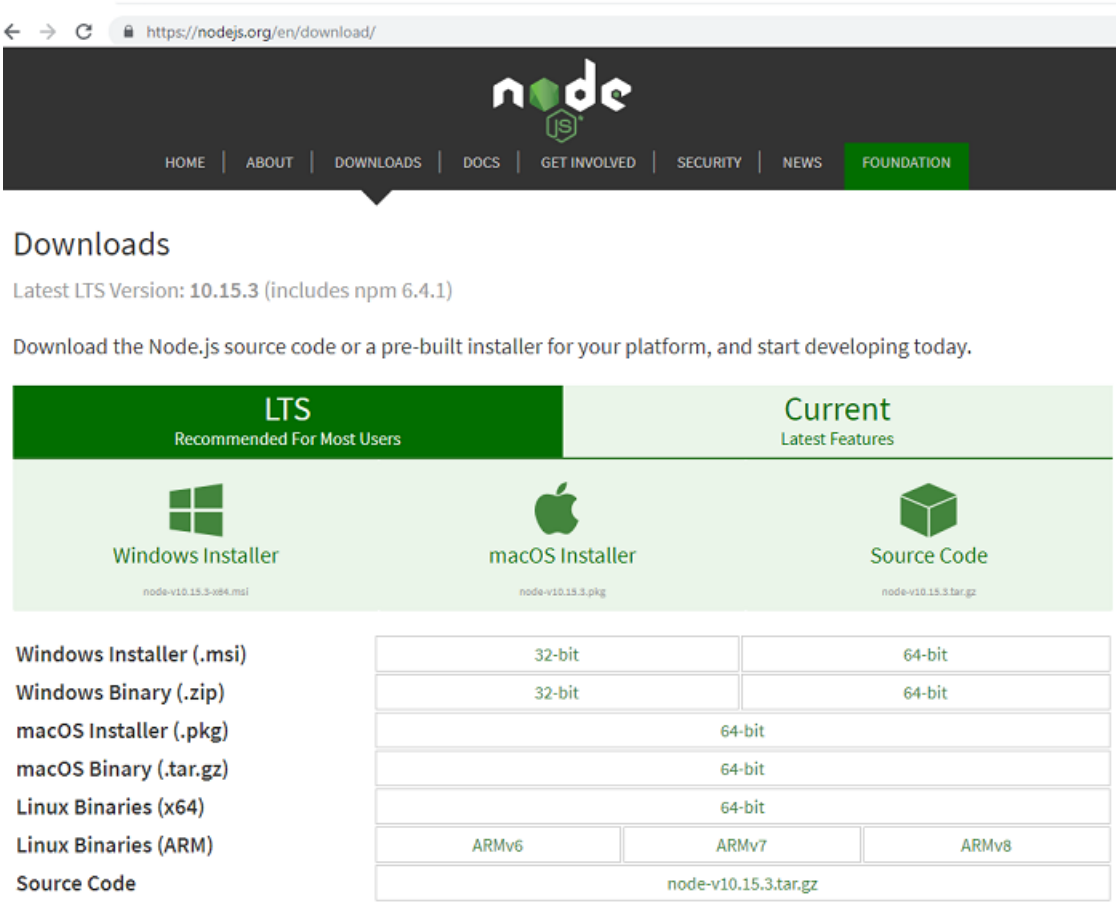
### Actors

Administrators

### Before you begin

Before you install wmcd-cli from the NPM registry, do the following:

- Install the NodeJS runtime environment version 10.13 or later on your computer along with NPM (Node's package manager).




**Note:** The Cloud Deployment CLI is available for the following operating system environments:

- Windows
- Linux
- macOS

- Open a command line interface and type the following command to install wmcd-cli:

```
npm install -g wmcd-cli
```

### Specify the credentials to connect to Cloud Deployment

To connect to Cloud Deployment, you must configure the credentials in one of the following ways:

- Specify the credentials in config.json file.
  - Create a config.json file under %appdata%/wmcd_cli/ location for Windows or /home/wmcd_cli/ in Linux.
  - Add new profiles in the config.json file.

The following is an example of default and referenced profile in the configuration file.

```
{
    "default": {
        "url": "https://{subdomain}.webmethodscloud.com",
        "userName": "userName",
        "password": "password”
    },
    "someotherProfile": {
        "url": "https://{subdomain}.webmethodscloud.com",
        "userName": "userName",
        "password": "password”
    }
}
```

By default, the 'default' settings are read from the configuration file. To enable the referenced profile, run the following command:

```
wmcd-cli -profile someotherProfile
```

- Specify credentials as runtime arguments.
  - When you start the CLI, credentials are passed as a runtime argument. For example:

```
wmcd --mode interactive --url <url> --userName <userName>
--password <password>
```

**Note:** Runtime arguments will have the highest priority. However, If runtime arguments are not passed, then the default profile from %appdata%/wmcdd_cli/config.json is used.

## Cloud Deployment CLI Reference

The following table describes the commands you use to perform various scenarios in the CLI interface.

### Commands for viewing alerts in Cloud Deployment

Run the following command to view alerts.

```
alert list [options]
```

where the options are:

| --Name | Description |
|---|---|
| --solutionName <solutionName> | Filter the alerts that belong to a particular solution. |
| --stageName <stageName> | Filter the alerts that belong to a particular stage. Supported values are development, test, live, prelive. |

| --Name | Description |
|---|---|
| --alertName <alertName> | Filter the alerts by name . |
| --runtime <runtime> | Filter the alerts by Instances / Node name in a solution. |
| --severity <severity> | Filter the alerts based on the severity. Default supported values are info, warning, and critical. |
| --view <view> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`alert list --severity critical` |
| --view <json> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`alert list --severity critical --view json`<br>**Note:** JSON output format loads complete information that contains additional field than the table output format. |

#### Commands for listing the assets in LAR

Run the following command to list all the assets available in the LAR.

```
asset-repo list-assets <solutionName> <nodeName> <stageName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| --view <view> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`asset-repo list-assets DemoSoln IS development` |

#### Commands for creating solutions in Development stage

Run the following command to create solutions in Development stage.

```
solution create [options]
```

where the options are:

| --Name | Description |
|---|---|
| --inputFile <fileName> | Provide the name of the file that contains the input data.<br>`solution create --inputFile /home/etc/createSolution.json`<br>The file must contain a valid solution name, the name of the webMethods Integration Server and Universal Messaging instances, number of CPU Cores, and Memory characteristics of the hardware to support each service in the solution.<br>Example 1<br>`{ "integration": { "landscapeDefinition": { "solutionName": "Demo", "description": "Demo Solution", "solutionType": 1, "productDefinitions": { "IS": [{ "name": "IS", "resources": { "limits": { "cpu": "1", "memory": "2" } }, "version": "10.4", "env": { "packages": ["packages.WmCloudStreams .enabled=true", "packages.WmJDBCAdapter.enabled=true"] } }] } } } }`<br>Example 2<br>`{ "integration": { "landscapeDefinition": { "solutionName": "Solution2", "description": "sample solution", "solutionType": 2, "productDefinitions": { "IS": [{ "name": "IS", "resources": { "limits": { "cpu": "1", "memory": "2" }`<br>`}` |

| --Name | Description |
| --- | --- |
| | |

```
      },
      "version": "10.4",
      "env": {
       "packages": ["packages.WmCloudStreams
.enabled=true",
 "packages.WmJDBCAdapter.enabled=true"]
      }
     }],
     "UNIVERSALMESSAGING": [{
      "name": "UM",
      "resources": {
       "limits": {
        "cpu": "1",
        "memory": "2"
       }
      },
      "version": "10.4"
     }]
    }
   }
  }
}
```

Example 3

```
{
 "integration": {
  "landscapeDefinition": {
   "solutionName": "Solution3",
   "description": "Sample Solution",
   "solutionType": 3,
   "productDefinitions": {
    "IS": [{
     "name": "IS1",
     "resources": {
      "limits": {
       "cpu": "1",
       "memory": "2"
      }
     },
     "version": "10.4",
     "env": {
      "packages": ["packages.WmCloudStreams.enabled=true",
 "packages.WmJDBCAdapter.enabled=true"]
     }
    }, {
     "name": "IS2",
     "resources": {
      "limits": {
       "cpu": "1",
       "memory": "2"
      }
     },
     "version": "10.4",
     "env": {
      "packages": ["packages.WmCloudStreams.enabled=true",
 "packages.WmJDBCAdapter.enabled=true"]
     }
    }],
    "UNIVERSALMESSAGING": [{
     "name": "UM",
```

| --Name | Description |
|---|---|
| | `"resources": { "limits": { "cpu": "1", "memory": "2" } }, "version": "10.4" }] } } } }` |

#### Commands for listing the solutions for a particular stage

Run the following command to list solutions for a particular stage.

```
solution list <stageName> [options]
```

where the valid stage name value includes development, test, prelive, and live.

Example:

```
solution list development --view json
```

#### Commands for getting the solution and runtimes for a particular stage

Run the following command to get the solution and all runtimes for a particular stage.

```
solution get <solutionName><stageName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| --view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`solution get DemoSoln development` |

#### Commands for deleting the solution for a particular stage

Run the following command to delete the solution for a particular stage.

```
solution delete <solutionName> <stageName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| --view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`solution delete DemoSoln development` |

#### Commands for activating or deactivating the solution for a particular stage

Run the following command to activate and deactivate a solution for a particular stage.

```
solution update-status <solutionName> <stageName> <actionName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| --view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`solution update-stats DemoSoln development deactivate`<br>■ Allowed stage names are `development`, `test`, `live`, `prelive`.<br>■ Allowed action names are `activate` or `deactivate` |

#### Commands for getting the status of all pods in a solution

Run the following command to get the status of all pods in a solution.

```
solution get-status <solutionName> <stageName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| --view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`solution get-status DemoSoln development` |

#### Commands for getting the license information for a particular tenant

Run the following command to get the license information for a particular tenant.

```
tenant get-license-info [options]
```

where the options are:

| --Name | Description |
|---|---|
| --view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`tenant get-license-info` |

#### Commands for getting the total CPU and memory utilization details for a particular tenant

Run the following command to get the total CPU and memory utilization details for a particular tenant.

```
tenant get-utilization-details [options]
```

where the options are:

| --Name | Description |
|---|---|
| --view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`tenant get-license-info` |

#### Commands for listing all the users

Run the following command to list all the users.

```
user list [options]
```

where the options are:

| --Name | Description |
|---|---|
| --view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`tenant get-license-info` |

#### Commands to get a particular user

Run the following command to get the user information.

```
user get <userId> [options]
```

where the options are:

| --Name | Description |
|---|---|
| --view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`user get <ID details>` |

#### Commands for promoting Integration Server and Universal Messaging configurations from one stage to another

Run the following command to promote Integration Server and Universal Messaging configurations from one stage to another.

```
runtime promote-configuration <fromSolutionName> <fromNodeName>
<fromStageName> [toSolutioName] [toNodeName] [options]
```

**Note:** If toSolutionName and toNodeName are provided as part of your command, then the CLI performs the cross solution promotion. However, if toSolutionName and toNodeName are not available, then the configuration will be promoted to higher stage on the sameIntegration Server or Universal Messaging node.

where the options are:

| --Name | Description |
|---|---|
| --propFile <fileName> | Properties file to perform variable substitution.<br>Example: Use the following command for promotion across same instance:<br>`runtime promote-configuration DemoSoln IS development`<br>Example: Use the following command for cross solution promotion:<br>`runtime promote-configuration DemoSoln IS development DemoSoln1 IS1`<br>Promotes the configuration from the solution DemoSoln IS node to DemoSoln1 and IS1 node. |

#### Commands for promoting Integration Server packages from one stage to another

Run the following command to promote Integration Server packages from one stage to another.

```
runtime promote-packages <fromSolutionName> <fromNodeName> <fromStageName>
[toSolutioName] [toNodeName] [options]
```

**Note:** If toSolutionName and toNodeName are provided then the CLI performs the cross solution promotion. If toSolutionName and toNodeName are not available, then the packages will be promoted to higher stage on the same Integration Server instance.

where the options are:

| --Name | Description |
|---|---|
| --include <comma separated packages names> | Promotes only specified packages.<br>Example:<br>`--include package1,package2,package3` |
| --exclude <comma separated packages names> | Promotes all the packages other than the packages specified in the options.<br>Example:<br>`--exclude package1,package2,package3` |
| --propFile <fileName> | Provide properties file to perform variable substitution.<br>Examples:<br>Use the following command for promotion across same instance:<br>`runtime promote-packages DemoSoln IS development -- propFile /home/etc/var_sub.properties`<br>Use the following command for cross solution promotion:<br>`runtime promote-packages DemoSoln IS development TestSoln IS2`<br>Use the following command to replace a property in a specific composite asset.<br>`<propertyName>/<compositeAssetName>`<br>**Note:** The replacement properties should be in the following format:<br>`<propertyName>/<compositeAssetName>/<assetName>` |

| --Name | Description |
|---|---|
| | Example for replacing properties of a package:<br>`activatePkgOnInstall/TestODataService=false`<br>where TestODataService is the package name whose property activatePkgOnInstall" is assigned with value "false".<br>Example for replacing properties of a service of a package:<br>`serverName/TestODataService/JDBC_Connection.ODataService=localhost`<br>where `JDBC_Connection.ODataService` is a service under package `TestODataService`, whose parameter `serverName` is assigned with value localhost |

#### Commands for printing the list of exposed services

Run the following command to print the list of exposed services.

```
runtime get-exposed-is-serices <solutionName> <nodeName> <stageName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| -view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`runtime get-exposed-is-services DemoSoln IS development` |

#### Commands for listing all the queues

Run the following command to list all the queues in the Universal Messaging instance.

```
um list-queues <solutionName> <nodeName> <stageName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| -view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format. |

| --Name | Description |
|---|---|
| | Example:<br>`um list-queues DemoSoln UM development` |

#### Commands for getting the queue information

Run the following command to retrieve pushed, popped, and memory usage of the queue.

```
um get-queue <solutionName> <nodeName> <stageName> <queueName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| -view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`um get-queue DemoSoln UM development dummyQueue` |

#### Commands for getting the queue details

Run the following command to retrieve the queue details.

```
um get-queue-details <solutionName> <nodeName> <stageName> <queueName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| -view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`um get-queue-details DemoSoln UM development dummyQueue` |

#### Commands for listing all the channels in the Universal Messaging instance

Run the following command to list all the channels in a Universal Messaging instance.

```
um list-channels <solutionName> <nodeName> <stageName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| -view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`um list-channels DemoSoln UM development` |

#### Commands for getting the channel information

Run the following command to get the channel information.

```
um get-channel <solutionName> <nodeName> <stageName> <channelName> [options]
```

where the options are:

| --Name | Description |
|---|---|
| -view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`um get-channel DemoSoln UM development dummyChannel` |

#### Commands for getting the channel details

Run the following command to get the channel details.

```
um get-channel-details <solutionName> <nodeName> <stageName> <channelName>
 [options]
```

where the options are:

| --Name | Description |
|---|---|
| -view <viewType> | By default, outputs the response in a table view mode. The CLI supports both table and JSON output format.<br>Example:<br>`um get-channel-details DemoSoln UM development dummyChannel` |