Contents

[**Egress endpoints** 1](#_Toc138403028)

[**Configure egress endpoints** 1](#_Toc138403029)

[**Egress endpoint configuration schema** 2](#_Toc138403030)

[**Egress endpoint parameters** 2](#_Toc138403031)

[**Special characters encoding** 4](#_Toc138403032)

[**Examples** 5](#_Toc138403033)

[**Egress data to AVEVA Data Hub** 5](#_Toc138403034)

[**Egress data to PI Web API** 5](#_Toc138403035)

[**Egress data to PI Web API using a valid secret Id** 5](#_Toc138403036)

[**Egress data to PI Web API using negotiate** 6](#_Toc138403037)

[**REST URLs** 6](#_Toc138403038)

[**Egress execution details** 7](#_Toc138403039)

[**Prepare egress destinations 8**](#_Toc138403040)

[**AVEVA Data Hub 8**](#_Toc138403041)

[**AVEVA Server 8**](#_Toc138403042)

**Egress endpoints**

AVEVA Adapters collect time series data, which they can send to a permanent data store (endpoint). This operation is called data egress. The following endpoints are available for data egress:

* AVEVA Data Hub (ADH)
* PI servers through PI Web API

For long term storage and analysis, you can configure any adapter to send time series data to one or several of these endpoints in any combination. An egress endpoint is comprised of the properties specified under [Egress endpoint parameters](https://github.com/osisoft/PI-Adapter/blob/main/shared-content/configuration/egress-endpoints.md#egress-endpoint-parameters).

Data egress to a PI server creates a PI point in the AVEVA Adapter configuration. Data egress to AVEVA Data Hub creates a stream in the AVEVA Adapter configuration.

The name of the PI point or AVEVA Data Hub stream is a combination of the StreamIdPrefix specified in the adapter data source configuration and the StreamId specified in the adapter data selection configuration.

**Configure egress endpoints**

Complete the following steps to configure egress endpoints. Use the PUT method in conjunction with the http://localhost:5590/api/v1/configuration/OmfEgress/dataendpoints REST endpoint to initialize the configuration.

1. Using a text editor, create an empty text file.
2. Copy and paste an example configuration for egress endpoints into the file.

For sample JSON, see [Examples](https://github.com/osisoft/PI-Adapter/blob/main/shared-content/configuration/egress-endpoints.md#examples).

1. Update the example JSON parameters for your environment.

For a table of all available parameters, see [Egress endpoint parameters](https://github.com/osisoft/PI-Adapter/blob/main/shared-content/configuration/egress-endpoints.md#egress-endpoint-parameters).

1. Save the file. For example, as ConfigureEgressEndpoints.json.
2. Open a command line session. Change directory to the location of ConfigureEgressEndpoints.json.
3. Enter the following cURL command (which uses the PUT method) to initialize the egress endpoints configuration.

curl -d "@ConfigureEgressEndpoints.json" -H "Content-Type: application/json" -X PUT "http://localhost:5590/api/v1/configuration/OmfEgress/dataendpoints"

**Notes:**

* + If you installed the adapter to listen on a non-default port, update 5590 to the port number in use.
  + For a list of other REST operations you can perform, like updating or replacing an egress endpoints configuration, see [REST URLs](https://github.com/osisoft/PI-Adapter/blob/main/shared-content/configuration/egress-endpoints.md#rest-urls).

**Egress endpoint configuration schema**

The full schema definition for the egress endpoint configuration is in the OmfEgress\_DataEndpoints\_schema.json file located in one of the following folders:

Windows: %ProgramFiles%\OSIsoft\Adapters\<AdapterName>\Schemas

Linux: /opt/OSIsoft/Adapters/<AdapterName>/Schemas

**Egress endpoint parameters**

The following parameters are available for configuring egress endpoints:

| **Parameter** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| **Id** | Optional | string | Unique identifier Allowed value: any string identifier Default value: new GUID |
| **Endpoint** | Required | string | Destination that accepts OMF v1.2 messages. Supported destinations include AVEVA Data Hub and AVEVA Server. Allowed value: well-formed http or https endpoint string Default: null |
| **Username** | Optional for PI server endpoint | string | Basic authentication to the PI Web API OMF endpoint *PI server:* Allowed value: any string Default: null **Note:** If your username contains a backslash, you must add an escape character, for example, type OilCompany\TestUser as OilCompany\\TestUser. **Note:** If neither Username nor ClientID is supplied, it will use Negotiation instead (i.e. Kerberos, NTLM) |
| **Password** | Optional for PI server endpoint | string | Basic authentication to the PI Web API OMF endpoint *PI server:* Allowed value: any string or {{<secretId>}} (see Reference Secrets) Default: null |
| **ClientId** | Required for AVEVA Data Hub endpoint | string | Authentication with the AVEVA Data Hub OMF endpoint Allowed value: any string, can be null if the endpoint URL schema is HTTP Default: null |
| **ClientSecret** | Required for AVEVA Data Hub endpoint | string | Authentication with the AVEVA Data Hub OMF endpoint Allowed value: any string or {{<secretId>}} (see Reference Secrets); can be null if the endpoint URL schema is HTTP Default: null |
| **DebugExpiration** | Optional | string | Enables logging of detailed information to disk for each outbound HTTP request pertaining to the egress endpoint. The value represents the date and time this detailed information should stop being saved. Examples of valid strings representing date and time: UTC: yyyy-mm-ddThh:mm:ssZ, Local: yyyy-mm-ddThh:mm:ss. For more information, see Egress debug logging. Default: null |
| **TokenEndpoint** | Optional for AVEVA Data Hub endpoint | string | Retrieves an AVEVA Data Hub token from an alternative endpoint Allowed value: well-formed http or https endpoint string Default value: null |
| **ValidateEndpointCertificate** | Optional | boolean | Disables verification of destination certificate. **Note:** Only use for testing with self-signed certificates. Allowed value: true or false Default value: true |

**Special characters encoding**

The adapter encodes special characters used in the data selection **StreamId** parameter string before sending it to configured endpoints. The encoded characters look as follows:

| **Special character** | **Encoded character** |
| --- | --- |
| \* | %2a |
| ' | %27 |
| ` | %60 |
| " | %22 |
| ? | %3f |
| ; | %3b |
| | | %7c |
| \ | %5c |
| { | %7b |
| } | %7d |
| [ | %5b |
| ] | %5d |

**Examples**

The following examples are valid egress configurations:

**Egress data to AVEVA Data Hub**

[{

"Id": "AVEVA Data Hub",

"Endpoint": "https://<AVEVA Data Hub OMF endpoint>",

"ClientId": "<clientid>",

"ClientSecret": "<clientsecret>"

}]

**Egress data to PI Web API**

[{

"Id": "PI Web API",

"Endpoint": "https://<pi web api server>:<port>/piwebapi/omf/",

"UserName": "<username>",

"Password": "<password>"

}]

**Egress data to PI Web API using a valid secret Id**

See Reference Secrets for more information on how to use a secret Id.

[{

"Id": "PI Web API",

"Endpoint": "https://<pi web api server>:<port>/piwebapi/omf/",

"UserName": "<username>",

"Password": "{{<secretId>}}"

}]

**Egress data to PI Web API using negotiate**

See Reference Secrets for more information on how to use a secret Id.

[{

"Id": "PI Web API",

"Endpoint": "https://<pi web api server>:<port>/piwebapi/omf/"

}]

**REST URLs**

| **Relative URL** | **HTTP verb** | **Action** |
| --- | --- | --- |
| api/v1/configuration/omfegress/DataEndpoints | GET | Gets all configured egress endpoints |
| api/v1/configuration/omfegress/DataEndpoints | DELETE | Deletes all configured egress endpoints |
| api/v1/configuration/omfegress/DataEndpoints | POST | Adds an array of egress endpoints or a single endpoint. Fails if any endpoint already exists |
| api/v1/configuration/omfegress/DataEndpoints | PUT | Replaces all egress endpoints |
| api/v1/configuration/omfegress/DataEndpoints | PATCH | Allows partial updating of configured endpoints. **Note:** The request must be an array containing one or more endpoints. Each endpoint in the array must include its *Id*. |
| api/v1/configuration/omfegress/DataEndpoints/{Id} | GET | Gets configured endpoint by *Id* |
| api/v1/configuration/omfegress/DataEndpoints/{Id} | DELETE | Deletes configured endpoint by *Id* |
| api/v1/configuration/omfegress/DataEndpoints/{Id} | PUT | Updates or creates a new endpoint with the specified *Id* |
| api/v1/configuration/omfegress/DataEndpoints/{Id} | PATCH | Allows partial updating of configured endpoint by *Id* |

**Egress execution details**

* After configuring an egress endpoint, egress is immediately run for that endpoint. Egress is handled individually per configured endpoint. When data is egressed for the first time, types and containers are egressed to the configured endpoint. After that only new or changed types or containers are egressed. Type creation must be successful in order to create containers. Container creation must be successful in order to egress data.
* If you delete an egress endpoint, data flow immediately stops for that endpoint. Buffered data in a deleted endpoint is permanently lost.
* Type, container, and data items are batched into one or more OMF messages when egressing. As per the requirements defined in OMF, a single message payload will not exceed 192KB in size. Compression is automatically applied to outbound egress messages. On the egress destination, failure to add a single item results in the message failing. Types, containers, and data are egressed as long as the destination continues to respond to HTTP requests.

# Prepare egress destinations

AVEVA Data Hub and AVEVA Server destinations may require additional configuration to receive OMF messages.

## AVEVA Data Hub

To prepare OCS to receive OMF messages from the adapter, create an OMF connection in AVEVA Data Hub. Creating an OMF connection results in an available OMF endpoint that can be used by the adapter egress mechanism. Complete the following steps to create an OMF connection:

1. Create a **Client**.

The Client Id and Client Secret will be used for the corresponding properties in the egress configuration.

1. Create an **OMF** type **Connection**.

The connection should link the created client to an existing [namespace](https://docs.osisoft.com/bundle/ocs/page/set-up/namespaces/namespaces-concept.html) where the data will be stored.

The **OMF Endpoint** URL for the connection will be used as the egress configuration Endpoint property.

## AVEVA Server

To prepare a AVEVA Server to receive OMF messages from the adapter, a PI Web API OMF endpoint must be available. Complete the following steps:

1. Install PI Web API and enable the **Open Message Format (OMF) Services** feature.
   * During configuration, choose an AF database and PI Data Archive where metadata and data will be stored.
   * The account used in an egress configuration needs permissions to create AF elements, element templates, and PI points.
2. Configure PI Web API to use Basic and/or Kerberos authentication.

For complete steps, as well as best practices and recommendations, see the following topic in the PI Web API User Guide: [Authentication methods](https://docs.osisoft.com/bundle/pi-web-api/page/authentication-methods.html).

**Notes:**

* The certificate used by PI Web API must be trusted by the device running the adapter, otherwise the egress configuration ValidateEndpointCertificate property needs to be set to false (this can be the case with a **self-signed certificate** but should only be used for testing purposes).
* To continue to send OMF egress messages to the PI Web API endpoint after upgrading PI Web API, restart the adapter service.