Configuring On-Premise Integration Servers for webMethods Cloud

Version 10.15

October 2022

WEBMETHODS

Document ID: IS-OP-1015-20231222

Table of Contents

About this Guide

This guide describes how to configure Integration Server as an on-premise server for use with webMethods Cloud. It contains information for administrators who configure and manage on-premise Integration Servers and for application developers who want to create services that will be accessed through webMethods Cloud.

webMethods Cloud is a generic term used in this document to represent webMethods.io Integration and webMethods Integration Cloud.

To use this guide effectively, you should understand the basic concepts described in the *webMethods Integration Server Administrator's Guide*. You should also be familiar with all the services you want to share with webMethods Cloud.

Important:
webMethods Integration Cloud is deprecated as of December 31, 2022 and will be discontinued as of December 31, 2023. webMethods Integration Cloud has been superseded by webMethods.io Integration, which offers a number of improvements and new capabilities reaching significantly beyond the capabilities of webMethods Integration Cloud, including code, low-code and no-code approaches to building integrations and automation.

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: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	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

Product Documentation

You can find the product documentation on our documentation website at https://documentation.softwareag.com.

In addition, you can also access the cloud product documentation via https://www.softwareag.cloud. Navigate to the desired product and then, depending on your solution, go to "Developer Center", "User Center" or "Documentation".

Product Training

You can find helpful product training material on our Learning Portal at https://learn.softwareag.com.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at https://techcommunity.softwareag.com. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at https://github.com/softwareag and https://containers.softwareag.com/products and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at https://empower.softwareag.com. Many services on this portal require that you have an account. If you do not yet have one, you can request it at https://empower.softwareag.com/register. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.

- Open and update support incidents.
- Add product feature requests.

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 About Configuring an On-Premise Integration Server

What Is an On-Premise Integration Server?

An *on-premise* Integration Server is any Integration Server that is configured to share service metadata with webMethods Cloud and enable execution of those services by integrations defined in webMethods Cloud. The service metadata uploaded from an on-premise Integration Server provides the accounts and applications required to create integrations on webMethods Cloud.

About Accounts, Stages, and Environments

webMethods Integration Cloud

An *account* on the on-premise Integration Server is a connection that you create to enable the communication of data from the on-premise Integration Server to Integration Cloud and vice versa. Once configured the on-premise Integration Server listens for requests from a Integration Cloud tenants. A single Integration Server can listen to requests from multiple tenants.

When the on-premise Integration Server receives requests, it processes the requests using the account configured to process requests on Integration Cloud.

A *stage* is an environment configured on Integration Cloud that represents a specific point in the development cycle of an integration service. When you configure a connection to an Integration Cloud server, the on-premise Integration Server fetches the stages you defined on the Integration Cloud server. When you configure the account, you define the stage on which the on-premise Integration Server listens for requests. For more information about configuring a connection to a Integration Cloud server, see "Configuring a Tenant Connection" on page 13.

For example, the Integration Cloud server might have one environment for development (Default), one for testing (Test), and another for production (Live). You would configure each on-premise account to listen for requests for a specific stage. So, if you develop an integration on the Default stage and invoke a service on the on-premise Integration Server, the account listening for requests on the Default stage receives and processes the request.

For more information about defining stages in Integration Cloud, see the *webMethods Integration Cloud Help*.

Important:
webMethods Integration Cloud is deprecated as of December 31, 2022 and will be discontinued as of December 31, 2023. webMethods Integration Cloud has been superseded by webMethods.io Integration, which offers a number of improvements and new capabilities reaching significantly beyond the capabilities of webMethods Integration Cloud, including code, low-code and no-code approaches to building integrations and automation.

webMethods.io Integration

webMethods.io Integration has environments and each environment has a different endpoint. You can configure one on-premise Integration Server to an environment. Click here for information on environments in webMethods.io Integration.

About Applications

A group of services that you share with webMethods Cloud is called an *application*. Applications are created on the on-premise Integration Server and uploaded to webMethods Cloud. webMethods Cloud can execute any service hosted by an on-premise Integration Server. When you share services in an application, you are sharing the *metadata* for the service. For Integration Server services, the metadata you share is the service name, service signature, display name, and service comments. Users of webMethods Cloud can then create integrations that invoke services defined in the applications.

When webMethods Cloud executes a service, the on-premise Integration Server returns all the results to webMethods Cloud. You can batch the results to limit the number of results returned to webMethods Cloud. For more information, see the **Batch Data** field in "Defining Applications" on page 34.

After you create applications, you upload them to webMethods Cloud. If the application changes on the on-premise Integration Server, you must upload the application again for the changes to be replicated to webMethods Cloud.

When you upload applications to the webMethods Cloud server, you associate one or more accounts that the application can use to access services on the on-premise Integration Server. If the account associated with an application changes, you can upload the account to the webMethods Cloud server without having to upload the application. For more information about uploading accounts, see "Uploading Accounts to webMethods Cloud " on page 26.

Deploying Assets

You can deploy settings, accounts, and applications as assets using webMethods Deployer. For more information, see *webMethods Deployer User's Guide*.

Steps to Configure an On-Premise Integration Server

The following table describes the process for using Integration Server Administrator to configure an on-premise Integration Server.

Step	Description
Step 1	Create a user account on the webMethods Cloud server.
Step 2	Configure one or more tenants for the on-premise Integration Server to use on the webMethods Cloud server. See "Configuring a Tenant Connection" on page 13.
Step 3	Create the accounts by which webMethods Cloud can connect to the on-premise Integration Server. See "Configuring Accounts for webMethods Cloud " on page 21.
Step 4	Define and upload the applications to share through webMethods Cloud. See "Managing Applications" on page 31.

Step	Description
Step 5	Define on-premise Integration Server services that can be invoked in webMethods Cloud running the on-premise Integration Server inside a Docker container. See "Managing Docker Services" on page 39.

Using Multiple On-Premise Integration Servers to Process Requests from webMethods Cloud

Multiple on-premise Integration Servers can receive and process requests from webMethods Cloud. The on-premise Integration Servers can be in a clustered environment (stateful or stateless) or in a non-clustered environment. When multiple Integration Servers are configured to receive and process requests from webMethods Cloud, requests are distributed to each Integration Server in a round-robin fashion.

To use multiple on-premise Integration Servers to process requests from webMethods Cloud, the following conditions must be true for all the on-premise Integration Servers:

- Configuration for webMethods Cloud tenants, accounts, and applications are the same across all the Integration Servers.
- The packages containing services to be called from webMethods Cloud are the same across all Integration Servers. If you change a package on one of the on-premise Integration Servers, you must propagate the changes to the other on-premise Integration Servers.
- Accounts and applications must be uploaded once from one of the Integration Servers. That is, you do not need to upload the accounts and applications from every Integration Server.
- Accounts must be enabled on all Integration Servers to receive and process incoming requests from webMethods Cloud.

2 Configuring a Tenant Connection

About Tenant Connections

In hybrid integration, an on-premise Integration Server uses a tenant connection to share account and application information with webMethods Cloud. Specifically, the tenant connection enables communication between the on-premise Integration Server and the tenant in webMethods Cloud.

A tenant alias contains the location and login details needed for the on-premise Integration Server to establish a connection with webMethods Cloud. An on-premise Integration Server can have multiple tenant aliases.

Each account on an on-premise Integration Server is associated with one tenant alias. A single tenant alias can be associated with multiple accounts.

An on-premise Integration Server contains a pre-defined tenant alias named "default". The default tenant alias can be edited to specify different credentials and a different URL for the webMethods Cloud server. However, the default tenant alias cannot be deleted.

Note:
Prior to Integration Server 10.11 or application of a fix or release that included PIE-68485, an on-premise Integration Server could have only one tenant alias. Tenant credentials specified on the **webMethods Cloud > Tenant connections** page prior to application of a fix that contains PIE-68485 or migrating to a release that includes PIE-68485 are preserved in the default tenant.

Creating a Tenant Alias

The tenant alias specifies the credentials used to listen for requests and the URL for the webMethods Cloud with which the on-premise Integration Server shares applications and accounts.

To create a tenant alias

1. Open Integration Server Administrator.
2. Go to **webMethods Cloud > Tenant connections**.
3. Click **Create Tenant Connection**.
4. Under **Settings**, complete the fields as follows:

Field	Description
Enable	Click **Yes** to enable the tenant once it is created.
User Name	User name for an account on webMethods Cloud.
Password	Password identified in the user account for **User Name**.

Field	Description
webMethods Cloud URL	The URL of webMethods Cloud with which to share accounts and applications created on the on-premise Integration Server. For example, the URL would be of the following format: `https://<sub-domain>.<domain-name>` For example, `https://softwareag-education.webmethodscloud.com`, or `https://sample.prod-int-aws-us.webmethods.io` **Note:** To set up a two-way SSL communication, add port 8443 in the URL. For example, `https://<sub-domain>.<domain-name>:8443`.

5. Under **Certificate Settings (optional)**, complete the fields if you want to set up a two-way SSL communication with webMethods Cloud. Before you begin, see "Setting Two-Way SSL Communication" on page 17.

   If you do not configure these settings, Integration Server uses one-way SSL communication with webMethods Cloud.

Field	Description
Keystore Alias	A user-specified, text identifier for the Integration Server keystore. The alias for the keystore that contains the client certificates that you want Integration Server to use when connecting to webMethods Cloud.
Key Alias	The alias for the private key, which must be stored in the keystore specified by the above keystore alias.
Truststore Alias	The alias for the truststore. The truststore must contain the trusted root certificate for the CA that signed Integration Server certificate associated with the key alias. The truststore also contains the list of CA certificates that Integration Server uses to validate the trust relationship.

6. Click **Update Settings**.

   Integration Server connects to webMethods Cloud specified in the **webMethods Cloud URL** and downloads the configuration information that is required to receive any incoming requests.

Changing the Tenant Password

When you change the tenant password, you must update the tenant password in webMethods Cloud and in the on-premise Integration Server. If you update the tenant password on webMethods Cloud, you must update the tenant information in the on-premise Integration Server that connects to webMethods Cloud. For information about how to change the tenant password with minimal

disruption to FlowServices running on webMethods Cloud, refer to the webMethods.io Integration documentation.

You can update the tenant password on the on-premise Integration Server when it is not connected to the tenant on webMethods Cloud. When you enable the tenant, Integration Server attempts to establish the tenant connection with the new password.

Note:
After changing the tenant password on the on-premise Integration Server, you must disable and then re-enable the cloud accounts for the affected on-premises servers.

Editing a Tenant Alias

You can edit the tenant details to specify a different webMethods Cloud server or change the credentials. For more information about changing the tenant password, see "Changing the Tenant Password" on page 15

To edit a tenant alias

1. Open the Integration Server Administrator.
2. Go to **webMethods Cloud > Tenant connections**.
3. Click the name of the tenant you want to edit settings.
4. Edit the tenant configuration.
5. Click **Update Settings**.
6. If you changed the webMethods Cloud URL, disable and then re-enable the accounts that use the tenant alias to ensure that information is sent to and received from the new destination. For more information about enabling and disabling accounts, see "Enabling and Disabling Accounts" on page 27.

Deleting a Tenant Alias

Before deleting a tenant alias, keep the following in mind:

- The pre-defined default tenant alias cannot be deleted.
- A tenant alias associated with an account cannot be deleted.

To delete a tenant alias

1. Open the Integration Server Administrator.
2. Go to **webMethods Cloud > Tenant connections**.

3. In the row for the tenant you want to delete, click ✕.

Enabling or Disabling a Tenant

A tenant alias must be enabled to upload account and application information to webMethods Cloud.

At times, Integration Server disables a tenant alias automatically. It will need to be enabled after resolution of the problem that caused the tenant alias to be disabled. For example, when an on-premise Integration Server receives an authorization failure when attempting to connect to webMethods Cloud, the on-premise Integration Server does not make repeated attempts to connect to the webMethods Cloud. Instead, Integration Server disables the tenant alias. This prevents the on-premise Integration Server and other clients from being locked out of the tenant.

Important:
Disabling a tenant alias does not disable the accounts associated with the alias. An enabled account can still process and respond to service execution requests from webMethods Cloud even if the associated tenant alias is disabled. To stop an account from processing requests from webMethods Cloud, disable the account.

To enable or disable a tenant alias

1. Open the Integration Server Administrator.

2. Go to **webMethods Cloud > Tenant connections**.

3. In the row for the tenant you want to enable or disable, click the text in the **Enabled** column.

Setting Two-Way SSL Communication

Integration Server supports two-way SSL communication between the on-premise Integration Server and webMethods Cloud. Integration Server, by default, supports one-way SSL communication in which the on-premise Integration Server acts as a client and validates the certificate issued by webMethods Cloud that acts as a server.

In two-way SSL communication, both the on-premise Integration Server and webMethods Cloud validate each other's certificate using private keys. If you want more secure communication between two business applications, you can set up two-way SSL communication.

Before you set up a two-way SSL communication, you need to download the webMethods Cloud signed certificate and generate a keystore file. Then, use the keystore file to generate a keystore alias on the on-premise Integration Server. When you set up a connection to webMethods Cloud, you need to use these keystore details so that webMethods Cloud can validate the identity of Integration Server.

Here are the high-level steps to set up two-way SSL communication:

To generate a keystore alias using the webMethods Cloud certificate

1. Go to the webMethods Cloud **Certificates** page and download the webMethods signed certificate file in JKS or p12 format, which contains the private key and the certificate. You can also upload your own CA signed certificate. Integration Server does not support self signed certificates.

   Note:
   You can either directly generate the JKS file or use JKS tools or utilities to generate the JKS file.

2. Add the JKS file in the **Security > Keystore** page in Integration Server Administrator and specify the keystore properties in the **Security > Keystore > Create Keystore Alias** page.

3. Go to **webMethods Cloud > Tenant connections** page in Integration Server Administrator and specify the details. See step 5 under Creating Tenants.

   For detailed information on how two-way SSL communication works, see the documentation of the respective webMethods Cloud products.

Setting Maximum Thread Pool Usage for Cloud Requests

When an application developed on webMethods Cloud invokes a service on an on-premise Integration Server, the on-premise Integration Server uses a thread from the server thread pool to process the service request. The processing thread invokes the service, posts the response, and then terminates, which frees up the thread to process other requests. The on-premise Integration Server can process multiple service requests from webMethods Cloud simultaneously.

The watt.server.threadPool.cloudRequests specifies the maximum percentage of the server thread pool that can be used for processing webMethods Cloud requests. The default is 5 which means that if the maximum size of the server thread pool is 1000, up to 50 server threads at a time can be used to process cloud requests.

At start up, Integration Server uses the specified percentage to calculate the number of threads that can be used and logs a message with that information to the server.log. At run-time, when Integration Server reaches the maximum number of threads that can be used to process requests from webMethods Cloud, Integration Server writes the following message and blocks new server requests from webMethods Cloud until a thread becomes available.

```
[ISS.0021.8040W] Server thread pool is using the maximum number of threads allowed for
processing requests from the webMethods Cloud. Requests from webMethods Cloud will block
until a thread becomes available.
```

Note:
If watt.server.threadPool.cloudRequests is set to 0, the on-premise Integration Server processes requests from webMethods Cloudone at a time. The on-premise Integration Server blocks processing of further requests from webMethods Cloud until the current service execution completes. This delays processing of requests from webMethods Cloud.

Configuring Retry Functionality for Sending Responses

Integration Server uses a transactional publish when sending responses to webMethods Cloud. The transactional publish includes a built-in mechanism to recover from intermittent network errors, which can help mitigate the impacts of timeout errors and connection leaks. Integration Server automatically retries any failed transactional publish.

Use the following server configuration parameters to configure the retry functionality for publishing responses to webMethods Cloud.

- **watt.wmcloud.sendResponse.retryCount** Specifies the maximum number of retry attempts that an on-premise Integration Server makes when a failure occurs when sending a response to webMethods Cloud. The default value is 3. A value of 0 indicates that Integration Server will not attempt any retries. Integration Serverdoes not need to be restarted for changes to this parameter to take effect.
- **watt.wmcloud.sendResponse.retryDelay** Specifies the number of milliseconds that an on-premise Integration Server waits before making another retry attempt after a publish attempt fails. The default value is 1000 (i.e., 1 second). Integration Serverdoes not need to be restarted for changes to this parameter to take effect.

If the maximum retry attempts have been made and publishing a response to the webMethods Cloud still fails, the on-premise Integration Server writes an error to the server log.

Note:
Setting the sever log facility 0021 Integration Cloud to the Info level or more verbose level may assist with debugging any publish failures.

Monitoring webMethods Cloud Tenants

The `wm.client.integrationlive.admin:scheduleTenantMonitoringTask` task runs in Integration Server at the specified interval of 10 minutes to monitor the tenant connections. If a tenant connection fails, Integration Server notifies the user through a notification in Integration Server Administrator followed by an email. Configure the hybrid activity notifications using the `watt.wmcloud.hybridConnectivityAlert.notifications` server configuration parameter and the recipients of the notification emails using the `watt.wmcloud.hybridConnectivityAlert.mail` server configuration parameter. Integration Server uses the email notification settings configured in **Integration Server Administrator > Settings > Resources**. For more information, see the *webMethods Integration Server Administrator's Guide*.

Note:
The hybrid connectivity alerts are introduced as part of PIE-81176 in IS_10.15_Core_Fix4.

3 Configuring Accounts for webMethods Cloud

Overview

When you create an account on the on-premise Integration Server, you specify the connection parameters that the on-premise Integration Server uses to access webMethods Cloud. For each account, you specify the a tenant alias that communicates with the on-premise Integration Server. Each account can be associated with a different tenant alias.

You create accounts on the on-premise Integration Server to allow them to serve any requests that originate from webMethods Cloud. If an account is disabled on the on-premise Integration Server, any requests sent from webMethods Cloud will time out depending on the amount of time specified in the **Request Timeout** field. For more information about the **Request Timeout** field, see "Creating Accounts on an On-Premise Integration Server " on page 23.

You can upload accounts in two ways: as part of an application or individually, separate from applications. In order to upload an account as part of an application, you must first create the account and then associate it to the application when you upload it to webMethods Cloud as described in "Uploading Applications" on page 35.

You can upload accounts separately from applications in the event that you need to override the settings of a previously uploaded account. You might do this if you associate an account while uploading an application to webMethods Cloud and later change the account details. This way, you can change the account details without having to upload the entire application.

Viewing the Accounts that Reside on the Server

The **Accounts** screen lists all of the webMethods Cloud accounts that reside on your on-premise Integration Server. It also displays whether each account is enabled.

To view the accounts that reside on the on-premise Integration Server

1. Open Integration Server Administrator.

2. Go to **webMethods Cloud > Accounts**.

   The **Accounts** page displays the list of accounts that were created on the on-premise Integration Server. Integration Server Administrator displays the following information for each account:

Column	Description
Alias	The name of the account.
Description	Description of the account.
Tenant alias	Tenant connection used to communicate with webMethods Cloud.
Stage	The webMethods Cloud environment from which the on-premise Integration Server receives requests.

Column	Description	
Last Uploaded Time	The last time the account was uploaded to webMethods Cloud.	
Upload	Displays one of the following icons to allow you to upload the account:	
	This icon...	**Indicates that the account is...**
		New or has been edited since the last time it was uploaded to webMethods Cloud.
		Is synchronized with the account that has already been uploaded to webMethods Cloud.
Test	Displays the ▶ icon to allow you to test the account.	
Enabled	Indicates whether the account is enabled.	
Delete	Displays the ✕ icon to allow you to delete the account from the on-premise Integration Server and webMethods Cloud.	

Creating Accounts on an On-Premise Integration Server

An account specifies the connection parameters needed to access webMethods Cloud. Each account on an on-premise Integration Server is associated with a single tenant, however, you can choose which tenant the account uses. Multiple accounts can use the same tenant.

Note:
Accounts existing on the on-premise Integration Server prior to the application of a fix that includes PIE-68485 or migration to a release that includes PIE-68485 can use the default tenant only. This cannot be changed. Only accounts created after PIE-68485 is applied (or the Integration Server is migrated to a release that includes PIE-68485) can use a tenant besides the default.

To create an account

1. Open Integration Server Administrator.
2. Go to **webMethods Cloud > Accounts**.
3. Click **Create On-Premise Account**.
4. Under **General Settings**, complete the fields as follows:

Field	Description
Enable	Enables or disables the webMethods Cloud account. Valid values:

Field	Description
	■ **Yes** - Enables the webMethods Cloud account. ■ **No** - Disables the webMethods Cloud account.
Alias Name	A unique name for the account.
Description	Description of the account.
Tenant Alias	The tenant used with the account. The tenant alias specifies the credentials used to connect to webMethods Cloud.
Stage	The webMethods Cloud environment from which the on-premise Integration Server receives requests. The list is populated by the stages or environments defined on webMethods Cloud. **Note:** The typical life cycle of an integration development involves creating integrations, testing them, and making them production worthy. Each of these activities can be termed as different stages of an integration life cycle development.

5. Under **Account Settings**, complete the fields as follows:

Field	Description
Maximum Reconnection Attempts	Specify the maximum number of reconnection attempts that Integration Server should make if the connection to webMethods Cloud fails. When the connection between webMethods Cloud and on-premise Integration Server is down, Integration Server attempts to reconnect to webMethods Cloud with a random interval between 100 milliseconds to 10 seconds for the first two minutes and then waits for a period of 30 to 60 seconds before attempting to retry the connection. The on-premise Integration Server continues in this manner until connectivity to webMethods Cloud is restored.
Request Timeout	Maximum amount of time (in milliseconds) that webMethods Cloud waits for the on-premise Integration Server to process a request. If the on-premise Integration Server is not listening for a request or if it takes longer to process the request than the specified time, webMethods Cloud issues an error and stops listening for a response. The default is 60000 milliseconds (1 minute). When the request timeout expires, webMethods Cloud shows this exception "[ISS.0021.8042E] Error occurred while executing service *<on-prem-servicename>* with request ID *<requestID>*. The on-premise Integration Server did not respond within the configured request timeout of *<requestTimeoutValue>* milliseconds. Check on-premise logs."

Field	Description
	If the invoked on-premise service takes more time to execute than the **Request Timeout** value, Integration Server writes the following error message to the server log: [ISS.0021.8041E] Error occurred while executing service <*serviceName*> with request ID <*requestID*>. Service execution took <*x*> milliseconds, which is more than the configured request timeout of <*y*> milliseconds. When responding to a request from webMethods Cloud, the on-premise Integration Server adds a time-to-live (TTL) property to the response. The on-premise Integration Server uses the **Request Timeout** value set for the webMethods Cloud account as the TTL value for the response. If the on-premise Integration Server sends the response after the timeout value for the request elapses on webMethods Cloud, the response remains on webMethods Cloud only until the response expires (that is, only until the TTL in the response elapses).
Time to Live	If you are batching the data from the on-premise Integration Server, the length of time in seconds that the execution results remain in the cache of the on-premise Integration Server. The value must be greater than 0. The default is 60 seconds.
Enable Compression	Specify whether on-premise Integration Server and webMethods Cloud must compress the request and response payloads during hybrid messaging. **Note:** Selecting the Enable Compression checkbox enables compression only if both the on-premise Integration Server and webMethods Cloud installations are configured to use the compression functionality. The compression process starts at webMethods Cloud. If webMethods Cloud compresses the request payload, on-premise Integration Server decompresses, processes, and compresses the payload in the response to webMethods Cloud. However, if webMethods Clouddoes not perform compression, then on-premise Integration Server returns the payload in its original, uncompressed form.
Allowed On-Premise Hosts	(Optional.) The on-premise Integration Server might use multiple addresses, depending on which network or proxy it uses to access webMethods Cloud. Specify a comma-separated list of IP addresses that can receive requests from webMethods Cloud. Only those IP addresses specified can receive requests. If no value is specified, webMethods Cloud derives the IP address of the on-premise Integration Server that uploads the account to webMethods Cloud and allows only that IP address to receive requests from webMethods Cloud.

Field	Description
Run As User	Specify the user name you want the on-premise Integration Server to use when running the service. Click 🔍 to search for a user and select the user. You can select users from the local or central directory. The on-premise Integration Server runs the service as if the user you specify is the authenticated user who invoked the service. If the service is governed by an ACL, ensure that you specify a user who is allowed to invoke the service.

6. If you want to test the account, click **Test Account Settings**.

   Integration Server Administrator displays a status line that indicates whether the account is successful or not. The status line is displayed at the top of the screen.

7. Click **Save Changes**.

Uploading Accounts to webMethods Cloud

After an account is created, you upload it to webMethods Cloud so that it can be used to execute services on the on-premise Integration Server. If you change the account after it is uploaded to webMethods Cloud, you must upload it again for the changes to become effective.

Note:
If you want to upload the account as part of an application, use the procedure described in "Uploading Applications" on page 35.

To upload accounts to webMethods Cloud

1. Open Integration Server Administrator.

2. In the **webMethods Cloud** menu of the navigation panel, click **Accounts**.

3. Click one of the following in the **Upload** column for the account you want to upload.

Click this icon...	If the account on the on-premise Integration Server is...
	New or has been edited since the last time it was uploaded to webMethods Cloud. ■ For a new account, this icon indicates that the account has not been uploaded to webMethods Cloud. ■ For an account that already exists on webMethods Cloud, this icon indicates that the account on the on-premise Integration Server is not synchronized with the one on webMethods Cloud.

Click this icon...	If the account on the on-premise Integration Server is...
	Synchronized with the account that has already been uploaded to webMethods Cloud.

When you upload the account, Integration Server Administrator:

- Displays a status line that indicates whether the account has been uploaded successfully. The status line is displayed at the top of the screen.
- Updates the **Last Uploaded Time** field to indicate the time that the account was uploaded and displays the icon to indicate that the account on webMethods Cloud is synchronized with the one on the on-premise Integration Server.

Note:
If a webMethods Cloud account is disabled during an upload, Integration Server uploads the account successfully. However, you will be notified to enable the account to serve any requests originating from webMethods Cloud.

Testing Accounts

After you add an account, you can test the account settings to ensure that the account is valid. Use the following procedure to test the account.

To test the account

1. Open Integration Server Administrator.
2. In the **webMethods Cloud** menu of the navigation panel, click **Accounts**.
3. Click the ▶ icon in the **Test** column for the account you want to test.

   Integration Server Administrator displays a status line that indicates whether the account is valid. The status line is displayed at the top of the screen.

Enabling and Disabling Accounts

When an account is enabled, Integration Server automatically establishes connectivity with webMethods Cloud at startup and is ready to serve any requests originating from webMethods Cloud.When an account is disabled, the on-premise Integration Server does not use the account to listen for requests from webMethods Cloud. Only an enabled account can listen for requests from webMethods Cloud to execute services on the on-premise Integration Server.

Note:
Disabling a tenant alias does not effectively disable the accounts associated with the alias. An account can still receive, process, and reply to requests for on-premise services from webMethods Cloud when the tenant alias associated with the account is disabled.

If you changed webMethods Cloud URL for a tenant alias since the account was uploaded or since you started Integration Server, you must disable and then re-enable the accounts that use the tenant alias to ensure that information is sent to and received from the new destination.

Note:
When an account is disabled, requests sent to the on-premise Integration Server remain in the queue until they are read or expire.

To enable or disable an account

1. Open Integration Server Administrator.
2. In the **webMethods Cloud** menu of the navigation panel, click **Accounts**.
3. Under the **Enabled** column of the **On-Premise Accounts** table, select one of the following:

Click...	To...
No	Enable the webMethods Cloud account.
Yes	Disable the webMethods Cloud account.

Deleting On-Premise Accounts

When you no longer need an account, you can delete it. Deleting an account from the on-premise Integration Server also deletes the account from webMethods Cloud.

Note:
If the account is in use by any of the integration flows in webMethods Cloud, the delete operation will fail.

To delete an on-premise account

1. Open Integration Server Administrator.
2. In the **webMethods Cloud** menu of the navigation panel, click **Accounts**.
3. Click the ✕ icon in the row that corresponds to the account you want to delete in the **Delete** column of the **On-Premise Accounts** table.
4. When Integration Server asks you to confirm that you want to delete the account, click **OK**.

Editing Accounts

After creating an account, you can edit the details. The tenant, alias, and stage of an account cannot be changed. If you edit an account, you must upload the account for the changes to take effect on webMethods Cloud.

To edit an account

1. Open Integration Server Administrator.
2. In the **webMethods Cloud** menu of the navigation panel, click **Accounts**.
3. Locate the account you want to edit and click on the name in the **Alias** column.
4. Update the information for the account.
5. Click **Save Changes**.

   Note:
   While editing, if you enable or disable compression for an existing account, disable the account, and then enable it before the next step. For more information, see "Enabling and Disabling Accounts" on page 27.

6. To upload the account, follow the procedure described in "Uploading Accounts to webMethods Cloud " on page 26.

Monitoring On-Premise Listeners for webMethods Cloud Accounts

Each enabled webMethods Cloud account should have an active listener on the on-premise Integration Server. The listener listens for requests from webMethods Cloud to execute services on the on-premises Integration Server. To ensure that each enabled webMethods Cloud account has an active listener, Integration Server uses a monitoring thread that executes at as specified interval. If the monitoring thread finds a listener that is not running, the monitoring thread attempts to start the listener. The watt.wmcloud.listeners.monitoringInterval server configuration parameter determines the interval at which the monitoring thread executes. The default value of the parameter is 300000 milliseconds (5 minutes).

Additionally, the monitoring thread looks for listeners that have not processed any requests in a specified period of time and recreates the listeners that have been idle too long. The watt.wmcloud.listeners.maxIdleTime server configuration parameter determines the length of time a listener can be idle before it gets recreated. The default value of this parameter is 1800000 milliseconds (30 minutes).

Integration Server stops and restarts all on-premise account listeners whenever the **Update Settings** button on the **webMethods Cloud > Tenant connections** page is clicked.

Note:
The monitoring thread exists only if the on-premise Integration Server has at least one enabled webMethods Cloud account.

When an account connection is disconnected or restored, Integration Server notifies the user through a notification in Integration Server Administrator followed by an email. Integration Server generates notifications for an active listener.

When you create an account, the on-premise Integration Server creates two listeners (request and response) for the account. Therefore, if an account is disconnected or reconnected, Integration Server generates two notifications per listener. Configure the hybrid activity notifications using the `watt.wmcloud.hybridConnectivityAlert.notifications` server configuration parameter and the recipients of the notification emails using the `watt.wmcloud.hybridConnectivityAlert.mail` server configuration parameter. Integration Server uses the email notification settings configured in **Integration Server Administrator > Settings > Resources**. For more information, see the *webMethods Integration Server Administrator's Guide*.

Note:
The hybrid connectivity alerts are introduced as part of PIE-81176 in IS_10.15_Core_Fix4.

4 Managing Applications

About Sharing Metadata on an On-Premise Integration Server

You create applications on the on-premise Integration Server to share services with webMethods Cloud. Keep the following points in mind while sharing services through an application:

- You can share only services running on the on-premise Integration Server configured to create applications on webMethods Cloud.
- You can share only services contained in custom packages.
- You can share services from different packages in the same application. For example, if service A is located in package A, and service B is located in package B, you can add both service A and service B to the same application.
- You can share only those services that have an input and/or output signature specified.
- You can share only those services whose signatures are of the following data types:
  - String
  - String List
  - Document
  - Document Reference
  - Document List
  - Document Reference List
  - Object
  - Object List
- Any Object or Object List placed into the service pipeline, and therefore any Object or Object List in a service signature, must be a serializable data type. If it cannot be serialized, the pipeline cannot be exchanged with a remote Integration Server. Further, the class names for the data type must be included in the Integration Server whitelist classes file or a package whitelist classes file for the pipeline to be deserialized at the destination Integration Server. Classes defined by Integration Server are already included in the whitelist class file. This requirement applies to all Object and Object List variables, including those contained in a Document or Document List.
- You can set the on-premise Integration Server to send service results to webMethods Cloud in batches.
- You cannot share service signatures that include:
  - Cyclical dependencies of document references.
  - Fields of type String Table, including fields of type String Table in a Document.
  - An empty Document or Document List.

- When the server log facility code **0021 webMethods Cloud** is set to the Debug log level, Integration Server writes log messages that indicate why an on-premise service is marked as not shareable.
- You must configure one or more accounts to associate with the application before you can upload the application to webMethods Cloud. For more information about configuring accounts, see "Configuring Accounts for webMethods Cloud " on page 21.
- You must upload the application for the updates to be shared with webMethods Cloud if you edit:
  - The application.
  - The signature or referenced Document of a service shared by the application.
- When you upload an application, it replaces the application and operations available on webMethods Cloud with the one that you upload.

Viewing the Applications that Reside on Your Server

The Applications screen lists all of the applications on the on-premise Integration Server that you can share with webMethods Cloud.

To view the applications you can share with webMethods Cloud from the on-premise Integration Server

1. Open the Integration Server Administrator if it is not already open.
2. In the **webMethods Cloud** menu of the navigation panel, click **Applications**.

   The Applications screen displays the list of applications that were created on the on-premise Integration Server. Integration Server Administrator displays the following information for each application:

Column	Description	
Name	Name of the application.	
Description	Description of the application.	
Last Uploaded Time	The last time the application was uploaded to webMethods Cloud.	
Upload	Displays one of the following icons to allow you to upload the application:	
	This icon...	**Indicates that the application is...**
		New or has been edited since the last time it was uploaded to webMethods Cloud.

Column	Description
	Synchronized with the application that has already been uploaded to webMethods Cloud.
Delete	Displays the ✕ icon to allow you to delete the application from the list and from webMethods Cloud.

Defining Applications

Perform the following procedure to define applications to share with webMethods Cloud.

To define an application to share with webMethods Cloud

1. Open the Integration Server Administrator if it is not already open.
2. In the **webMethods Cloud** menu of the navigation panel, click **Applications**.
3. Click **Define webMethods Cloud Application**.
4. Under **webMethods Cloud Application**, complete the following fields:

Field	Description
Name	A unique name for the application. The application name cannot exceed 32 characters, contain reserved words and characters that are used in Java or C/C++ (such as *for*, *while*, and *if*), or the following illegal characters: "#-&@^!%*:$./\ \ `;,~+=)(|}{][><"
Description	Description of the application.

5. In the **Assign Services to Application** area, specify the services to expose to webMethods Cloud as follows:
   a. Under **Package/Services**, click ⊞ to expand the services available in the package.
   b. Select each service you want to expose to webMethods Cloud.
   c. If you want the service to return results to webMethods Cloud in batches, rather than returning the results all at once, click **Batch Data**.

      If the top level output signature of the service contains only one field, and the field is a document list or document reference list (both of which are IData arrays), **Batch Data** is selected by default. Otherwise, it will be cleared.

By selecting this option, Integration Server uses pub.flow:iterator to batch service results. For more information about the pub.flow:iterator service, see *webMethods Integration Server Built-In Services Reference*.

d. Under **Display Name**, specify the name of the service as it should appear in webMethods Cloud or accept the default.

The **Display Name** field cannot contain reserved words and characters that are used in Java or C/C++ (such as *for*, *while*, and *if*) or the following illegal characters:

"#-&@^!%*:$./\ \ `;,~+=)(|}{][><"

Note:
Applications are displayed in webMethods Cloud as *operations*. Operations are named according to the name defined by the **Display Name** field.

6. Click **Save Changes**.

Uploading Applications

After you define an application, you must upload it to webMethods Cloud before using it in an integration. The account associated with an application must use an enabled tenant alias for an application to upload successfully.

To upload an application

1. Open the Integration Server Administrator if it is not already open.

2. In the **webMethods Cloud** menu of the navigation panel, click **Applications**.

3. Click one of the following icons in the row that corresponds to the application you want to upload in the **Upload** column of the **webMethods Cloud Applications** table.

This icon...	Indicates that the application on the on-premise Integration Server is...
	New or has been edited since the last time it was uploaded to webMethods Cloud. ■ For a new application, this icon indicates that the application has not been uploaded to webMethods Cloud. ■ For an application that already exists on webMethods Cloud, this icon indicates that the application on the on-premise Integration Server is not synchronized with the one on webMethods Cloud.
	Synchronized with the application that has already been uploaded to webMethods Cloud.

4. When the **Upload Application** screen appears, from the **Select** column of the **Select Accounts** area, select one or more accounts to associate with the application.

5. Click **Upload**.

6. When Integration Server asks you to confirm that you want to upload the application, click **OK**.

   When you upload the application, the on-premise Integration Server:

   - Uploads the application to webMethods Cloud, replacing the existing application.
   - Updates the **Last Uploaded Time** column of the **webMethods Cloud Applications** screen to indicate that the application on webMethods Cloud is synchronized with the one on the on-premise Integration Server.
   - Shares the service name, service signature, display name, and service comments with webMethods Cloud.

   Note:
   If the application has been uploaded previously to webMethods Cloud, uploading the application again overwrites the existing application.

Note:
If any account associated with an application is disabled during an upload, Integration Server uploads the application successfully. However, you will be notified to enable the account to serve any requests originating from webMethods Cloud.

Deleting Applications

When you no longer want to share applications with webMethods Cloud, you can delete them. Deleting an application from the on-premise Integration Server also deletes the application and its corresponding operations from webMethods Cloud.

To delete an application

1. Open the Integration Server Administrator if it is not already open.

2. In the **webMethods Cloud** menu of the navigation panel, click **Applications**.

3. Click the ✕ icon in the row that corresponds to the application you want to delete in the **Delete** column of the **webMethods Cloud Applications** list.

4. When Integration Server asks you to confirm that you want to delete the application, click **OK**.

Editing Applications

After creating an application, you can edit the application details. If you edit an application, you must upload the application for the changes to take effect on webMethods Cloud.

To edit an application

1. Open the Integration Server Administrator if it is not already open.
2. In the **webMethods Cloud** menu of the navigation panel, click **Applications**.
3. Locate the application you want to edit and click on the name in the **Name** column.
4. Update the information for the application.
5. Click **Save Changes**.

Note:
The updated application is not available for use on webMethods Cloud until you upload it to webMethods Cloud. Until then, webMethods Cloud uses the last version of the application that you uploaded. To upload the application, follow the procedure described in "Uploading Applications" on page 35.

5 Managing Docker Services

Overview

You can create a Docker image from an installed and configured Integration Server instance and then run the Docker image inside a Docker container. Integration Server provides a script to create Docker files, build Docker images, load or push the Docker image to a Docker registry hosted on-premise, and push the Docker image to a Docker registry hosted on webMethods Cloud.

Docker support is available on Linux and UNIX systems for which Docker provides native support.

Note:
You must enable CSRF Guard in Integration Server before you create the Docker image and upload it to webMethods Cloud. Enabling the CSRF security feature will prevent CSRF attacks. See the *webMethods Integration Server Administrator's Guide* on the Software AG Documentation website at http://documentation.softwareag.com for information on how to enable CSRF Guard.

Specifying Services to Expose to Consumers in webMethods Cloud

If you want to expose Integration Server services running inside a Docker container in webMethods Cloud to consumers, you must specify the Integration Server services before creating the Docker image for Integration Server. The **webMethods Cloud Docker Services** screen lists all the services to expose to consumers from a Docker container.

To specify the services to expose in webMethods Cloud

1. Open the Integration Server Administrator if it is not already open.

2. In the **webMethods Cloud** menu of the navigation panel, click **Docker services**.

   The **webMethods Cloud Docker Services** screen is displayed. The **Package** column lists all packages in Integration Server.

3. In the **Docker Services** column, specify the services to expose to consumers from your custom packages. You can specify the services using one or more of the following ways:

   - Directive and fully qualified name of the service in the following format:

     /directive/folder.subfolder:serviceName

     For example: /invoke/is.assets:getChecksums. Use a comma to separate the services.

   - The URL for invoking the service. For the URL, specify a snippet that is unique enough for webMethods Cloud to accurately map it to the service in the Docker container. Use a comma to separate the services.

   - A URL alias created using **Settings > URL aliases > Create URL Alias.** Use a comma to separate the services.

4. Click **Save Changes**.

> **Note:**
> For information about how to create, run, and push Docker images to webMethods Cloud, see the *webMethods Integration Server Administrator's Guide*.

6 Connecting to webMethods Cloud through a Proxy

About Configuring Integration Server

If your on-premise Integration Server is to connect to webMethods Cloud through an Internet proxy, you may need to perform additional configuration which falls into the following categories:

- Creating a proxy server alias.
- Adding proxy-related Java system properties to the Java Virtual Machine in which the on-premise Integration Server runs.
- Configuring the Internet proxy for use with the on-premise Integration Server and webMethods Cloud.

Creating a Proxy Server Alias

If your company requires use of a proxy server for establishing outbound connections to applications located in the cloud, then you need to configure a proxy server alias for your on-premise Integration Server. A proxy server alias identifies the proxy server and the port on the proxy server through which you want to route requests and any credentials needed to access the proxy server.

When creating a proxy server alias for use for connection to webMethods Cloud, you must do the following:

- Specify the proxy server alias as the default proxy server alias.
- Specify the proxy host and port number.
- Set the protocol to HTTPS.
- If the proxy server is configured for basic authentication, the proxy server alias must specify the user name and password required to access the proxy server.

You configure a proxy server alias using the **External servers > Proxy servers** page in Integration Server Administrator. For detailed information about proxy servers and how to configure a proxy server, see the *webMethods Integration Server Administrator's Guide*.

Updating JVM Configuration Settings for Proxies

Depending on your networking and security requirements, you may need to set Java system properties related to proxy servers. Because the connection between the on-premise Integration Server and webMethods Cloud is over HTTPS, you need to set some or all of the following:

- `https.proxyHost`
- `https:proxyPort`
- `http:nonProxyHosts`
- `https.proxyUser`
- `https.proxyPassword`

Because the proxy property values need to be set for the Java Virtual Machine (JVM) in which Integration Server runs, you update the custom_wrapper.conf to ensure the proxy parameter values are supplied when the JVM launches. Specifically, you add a `wrapper.java.additional.` *n* property that specifies the property name and value that you want to pass to Integration Server, where *n* is a unique sequence number. The property name must be preceded by `-D`.

For example, the `wrapper.java.additional properties` in the custom_wrapper.conf file might look similar to the following:

```
wrapper.java.additional.204=-Dhttps.proxyHost=YOUR_PROXY_HOST
wrapper.java.additional.205=-Dhttps.proxyPort=YOUR_PROXY_PORT
wrapper.java.additional.206=-Dhttps.proxyUser=YOUR_PROXY_USER
wrapper.java.additional.207=-Dhttps.proxyPassword=YOUR_PROXY_PASSWORD
wrapper.java.additional.208=-Dhttp.nonProxyHosts=localhost|127.0.0.1
wrapper.java.additional.209=-DHPROXY=YOUR_PROXY_HOST:YOUR_PROXY_PORT
wrapper.java.additional.210=-DHAUTH=YOUR_PROXY_USER:YOUR_PROXY_PASSWORD
```

For instructions about passing Java system properties to Integration Server and the JVM used by Integration Server, see the *webMethods Integration Server Administrator's Guide*.

To pass system properties to Microservices Runtime, you must update the JAVA_CUSTOM_OPTS property in *Integration Server_directory* /bin/server.bat(sh). For more information about passing system properties to Microservices Runtime, see *Developing Microservices with webMethods Microservices Runtime Developing Microservices with webMethods Microservices Runtime*.

Note:
If Integration Server uses Java SE Development Kit 8, Update 111 or later and the Internet proxy is configured to enforce HTTPS Basic authentication, you may need to add the following system properties to the custom_wrapper.conf:

```
jdk.http.auth.proxying.disabledSchemes

jdk.http.auth.tunneling.disabledSchemes
```

Configuring the Internet Proxy

When using a proxy server for outbound requests from on-premise Integration Server to webMethods Cloud, you may need to configure the following on the proxy server:

- Whitelist entries. If the proxy server employs a whitelist of allowed URLs, you need to add the URL for webMethods Cloud. A URL for the webMethods Cloud uses the following format:
  - `https://softwareag-education.webmethodscloud.com` or
  - `https://sample.prod-int-aws-us.webmethods.io`

  For more information about specifying whitelist entries for an Internet proxy, refer to the documentation for the Internet proxy used by your company.

  For information on specific IP addresses to whitelist, refer to the documentation of the respective webMethods Cloud products.

- Tunneling with HTTP CONNECT. To ensure end-to-end encryption for TLS over HTTPS, configure the Internet proxy to accept the HTTP CONNECT method. If the Internet proxy does not accept the HTTP CONNECT, a connection from the on-premise Integration Server to webMethods Cloud through the proxy server can still be established but end-to-end encryption with TLS will not be provided.
- Long running HTTP connections. Configure the proxy server to be friendly to long running HTTP 1.0 and 1.1 connections. This may improve performance and limit latencies.