Hyland Email Broker

Configuration Guide

Version: 4.0

Written by: Product Knowledge, R&D
Date: December 2025

HYLAND™

Documentation Notice

Table of Contents

Overview

The Hyland Email Broker (Broker) has three configuration points, start parameters, appsettings.json file, and remote configuration. The start parameters allow you to specify the location of the app settings and crash information files. The appsettings.json file controls the remote configuration access of the service, and the remote configuration contains the settings controlled by your Hyland Remote Configuration (HRC) Server (HRCS).

Start parameters

The Broker provides two optional parameters, **--config** and **--crashfile**.

The **--config** parameter allows you to specify the file path to use for your app settings file. This parameter overwrites both the default location, ./**appsettings.json**, and the **ECS_BROKER_provider_source** environment variable. You may provide either a full or relative file path. To set the app settings file, enter **--config** followed by the desired file path. If the path contains spaces, enclose it with quotation marks.

For example, if your settings file is located at **C:/My Settings/myBrokerSettings.json**, you would add the following to your start parameters:

--config "C:/My Settings/myBrokerSettings.json"

The **--crashfile** parameter allows you to specify the file path to use for service crash files. Service crash files occur when starting the service with an invalid or missing app settings file. This setting overwrites the default location, **./HylandEmailBroker.err**. To set the crash file, enter **--crashfile** followed by the desired file path.

For example, if your crash file is located at **C:/CrashFiles/MyBrokerCrash.txt**, you would add the following start parameters

--crashfile C:/CrashFiles/MyBrokerCrash.txt

Windows service start parameters

If you want to use start parameters with an installed Windows service, then you must run the service controller command, **sc config,** to configure the start parameters.

To modify the service's start parameters, complete the following steps:

1. Open an elevated command prompt.
2. Run the following command:

   ```
   sc config Hyland.Email.Broker.exe binPath=
   "\"{INSTALL_DIR}\Hyland.Email.Broker.exe\" --crashfile \"{CRASHFILE_PATH}\" --
   config \"{CONFIG_PATH}\""
   ```

   Where

   {INSTALL_DIR} is the location where the Broker is installed. By default, this location is **C:\Program Files\Hyland\Hyland Email Broker**.

   {CRASHFILE_PATH} is the location where the system writes the crashfile.

   {CONFIG_PATH} is the name and location of the settings.json file.
3. Restart the service.

Linux service start parameters

The Linux installer provides arguments to set the start parameters at installation time. If you want to modify the start parameters after installation, then you must modify the installed **.service** file and restart the service.

To modify the service's start parameters, complete the following steps:

1. Locate and open the installed **.service** file. The default service name is **HylandEmailBroker.service** with **/etc/systemd/system/** as the default location. You can change the name and location during installation.
2. Locate and modify the desired **Environment** entry in the **.service** file. If the desired entry is missing, then you must add the entry yourself.
    - The Environment="ECS_BROKER_configfile= entry controls the --config start parameter.
    - The Environment="ECS_BROKER_crashfile= entry controls the --crashfile start parameter.

To use the system environment variable **ECS_BROKER_provider__source**, complete the following steps:

1. Remove the setting for the **--config** start parameter by setting the value to nothing. For example, **Environment="ECS_BROKER_configfile=**.
2. Add the system environment variable **ECS_BROKER_provider__source=/full/path/to/settings/file.json**.
3. Restart the service.

For more information on basic **systemctl** commands, refer to Linux Service Control Commands.

Proxy support

The Broker has the capability to utilize a proxy server. Since some URLs within the Broker use the imap:// protocol, we recommend setting the **ALL_PROXY environment** variable to ensure optimal performance. The Broker's imap connection can also establish connections through an HTTP proxy server. For more information, see the HttpClient.DefaultProxy Property topic on learn.microsoft.com.

App settings

The app settings file controls how your Broker connects to the Hyland Remote Configuration Server (HRCS). The remote connections are controlled by the Remotes section in your app settings. You may also include an optional Hyland Logging section to debug issues with your remote connections. However, any debug logging should be removed once the **Remotes** configuration is working. All production logging should be configured through HRCS. See the Remote configuration: Hyland.Logging section for information on available logging settings.

Remotes

The **Remotes** section contains the connection information for your HRCS configurations.

The **Remote Configuration Servers** subsection is a map containing the various server configurations for the Broker. Each **<User Defined Remote Name>** entry of the **Remote Configuration Servers** map represent a unique server configuration. The **<User Defined Remote Name>** keys do not matter to the Broker, and they are used to easily track the various servers with human-readable identifiers. Typically, a

Broker instance should only have a single **<User Defined Remote Name>**. For more information on Remote HRCS Connections and behavior, see Appendix B: Remote HRCS connection best practices and behavior.

For a full list of remotely configured options see Remote Configuration.

<User Defined Remote Name>

Each **<User Defined Remote Name>** is a subsection of the **Remote Configuration Servers** section and specifies the settings for each configured server. The **<User Defined Remote Name>** is user-defined and only serves as a human-readable identifier for the server.

The following table lists the settings for the **<User Defined Remote Name>** sub-section.

Setting	Description
Configuration Reference Ids	Specifies a map of **<User Defined Reference Names>** to references IDs on the HRCS.
Host	Specifies the host IP address of the configured server.
Port	Specifies the port number of the configured server.
Route	Specifies the client route of the HRCS.
Bearer Token	Specifies the bearer token that will be used when interacting with the configured HRCS.
Reconnect Delay	Specifies how long to wait before reattempting to connect to the HRCS after a failed attempt.

ConfigurationReferenceIds <User Defined Reference Names>

The <User Defined Reference Names> keys of the **ConfigurationReferenceIds** control which configurations the Broker should retrieve from the HRCS server. The Broker support Hyland Logging HRCS configurations identified with a **HylandLogging** prefix, and Email Broker HRCS configurations identified with an **EmailBroker** prefix. Typically, a Broker instance should only have a single **EmailBroker** HRCS configuration with an optional **HylandLogging** HRCS configuration. Any entries without a **HylandLogging** or **EmailBroker** prefix are ignored by the Broker. For more information on Remote HRCS connections and behavior, see Appendix B: Remote HRCS connection best practices and behavior. For more information on best practices for logging, see Appendix C: Logging best practices.

Example

The following is an example of the **Remotes** section.

```
"Remotes": {
    "RemoteConfigurationServers": {
        "<User Defined Remote Name>": {
            "ConfigurationReferenceIds": {
                “HylandLogging”: “example-hyland-logging-id”,
                "EmailBroker": "example-broker-id",
                “xEmailBroker”: “example-ignored-broker-id”
            },
            "Host": "example-hrcs-host",
            "Port": "5001",
```

```
            "BearerToken": "example-bearer-token",
            "ReconnectDelay": "00:00:10"
        }
    }
}
```

Configuration

The Broker requires a **Hyland-Email-Broker** type configuration with version **HylandEmailBrokerV3** to support remote configuration. Your HRCS administrator must create the configuration. There are three ways to create configuration for editing.

- To start from an empty config, the HRCS administrator will create configuration by navigating to **Manage Config** > **Hyland-Email-Broker** and clicking the plus icon. After giving the config instance a description, the administrator can click the **Save and Open** button and then click the **Create Empty Config** button under **Version: HylandEmailBrokerV3**.
- To start from a pre-existing json file (appsettings.json), the HRCS administrator will navigate to **Manage Config** > **Hyland-Email-Broker** and click the plus icon. After giving the config instance a description, the administrator can click the **Save and Open** button and then click the **Upload** button under **Version: HylandEmailBrokerV3.**
- To start from an Email Agent ini file, the HRCS administrator will navigate to **Manage Config** > **Hyland-Email-Broker** and click the plus icon. After giving the config instance a description, the administrator can click the **Save and Open** button and then follow the instructions in Appendix E: Email Agent Upgrade.

After creating the configuration, the administrator must configure the ID in your **Remotes > RemoteConfigurationServers > {User Defined Remote Name} > ConfigurationReferenceIds** settings in your Broker **appsettings.json** file.

Remote configuration

The following sections contain the settings controlled by each of your HRCS connections.

Instance Options

The **Instance Options** section controls the Broker's immutable global settings.

Note: You must restart the Broker before the Broker recognizes any changes within the **Instance Options** section. All other settings can be changed without restarting the Broker.

The following table lists the settings for the **Instance Options** section.

Setting		Description
Instance Name		Controls the name of this instance. This name appears on all log events from this instance as the **Instance Name** logger context field, and it modifies the name of the temporary files generated by this instance. If you only use one instance, then you may omit this setting. The Broker uses **HylandEmailBroker** as the default broker name when the **Instance Name** is not specified.
Temp Directory		Controls the path to the directory used by this Broker instance to store temporary files. If you omit this setting, it defaults to the temporary path defined by your operating system. On Windows, this path is `C:\Users\{Username}\AppData\Local\Temp` where `{Username}` is the name of the user running the Broker instance. On Linux, this default path is `/tmp`.

System

The **System** section controls the Broker's mutable global settings.

The following table lists the settings for the **System** section.

Setting		Description
Default Worker Execution Interval		Controls how often a worker should fire. If the execution interval elapses before a worker finishes processing its inbox, then the Broker allows the worker to continue without interruption and does not attempt to start the worker until after the next interval. Should match the **[-][d'.']hh':'mm':'ss['.'fffffff]** timespan format. If you omit this setting, the app uses **1** minute as the default. If the time elapses and a worker is still working, then the worker will simply continue its current task.
Worker Shutdown Timeout		Controls how long the app should wait for a worker to shut down during a configuration change or during shutdown. The value should match the **[-][d'.']hh':'mm':'ss['.'fffffff]** timespan format. If you omit this setting, the app uses **10** seconds as the default. If a worker does not shut down gracefully within the timeout, then the broker will forcibly stop the worker, regardless of its current task. This may result in partially captured messages on your S3 server.
Stagger Startup		Controls how the app starts its workers. If **Stagger Startup** is true, then each worker starts after its **Execution Interval**. If

		Stagger Startup is **false**, then the app starts all workers immediately. Defaults to **false**.
Debug Flags		The debug flags control behaviors for troubleshooting issues with the Broker. These settings require an attached debugger, and they may cause unsafe or unexpected behavior. They should never be used in a production environment.

Hyland Logging

The **Hyland Logging** section controls the logging behavior for the Broker. Typically, a deployment only uses the **Routes** settings, but the Broker supports all Hyland Logging settings.

Routes

The **Routes** section is a subsection of **Hyland Logging** and specifies the configuration information for each configured route.

<Name of Route>

The **<Name of Route>** section is a subsection of the **Routes** section and specifies the settings for each configured route. The **<Name of Route>** name is user-defined.

The following table lists some of the common settings for the **<Name of Route>** section.

Setting	**Description**
Console	Specifies logging to the console. Leave this value empty.
File	Specifies the path and name of the log file to where the logger writes.
Splunk	Specifies the http address of the Splunk server. Omit to disable Splunk logging.
Splunk Token	Specifies the authentication token to use for the Splunk server
Exclude Profiles	Specifies which profiles are not written to this route. Profiles not listed are written to the route. The **include-profiles** setting overrides this setting. The default is **Empty List**.
Include Profiles	Specifies which profiles are written to this route. Profiles not listed are not written. The default is **Empty List**.
Minimum Level	Specifies the minimum level of logging, **Trace**, **Debug**, **Information**, **Warning**, **Error**, **Critical**, or **None**, you want the system to log.
Maximum Level	Specifies the maximum level of logging, **Trace**, **Debug**, **Information**, **Warning**, **Error**, **Critical**, or **None**, you want the system to log.
File Roll on Size	Specifies if the logger should rollover based on size. The default is **false**.

File Count Limit	Specifies the number of rollover log files to keep. Setting **File Count Limit** to **“”** (empty string) specifies that the logger should keep an infinite number of log files. The default is **31**.
File Byte Limit	Specifies the maximum size a log file may reach before the logger stops writing to the file. If **File Roll on Size** is set to **false**, then the logger will not write to the file until another log rollover event occurs, or the file is deleted. The default is **1 GB**.
Output Format	Specifies the output format to use when logging. Valid values are **json**, **text**, and **minimal**. The default is **json**. Note that the minimal format does not include logging context data, so some logging information may be missing.

Email Servers

The **Email Servers** section consists of a list of email servers used by the Workers' configurations. Each sub-section of **Email Servers** represents a unique server that is named by its section property. These names are user defined and are referenced by the Workers' **Email Server > Server** field. Each server can use the **IMAP**, **Microsoft 365**, **EWS** or **Gmail API** protocol. Typically, a Broker will only have one configured Email Server with a Connection Option for each inbox that the Broker should monitor. However, the Broker can support multiple Email Servers with any combinations of protocols.

IMAP

The **IMAP** section specifies the settings necessary to connect to an Internet Message Access Protocol (IMAP) email server. An IMAP server can use one of the following connection mechanisms.

- Basic Authentication IMAP
- Gmail 2LO IMAP
- Gmail 3LO IMAP

For the Gmail IMAP servers, you must work with your G Suite administrators to register a project with IMAP access for your Broker instance. This project manages the **Google OAuth** settings used by Gmail 2LO IMAP, and it manages the Client ID and Client Secret used by Gmail 3LO IMAP.

Note that if you are connecting to a Gmail server, you may want to use a Gmail Api server instead of one of the Gmail IMAP protocols. The Gmail API protocol has faster integration with Gmail and provides **Delete Behavior** settings. The Gmail IMAP options exist to provide retrieval behavior that is identical to the legacy Email Agent (Agent) **Gmail IMAP protocols.**

Base Authentication IMAP

The **Base Authentication IMAP** section specifies the settings for a generic IMAP server using basic authentication connection options. Note that Gmail prevents access using basic authentication, so you must use one of the Gmail **IMAP** protocols, or the Gmail API protocol.

The following table lists the settings for the **Base Authentication IMAP** section.

Setting	**Description**
Host	Specifies the host server of this IMAP email server.

Port	Specifies the port number exposed by the email server for IMAP protocol connections.
Connection Options	Specifies the connection options available with this server. Each connection option has a user defined name. These options are referenced by the Workers' **Email Server** > **Connection Option** property. Multiple workers should not target the same option.

Connection Options

The **Connection Options** for Basic Authentication IMAP contain the following.

Setting	Description
Username	Specifies the username for the account that owns the monitored inbox. Multiple options should not target the same inbox.
Password	Specifies the password for the specified username

Gmail 2LO IMAP

The **Gmail 2LO IMAP** section specifies the settings for a Gmail IMAP server using two-legged OAuth. Alternatively, you may wish to configure a Gmail API > Gmail 2LO server for better performance and for access to the Delete Behavior setting. The **Gmail 2LO IMAP** protocol preserves the legacy behavior and permissions of the Email Agent's **GIMAP** protocols.

The following table lists the settings for the **Gmail 2LO IMAP** section.

Setting	Description
Host	Specifies the host server of this IMAP email server.
Port	Specifies the port number exposed by the email server for IMAP protocol connections.
Google OAuth	Specifies the information for the server's Google OAuth configuration. This section matches the JSON object from a service account's JSON private setting.
Connection Options	Specifies the connection options available with this server. Each connection option has a user defined name. These options are referenced by the Workers' **Email Server** > **Connection Option** property. Multiple workers should not target the same option.

Connection Options

The **Connection Options** for Gmail 2LO IMAP contain the following:

Setting	Description
Username	Specifies the username for the account that owns the monitored inbox. Multiple options should not target the same inbox.

Gmail 3LO IMAP

The **Gmail 3LO IMAP** specifies the settings for a Gmail IMAP server using three-legged OAuth. Alternatively, you may wish to configure a Gmail API > Gmail 3LO server for better performance and for access to the Delete Behavior setting. The Gmail 3LO IMAP protocol preserves the legacy behavior and permissions of the Email Agent's **GTOKENIMAP** protocol.

The following table lists the settings for the **Gmail 3LO IMAP** section.

Setting	Description
Host	Specifies the host server of this IMAP email server.
Port	Specifies the port number exposed by the email server for IMAP protocol connections.
Client ID	Specifies the ID generated when registering your Broker.
Connection Options	Specifies the connection options available with this server. Each connection option has a user defined name. These options are referenced by the Workers' **Email Server** > **Connection Option** property. Multiple workers should not target the same option.

Connection Options

The Connection Options for Gmail 3LO IMAP contain the following

Setting	Description
Username	Specifies the username for the account that owns the monitored inbox. Multiple options should not target the same inbox.
Client Secret	Specifies the client secret for Gmail refresh token connections.
Refresh Token	Specifies the refresh token used for Gmail 3LO authentication. You must use the **Hyland Gmail Auth Client** to acquire your refresh tokens.

Microsoft 365

The **Microsoft 365** section contains the **Client Secret Server** settings necessary to connect to an **Microsoft 365** email server. When configuring a **Microsoft 365** server, you must work with an Azure administrator to register the Broker under **App Registrations** in your tenant's Active Directory. This registration generates the Application (client) ID. The Directory (tenant) ID is the ID of the active directory. You can find both IDs on the **Overview** page of the Broker's registration. The Broker needs the **Application - Mail.ReadWrite** permission with admin consent to perform its operations.

Client Secret Server

The following table lists the settings for the Microsoft 365 **Client Secret Server** section.

Setting	Description
Host	Specifies the host server of this Office365 email server.

Client ID	Specifies the ID generated by Azure Active Directory when registering your Broker.
Tenant ID	Specifies the ID of your organization (or Active directory) within Azure.
Delete Behavior	Specifies the action to perform when choosing to delete the message on successful processing. For more information on the available settings, see the Delete Behavior table below.
Connection Options	Specifies the connection options available with this server. Each connection option has a user defined name. These options are referenced by the Workers' **Email Server > Connection Option** property. Multiple Workers should not target the same option.

Delete Behavior

The following table lists the options for the Microsoft 365 **Delete Behavior** setting.

Setting	Description
Move to deleted items	Moves the message to the Deleted Items folder.
Soft delete	Moves the message to the Recoverable Items Deletions folder using the key **RecoverableItemsDeletions**. **Note** Review Outlook Online configuration to verify the behavior for your configured instance. For example, behavior may change based on **In-Place Hold**, **Litigation Hold**, and **Single-Item Recovery** settings.
Hard delete	Moves the message to the Recoverable Items Purges folder using the key, **RecoverableItemsPurges**. **Note** Review Outlook Online configuration to verify the behavior for your configured instance. For example, behavior may change based on **In-Place Hold**, **Litigation Hold**, and **Single-Item Recovery** settings.

Connection Options

The Connection Options for Microsoft 365 Client Secret Server contain the following

Setting	Description
Username	Specifies the username for the account that owns the monitored inbox. Multiple options should not target the same inbox.
Client Secret	Specifies the client secret for the registered Azure application's Client ID.

EWS

The **EWS** section specifies the settings necessary to connect to an Exchange Web Services (EWS) email server. The EWS protocol only works for on-premise EWS servers. Hybrid EWS or Microsoft 365 servers must use the Microsoft 365 protocol instead.

The following table lists the settings for the **EWS** section.

Setting	Description
Host	Specifies the host server of this EWS email server.
Delete Behavior	Specifies the action to perform when choosing to delete the message on successful processing. For more information on the available settings, see the Delete Behavior table below.
Connection Options	Specifies the connection options available with this server. Each connection option has a user defined name. These options are referenced by the Workers' **Email Server > Connection Option** property. Multiple workers should not target the same option.

Delete Behavior

The following table lists the settings for the EWS **Delete Behavior** section.

Setting	Description
Move to deleted items	Moves the message to the Deleted Items folder.
Soft delete	Moves the message to the Recoverable Items Deletions folder using the key **RecoverableItemsDeletions**.
Hard delete	Moves the message to the Recoverable Items Purges folder using the key, **RecoverableItemsPurges**.

Connection Options

The Connection Options for EWS contain the following.

Setting	Description
Username	Specifies the username for the account that owns the monitored inbox. Multiple options should not target the same inbox.
Password	Specifies the password for the specified username.

Gmail Api

Gmail Api specifies the settings necessary to connect to a Gmail email server using the Gmail API. The Gmail Api has faster integration with Gmail than a Gmail 2LO or 3LO IMAP server and provides **Delete Behavior** options. A Gmail Api server can use one of the following connection mechanisms.

- Gmail 2LO
- Gmail 3LO

You must work with your G Suite administrators to register a project with Gmail API access for your Broker instance. This project manages the **Google OAuth** settings used by Gmail 2LO, and it manages the Client ID and Client Secret used by Gmail 3LO.

Gmail 2LO

The **Gmail 2LO** section specifies the settings for a Gmail server using two-legged OAuth and the Gmail API.

The following table lists the settings for the **Gmail 2LO** section.

Setting	Description
Uri	Specifies the Gmail URL. This setting is optional.
Delete Behavior	Specifies the action to perform when choosing to delete the message on successful processing. For more information on the available settings, see the Delete Behavior table below.
Google OAuth	Specifies the information for the server's Google OAuth configuration. This section matches the JSON object from a service account's JSON private setting.
Connection Options	Specifies the connection options available with this server. Each connection option has a user defined name. These options are referenced by the Workers' **Email Server > Connection Option** property. Multiple workers should not target the same option.

Delete Behavior

The following table lists the options for the Gmail Api 2LO **Delete Behavior** section.

Setting	Description
Move to trash	Moves the message to trash.
Archive	Removes the inbox label from the message.
Delete	Permanently deletes the message.

Connection Options

The **Connection Options** for Gmail Api 2LO contain the following

Setting	Description
Username	Specifies the username for the account that owns the monitored inbox. Multiple options should not target the same inbox.

Gmail 3LO

Gmail 3LO specifies the settings for a Gmail server using three-legged OAuth and the Gmail API.

The following table lists the settings for the **Gmail 3LO** section.

Setting	Description
Uri	Specifies the Gmail URL. This setting is optional.

Delete Behavior	Specifies the action to perform when choosing to delete the message on successful processing. For more information on the available settings, see the Delete Behavior table below.
Client ID	Specifies the ID generated when registering your Broker.
Connection Options	Specifies the connection options available with this server. Each connection option has a user defined name. These options are referenced by the Workers' **Email Server** > **Connection Option** property. Multiple workers should not target the same option. For more information on the available settings, see the Connection Options table below.

Delete Behavior

The following table lists the options for the Gmail API 3LO **Delete Behavior** section.

Setting	Description
Move to trash	Moves the message to trash.
Archive	Removes the inbox label from the message.
Delete	Permanently deletes the message.

Connection Options

The Connection Options for Gmail API 3LO contain the following

Setting	Description
Username	Specifies the username for the account that owns the monitored inbox. Multiple options should not target the same inbox.
Client Secret	Specifies the client secret for Gmail refresh token connections.
Refresh Token	Specifies the refresh token used for Gmail 3LO authentication. You must use the **Hyland Gmail Auth Client** to acquire your refresh tokens.

Storage Servers

The **Storage Servers** section specifies the set of storage servers used by the Workers configurations. Each section under Storage Servers represents a unique server that is named by its section property. This name is referenced by the Workers' **Storage Server > Server** property. Each server can be an S3 Server or a Local Server. Typically, a Broker will have one storage server using the S3 protocol with connection options for each desired storage bucket. The Broker can support multiple storage servers with any valid protocol. The bucket layout does not matter to the Broker, so there is no best practice for bucket layouts. Your layout only depends on your S3 server's best practices. Multiple workers can share a bucket, or each worker can have their own bucket.

Note: The Storage Servers must match your Perceptive Content Email Broker Connector Configuration.

S3 Server

The **S3 Server** section specifies the settings for storing email data on an Amazon S3 compatible server. See Appendix D: S3 permissions for information on the required S3 permissions.

The following settings are available for the **S3 Server** section.

Setting	Description
Endpoint	Specifies the IP address or DNS name of the storage server. This endpoint must include the server's port number. It must not include a **http://** or **https://** prefix. Use the **Enable Https** setting to control the endpoint's SSL behavior. Either the endpoint must specify a region, or the **Region** field must be configured. The HRCS plugin cannot validate this for you.
Region	Specifies the physical location of the server. By default, it is blank. If you are unsure of the location, or if you are using a local S3 server, leave it unset. Either the endpoint must specify a region, or the **Region** field must be configured. The HRCS plugin cannot validate this for you.
Enable Https	Controls the SSL settings used when communicating with the endpoint.
Connection Options	Specifies the connections options available with this server. Each connection option has a user-defined name. These options are referenced by the Workers' Storage Server Connection Option property. Multiple options may share their settings, and multiple workers may reference the same option. The setting is in the **Connection Options** section. • **Bucket** – Specifies the storage bucket to use on the Amazon S3 compatible server • **Authentication** – Specifies the authentication used by this connection option. This section contains a **Basic Authentication** sub-section that contains the settings **Username** and **Password**.

Local Server

The **Local** section specifies the settings for storing email data on the local machine.

The following setting is available for the **Local** section.

Setting	Description
Location	Specifies the path to the local disk storage directory. Because of directory control concerns, we recommend that multiple local servers do not share their storage directory. However, if you have multiple workers that should write to the same directory, then multiple workers may reference the same local storage server. Note that workers referencing a local server do not use a **Storage Server Connection Option**.

Message Queuing Servers

The **Message Queuing Servers** section specifies the set of message queuing servers used by the **Workers** configurations. Each section under **Message Queueing Servers** represents a unique server

that is named by its section property. The **Workers' RabbitMQ Message Queueing Server > MQServer Server** property references this name.

The Broker only supports **Advanced Message Queuing Protocol (AMQP)** compatible connections. Typically, the Broker will only have one **MQ Server** with a single **Connection Option** for the Workers. However, the Broker can support multiple Message Queuing Servers and multiple Connection Options.

RabbitMQ

The **RabbitMQ** section specifies the settings necessary to connect to an AMQP server. All settings in this section are optional. Any omitted settings will use the default behavior as described by RabbitMQ's API documentation. Note that if you wish to use RabbitMQ's API default authentication, you must configure and reference a **Connection Option** set to use **No Authentication**. The authorized accounts must have read and write permissions. Depending on your AMQP server, you may also need to give the users explicit permission to use the configured Virtual Host.

The following settings are available for the **RabbitMQ** section.

Setting	Description
Host Name	Specifies the name of the host used by this AMQP server.
Virtual Host	Specifies the virtual host used by this AMQP server.
Port	Specifies the port number exposed by your AMQP server. Type **-1** to use the RabbitMQ API's default port number.
Operation Timeout	Specifies the number of milliseconds past the expected operation durations the server should wait for AMQP operations. This setting must be between 0 and 60000 milliseconds (1 minute) or it must be completely omitted. If the setting value is 0 or omitted, then the server only waits for the expected duration of each operation. Should match the **[-][d.']hh':'mm':'ss['.'fffffff]** timespan format.
SSL	Specifies the SSL section contains the settings that control the SSL options for this server. For more information on the available settings, see the **SSL table** below.
Connection Options	Specifies the connections options available with this server. Each connection option has a user defined name. These options are referenced by the Workers' Message Queueing Server Connection Option property. Multiple options may share their settings, and multiple workers may reference the same option. For more information on the available settings, see the **Connection Options** table below.

SSL

The **Ssl** section is a subsection of the **RabbitMQ** section and specifies the settings that control the SSL options for the configured AMQP server connection.

The following settings are available for the **SSL** section.

<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>Enabled</td><td>Controls if this server should use the SSL settings during AMQP connections. If Enabled is false, then the server will not use SSL to connect to the AMQP server.</td></tr>
<tr><td>Certificate Path</td><td>The file path to the SSL certificate used by this server.</td></tr>
<tr><td>Certificate Passphrase</td><td>The password required to access the SSL certificate found at the Certificate Path. If your cert is not protected by password, then you may omit this setting.</td></tr>
<tr><td>Server Name</td><td>The name of the server used to generate the certificate found at the Certificate Path.</td></tr>
<tr><td>Acceptable Policy Errors</td><td>Specifies the for controlling any allowed SSL Policy errors If none of the flags are selected, then the server will not allow any certificate errors. If you want to maintain the safest connection, you should not enable any of these flags. However, you may need these flags if your AMQP server uses self-signed certs.
The following are the available flags for the Errors section.
• Remote Certificate Not Available
• Remote Certificate Name Mismatch
• Remote Certificate Chain Errors</td></tr>
<tr><td>SSL Protocols</td><td>Specifies the flags for controlling the allowed SSL protocol versions. If none of the flags are selected, then the server will not be able to connect using SSL.
The following table lists the available flags for the Protocols section.
<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>SSL 2.0</td><td>Specifies the SSL 2.0 protocol. SSL 2.0 has been superseded by the TLS protocol and is provided for backward compatibility only.</td></tr>
<tr><td>SSL 3.0</td><td>Specifies the SSL 3.0 protocol. SSL 3.0 has been superseded by the TLS protocol and is provided for backward compatibility only.</td></tr>
<tr><td>TLS 1.0</td><td>Specifies the TLS 1.0 security protocol. The TLS protocol is defined in IETF RFC 2246.</td></tr>
<tr><td>TLS 1.1</td><td>Specifies the TLS 1.1 security protocol. The TLS protocol is defined in IETF RFC 4346.</td></tr>
<tr><td>TLS 1.2</td><td>Specifies the TLS 1.2 security protocol. The TLS protocol is defined in IETF RFC 5246.</td></tr>
</table>
</td></tr>
</table>

Connection Options

The **Connection Options** section specifies the connections options available with this server. Each connection option has a user defined name. These options are referenced by the Workers' **Message Queueing Server** > **Connection Option** property. Multiple options may share their settings, and multiple workers may reference the same option.

<table>
<tr><th>Setting</th><th></th><th>Description</th></tr>
<tr><td>Authentication</td><td></td><td>The authentication used by this connection option. This section either contains a No Authentication section or a Basic Authentication section.

The following table lists the settings for the Authentication section.
<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>No Authentication</td><td>Represents a connection option that uses RabbitMQ's default authentication</td></tr>
<tr><td>Basic Authentication</td><td>Represents a connection option that uses basic authentication and includes the settings Username and Password.</td></tr>
</table>
</td></tr>
</table>

Workers

The **Workers** section specifies the set of configured workers that monitor email addresses for messages to store in each workers' storage server. Each section under **Workers** represents a unique worker that is named by its section property.

Each Worker needs a configured Message Queueing Server, Storage Server, and Email Server. Workers may share each server. Workers may share the **Connection Options** for **Message Queue Servers** and **Storage Servers**, but Workers cannot share **Connection Options** for **Email Servers** since this would cause multiple workers to monitor the same inbox. Every mail protocol was designed to only support one active user at a time, so the Broker would have indeterminate behavior if multiple workers attempted to use an inbox at the same time.

The following settings are available in the **Workers** section.

Setting	Description
Enabled	Controls whether the Broker should run this worker. If **Enabled** is **false**, then the worker will not execute.
Execution Interval	Controls how long the Broker waits between worker executions. If the execution interval elapses before a worker finishes processing its inbox, then the Broker allows the worker to continue without interruption and does not attempt to start the worker until after the next interval. Should match the **[-][d.']hh':'mm':'ss['.'fffffff]** timespan format. Defaults to the configured **System > Default Worker Execution Interval.**

Storage Server	A subsection that controls the server that this worker uses to store emails. For more information, see Storage Server.
Message Queueing Server	Controls the Message Queue server monitored by this worker. For more information, see Message Queueing Server.
Email Server	A subsection that controls the server that this worker monitors for emails. For more information, see Email Server.

Storage Server

The **Storage Server** section controls which storage server and connection option the worker uses for file storage for consumer services. The referenced connection option specifies which authentication to use and which bucket to store files in. The bucket layout does not matter to the Broker, so there is no best practice for bucket layouts. Your layout only depends on your S3 server's best practices. Multiple workers can share a bucket, or each worker can have their own bucket.

The following settings are available in the **Storage Server** section.

Setting		**Description**
Server		Specifies the name of the targeted storage server. This name must match a configured **Storage Server**. Multiple workers may share the same **Storage Server**.
Connection Option		Specifies the name of the connection option to use in the targeted Storage Server. Multiple workers may share the same Storage Server connection options to authenticate as the same user. If the worker targets a local storage server, then this Connection Option is unused.

Message Queueing Server

The **Message Queueing Server** section controls which MQ server, connection options, and queue the worker uses for message queuing to consumers consumer services. The referenced connection option specifies which authentication to use. The queue layout controls how the message data reaches the consuming services. Any workers that sends messages to the same queue are captured using the same consuming services, so any workers that should be captured in the same way can go to the same queue. You may specify a unique queue per worker, but you may need to duplicate your consumer configurations if they have the same capture configuration.

The following settings are available in the **MQ Server** section.

Setting		**Description**
Server		**Specifies** which Message Queuing Server to use for this worker's server connection. This name must match a configured **MQ** Server from the Message Queuing Servers section. Multiple workers may share the same **MQ** Server.
Connection Option		Specifies the name of the Connection Option to use in the targeted **MQ Server**. Multiple workers may share the same **MQ Server Connection Option**.

Queue		Specifies the name of the queue to use with the target **MQ Server**. This queue controls which Consumer worker captures the messages from this Broker worker. If multiple workers need the same capture behavior, then they may share a queue. Otherwise, the workers must have their own queues.

Email Server

The **Email Server** section controls which Email Server and connection option to use and specifies how the Email Account should behave during success and failure events.

The following settings are available in the **Email Server** section.

Setting		**Description**
Server		Specifies the name of the targeted **Email Server**. Multiple workers may share the same **Email Server**.
Connection Option		Specifies the name of the Connection Option to use in the targeted **Email Server**. Multiple workers may not share the same **Email Server Connection Option**.
Account Behavior		Controls the behavior of the email server account used by this worker. See the Account Behavior section for more information.

Account Behavior

The **Account Behavior** section controls the behavior of the email server account used by this worker. This section contains the following sub-sections.

- Failure Action
- Success Action
- Debug Flags

Failure Action

This section controls the behavior of the email server account when the Broker encounters a failure when processing a message.

The following options are available in the **Failure Action** section.

Setting		**Description**
No Action		Specifies that the Broker leaves failed message in the inbox to be reprocessed instead of moving a failed message to a failure archive. This option does not contain any settings.
Archive Message		The Broker moves failed messages to the specified failure archive instead of leaving a failed message in the inbox for reprocessing. This section contains a single field, **Archive**, which controls where the Broker should store the failed messages on the email server.

Success Action

This section controls the behavior of the email server account when the Broker successfully finishes processing a message.

The following options are available in the Success Action section.

Setting		Description
Delete Message		Specifies that the Broker deletes the successful message from the inbox. This section does not contain any settings.
Archive Message		The Broker moves successful messages to the specified archive instead of deleting the message from the inbox. This section contains the single field, **Archive**, which controls where the Broker should store the successful messages on the email server.

Debug Flags

The debug flags control behaviors for troubleshooting issues with the Broker. These settings require an attached debugger, and they may cause unsafe or unexpected behavior. They should never be used in a production environment.

Appendix A: Hyland Application Settings Utility (HASU)

This section gives a brief overview of how to use HASU, a command line utility, to register certificates to the required certificate store and write values to an existing property within a configuration file.

Register

The register command registers a valid x509Certificate2 into the cert store in CurrentUser:My store location for future encryption and decryption. This is required for Linux based environments.

The following table lists the available actions for register.

Action	Description
-p --password	Required. Specifies the password that protects the certificate.
--f --filePath	Required. Specifies the full filepath to the PFX certificate.
--verbose	Sets minimum LogLevel to Trace. The default is Error.

Example

```
./Hyland.Application.Settings.Utility register --filePath /etc/pki/tls/certs /sample-certificate.pfx --password YourPassword –verbose
```

Write

The write command stores the value to an existing property within the configuration file.

The following table lists the available actions for write.

Action	Description
-p, --property	Required. Property of the config file. Nested objects should be separated by a colon ':'.
-a, --applicationRoot	Required. Specifies the path to the application root.
--file	Required. Specifies the relative filepath to the JSON configuration from the application root.
-v, --value	Required. Specifies the value to protect.
-i, --inline	Required. Encrypts the value of the provided property in the provided file.
-r, --recursive	Required. Encrypt all of the values under the provided property in the provided file.
--verbose	Sets minimum LogLevel to Trace. The default is Error.
-f, --force	Forces creation of value if not present. The default is false.
--encrypt	Boolean flag to determine if the value should be encrypted before being stored. The default is false. If needed, you can read the encrypted settings stored in the configuration file.
-t, --thumbprint	Required for Linux. Specifies the certificate thumbprint used to hash the encryption key.

Example

```
./Hyland.Application.Settings.Utility write -a /opt/Hyland_Email_Broker --file
appsettings.json -p Remotes:RemoteConfigurationServers:Remote1:BearerToken -i --
thumbprint 03D724DD2666B9D858CAB84808372BAE82F89A36 --encrypt
```

Read

The read command displays the property value in the console output.

The following table lists the available actions for read.

Action	Description
-p, --property	Required. Property of the config file. Nested objects should be separated by a colon ':'.
-a, --applicationRoot	Required. Specifies the path to the application root.
--file	Required. Specifies the relative file path to the JSON configuration from the application root.
--verbose	Sets minimum Log Level to Trace. The default is Error.
--decrypt	Boolean flag to determine if the value should be decrypted before being retrieved. The default is false.

Example

```
./Hyland.Application.Settings.Utility read -a /opt/broker --file appsettings.json -p
Remotes:RemoteConfigurationServers:Remote1:BearerToken
```

Appendix B: Remote HRCS connection best practices and behavior

Behavior

The Broker adds a connection for all configuration references that start with the **HylandLogging** prefix. These configurations are added under the **Hyland Logging** section, and they must have a **Hyland.Logging** configuration type or the Broker will fail to retrieve the configurations.

The Broker adds a connection for all configuration references that start with the **EmailBroker** prefix. These configurations are added directly to the root of the configuration. They are expected to be reference instances version **HylandEmailBrokerV3** of the **Hyland-Email-Broker** configuration type.

The order of the connections and the order of the reference IDs control the order that the remote configurations are applied to the Broker, so for any conflicting settings the Broker uses the value from the last connection. Unique values are preserved from each connection.

All the **Hyland Logging** configurations are added before the **EmailBroker** configurations, so any Hyland.Logging settings in any **EmailBroker** configuration take precedent over the settings from any **HylandLogging** configuration.

Example

Given:

```
"Remotes": {
    "RemoteConfigurationServers": {
        "<UserDefined-Remote>": {
            "ConfigurationReferenceIds": {
                "EmailBroker-Config1": "example-broker-id-1",
                “EmailBroker-Config2”: “example-broker-id-2”
            },
            "Host": "example-hrcs-host",
            "Port": "5001",
            “Route”: “example-hrcs-client-route”,
            "BearerToken": "example-bearer-token",
            "ReconnectDelay": "00:00:10"
        }
    }
}
```

Or given:

```
"Remotes": {
    "RemoteConfigurationServers": {
        "<UserDefined-Remote-1>": {
            "ConfigurationReferenceIds": {
                "EmailBroker ": "broker-id-1"
            },
            "Host": "example-hrcs-host-1",
            "Port": "5001",
            “Route”: “example-hrcs-client-route-1”,
            "BearerToken": "example-bearer-token-1",
            "ReconnectDelay": "00:00:10"
        },
        "<UserDefined-Remote-2>": {
            "ConfigurationReferenceIds": {
```

```
                “EmailBroker”: “broker-id-2”
            },
            "Host": "example-hrcs-host-2",
            "Port": "5001",
            “Route”: “example-hrcs-client-route-2”,
            "BearerToken": "example-bearer-token",
            "ReconnectDelay": "00:00:10"
        }a
    }
}
```

With `broker-id-1` settings:

```
{
  "key1": "b1-v1",
  "key2": {
    "subkey1": "b1-v2-1",
    “subkey2”: “b1-v2-2”
  },
  “key3”: “b1-v3”
}
```

With `broker-id-2` settings:

```
{
  "key1": "b2-v1",
  "key2": {
    “subkey2”: “b2-v2-2”
  },
  “key4”: “b2-v4”
}
```

The resulting configuration would be

```
{
  "key1": "b2-v1",
  "key2": {
    "subkey1": "b1-v2-1",
    “subkey2”: “b2-v2-2”
  },
  “key3”: “b1-3”,
  “key4”: “b2-v4”
}
```

Best practices

The Broker allows multiple Remotes and ConfigurationReferenceIds to have as much flexibility as possible. However, you must be careful of the layer behavior when using more than one of each. Typically, your Broker should have a single **RemoteConfigurationServers** entry with a single **EmailBroker ConfigurationReferenceIds** and a single, optional **HylandLogging** section. Using only one configuration reference with one remote prevents unintended configuration layering behavior. See Appendix C for information on when you should include a **HylandLogging** section.

You may use the **prefix** behavior to store multiple references for easy configuration switching **while testing**. However, you should remove any unused references and remotes from production environments to prevent accidentally using the wrong configuration.

Example

The following example demonstrates a **Remotes** section with two reference IDs. You can switch between them easily by adding or removing the `x` from the reference ID names. Note that you must restart your Broker when switching for the changes to take effect.

```
"Remotes": {
    "RemoteConfigurationServers": {
        "<UserDefined-Remote>": {
            "ConfigurationReferenceIds": {
                "EmailBroker-TestEnv1": "example-test-settings-1",
                “xEmailBroker-TestEnv2”: “example-test-settings-2”
            },
            "Host": "example-hrcs-host",
            "Port": "5001",
            "BearerToken": "example-bearer-token",
            "ReconnectDelay": "00:00:10"
        }
    }
}
```

Appendix C: Logging best practices

Typically, all logging settings should be configured in HRCS using either the **Email Broker** plugin or the **Hyland.Logging** plugin. The **Hyland Logging** section of the **Email Broker** configuration should have any instance or service specific logging settings. The **Hyland.Logging** plugin configuration should have any logging settings that you wish to share across multiple instances or services such as a shared Splunk logging configuration.

If you do not have any cross service/instance logging settings, then you should omit **HylandLogging** from your remotes. If you have cross service/instance logging settings, then you need to be mindful of the layering behavior when configuring both **Email Broker** plugin logging and **Hyland.Logging.** Generally, it is safest to only have **Email Broker** plugin logging settings **or Hyland.Logging** plugin settings.

Additionally, the Broker supports configuring Hyland.Logging directly in the **appsettings.json** to facilitate debugging your **Remotes** connections. However, the **appsettings.json** logging settings should be removed once you properly establish your **Remotes** connections since the appsettins.json settings may interfere with the remote settings due to layering behavior.

Example

The following is an example of the **Hyland.Logging** section for your appsettings.json.

```
"Hyland.Logging": {
    "Routes": {
        "Console": {
            "Console": "",
            "minimum-level": "Information",
            "maximum-level": "Critical"
        },
        "All-Logs": {
            "file": "example-valid-path/file-name",
            "minimum-level": "Debug",
            “FileRollOnSize”: true,
            “FileRollInterval”: “day”,
            “FileByteLimit”: 100000000,
            “FileCountLimit”: 50,
            “OutputFormat”: “text”
        }
```

```
    }
}
```

Appendix D: S3 permissions

Appendix D: S3 permissions

Your S3 account needs to be able to access the following S3 APIs to function.

- HeadBucket - https://docs.aws.amazon.com/AmazonS3/latest/API/API_HeadBucket.html
- ListBuckets - https://docs.aws.amazon.com/AmazonS3/latest/API/API_ListBuckets.html
- ListObjects - https://docs.aws.amazon.com/AmazonS3/latest/API/API_ListObjects.html
- CreateBucket - https://docs.aws.amazon.com/AmazonS3/latest/API/API_CreateBucket.html
- GetObject - https://docs.aws.amazon.com/AmazonS3/latest/API/API_GetObject.html
- PutObject - https://docs.aws.amazon.com/AmazonS3/latest/API/API_PutObject.html
- DeleteObject - https://docs.aws.amazon.com/AmazonS3/latest/API/API_DeleteObject.html
- HeadObject - https://docs.aws.amazon.com/AmazonS3/latest/API/API_HeadObject.html
- GetBucketLocation - https://docs.aws.amazon.com/AmazonS3/latest/API/API_GetBucketLocation.html

The permissions mechanisms vary based on your S3 server implementation, so contact your S3 administrator to properly configure permissions.

Appendix E: Email Agent upgrade

The Broker's HRCS plugin upgrades any Agent ini settings with an equivalent Broker Hyland Remote Configuration (HRC) setting. These upgraded settings match the original behavior as close as possible, and the upgrader generates settings for new options that do not exist in the original **ini.** This upgrade creates a worker for each Agent ini Profile ({Profile}) and creates shared Email Servers for each unique **server.incoming** in the Profiles.

The Connector Plugin upgrades all possible Email Agent ini settings as they are written in the ini file, so validity is not guaranteed. You may need to correct your upgraded configuration before using it if the original settings were not valid.

{profile} is the section heading for a profile in the Email Agent ini file.

The Broker upgrader uses the following Email Agent ini sections and settings groups.

- [Logging] > *
  - With [{profile }] > createprofilelog
- [{profile}] > server.incoming.*

The upgrade process requires an HRCS administrator to create a **Hyland-Email-Broker** type configuration. During the upgrade you may encounter upload or upgrade failures, see Upload and Upgrade Failures for information on resolving these issues. To upgrade an Email Agent ini file to Email Broker configuration using a newly created **Hyland-Email-Broker** type configuration, perform the following steps.

1. Open your created configuration.
2. Upload a **Version: EmailAgent** configuration.
3. Upgrade to the **Version: HylandEmailBrokerV3** configuration from Source Version **EmailAgent**.

Upload and upgrade failures

While attempting to upload an Agent ini file or upgrading to a Broker configuration, you may encounter failures. The following tables outline some common failures and how to resolve them

Upload failures

Failure	Solution
"Invalid ini key: {key}" where **{key}** contains **':'**	Remove the **':'** character from the ini section key. For example, "[Section:1]" would need to be "[Section – 1]" or something similar. The HRCS ini parser does not support ini sections containing **':'**. If you encounter this error, you will likely see the error multiple times since an invalid ini section also invalidates all the properties in that section.

Upgrade failures

Failure	Solution
Logger has an unsupported rollover period "WEEK". Supported periods are: MINUTE, HOUR, DAY, MONTH.	Reconfigure the Agent logger to use one of the supported periods or switch to a size-based rollover. Hyland Logging does not support the WEEK rollover period. HRCS requires you to switch to a supported period or to size-based rollover prior to upgrading to avoid any unintended log rolling behavior.
Failed to upgrade "{Profile}" email server. Failure: "Protocol "POP3" is not supported. Supported Protocols are: [EWS, EWSCS, EWSROPC, GIMAP, GTOKENIMAP, IMAP, IMAPS]."	Reconfigure the **{Profile}** profile in the Agent ini to use a supported protocol or remove the offending **{Profile}** profile from the Agent ini. The Broker does not support POP3. HRCS requires you to switch to a supported protocol or to remove the unsupported profile prior to upgrade to avoid any unintended Email Server interactions.

Upgraded – [Logging]

The **Hyland Logging** upgrade creates new routes that best match the Agent's original [Logging] section. If your agent has logging disabled, then the upgrade will not create any routes. The only way to disable Hyland Logging is to remove all Routes. If your agent has logging enabled, then the upgrade creates a

Route named **EmailAgentRoute** to handle the global logging. Additionally, the upgrade creates a route for each Email Agent profile, **[{profile}]**, with **createprofilelog=true** to maintain the Agent's original logging behavior. The profile Routes are clones of the global **EmailAgentRoute** with the original **{profile}** as their name and **Included Profiles** to only include the **{profile}**. If your deployment intends to only use Splunk for logging, you must delete these routes to prevent the Broker from writing log files. All logging settings may be configured to meet your deployment requirements. These initial settings emulate the Agent's original behavior.

The upgrade process configures some common settings along with additional settings based on if the Agent's **[Logging] > policy.type** field was size or time.

Common logger upgrade

HRC Path: Hyland Logging > Routes > {Route}

[Logging] > Ini Setting	HRC Setting	Notes
level	Minimum Level	Note that if the Agent's **[Logging] > level** is off, then the upgrader will not create any routes.
	File	Based on the Agent's log file name convention. • If the route is the global **EmailAgentRoute**, then the value is email.agent.all.log. • If the route is a **{Profile}** route, then the value is **email.agent.{Route}.log**.
	Maximum Level	The upgrader sets **critical** to match the Agent's Logging behavior
	Output Format	The upgrader sets **json** to support machine readable logs.
	Include Profiles	• If the route is the global **EmailAgentRoute**, then the value is empty. • If the route is a **{Profile}** route, then the value is **{Profile}**.

[Logging] > policy.type - size upgrades

HRC Path: Hyland Logging > Routes > {Route}

[Logging] > Ini Setting	HRC Setting	Notes
policy.size.maxlogstokeep	File Count Limit	
policy.size.maxmbsize	File Byte Limit	The Agent uses a megabyte size while Hyland Logging uses a byte size, so the upgraded value will be 1,000,000 times the Agent's value.
	File Roll on Size	The upgrader sets **true** to enable size rolling.

[Logging] > policy.type - time upgrades

HRC Path: Hyland Logging > Routes > {Route}

[Logging] > Ini Setting	HRC Setting	Notes

policy.time.maxhistory	File Count Limit	
policy.time.rolloverperiod	File Roll Interval	Note the upgrade will fail if **[Logging] > policy.time.rolloverperiod** is **week**. Hyland logging does not support week rollover periods, so you must change your rollover period prior to upgrading.

Upgraded - [{Profile}] > server.incoming

The **Email Servers** upgrade creates a single Email Server section for each unique server in the original **ini** to eliminate any duplicate Email Server configurations. Each new server has a **Connection Option** for each unique account for the original server. These servers are referenced by the Workers' **Email Server > Server** field, and these **Connection Options** are referenced by the Workers' **Email Server > Connection Option** field.

The exact field migrations vary by the original email protocols. The following sections list the migration mappings for each supported protocols' Email Server and Connection Options.

[{Profile}] > server.incoming.protocol - EWS

The original **EWS** protocol upgrades to the new UI's **EWS** option. The following tables list the new option field and the original ini setting for the Email Server and Connection Option upgrade.

HRC Path: Email Servers > {Email Server} > EWS

[{Profile}] Ini Setting	HRC Setting	Notes
server.incoming	Host	
	Delete Behavior	Delete Behavior is not configurable in the Agent ini. The upgrade sets this to **Move to deleted items** to match the original Agent's behavior. You may change this setting at any time.

HRC Path: Email Servers > {Email Server} > EWS > Connection Options > {Email Server Connection Option}

[{Profile}] Ini Setting	HRC Setting
server.incoming.username	Username
server.incoming.password	Password

[{Profile}] > server.incoming.protocol - EWSCS and EWSROPC

The original EWSCS protocol upgrades to the UI's **Microsoft 365** option. The following tables list the new option field and the original ini setting for the Email Server and Connection Option upgrade.

Note that the Broker does not support the ROPC authentication mechanism, so you must convert your Azure application to use Client Secret authentication.

HRC Path: Email Servers > {Email Server} > Microsoft 365

[{Profile}] Ini Setting	HRC Setting	Notes

server.incoming	Host	
server.incoming.clientid	Client ID	
server.incoming.tenantid	Tenant ID	
	Delete Behavior	Delete Behavior is not configurable in the Agent ini. The upgrade sets this to **Move to deleted items** to match the original Agent's behavior. You may change this setting at any time.

HRC Path: Email Servers > {Email Server} > Microsoft 365 > Connection Options > {Email Server Connection Option}

[{Profile}] Ini Setting	HRC Setting
server.incoming.username	Username
server.incoming.clientsecret	Client Secret

[{Profile}] > server.incoming.protocol - GIMAP

The original GIMAP protocol upgrades to the UI's **IMAP** option with the **Gmail 2LO IMAP** option. The following tables list the new option field and the original ini setting for the Email Server and Connection Option upgrade.

The **GIMAP** upgrade requires manual configuration when upgrading. All other protocols do not require manual configuration. The upgraded GIMAP protocol uses an IMAP Email Server with the Gmail 2LO authentication mechanism. This mechanism requires authentication information that was stored outside of the Agent ini. To finish your Email Server's configuration, you must configure **Email Servers > {Email Server Name} > IMAP > Gmail 2LO IMAP > Google OAuth** to contain the contents of the file referenced by the Agent **[{Profile}] > server.incoming.credentialspath**.

HRC Path: Email Servers > {Email Server} > IMAP > Gmail 2LO IMAP

[{Profile}] Ini Setting	HRC Setting	Notes
server.incoming	Host	
server.incoming.port	Port	
	Google OAuth	The upgrader cannot populate the **Google OAuth** setting for you since the Agent's ini did not contain the actual data, and it only contained a file path to the OAuth settings. You must manually add the content from the file referenced by the ini's **[{Profile}] > server.incoming.credentialspath** setting to the **Google OAuth** field.

HRC Path: Email Servers > {Email Server} > IMAP > Gmail 2LO IMAP > Connection Options > {Email Server Connection Option}

[{Profile}] Ini Setting	HRC Setting

server.incoming.username	Username

[{Profile}] > server.incoming.protocol - GTOKENIMAP

The original GTOKENIMAP protocol upgrades to the UI's **IMAP** option with the **Gmail 3LO IMAP** option. The following tables list the new option field and the original ini setting for the Email Server and Connection Option upgrade.

HRC Path: Email Servers > {Email Server} > IMAP > Gmail 3LO IMAP

[{Profile}] Ini Setting	HRC Setting
server.incoming	Host
server.incoming.port	Port
server.incoming.clientid	Client ID

HRC Path: Email Servers > {Email Server} > IMAP > Gmail 3LO IMAP > Connection Options > {Email Server Connection Option}

[{Profile}] Ini Setting	HRC Setting
server.incoming.username	Username
server.incoming.clientsecret	Client Secret
server.incoming.refreshtoken	Refresh Token

[{Profile}] > server.incoming.protocol - IMAP and IMAPS

Both IMAP and IMAPS protocols upgrade to the UI's **IMAP** option with the **Basic Authentication IMAP** option. The following tables list the new option field and the original ini setting for the Email Server and Connection Option upgrade.

HRC Path: Email Servers > {Email Server} > IMAP > Basic Authentication IMAP

[{Profile}] Ini Setting	HRC Setting
server.incoming	Host
server.incoming.port	Port

HRC Path: Email Servers > {Email Server} > IMAP > Basic Authentication IMAP > Connection Options > {Email Server Connection Option}

[{Profile}] Ini Setting	HRC Setting
server.incoming.username	Username
server.incoming.password	Password

Email Servers Naming

The new Email Server names are derived from their settings to prevent naming collisions. The server naming format is
`"{server.incoming.protocol}__host{server.incoming}__port{server.incoming.port}__clientId{server.incoming.clientid}__tenantId{server.incoming.tenantid}"`
where each `{server.incoming.*}` is the value of the ini setting for a [Profile}] with any ':' characters replaced by "__". Any empty ini settings are omitted from the derived name.

You may rename Email Servers at any time. However, you must also update any **Workers > {Profile} > Email Server > Server** fields that referenced the original setting.

Email Server Connection Options Naming

The Connection Option derived naming format is "`{server.incoming.username}-settings`".

You may rename Email Server Connection Options at any time. However, you must also update any **Workers > {Profile} > Email Server > Connection Option** fields that referenced the original setting.

Example Email Server and Connection Option Names

The following scenarios demonstrate a few examples of upgrade Email Server and Connection Option names.

Senario 1 – Minimal IMAPS setting

- Given the following ini settings.
  ```
  server.incoming.protocol=imaps
  server.incoming=example1.com
  server.incoming.username=account1@example1.net
  ```
- The derived Email Server name would be `IMAPS-host__example1.com`
- The derived connection option name would be `account1@example1.net-settings`

Scenario 2 – Full EWSCS settings

- Given the following ini settings.
  ```
  server.incoming.protocol=ewscs
  server.incoming=https://outlook.office365.com/EWS/Exchange.asmx
  server.incoming.clientid=example-client-id
  server.incoming.tenantid=example-tenant-id
  server.incoming.username=account2@example2.onmicrosoft.com
  ```
- The derived Email Server name would be `EWSCS-host__https__//outlook.office365.com/EWS/Exchange.asmx-clientid__example-client-id-tenantid__example-tenant-id`
- The derived connection option name would be `account2@example2.onmicrosoft.com-settings`

Generated - Storage Servers

The **Storage Servers** do not have an equivalent ini setting, so the upgrade process generates a single storage server named **Storage Server 1** with the **S3** option, and it generates a single **Connection Option** named **Storage Option 1** for **Storage Server 1.** The upgraded Workers are automatically configured to use this server and option. Also, the **Perceptive Content Email Broker Connector** uses the same storage server name and option during its upgrade process. You may rename this server and

option at any time, but you must also update any workers and any settings in the Perceptive Content Email Broker Connector that reference the original values.

You must manually configure the generated server, **Storage Servers > Storage Server 1 > S3**, and the generated Connection Option, **Storage Servers > Storage Server 1 > S3 > Connection Options > Storage Option 1**, to match your S3 server and account before you can use the broker. If your deployment uses a **Local** server, then you must reconfigure **Storage Servers > Storage Server 1** to use the **Local** option and configure a **Location** (file system location) on the **Local** server.

Generated - Message Queueing Servers

The **Message Queueing Servers** do not have an equivalent ini setting, so the upgrade process generates a single MQ server named **MQ Server 1** with a single **Connection Option** named **MQ Option 1**. The upgraded Workers are automatically configured to use this server and option. Also, the **Perceptive Content Email Broker Connector** uses the same MQ server name and option during its upgrade process. You may rename this server and option at any time, but you must also update any workers and any settings in the Perceptive Content Email Broker Connector that reference the original values.

You must manually configure the generated server, **Message Queueing Servers > MQ Server 1**, and the generated Connection Option, **Message Queueing Servers > MQ Server 1 > Connection Options > MQ Option 1**, to match your AMQP Server and account before you can use the broker.

Generated - Workers

The **Workers** upgrade creates a Broker Worker for each Email Agent Profile.

Worker Naming

The generated Workers use the same names as the Agent Profiles.

Storage Server

The **Storage Server** upgrade configures the Worker to target the generated Storage Server and Connection Option.

HRC Path: Workers > {Profile} > Storage Server

HRC Setting	Value	Notes
Server	Storage Server 1	References the generated placeholder configuration **Storage Servers > Storage Server 1**
Connection Option	Storage Option 1	References the generated placeholder configuration **Storage Servers > Storage Server 1 > Connection Options > Storage Option 1**

Message Queueing Server

The **Message Queueing Server** upgrade configures the Worker to target the generated Message Queueing Server and Connection Option, and configures a queue name. This Queue Name is derived from the Agent's Profile name using the format of `"EmailBroker_{Profile}"` where {Profile} is the name of the upgraded Agent Profile. To maintain the Agent's original behavior, this queue name matches the queue name generated by the Perceptive Content Email Broker Connector's Worker upgrade. You may change this queue name at any time, but if you wish to maintain the original behavior, you must also

update the Broker Connectors configuration. Alternatively, this queue may be shared across multiple Broker workers to allow the Broker workers to share the Broker Connector workers.

HRC Path: Workers > {Profile} > Message Queueing Server

HRC Setting	Value	Notes
Server	MQ Server 1	References the generated placeholder configuration **Message Queueing Servers > MQ Server 1**
Connection Option	MQ Option 1	References the generated placeholder configuration **Message Queueing Servers > MQ Server 1 > Connection Options > MQ Option 1**
Queue	EmailBroker_`{Profile}`	Where `{Profile}` is the name of the upgraded Agent Profile.

Email Server

The **Email Server** upgrade configures the Worker to target the upgraded Email Server and Connection Option that was used by the original Agent Profile. Additionally, the upgrade configures the **Account Behavior** to match the Agent's original behavior.

HRC Path: Workers > {Profile} > Email Server

HRC Setting	Value	Notes
Server	{Email Server}	References the upgraded configuration **Email Servers > {Email Server}**
Connection Option	{Email Server Connection Option}	References the upgraded configuration **Email Servers > {Email Server} > Connection Options > {Email Server Connection Option}**

HRC Path: Workers > {Profile} > Email Server > Account Behavior

HRC Setting	Value	Notes
Failure Action	No Action or Archive Message	The value for Failure Action depends on whether the Agent ini setting, **movefailedmessage** for the profile is true or false prior to the upgrade. If **movefailedmessage** = **false**, then the HRC setting, **Failure Action**, will be equal to **No Action** after the upgrade. If **movefailedmessage** = **true**, then the HRC setting, **Failure Action**, will be equal to **Archive Message** after the upgrade. Also, in this scenario, the value of **Failure Action** > **Archive** will be equal to the value of **movefailedmessage.foldername** (ini setting) after the upgrade. See the new Failure Action setting for more information.

Success Action	Delete Message	**Delete Message** matches the original Agent behavior to delete successful messages. See the new Success Action setting for more information.