TIBCO ActiveMatrix® Service Grid - Container Edition

Administration

Version 1.0.0
December 2020

Document Updated: May 2021

Contents

Introduction to Application Configurator

Application Configurator is a web application to configure different application entities and generate application configuration YAML file. Application configuration YAML file is required to containerize an application. You can also generate the Log4j configuration file and the JVM configuration file for an application.

The generated configuration is used to create a Docker image of the application. Application Configurator runs as a Docker container. The script to build Application Configurator image is provided at `amsgce-runtime-<version>\applicationConfigurator\build`. For more information about building Application Configurator container and running it, see *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*. You must upload the application DAA to generate its configuration file in YAML format.

By using the Application Configurator, you can configure service bindings, resource templates, substitution variables, and so on. You can download the generated YAML file, DAA, Log4j configuration, and JVM configuration files as a `.zip` file.

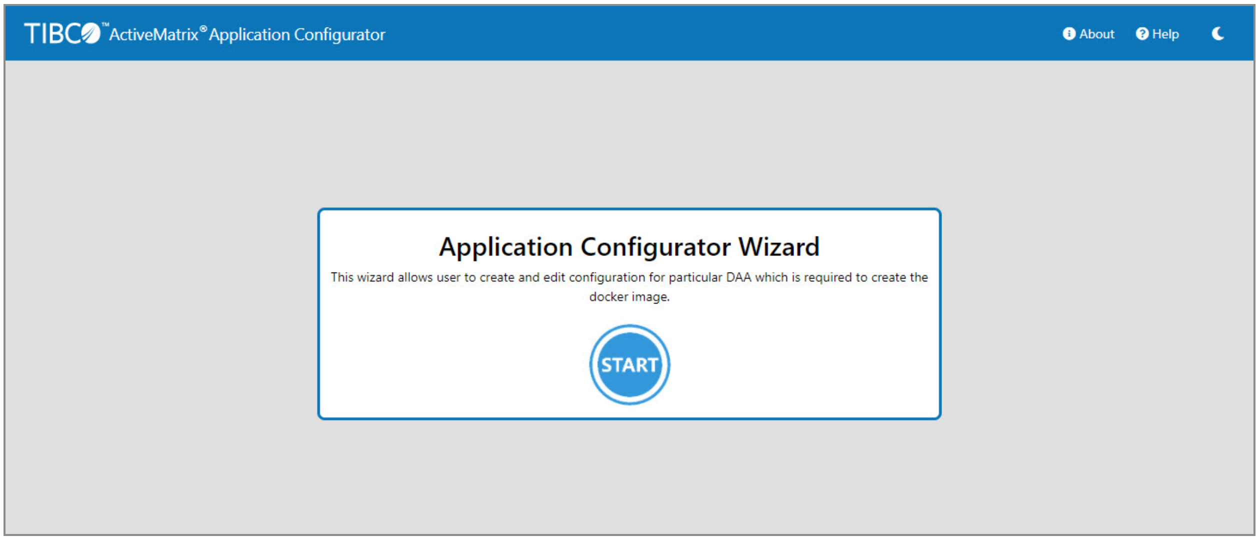
Accessing Application Configurator

The Application Configurator provides access to all application configuration functions.

1. Ensure that the Application Configurator container is running. For more information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*.
2. To open the Application Configurator, access the following URL: `http://`*`hostname`*`:`*`port`*`/appconfig`.
    - Use the same host port you specified when running the Application Configurator container.
    - **Default URL:** `http://<hostname>:8087/appconfig` or `http://<`*`IP Address`*`>:8087/appconfig`.
    - To run the Application Configurator on Kubernetes cluster setup, access the following URL: `http://`*`hostname`*`:31087/appconfig/`. Here, 31087 is the nodePort configured in the sample Kubernetes deployment file, which you can

change.

- Login credentials are not required because the Application Configurator does not store any confidential information.



Application Configurator Wizard

This wizard allows user to create and edit configuration for particular DAA which is required to create the docker image.

To get more information about Application Configurator like version, Java version, Operating System version, Custom Features, click **About** in the upper-right corner.

Miscellaneous Details

Guided Tour

To get started, you can take a guided tour of the Application Configurator UI. To start the guided tour, click **Help > Start Tour** in the upper-right corner.

To get more information about any of the features, click **Help > Documentation** in the upper-right corner. You can also view information about the fields on the UI from the right sidebar.

Light and Dark Theme

You can switch between light and dark theme by clicking the theme icon in the upper-right corner. By default, the light theme ☾ is enabled.

Session Timeout

By default, an Application Configurator session times out after 30 minutes. If the session remains idle for 25 minutes, a warning is displayed to reset the timeout. The configuration is not saved; you must download the configuration before session timeout.

To update the default session timeout value, specify the `sys.sessionTimeout` environment variable when running the Application Configurator container. The `sessionTimeout` takes milliseconds as unit.

Example:

```
docker run --rm -p 8087:8087 -p 9998:9998 -e sys.sessionTimeout=3600000 --name
amxce_ac amxce_ac:1.0
```

In this example, Application Configurator session timeout of 60 minutes is set. To check the sessionTimeout is enabled, you can view the container logs by checking the following logging lines:

```
09 Dec 2020 20:29:32,718 a9d4ffaba4b0 [ComponentFrameworkDelegate] [INFO ]
com.tibco.amx.admin.api.application - Set session timeout to 3600000 using
environment variable.
```

Generating Application Configuration

In TIBCO ActiveMatrix Service Grid, applications are configured from the Administrator UI and all configurations are stored in the Administrator database. ActiveMatrix Service Grid - Container Edition does not have the Administrator database. All configurations are stored in a YAML file.

YAML file is required to build the application Docker image. The YAML file is generated from the Application Configurator UI or the DAA2Config command-line tool.
When migrating your application running in TIBCO ActiveMatrix Service Grid 3.x to ActiveMatrix Service Grid - Container Edition, you can use the Application Extractor to generate the YAML file with all runtime configuration. See the following topics for more information:

- Generating Application Configuration YAML File from Application Configurator
- Generating Application Configuration YAML File from DAA2Config Command-Line Tool
- To generate the YAML file for applications that you want to migrate from TIBCO ActiveMatrix Service Grid 3.x to ActiveMatrix Service Grid - Container Edition, see the "Migrating Applications from ActiveMatrix Service Grid 3.x to ActiveMatrix Service Grid - Container Edition" topic in the *TIBCO ActiveMatrix® Service Grid - Container Edition Quick Start*.

Generating Application Configuration YAML File from Application Configurator

In the Application Configurator, you can configure entities for an application such as properties, substitution variables, resource templates, services, and reference bindings. When the configuration is complete, you can download the configuration in a `.zip` format.

Before you begin

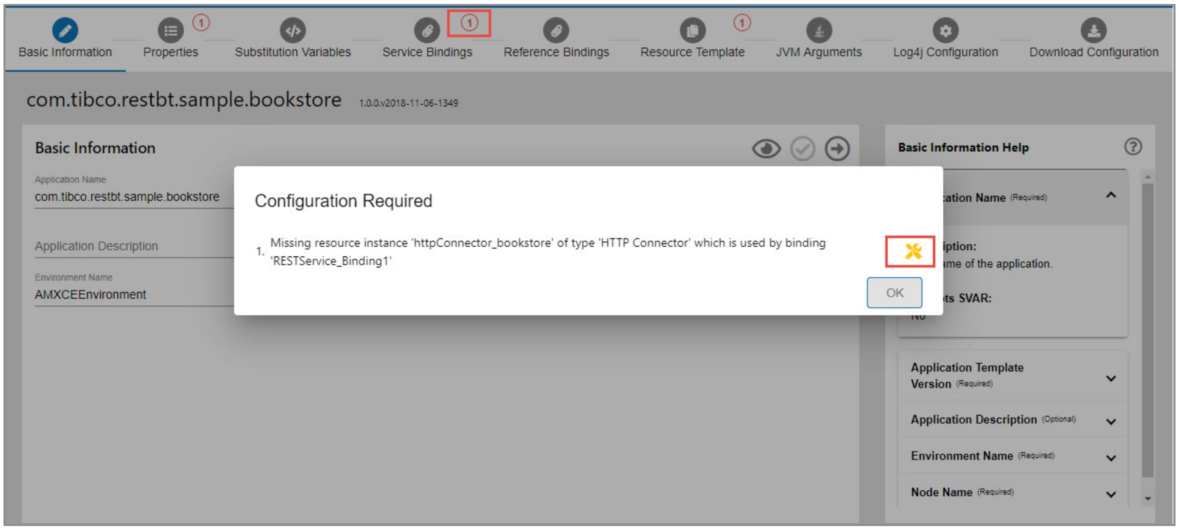
The application DAA, a Log4j configuration file (optional), and an input YAML file are required to start with the application configuration.

Procedure

1. Upload the application DAA, and optionally, the Log4j configuration file. If you have previously downloaded the YAML file, you can upload it as an input file. See Uploading Application DAA.

   A red indicator icon ① with the count is displayed on the tabs and entities which have incomplete configuration.

2. You can navigate to the tabs or entities which have incomplete configuration and complete the configuration. Alternatively, click the red indicator icon ①. The list of required configuration is displayed. Click **Fix** ✕ to complete the required configuration.



com.tibco.restbt.sample.bookstore 1.0.0.v2018-11-06-1349

Basic Information

Application Name
com.tibco.restbt.sample.bookstore

Application Description

Environment Name
AMXCEEnvironment

Configuration Required

1. Missing resource instance 'httpConnector_bookstore' of type 'HTTP Connector' which is used by binding 'RESTService_Binding1'

3. If your application contains configured properties, specify values for them. See Properties.

4. If your application contains substitution variables, assign values to them. See Substitution Variables.

5. Configure the service bindings. See Managing Service and Reference Bindings.

   - For the bindings that have been configured at design time:

Task	Steps
To edit a service binding	i. In the list of

Task	Steps
	bindings, click **Edit** ✎ next to the binding. ii. In the **Edit Binding** dialog box, edit the configuration information and click **Save** ✓.
To delete a service binding	Click **Delete** 🗑 next to the binding.

- For the bindings that have not been configured at design time:

Task	Steps
To add a service binding	i. Select a binding type from the drop-down list and click the **Add Service Binding** button. ii. In the **Add Binding** dialog box, edit the configuration information and click **Save** ✓.

6. Configure the reference bindings, if any. See Managing Service and Reference Bindings.

   - For the reference bindings that have been configured at design time:

Task	Steps
To edit a reference binding	i. In the list of bindings, click **Edit**

Task	Steps
	✎ next to the binding. ii. In the **Edit Binding** dialog box, edit the configuration information and click **Save** ✓.
To delete a reference binding	Click **Delete** 🗑 next to the binding.

- For the reference bindings that have not been configured at design time:

Task	Steps
To add a reference binding	i. Select a binding type from the list and click the **Add Reference Binding** button. ii. In the **Add Binding** dialog box, edit the configuration information and click **Save** ✓.

7. If your application refers to resource templates, configure the resource template. See Managing Resource Templates.
8. You can change the default JVM parameters on the JVM Arguments tab.
9. You can change the default Log4j configuration on the Logging Configurations (Log4j) tab.
10. When the configuration is complete, you can download the configuration as a `.zip` file by navigating to the **Download Configuration** tab. The `.zip` file contains the YAML file, DAA, Log4j configuration file, and JVM parameters file. See Downloading Application Configuration.

What to do next

You can use the downloaded application configuration to containerize an application. For more information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*.

Generating Application Configuration YAML File from DAA2Config Command-Line Tool

The Command Line tool DAA2Config is a standalone tool to generate a YAML file with default values, Log4j configuration file (`node-log4j.xml`), and node jvm arguments (`node_jvm_parameters.config.yaml`) with the default configuration. You need to pass application DAA as an input.

If the DAA does not have all required resource templates, DAA2Config tool creates those resource templates with the default values. You can update the configuration by editing the file in a text editor or by uploading to the Application Configurator.

Before you begin

- Ensure that you have Java 11 installed on your machine and you have set `JAVA_HOME` as environment variable.
- You must have write permission on the folder that contains the DAA and the output location folder; otherwise, a permission error is displayed.

Procedure

1. Navigate to the `amsgce-runtime-<version>\daautil\bin` folder.
2. Run the following command to generate the YAML configuration file from DAA:

   For Linux Platform:

   ```
   ./DAA2Config.sh -daaFile <daa file location> -outputLocation  <output
   location>
   ```

   For Windows Platform:

   ```
   DAA2Config.bat -daaFile <daa file location> -outputLocation  <output
   location>
   ```

   Example (For Linux Platform):

   ```
   ./DAA2Config.sh -daaFile /home/amsgce-runtime-
   <version>/samples/bookstore/com.tibco.restbt.sample.bookstore/Deploy
   mentArtifacts/com.tibco.restbt.sample.bookstore.daa
   ```

   Example (For Windows Platform):

```
DAA2Config.bat -daaFile C:\amsgce-runtime-
1.0.0\samples\bookstore\com.tibco.restbt.sample.bookstore\Deployment
Artifacts\com.tibco.restbt.sample.bookstore.daa
```

For more information about the arguments that you can specify when running the command, see DAA2Config Command Reference. Output location is an optional argument. If it is not provided, the DAA2Config tool generates all artifacts in the same folder where the DAA file is.

Sample CLI Output:

```
HP-Z230-SFF-Workstation:~/amxce/amsgce-runtime-1.0.0/daautil/bin$
./DAA2Config.sh -daaFile /home/amxce/amsgce-runtime-
1.0.0/samples/amxce/bookstore/daas/com.tibco.restbt.sample.bookstore
.daa
----------------------------------------------------------------------
------------------------
Checking java executable
/usr/bin/java
found java executable in PATH
----------------------------------------------------------------------
------------------------

15 Jul 2020 15:21:17,863 [main] [INFO ] com.tibco.amxce.rdacompiler -
RDACompiler 5.0.0
Invoking createConfigFile
 -daaFile /home/amxce/amsgce-runtime-
1.0.0/samples/amxce/bookstore/daas/com.tibco.restbt.sample.bookstore
.daa

15 Jul 2020 15:21:20,253 [main] [INFO ] com.tibco.amxce.rdacompiler -
Successfully created application config file for DAA
'com.tibco.restbt.sample.bookstore.daa' at '/home/amxce/amsgce-
runtime-
1.0.0/samples/amxce/bookstore/daas/com.tibco.restbt.sample.bookstore
.config.yaml'
```

Result

The application configuration YAML file, node JVM parameters configuration file, and Log4j configuration file with default values are generated in the output location provided. If no output location is provided, all files are generated in the same folder where the DAA file is located. For more information about the YAML file, see Understanding Application Configuration YAML File.

Sample `com.tibco.restbt.sample.bookstore.config.yaml`

```
configVersion: 1.0.0
name: com.tibco.restbt.sample.bookstore
daaName: com.tibco.restbt.sample.bookstore.daa
appTemplate: com.tibco.restbt.sample.bookstore
appTemplateVersion: 1.0.0.v2018-11-06-1349
description: Add description here
environment: AMXCEEnvironment
node: AMXCENode
properties:
- name: '[Service]BookStoreResource/RESTService_
Binding1/httpConnectorName'
  type: HTTPConnector
  value: httpConnector_bookstore
services:
- name: BookStoreResource
  restBindings:
  - name: RESTService_Binding1
    applicationPath: /bookstore
    interfaceName: '
{http://ns.tibco.com/BookStoreResource/}BookStoreResource'
    restServiceOperationConfiguration:
    - httpMethod: GET
      mediaType: STANDARD_JSON
      operationName: getBookList
      path: /books
    - httpMethod: GET
      mediaType: STANDARD_JSON
      operationName: getBookByTitle
      path: /book/{title}
    - httpMethod: POST
      mediaType: STANDARD_JSON
      operationName: addBook
      path: /addbook
    - httpMethod: GET
      mediaType: STANDARD_JSON
      operationName: getBookByTitleCategory
      path: /book/{title}/{category}
    restTransportConfigDesc:
      httpConnectorJNDIName: httpConnector_bookstore
    skipJsonRespNamespace: false
resources:
  httpConnectors:
  - name: httpConnector_bookstore
    operation: ADD
    host: 0.0.0.0
    port: '7777'
```

What to do next

You can use the generated configuration to containerize an application. For more information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*. To update the YAML file, you can use a text editor or you can upload the configuration file in the Application Configurator.

DAA2Config Command Reference

The following table lists the options you can specify when running the DAA2Config command line tool. To get the list of options, run the following command:

```
./DAA2Config.sh help
```

Option	Required?	Description
`-daaFile <DAA File Location>`	Y	Location of the DAA file from which to generate the YAML file. Only one DAA file must be specified as an argument.
`-outputLocation <Output Location>`	N	Location of the folder where you want to generate the output of DAA2Config command line utility. If unspecified, output files are generated in the same folder where the DAA file is.
`-envName <Environment Name>`	N	Name of the ActiveMatrix Service Grid - Container Edition environment. **Default:** AMXCEEnvironment
`-nodeName <Node name>`	N	Name of the ActiveMatrix Service Grid - Container Edition node. **Default:** AMXCENode
`-format`	N	Format of the generated YAML file can be `full` or `brief`.

Option	Required?	Description
		• In the `full` format, all available fields of each entity are provided in the configuration file, even when the field value is null. • In the `brief` format, only the required fields and the fields which are different from the default values are provided in the configuration file. For more information, see Understanding Application Configuration YAML File. **Default:** `full`

Application Configuration Workflow

You can upload an application DAA in the Application Configurator and configure different entities. This section describes how to configure your applications and generate the application configuration YAML file required to containerize application.

See the following topics for more information:

- Uploading Application DAA
- Viewing and Updating Application Configuration in Application Configurator
- Downloading Application Configuration
- Viewing and Updating Application Configuration in a Text Editor
- Validating Application Configuration

Uploading Application DAA

To generate the application configuration YAML file by using Application Configurator, you must upload the application DAA in the Application Configurator.

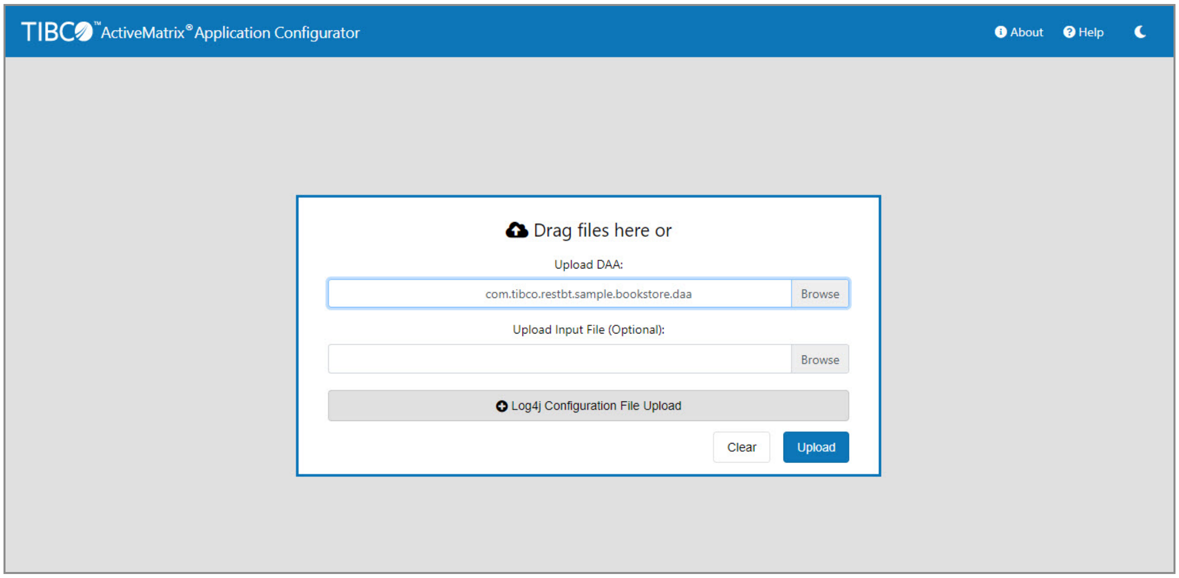
Procedure

1. In a browser, open the following URL: `http://`*`hostname`*`:`*`port`*`/appconfig`.

   Use the same host port that you specify when you run the Application Configurator container. The default container port for Application Configurator is 8087. For example, `http://<IP Address>:8087/appconfig`.

2. Click **Start**.

3. Upload the application DAA file. You can drag application DAA, configuration YAML file, and Log4j configuration file at a time.


TIBCO ActiveMatrix® Application Configurator
About
Help
Drag files here or
Upload DAA:
com.tibco.restbt.sample.bookstore.daa
Browse
Upload Input File (Optional):
Browse
Log4j Configuration File Upload
Clear
Upload

4. (Optional) If you already have a YAML file and want to configure in the same file, you can upload the YAML file by using the **Upload Input File** option. In this case, the application configuration in the YAML file is used instead of application DAA.

   Note: Application configuration file with the `.yml` extension is not supported in Application Configurator and when creating an application Docker image. You must use application configuration file with the `.yaml` extension.

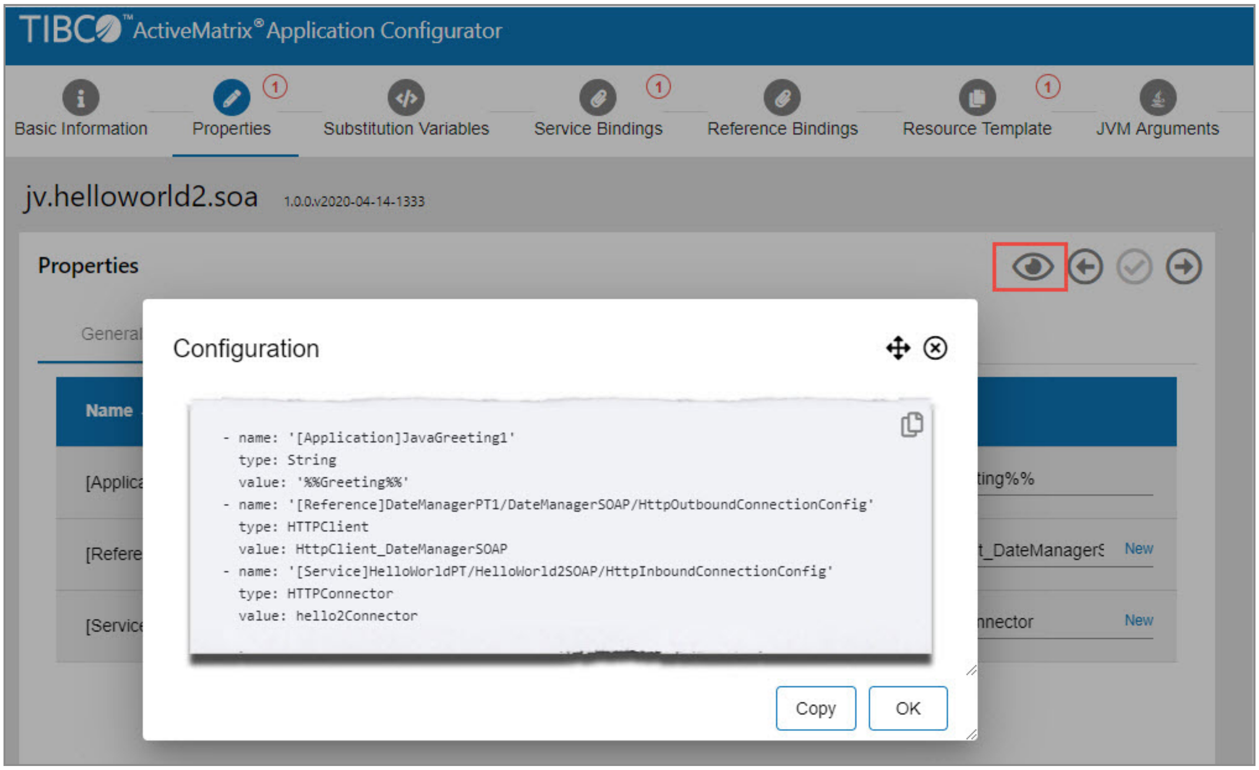
5. (Optional) Upload the Log4j configuration file for an application. If no Log4j configuration file is uploaded, the default Log4j configuration is used.
6. Click **Upload**.

You are redirected to the Application Configuration Basic Information page where you can start configuring an application. To configure other DAA, click the **Start Over** button in the upper-right corner and navigate to the Upload DAA page.

Viewing and Updating Application Configuration in Application Configurator

You can view and update the application configuration in the Application Configurator.

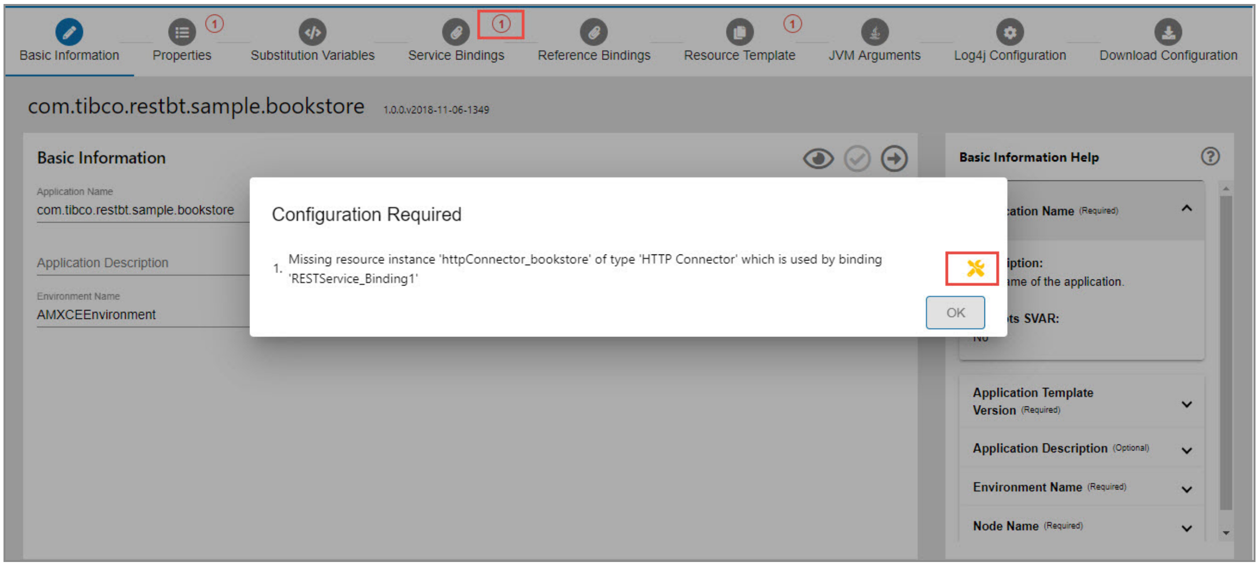
To view the configuration in YAML format, click **Show YAML** 👁. To copy the configuration in the YAML format, click the **Copy** button. If you are editing any of the fields, **Show YAML** 👁 is disabled until you save the changes.



Configuration

```
- name: '[Application]JavaGreeting1'
  type: String
  value: '%%Greeting%%'
- name: '[Reference]DateManagerPT1/DateManagerSOAP/HttpOutboundConnectionConfig'
  type: HTTPClient
  value: HttpClient_DateManagerSOAP
- name: '[Service]HelloWorldPT/HelloWorld2SOAP/HttpInboundConnectionConfig'
  type: HTTPConnector
  value: hello2Connector
```

A red indicator icon ① with the count is displayed on the tabs and entities which have incomplete configuration. You can navigate to those tabs or entities and complete the configuration. Alternatively, click the red indicator icon ①. The list of required configuration is displayed. Click **Fix** to complete the required configuration.



Configuration Required

1. Missing resource instance 'httpConnector_bookstore' of type 'HTTP Connector' which is used by binding 'RESTService_Binding1'

Downloading Application Configuration

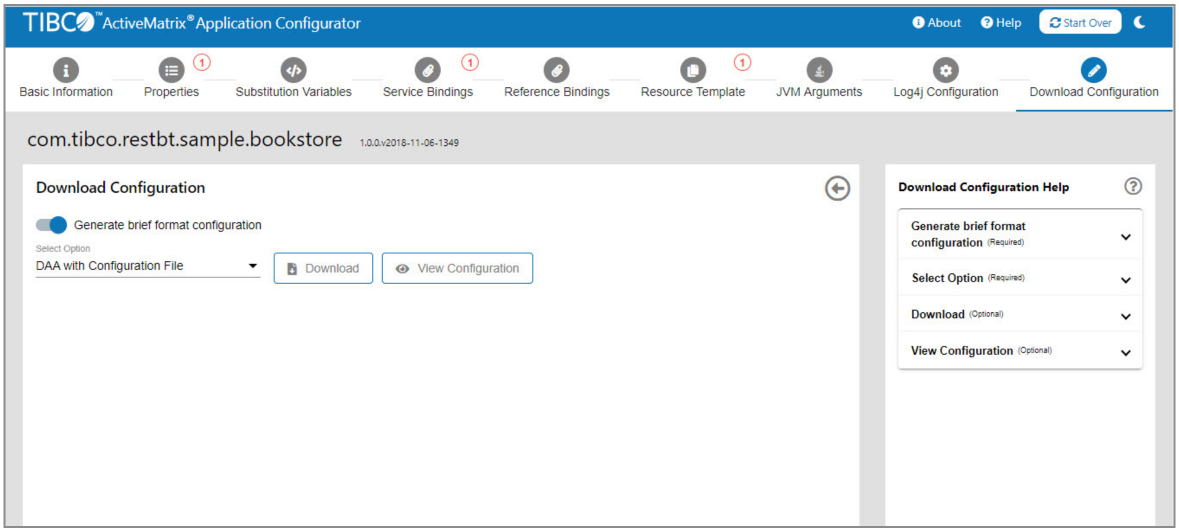
After the application configuration is completed, you can download the application configuration as a `.zip` file from the **Download Configuration** tab.

Before you begin

Before downloading the configuration, you must complete the required configuration with all dependencies resolved. The download configuration option is enabled only if all the required configuration is completed. You can view the configuration, even when it is in progress.

Procedure

1. Click the **Download Configuration** tab.



2. Use the toggle to generate configuration in `full` or `brief` format.

   - In the `full` format, all available fields of each entity are provided in the configuration file, though some of the fields may have null value.

   - In the `brief` format, only the required fields and the fields which are different from the default values are provided in the configuration file.

3. (Optional) Verify the configuration before downloading it by clicking **View Configuration**. The format of the generated configuration is `full` or `brief` depending on the option you have selected in the previous step.

> **Note:** If you click **View Configuration** and save the configuration file from the browser, the file is saved with the underscore (_) instead of dot (.) in the application name. For example, if application name is `com.tibco.sample.bookstore`, then the file is saved with the name `com_tibco_sample_bookstore`.

4. Select one of the following options from the drop-down list:
   - **DAA with Configuration File**: The downloaded `.zip` file contains application DAA, YAML file, JVM configuration (YAML), and Log4j configuration file.
   - **Only Configuration File:** The downloaded `.zip` file contains only YAML file, JVM configuration (YAML), and Log4j configuration file.
   - **Only DAA:** The downloaded `.zip` file contains only application DAA.
5. Click **Download**.

Result

The `.zip` file containing configuration files are downloaded to your machine.

- The name of the `.zip` file is `[Application name]_[Timestamp].zip`.
- The name of the application configuration YAML file that is downloaded is `<Application Name>.config.yaml`.
- The name of the Log4j configuration file that is downloaded is `node-log4j.xml`.
- The name of the node parameters file is `node_jvm_parameters.config.yaml`.If you have uploaded the Log4j configuration file with any custom name, the downloaded Log4j file will be `node-log4j.xml`.

If you have set the default Log4j configuration and node JVM parameters for Application Configurator and do not modify them in the Application Configurator, then the default Log4j configuration file and node JVM parameters file are downloaded as part of the .zip file.

If you have uploaded the keystore files to Application Configurator, the .zip file create "certs" folder with used keystore files.

> **Note:** If the uploaded application DAA contains already configured Keystore resource template, the keystore file is not downloaded in the `.zip` file.

Viewing and Updating Application Configuration in a Text Editor

You can view and update Application configuration YAML file in any text editor. To understand the structure of application configuration YAML file, see Understanding Application Configuration YAML File.

You can also use JSON schema provided in the `amsgce-runtime<version>.zip` file to use auto-completion feature when updating application configuration YAML file in the visual studio code editor. For more information, see Auto-Completion Feature for Application Configuration YAML File.

Understanding Application Configuration YAML File

In TIBCO ActiveMatrix Service Grid, applications are configured from the Administrator UI and all configurations are stored in the Administrator database. ActiveMatrix Service Grid - Container Edition does not have the Administrator database. All configurations are stored in the configuration file.

To create the Docker image of an application, this configuration file is required. Configuration file has a higher priority compared to DAA when creating the Docker image of an application, which means the configuration file overrides the configuration in DAA.

The application configuration file is a text file in YAML format and has the following sections:

Application Basic Information

```
configVersion: 1.0.0
name: com.tibco.restbt.sample.bookstore
daaName: com.tibco.restbt.sample.bookstore.daa
appTemplate: com.tibco.restbt.sample.bookstore
appTemplateVersion: 1.0.0.v2018-11-06-1349
description: Add description here
environment: AMXCEEnvironment
node: AMXCENode
```

Here:

- configVersion (optional): Configuration file version. It is 1.0.0 by default.

- name (required): Application name
- daaName (optional): DAA name which is used to create the YAML configuration and Docker image.
- appTemplate (required): Application template name
- appTemplateVersion (optional): Application template version
- description (optional): Description of an application
- environment (optional): Environment name. Default value is AMXCEEnvironment.
- node (optional): Node name. Default value is AMXCENode.

Properties

All properties listed are editable. The properties include application level properties and resources which are required by binding, and resources which are required by policy. Property can only be updated and cannot be added or removed.

```
properties:
- name: '[Service]BookStoreResource/RESTService_
Binding1/httpConnectorName'
  type: String
  value: httpConnector_bookstore
```

Properties display a prefix indicating the context as follows:

- Application level properties are displayed with the prefix `[Application]`.
- Binding level properties are displayed with the prefix `[Service]` or `[Reference]`.
    - The format of the reference binding property name is `[Reference]REFERENCE_NAME/REFERENCE_BINDING_NAME/PROPERTY_NAME`.
    - The format of service binding property is `[Service]SERVICE_NAME/SERVICE_BINDING_NAME/PROPERTY_NAME`.
- Properties for certain policy sets such as threading policy display with the preceding prefix `[Policy]`.

  Example: `[Policy]POLICY_NAME/PROPERTY_NAME`.

Substitution Variables

All substitution variables function at application level. They are used to resolve substitution variables that are used in the application. Substitution variables can be added, updated, or removed.

Substitution variables types are: String, Integer, Boolean, and Password.

For example, the following substitution variable `clientport` is of type `String` with a value of `9000`. The operation `ADD` means this substitution variable is not in DAA and it is added to the application from the YAML file.

```
- name: clientport
  operation: ADD
  type: String
  value: ‘9000’
```

The `httpClientPort` substitution variable has only the `operation` attribute with value `REMOVE`, which means this substitution variable has been defined in DAA, but it is not needed during application deployment and is removed by using the configuration file.

```
-name: httpClientPort
operation: REMOVE
```

Services

Service name and the Service interface (contract) cannot be updated. Bindings are editable. Service describes a "contract" between providers and consumers, therefore it is not editable at deployment time and bindings describe how the communication happens, therefore Service bindings can be updated, added, or removed. Service has three types of bindings: JMS binding, SOAP binding, and REST binding. There are three sections under services: jmsBindings, soapBindings, and restBindings. Each binding is listed in respective section based on its type.

```
services:
- name: BookStoreResource
  restBindings:
  - name: RESTService_Binding1
    applicationPath: /bookstore
    interfaceName: '
{http://ns.tibco.com/BookStoreResource/}BookStoreResource'
    restServiceOperationConfiguration:
    - httpMethod: GET
      mediaType: STANDARD_JSON
      operationName: getBookList
      path: /books
    - httpMethod: GET
      mediaType: STANDARD_JSON
      operationName: getBookByTitle
      path: /book/{title}
    - httpMethod: POST
      mediaType: STANDARD_JSON
      operationName: addBook
      path: /addbook
    - httpMethod: GET
      mediaType: STANDARD_JSON
      operationName: getBookByTitleCategory
      path: /book/{title}/{category}
    restTransportConfigDesc:
      httpConnectorJNDIName: httpConnector_bookstore
    skipJsonRespNamespace: false
```

References

Reference name cannot be updated. Reference bindings can be updated, added, or removed. Reference describes a "contract" between providers and consumers, therefore it is not editable at deployment time and bindings describe how the communication happens, therefore bindings can be updated, added, or removed. Only SOAP reference binding is supported for editing in ActiveMatrix Service Grid - Container Edition. Because reference has only one binding, there is only one section under each reference based on the binding type.

```
references:
- name:NewReference
  soapBindings:
  - name:SOAPReference_Binding1
    operation: REMOVE
  - name:SoapBinding
    operation: ADD
    encoding : LITERAL
    httpOutboundConnectionJNDIName: httpClient
    httpTransportConfig:
      endpointURI : /test/
      httpOutboundConnectionJNDIName: httpClient
      overrideHttpClientTimout: false
    soapVersion : '1.1'
    style: DOCUMENT
    transportBindingType: HTTP
    validateRequest: 'false'
    validateResponse: 'false'
```

Resources

In the resources section, resource templates are listed by type. Each type can have multiple resources with the same type. Resource can be updated, added, or removed.

```
resources:
  httpConnectors:
  - name: httpConnector_bookstore
    operation: ADD
    description: This is created using Application Configurator
(1.0.0.000)
    acceptQueueSize: '50'
    acceptors: '20'
    host: 0.0.0.0
    idleTimeout: '200000'
    lingerTime: '-1'
    lowResourceMaxIdleTime: '-1'
    outputBufferSize: '24576'
    port: '7777'
    requestHeaderSize: '4096'
    responseHeaderSize: '4096'
    sslEnabled: false
    useDirectBuffers: 'false'
    useNonBlockingIOSockets: 'false'
```

Operation Property

In the YAML file, you can see the operation property for some entities which specifies actions to be performed on the DAA for the entity. Operations can be ADD, UPDATE, and REMOVE. When unspecified, the default operation is UPDATE.

- ADD: For the ADD operation, the entity is added to the application during deployment. If the entity already exists in DAA, then this entity in the configuration file is ignored.
- UPDATE: For the UPDATE operation, the entity in the configuration file overwrites the entity in DAA. If the entity does not exist in DAA, then this entity in configuration file is ignored during deployment.
- REMOVE: For the REMOVE operation, the entity with the same type and name in DAA is removed if it exists. If the entity doesn't exist in DAA, then this entity in configuration file is ignored during deployment.

In the following example, the svar clientport is of type String with a value of 9000. The operation ADD means this svar does not exist in the DAA and it is added to application from the YAML file.

```
- name: clientport
  operation: ADD
  type: String
  value: ‘9000’
```

In the following example, the first reference binding `SOAPReference_Binding1` has `REMOVE` operation. This binding is removed from the application during deployment, although it is specified in the DAA. The second reference binding `SoapBinding` has `ADD` operation; it is added to the application during deployment.

```
references:
- name:NewReference
  soapBindings:
  - name:SOAPReference_Binding1
    operation: REMOVE
  - name:SoapBinding
    operation: ADD
    encoding : LITERAL
    httpOutboundConnectionJNDIName: httpClient
    httpTransportConfig:
      endpointURI : /test/
      httpOutboundConnectionJNDIName: httpClient
      overrideHttpClientTimout: false
    soapVersion : '1.1'
    style: DOCUMENT
    transportBindingType: HTTP
    validateRequest: 'false'
    validateResponse: 'false'
```

Brief or Full Format

Format of the generated YAML file can be `full` or `brief`. You can specify the format by using -format parameter when running `DAA2Config` command. You can also specify the format when downloading configuration from Application Configurator. When unspecified, the default value is `full`.

- In the `full` format, all available fields of each entity are generated in the configuration file, though the values of some of the fields are null.
- In the `brief` format, only the required fields and the fields which are updated are generated in the configuration file.

Following is the resources section of a sample YAML file with the `full` format.

When generating the application configuration, if you specify the format as `full`, fields with null values are also generated in the YAML file as shown in the following example:

```
resources:
  httpClients:
  - name: HttpClient_SampleSOAP
    operation: null
    description: null
    acceptRedirect: 'false'
    basicAuthConfigured: false
    basicAuthPassword: null
    basicAuthUsername: null
    connectionRetrivalTimeout: '0'
    connectionTimeout: '0'
    emptyPwd: false
    enableConnectionPooling: 'true'
    host: www.example.org
    localSocketAddress: null
    maxPoolSize: '20'
    maximumTotalConnections: '20'
    maximumTotalConnectionsPerHost: '2'
    port: '%%httpClientPort%%'
    proxyConfigured: false
    proxyHost: null
    proxyPort: null
    proxyType: null
    soBufferSize: '-1'
    soLinger: '0'
    soReceiveBufferSize: '-1'
    soReuseAddress: 'false'
    soSendBufferSize: '-1'
    soTimeout: '0'
    sslEnabled: false
    sslJndiName: null
    staleCheck: 'false'
    tcpNoDelay: 'true'
  - name: RESTReference_Binding1_HTTPClient
    operation: null
    description: null
    acceptRedirect: 'false'
    basicAuthConfigured: false
    basicAuthPassword: null
    basicAuthUsername: null
    connectionRetrivalTimeout: '0'
    connectionTimeout: '0'
    emptyPwd: false
    enableConnectionPooling: 'true'
    host: www.example.org
    localSocketAddress: null
    maxPoolSize: '20'
    maximumTotalConnections: '20'
    maximumTotalConnectionsPerHost: '2'
    port: '80'
    proxyConfigured: false
    proxyHost: null
    proxyPort: null
    proxyType: null
    soBufferSize: '-1'
    soLinger: '0'
    soReceiveBufferSize: '-1'
    soReuseAddress: 'false'
    soSendBufferSize: '-1'
    soTimeout: '0'
    sslEnabled: false
    sslJndiName: null
    staleCheck: 'false'
    tcpNoDelay: 'true'
  httpConnectors:
  - name: '[httpConnector]'
    operation: ADD
    description: null
    acceptQueueSize: '50'
    acceptors: '20'
    host: 0.0.0.0
    idleTimeout: '200000'
    lingerTime: '-1'
    lowResourceMaxIdleTime: '-1'
    outputBufferSize: '24576'
    port: '7778'
    requestHeaderSize: '4096'
    responseHeaderSize: '4096'
    sslEnabled: false
    sslJNDIName: null
    threadPoolJndiName: null
    useDirectBuffers: 'false'
    useNonBlockingIOSockets: 'false'
  - name: httpConnector
    operation: ADD
    description: null
    acceptQueueSize: '50'
    acceptors: '20'
    host: 0.0.0.0
    idleTimeout: '200000'
    lingerTime: '-1'
    lowResourceMaxIdleTime: '-1'
    outputBufferSize: '24576'
    port: '7777'
    requestHeaderSize: '4096'
    responseHeaderSize: '4096'
    sslEnabled: false
    sslJNDIName: null
    threadPoolJndiName: null
    useDirectBuffers: 'false'
    useNonBlockingIOSockets: 'false'
  jmsConnectionFactories:
  - name: JMSReference_Binding1_ConnectionFactory
    operation: null
    description: null
    enableAuthentication: false
    identityJNDIName: null
    jndiConnectionConfigurationName: JMSReference_Binding1_
JNDIConnectionConfiguration
    jndiName: connfactory2
    loginCredUserPwd: false
    maxPoolSize: '20'
    password: null
    sslEnabled: false
    sslJNDIName: null
    username: null
  - name: JMSService_Binding1_ConnectionFactory
    operation: null
    description: null
    enableAuthentication: false
    identityJNDIName: null
    jndiConnectionConfigurationName: JMSService_Binding1_
JNDIConnectionConfiguration
    jndiName: connectionFactory
    loginCredUserPwd: false
    maxPoolSize: '20'
    password: null
    sslEnabled: false
    sslJNDIName: null
    username: null
  jndiConnections:
  - name: JMSReference_Binding1_JNDIConnectionConfiguration
    operation: null
    description: null
    emptyPwd: false
    enableAuthentication: false
    identityJNDIName: null
    initialContextFactory:
com.tibco.tibjms.naming.TibjmsInitialContextFactory
```

```
    loginCredUserPwd: false
    maxPoolSize: '20'
    password: null
    providerUrl: tibjmsnaming://locahost:7222
    sslEnabled: false
    sslJNDIName: null
    username: null
  - name: JMSService_Binding1_JNDIConnectionConfiguration
    operation: null
    description: null
    emptyPwd: false
    enableAuthentication: false
    identityJNDIName: null
    initialContextFactory:
com.tibco.tibjms.naming.TibjmsInitialContextFactory
    loginCredUserPwd: false
    maxPoolSize: '20'
    password: null
    providerUrl: tibjmsnaming://locahost:7222
    sslEnabled: false
    sslJNDIName: null
    username: null
```

Following is the resources section of a sample YAML file with the `brief` format.

When generating the application configuration, if you specify the format as `brief`, only the required fields and the fields which are different from the default values are generated in the configuration file.

```
resources:
  httpClients:
  - name: HttpClient_SampleSOAP
    connectionRetrivalTimeout: '0'
    enableConnectionPooling: 'true'
    host: www.example.org
    maximumTotalConnections: '20'
    maximumTotalConnectionsPerHost: '2'
    port: '%%httpClientPort%%'
  - name: RESTReference_Binding1_HTTPClient
    connectionRetrivalTimeout: '0'
    enableConnectionPooling: 'true'
    host: www.example.org
    maximumTotalConnections: '20'
    maximumTotalConnectionsPerHost: '2'
    port: '80'
```

```
 httpConnectors:
 - name: '[httpConnector]'
   operation: ADD
   host: 0.0.0.0
   port: '7778'
 - name: httpConnector
   operation: ADD
   host: 0.0.0.0
   port: '7777'
 jmsConnectionFactories:
 - name: JMSReference_Binding1_ConnectionFactory
   jndiConnectionConfigurationName: JMSReference_Binding1_
JNDIConnectionConfiguration
   jndiName: connfactory2
 - name: JMSService_Binding1_ConnectionFactory
   jndiConnectionConfigurationName: JMSService_Binding1_
JNDIConnectionConfiguration
   jndiName: connectionFactory
 jndiConnections:
 - name: JMSReference_Binding1_JNDIConnectionConfiguration
   initialContextFactory:
com.tibco.tibjms.naming.TibjmsInitialContextFactory
   providerUrl: tibjmsnaming://localhost:7222
 - name: JMSService_Binding1_JNDIConnectionConfiguration
   initialContextFactory:
com.tibco.tibjms.naming.TibjmsInitialContextFactory
   providerUrl: tibjmsnaming://localhost:7222
```

Auto-Completion Feature for Application Configuration YAML File

ActiveMatrix Service Grid - Container Edition ships JSON schema which gives you the auto-completion feature when editing the application configuration YAML file in an editor. This schema is generated according to the YAML structure that is required to create the image. For the current release, this feature does not do schema validation to display errors in YAML file, but for ADD or UPDATE operation it displays required parameter error for an entity.

When editing the application configuration YAML file, you can get suggestions of the remaining parameters of an entity. You can add or update the required entity with auto-completion. For more information about the application configuration YAML file, see Understanding Application Configuration YAML File.

Using the JSON Schema in Visual Studio Code Editor

Currently, JSON schema supports the Visual Studio Code editor only. The JSON Schema `.zip` file is provided in the folder `amsgce-runtime-<version>/jsonschema/JsonSchema.zip`.

Procedure

1. Extract `JsonSchema.zip` to your local machine. Note the absolute path to `amxce_schema.json`.
2. Install the YAML plugin in Visual Studio Code. Restart Visual Studio Code.
3. In Visual Studio Code, navigate to **File > Preferences > Settings**. Search `yaml` in the search bar. Click the **Edit in setting.json** link.
4. Add the following mapping in the `settings.json`. Location must be the absolute path to the schema file on your system. Replace the `<location>` with the path that you noted in Step 2, which is the absolute path to `amxce_schema.json` file.

   ```
   Add below mapping in settings.json

   {
       "yaml.schemas": {
           "<location>/amxce_schema.json": ["/*.config.yaml"],
       }
   }
   ```

   If you are using Microsoft Windows Operating System, add slash'/' before absolute path. For example: `/C:/schema/amxce_schema.json`
5. Save `settings.json` and restart Visual Studio Code.

When you create a new `.config.yaml` file and press **CTRL + Space**, Visual Studio Code displays the suggestions based on the schema `amxce_schema.json` and parameters specific to the application DAA.

Validating Application Configuration

The application configuration artifacts (YAML file, JVM arguments, Log4j configuration) generated directly by Application Configurator, DAA2Config, or Application Extractor are error free. You can use a text editor to further update these artifacts. This manual step may introduce inconsistencies in the configurations. In such a case, you can use checkConfig command to validate the configuration. You must fix errors reported by the checkConfig command, otherwise, the Image building step fails.

The checkConfig command validates the application configuration including the YAML file, JVM arguments file, and Log4j configuration file. You can validate the configuration before containerizing an application. When running the checkConfig command, you must specify the location of the folder that contains the DAA, YAML file, JVM arguments file, and Log4j configuration file.

Before you begin

Ensure that you have Java 11 installed on your machine and you have set <JAVA_HOME> as environment variable.

Procedure

1. Navigate to amsgce-runtime-<version>\daautil\bin folder.
2. Run the following command to validate an application configuration:

   For Linux Platform:

   ```
   ./checkConfig.sh -daaFolder <location of folder that contains DAA and
   configuration files> -debug  <Boolean (true/false)>
   ```

   For Windows Platform:

   ```
   checkConfig.bat -daaFolder <location of folder that contains DAA and
   configuration files> -debug  <Boolean (true/false)>
   ```

   Example:

   ```
   ./checkConfig.sh -daaFolder "/home/apps/helloworld" -debug true
   ```

   For more information about arguments, which you can specify when running the command, see checkConfig Command Reference.

   Sample CLI Output

   In the following sample CLI output, you can see that the validation of application configuration is successful.

   ```
   C:\tibco\amxce\amsgce-runtime-1.0.0\daautil\bin>checkConfig.bat -
   daaFolder C:\tibco\amxce\helloWorld_REST_SOAP_SSL_GlobalTransation
   ```

```
DDA2Config Path : C:\tibco\amxce\amsgce-runtime-1.0.0\daautil\bin\
------------------------------------------------------
Checking Java installation on machine.
------------------------------------------------------
Found java executable in PATH

15 Sep 2020 12:08:03,805 [main] [INFO ] com.tibco.amxce.rdacompiler -
RDACompiler 5.0.0
Invoking validateConfigFile
-daaFolder C:\tibco\amxce\helloWorld_REST_SOAP_SSL_GlobalTransation

15 Sep 2020 12:08:11,394 [main] [WARN ] com.tibco.amxce.rdacompiler -
Property '[Reference]HelloWorldPT/EntryREST/httpConnectorName' has
been skipped since it is not used.
15 Sep 2020 12:08:11,458 [main] [INFO ] com.tibco.amxce.rdacompiler -
--------------------------------------------------
15 Sep 2020 12:08:11,458 [main] [INFO ] com.tibco.amxce.rdacompiler -
1 DAA(s) are found under application folder:
15 Sep 2020 12:08:11,461 [main] [INFO ] com.tibco.amxce.rdacompiler -
AMXCE-SOA_sslClient_sslServer.daa
15 Sep 2020 12:08:11,462 [main] [INFO ] com.tibco.amxce.rdacompiler -
The following DAA will be deployed,
15 Sep 2020 12:08:11,471 [main] [INFO ] com.tibco.amxce.rdacompiler -
DAA: AMXCE-SOA_sslClient_sslServer.daa
15 Sep 2020 12:08:11,478 [main] [INFO ] com.tibco.amxce.rdacompiler -
Config File: AMXCE-SOA_sslClient_sslServer.config.yaml
15 Sep 2020 12:08:11,480 [main] [INFO ] com.tibco.amxce.rdacompiler -
--------------------------------------------------
15 Sep 2020 12:08:11,635 [main] [INFO ] com.tibco.amxce.config.lint -
Config file has been validated successfully.
```

checkConfig Command Reference

The following table lists the options you can specify when running the checkConfig command. To view the list of options, run the following command:

For Linux Platform

```
./checkConfig.sh help
```

For Windows Platform

```
./checkConfig.bat help
```

Option	Required?	Description
-daaFolder <Directory path>	Yes	Location of the folder containing the DAA file, YAML file, Node_JVM arguments file, and log4j configuration file.
-debug <Boolean (true/false)>	No	Prints verbose output when set to true. **Default:** false.

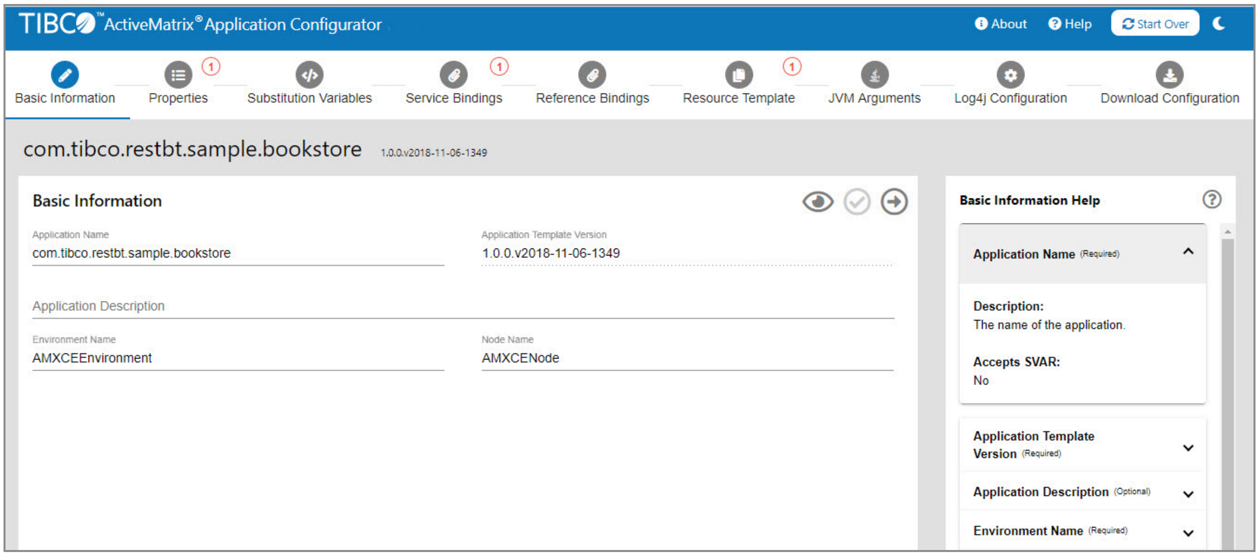
Configuring Application using Application Configurator

This section describes how you can configure different application entities such as Properties, Substitution variables, Resource Templates, Bindings, JVM arguments, and Log4j configuration by using Application Configurator.

Application Basic Information

The **Basic Information** tab displays basic information about application such as application name, application template version, application description, environment name, and node name. All the fields are editable except the application template version.

The application name with the application template version is displayed on each tab.



com.tibco.restbt.sample.bookstore 1.0.0.v2018-11-06-1349

Basic Information

Application Name
com.tibco.restbt.sample.bookstore

Application Template Version
1.0.0.v2018-11-06-1349

Application Description

Environment Name
AMXCEEnvironment

Node Name
AMXCENode

Basic Information Help

Application Name (Required)

Description:
The name of the application.

Accepts SVAR:
No

Application Template Version (Required)

Application Description (Optional)

Environment Name (Required)

To upload the DAA and start configuration again from scratch, click **Start Over** in the upper-right corner. Application Configurator does not store configuration data, clicking the **Start Over** button will remove all your current configuration.

Basic Information Reference

The **Basic Information** tab displays general information about an application.

Field	Required?	Editable?	Description
Application Name	Y	Y	The name of the application.
Application Template Version	Y	N	The version of the application template from which the application was created.
Application Description	N	Y	Description of the application.
Environment Name	Y	Y	Name of the ActiveMatrix Service Grid - Container Edition environment. **Default:** AMXCEEnvironment
Node Name	Y	Y	Name of the ActiveMatrix Service Grid - Container Edition node. **Default:** AMXCENode

Properties

A property is an externally visible data value. Properties enable object behavior to be configured at deployment time. A property has a Name, a Type, and a Value. The type of a property may be either simple or complex. Implementations, components, composites, bindings, and resource templates can have properties. Implementations, components, and composite properties are defined in TIBCO ActiveMatrix Business Studio. Binding and resource template properties are defined either by TIBCO ActiveMatrix Business Studio or by configuration YAML file.

Properties can have explicit values or may be bound to substitution variables, which can be set at application configuration time. Depending on the object possessing the property, the property value can be bound at design time, deployment time, or both:

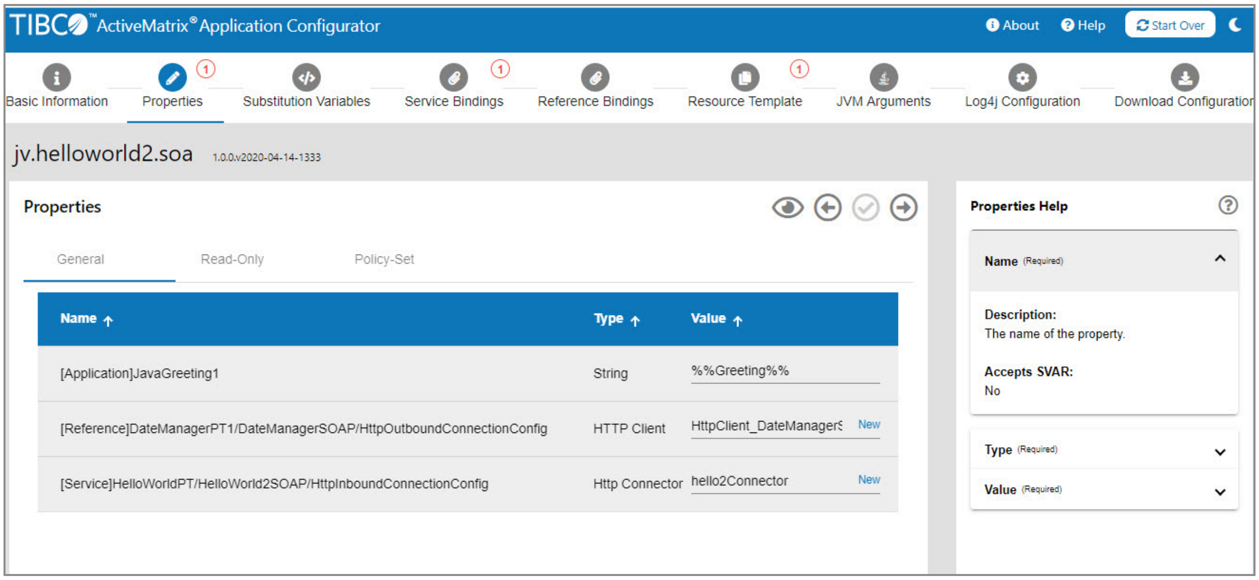
- At design time, you can provide default values and indicate whether a composite or component property value must be set at deployment time.
- Some properties can be bound to substitution variables.

At design time, a composite property value can be set to a constant or bound to a substitution variable.

Either type of binding can be overridden at application configuration time. However, only the properties of the root composite of an application or those on bindings associated with application level services and references can be overridden. Application Configurator cannot change the property values of nested composites (component of type composite).

Properties display a prefix indicating the context as follows:

- Application level properties are displayed with the prefix `[Application]`.
- Binding level properties are displayed with the prefix `[Service]` or `[Reference]`.
- Component level properties are displayed with the prefix `[Component]`.
- Properties at nested composites are displayed with the prefix `[Composite]`.
- Properties for certain policy sets such as threading policy are displayed with the prefix `[Policy]`.



jv.helloworld2.soa 1.0.0.v2020-04-14-1333

Properties

Name ↑	Type ↑	Value ↑
[Application]JavaGreeting1	String	%%Greeting%%
[Reference]DateManagerPT1/DateManagerSOAP/HttpOutboundConnectionConfig	HTTP Client	HttpClient_DateManagerS New
[Service]HelloWorldPT/HelloWorld2SOAP/HttpInboundConnectionConfig	Http Connector	hello2Connector New

Properties Help

Name (Required)

Description:
The name of the property.

Accepts SVAR:
No

Type (Required)

Value (Required)

Setting a Property Value

You can set a property value in the Application Configurator. A property value can be set to a constant, a substitution variable, or the name of a resource template.

To bind a property value to a substitution variable, set the value to `%%variableName%%`, where `variableName` is the name of the substitution variable.

Procedure

1. In the Application Configurator, click the **Properties** tab. For more information about the fields, see Properties Reference.
2. Specify a value according to the property type:
   - Type a value or substitution variable string.
   - If you have already created a resource template, start typing and select from the suggestions.
   - To create a new resource template:
     a. Click the **New** link. The **Add Resource Template** dialog box is displayed.
     b. Complete the dialog box and click **Save** ✓. The property value is filled with the name of the resource template.
3. Click **Save** ✓.

Properties Reference

General Tab

To edit property value for an application, select the **Properties** tab and navigate to the **General** tab.

Field	Read-only?	Description
Property Name	Y	Name of the property.
Property Type	Y	Type of the property: String, boolean, and a resource template type.
Property Value	N	Value of the property.

Read-only and Policy Set Tabs

To access the read-only and policy set properties for an application, on the **Properties** tab, click the **Read-only** or **Policy Set** tab.

Field	Read-only?	Description
Property Name	Y	Name of the property.
Property Type	Y	Type of the property: String, boolean, and a resource template type.
Property Value	Y	Value of the property.

Substitution Variables

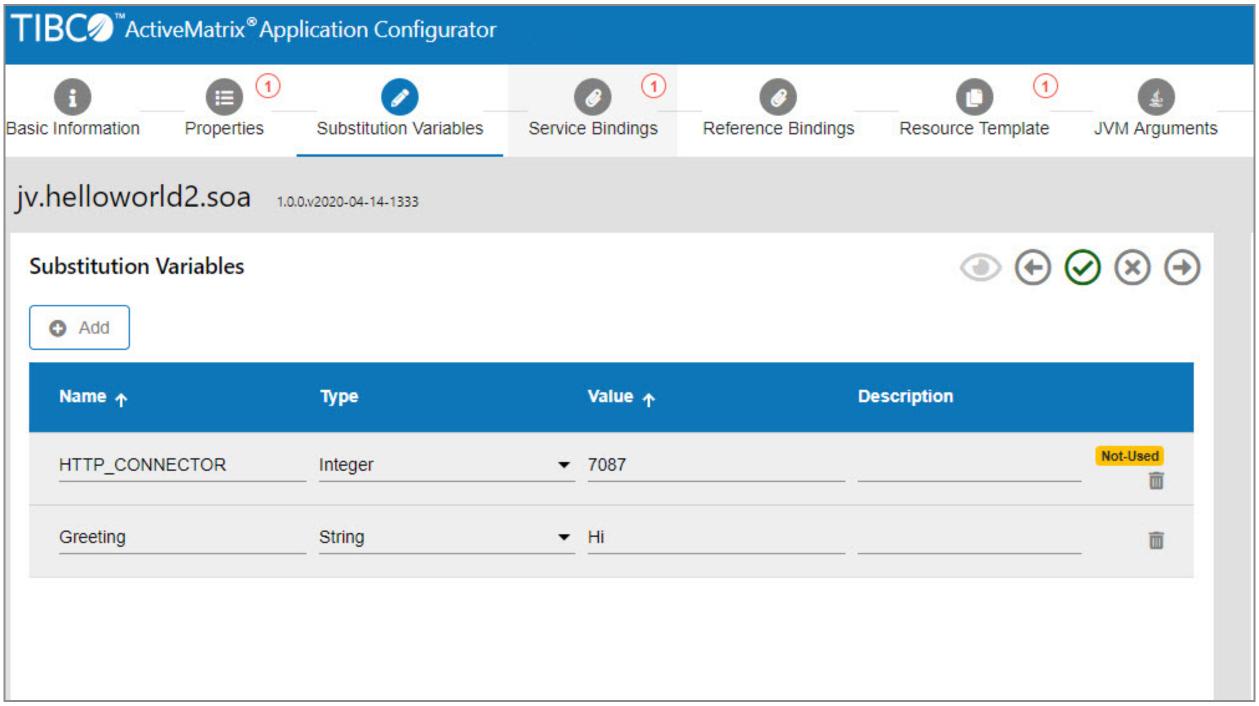
A substitution variable is a variable that you can reuse in the resource and application configurations. Substitution variables enable binding of property values to values set at runtime. For example, you can create an HTTP client resource template and bind its port property to a substitution variable that is set when the template is instantiated. The types of substitution variables are:

- String (default type)
- Integer
- Boolean
- Password

You can create substitution variables at design time or from the Application Configurator. At design time, instead of explicitly setting property values, you can bind them to substitution variables. During application configuration, set the substitution variables values to values supported by the resources available.

A substitution variable is identified by a name. Names must not contain whitespace. When a property value is bound to a substitution variable, the property value is a string containing the substitution variable name surrounded by two pairs of percent signs.

If there is a label **Not-Used** next to the substitution variable row, it indicates that the substitution variable is not used in the current application configuration.



jv.helloworld2.soa 1.0.0.v2020-04-14-1333

Substitution Variables

Name ↑	Type	Value ↑	Description
HTTP_CONNECTOR	Integer	7087	
Greeting	String	Hi	

Creating a Substitution Variable

On the **Substitution Variable** tab in the Application Configurator, you can create a substitution variable.

1. In the Application Configurator UI, click the **Substitution Variable** tab.
2. On the **Substitution Variable** tab, click **Add**. A row is added to the table. For more information about the fields, see Application Substitution Variables Reference.
3. In the Name column, type a name for the variable.
4. In the Type column, select the variable type from the drop-down list.
5. Click the Value column and specify a value.
6. Click **Save** ✓.

Application Substitution Variables Reference

The **Substitution Variables** tab displays the list of the application's substitution variables.

Substitution Variables

Field	Required?	Editable?	Description
Name	Y	Y	Name of the substitution variable.
Type	Y	Y	Type of the substitution variable. • String • Integer • Boolean • Password **Default:** String
Value	Y	Y	Value of the substitution variable.

Managing Service and Reference Bindings

Applications interact via services and references. A *Service* is a set of operations and the messages required by the operations. A *Reference* identifies the service consumed by a component or composite. Applications offer services and invoke references to other services.

An application's services and references are promoted from the services and references of the components it contains.

The component services can be consumed by other components within the composite or promoted as composite services for use by consumers outside the composite. A composite service has an interface and one or more bindings.

The component references consume services provided by other components in the same composite or services provided outside the composite. A composite reference has an interface and one binding.

Viewing and Managing Service Bindings

A binding specifies how communication happens between a reference and a service. A service binding describes the mechanism a client uses to access a service. A reference binding describes the access mechanism a reference uses to invoke a service.

ActiveMatrix Service Grid - Container Edition supports the following Binding Types (BT):

- REST
- SOAP
- JMS

The following bindings are explicitly created by architects and developers only on promoted services and references:

- SOAP
- JMS
- REST

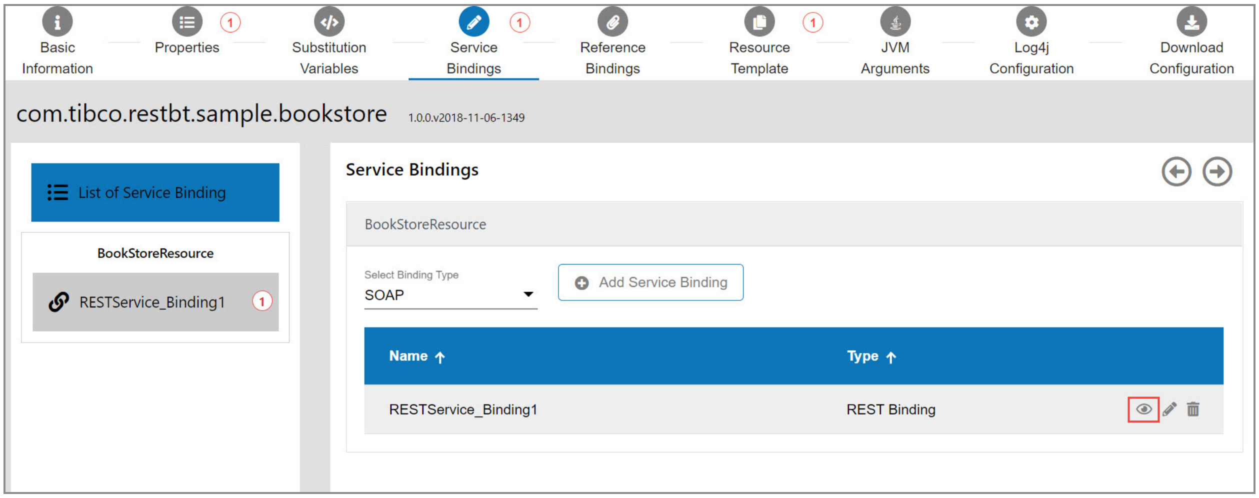
Note: TIBCO Business Studio and the Application Configurator provide the option to choose between TIBCO SOAP/JMS and W3C SOAP/JMS for SOAP binding type while adding a binding to a service.

Note: SOAP bindings support both HTTP and JMS transport types.

Viewing Bindings for a Service

You can view the bindings for a service on the **Service Bindings** tab. The **Service Bindings** tab lists services and bindings configured in a service. The left pane lists the services and respective bindings. The red indicator icon with count displays the missing configuration. Click the binding name to update or view the configuration. The right pane of this page displays the services and respective bindings. You can add, update, or delete a binding.

The following figure shows the bindings for a service:



com.tibco.restbt.sample.bookstore 1.0.0.v2018-11-06-1349

BookStoreResource

Select Binding Type: SOAP

Name	Type
RESTService_Binding1	REST Binding

You can view YAML configuration for a Binding by clicking the **View YAML** next to the binding as highlighted in the preceding image.

Adding a Binding to a Service

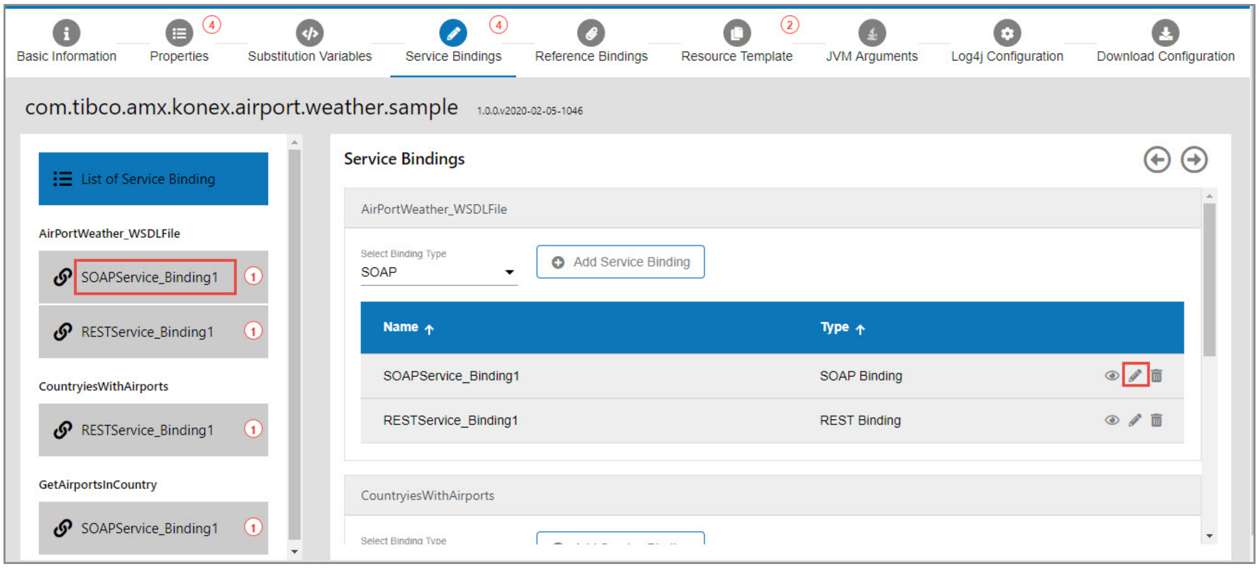
On the **Service Bindings** tab, you can add a binding to a service.

1. In the Application Configurator, click the **Service Bindings** tab. The list of services and their bindings is displayed.
2. Select the type from the drop-down list for the service in the right pane.
3. Click **Add Service Binding**.
4. Enter a name for the binding.
5. Specify binding properties (which are across multiple tabs).
6. Click **Save** .

Editing a Binding

On the **Service Bindings** tab, you can edit a binding.

1. In the Application Configurator, click the **Service Bindings** tab.
2. Click the name of the binding in the left pane or click **Edit** next to the binding in the right pane.



Name ↑	Type ↑
SOAPService_Binding1	SOAP Binding
RESTService_Binding1	REST Binding

3. Edit the configuration information and click **Save** ✓. For information about the fields, see Viewing and Managing Service Bindings.

Deleting a Binding

On the **Service Bindings** tab, you can delete a binding.

1. In the Application Configurator, click the **Service Bindings** tab.
2. Click **Delete** 🗑 next to the binding that you want to delete.

SOAP Bindings

SOAP bindings serve as a gateway for inbound and outbound SOAP messages. SOAP bindings expose endpoints that accept requests from SOAP consumers and allow composites to invoke external SOAP providers.

SOAP bindings support the following features:

- SOAP 1.1 and SOAP 1.2 specifications.
- Encoding: Document-literal and RPC-literal
- Message exchange patterns: one-way, request-response, and fault
- HTTP and JMS transport
- SOAP headers

- WS-Addressing
- WS-Reliable Messaging

See the following topic for more information about the fields that you can configure in the SOAP Binding:

- SOAP Binding Reference

SOAP Binding Reference

You can specify the endpoint, SOAP defaults, service transport, and reference transport for the binding node. You can specify the SOAP general configuration for the operation node and the part list for the input or output message node.

Binding - Service

Field	**Required?**	**Editable?**	**Description**
Name	Y	Y	The name of the binding.

Transport Configuration

Field	**Required?**	**Editable?**	**Accepts SVARs?**	**Description**
HTTP				
Transport Type	Y	Y	N	Type of transport supported by the binding: HTTP or JMS.
Endpoint URI	Y	Y	N	The endpoint URI. This field is populated from the SOAP address element of the WSDL port associated with the SOAP-HTTP service binding.
Connector Name	Y	Y	Y	The name of the HTTP connector resource that

Transport Configuration(Continued)

Field	Required?	Editable?	Accepts SVARs?	Description
				provides incoming transport services. Default: None
Session Inactivity Timeout (s)	N	Y	N	The time before an invocation of the endpoint times out. **Default:** 60
JMS				
Transport Type	Y	Y	N	The type of transport supported by the binding: HTTP or JMS.
Message Type	Y	Y	N	The type of the message content: Text or Bytes. **Default:** Text
Binding Specification	Y	Y	N	The binding specification supported: TIBCO SOAP/JMS or W3C SOAP/JMS. **Default:** TIBCO SOAP/JMS
JMS - Inbound Configuration				
Acknowledgement Mode	Y	N	N	The acknowledgment mode for incoming messages. Set to Auto, meaning that the message is automatically acknowledged when it is

Transport Configuration(Continued)

Field	Required?	Editable?	Accepts SVARs?	Description
				received. Default: AUTO_ ACKNOWLEDGE
JMS Connection Factory	Y	Y	N	A JMS Connection Factory
JMS Destination	Y	Y	N	A JMS Destination **Note:** Only queues are supported for SOAP/JMS. Topics are not supported.
JMS - Outbound Configuration				
JMS Connection Factory	Y	Y	N	A JMS Connection Factory
Delivery Mode	N	Y	N	The delivery mode of messages: • Persistent Messages are stored and forwarded. • Non-Persistent Messages are not stored and may be lost due to failure. **Default:** Persistent
Message Priority	N	Y	N	The priority of the message. **Valid value range:** 0-9

Transport Configuration(Continued)

Field	Required?	Editable?	Accepts SVARs?	Description
				Higher numbers signify a higher priority (that is, 9 is a higher priority than 8). **Default:** 4
Message Expiration	N	Y	N	The time for which a message can remain active. **Default:** 0 (that is, the message does not expire)
Correlation Scheme	Y	Y	N	Scheme which identifies the correlation scheme used when sending reply messages. • MessageID to CorrelationID — Message ID of the request message is copied to the Correlation ID of the response message. • CorrelationID to CorrelationID — Correlation ID of the request message is copied to the Correlation ID of the response message. • Infer from Request — If the CorrelationID is present in the incoming Request

Transport Configuration(Continued)

Field	Required?	Editable?	Accepts SVARs?	Description
				Message, the CorrelationID of the incoming Request Message is copied to the CorrelationID of the outgoing Response Message. If the CorrelationID is absent in the incoming Request Message, the MessageID of the incoming Request Message (which is always present) is copied to CorrelationID of the outgoing Response Message. **Default:** MessageID to CorrelationID

SOAP Default Configuration

Field	Required?	Editable?	Accepts SVARs?	Description
Description	N	Y	N	A description of the binding.
SOAP Version	Y	Y	N	The version of the SOAP specification: 1.1 or 1.2 **Default:** 1.1
Style	Y	Y	N	The SOAP binding style:

SOAP Default Configuration(Continued)

Field	Required?	Editable?	Accepts SVARs?	Description
				Document or RPC **Default:** Document
Encoding	Y	N	N	The encoding type for the body of the SOAP input and output messages. Set to Literal.
Target NameSpace	N	Y	N	The target namespace for a concrete WSDL file for the service.
Enable Request Message Validation	N	Y	Y	Schema validations can be enabled for SOAP Binding by using this field. The validations can be enabled or disabled by selecting true and false values from the drop-down list respectively. The field can also be configured as Substitution Variables by selecting the SVAR option. Selecting the SVAR option enables a text box where you can enter the name of the Substitution Variable. Default: FALSE
Enable Response Message Validation	N	Y	Y	Schema validations can be enabled for SOAP Binding by using this field. The validations can be enabled or disabled by selecting true and false values from the drop-down list respectively. The field can also be configured as Substitution Variables by selecting the SVAR

SOAP Default Configuration(Continued)

Field	Required?	Editable?	Accepts SVARs?	Description
				option. Selecting the SVAR option enables a text box where you can enter the name of the Substitution Variable. Default: FALSE
Sender Identifier Expression	N	Y	Y	This field contains an XPath expression, which identifies the sender (client or reference application). This XPath expression is used at runtime (when a SOAP Request is sent to the SOAP service endpoint) to extract the sender identification information.

Operation Configuration

Field	Required?	Editable?	Accepts SVARs?	Description
Style	Y	Y	N	The SOAP binding style: Document or RPC. **Default:** Document
Encoding	N	N	N	The encoding type for the body of the SOAP input and output messages. Set to Literal.
Sender Identifier Expression	N	Y	Y	This field contains an XPath expression, which identifies the sender (client or reference application). This XPath expression is used at runtime (when a SOAP Request is sent to the SOAP service endpoint) to extract the sender identification information.

The following is displayed only when you click INPUT/OUTPUT:

Part List in the Input/Output Message Node

Field	Description
Part Name	The name of the message part.
Part Type	The type of the message part: Body or Header.

JMS Bindings

JMS bindings integrate JMS applications with TIBCO ActiveMatrix. The JMS bindings convert JMS messages to TIBCO ActiveMatrix messages and TIBCO ActiveMatrix messages to JMS messages.

Java Message Service (JMS) is a Java specification for messaging between applications. JMS is based on the creation and delivery of messages. The creator of the message is known as the publisher and the receiver of the message is known as the subscriber. A JMS server acts as an intermediary for the message and manages its delivery to the correct destination.

Configuration Overview

JMS bindings enable you to establish request and response message communication with a JMS server. In other words, adding a JMS binding enables a particular application to receive JMS messages or to send messages to the JMS server (JMS destination).

For an application to receive messages, for example, it must subscribe to a JMS server on a destination, which is defined by the JMS Connection Factory, JMS Destination, and JNDI Connection resources.

For the application to send messages, configuration details must be provided for the runtime library through the JMS Connection Factory, JMS Destination, and JNDI Connection configuration resources.

Use Cases

TIBCO ActiveMatrix supports the following JMS use cases and corresponding Message Exchange Pattern (MEPs):

- Service binding - You can create a service referencing port types of a component hosted inside TIBCO ActiveMatrix. The component hosted inside TIBCO ActiveMatrix dictates the WSDL file and provides services.

- TIBCO ActiveMatrix subscriber communicating with a JMS publisher - In-Only
- TIBCO ActiveMatrix server communicating with a JMS requestor - In-Out

JMS Binding Reference

JMS bindings include properties. You can configure most properties and several properties accept substitution variables.

Binding

Field	Required?	Editable?	Accepts Svar?	Description
Name	Y	Y	N	Name of the JMS binding.
Description	N	Y	N	Description of the JMS binding.
Connection Factory	Y	Y	N	The name of a JMS Connection Factory. Required for MEP: • In-Out (service, reference) • In-Only (service, reference)

Configuration for JMS Binding Request Communication

Field	Required?	Editable?	Accepts Svar?	Description
Destination Type	Y	Y	N	The type of JMS destination, queue, topic, or JNDI. For direct destinations, use queue or topic. For JNDI resource template, use the JMS

Field	Required?	Editable?	Accepts Svar?	Description
				destination resource template. Required for MEP: - In-Only (service, reference) - In-Out (service, reference)
Destination	Y	Y	N	(This property is only applicable for the JNDI destination type.) The name of a JMS destination in case of JMS destination resource template. Required for MEP: - In-Only (service, reference) - In-Out (service, reference)
Queue Name	N	Y	Y	Name of the queue if destination type is selected as queue.
Topic Name	N	Y	Y	Name of the topic if destination type is selected as topic.

Configuration for Reply JMS message, applicable for In-Out MEP

Field	Required?	Editable?	Accepts Svar?	Description
Destination Type	Y	Y	N	The type of JMS destination, queue, topic or JNDI. For direct destinations, use queue or topic. For JNDI resource template, use the JMS destination resource template. **Default:** Same as Request Message

Field	Required?	Editable?	Accepts Svar?	Description
				Note: The 'Same as Request Message' option indicates that the Reply Message destination type is same as the Request Message destination type. In CLI script, there is no such option. You must select queue, topic, or JNDI. Required MEP: In-Out (service, reference)
Destination	Y	Y	N	(This property is only applicable for JNDI destination type.) The name of a JMS destination in case of JMS destination resource template. If unspecified, a temporary destination name derived from the value of JMSReplyTo JMS header is used.
Queue Name	N	Y	Y	Name of the queue if destination type is selected as queue.
Topic Name	N	Y	Y	Name of the topic if destination type is selected as topic.

Note: In case of In-Out MEP, even when Reply Message is configured, priority is given to JMSReplyTo JMS Message header and reply is sent on the destination represented by the JMSReplyTo header value. Clients must not set this header field when fixed reply destination is used.

Advanced Settings for JMS Binding

Field	Required?	Editable?	Accepts Svar?	Description
Reply Message **Note:** If Request or Reply message destination type is set to queue or topic and the JMS provider does not support dynamic queue or topic creation, or the user of provider does not have create permissions, create a queue or topic before deploying the application.				
Connection Factory	Y	Y	N	Name of the JMS connection factory resource template. **Default:** Same as Request Message **Note:** The 'Same as Request Message' option indicates that the Reply Message connection factory is same as the Request Message connection factory. In CLI script, there is no such option. Required MEP: In-Out (service, reference)
Correlation Scheme	Y	Y	Y	Scheme which identifies the correlation scheme used when sending reply messages. Required if the reply destination is set. The correlation schemes are: • `RequestCorrelIDtoCorrelID` - Correlation ID of the request message is copied to the Correlation ID of the response message. • `RequestMsgIDtoCorrelID` - Message ID of the request message is copied to the Correlation ID of the response message.

Field	Required?	Editable?	Accepts Svar?	Description
				Default: `RequestCorrelIDtoCorrelID` **Note:** `RequestMsgIDtoCorrelID` correlation scheme is not supported for topic set as static reply destination.
Operation Selection				
Type	Y	Y	N	(Applicable only in case of multiple operations.) Operation selection scheme in case of multiple operations. SCA and custom are supported. In case of custom scheme, other properties (JMS Property Name and Error Action) are not editable but Message Selector configuration on each operation is mandatory. See "Operation Node" for more details.
JMS Property Name	Y	Y	Y	(Applicable only in case of multiple operations.) Name of the JMS property to be used for operation selection in case of multiple operations and `SCA` operation selection type. **Default property name:** `scaOperationName`
Error Action	Y	Y	N	(Applicable only in case of multiple operations.) Action to trigger when operation selection from multiple operations fails. • `Discard Message` - When selected,

Field	Required?	Editable?	Accepts Svar?	Description
				runtime discards the message when operation selection fails. - `Send Message To Operation` - When selected, the user can inform runtime to send the message to a particular configured operation when operation selection fails. - `Send Message To Error Queue` - When selected, the user can inform runtime to send the message to a configured error queue when operation selection fails. - `Retain Message in Service Destination` - When selected, the user can inform runtime to retain the message in the service request destination configured in Request Message section. **Default error action:** `Discard Message`
Operation Name	Y	Y	Y	Displayed when the `Send Message to Operation` error action is selected. Operation name to send the message in case the operation selection fails and `Send Message to Operation` is configured.
Error Queue Name	N	Y	Y	Displayed when the `Send Message to Error Queue` error action is selected. Error queue to send the JMS message in case the operation selection fails and `Send Message to Error Queue` error action is configured.

Field	Required?	Editable?	Accepts Svar?	Description
Fault Selection				
JMS Property Name		Y	Y	JMS property name used to send the fault as a value. **Default property name:** `faultName`

Interface Settings

Field	Required?	Editable?	Accepts Svar?	Description
Operation Selection				
Message Selector	N	Y	Y	A JMS message selector allows the client to specify, by message header and properties, the messages it is interested in. Message selector on Interface Settings is configurable when the Error Action in the Operation Selection settings is other than `Retain Message in Service Destination` and Operation Selection Type is `SCA`.
Request Message				
Message Type	Y	Y	Y	Message type used for request messages. The type is one of the following: • XML Text - A text message carrying XML payload that confirms to specified schema. • Bytes - Binary data • Text - A text message carrying a

Field	Required?	Editable?	Accepts Svar?	Description
				payload of type `xsd:string`. - XML Bytes - XML content sent as bytes. (JMS resources treat this type as bytes but JMS bindings expect content in XML.) **Default:** XML Text
Durable Subscription	N	Y	Y	(Applicable only if the Request Message Destination type is Topic.) Configurable only in JMS binding on Promoted Service. Specifies a durable subscription. You must specify a name in the Durable Subscription field which gets registered with the JMS application as the durable subscription name.
Subscription Name	Y	Y	Y	(Applicable only if the Request Message Destination type is Topic.) Configurable only in JMS binding on Promoted Service. The subscription name registered with the JMS application for durable subscriptions. This field is only available when the Durable Subscription field is selected.
Delivery Mode	Y	Y	Y	(Configurable only in JMS binding on Promoted Reference.) The delivery mode of the message. The mode is one of the following:

Field	Required?	Editable?	Accepts Svar?	Description
				• Persistent - Messages are stored and forwarded. • Non-Persistent - Messages are not stored and could be lost due to failures in transmission. • TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic enabling higher message rates. **Default:** Non-Persistent
Message Priority	Y	Y	Y	(Configurable only in JMS binding on Promoted Reference.) Priority of the message. **Valid value range:** 0-9
Message Expiration	Y	Y	Y	(Configurable only in JMS binding on Promoted Reference.) The time, in milliseconds, for which request message is retained by the JMS provider.
Operation Timeout	N	Y	Y	(Configurable only in JMS binding on Promoted Reference.) The period that the JMS binding waits for the response to arrive.

Field	Required?	Editable?	Accepts Svar?	Description
				Default: If the MEP is In-Out, the default values are 6000 ms at the port type and operation levels. If other values (non-default values) are specified, these values take effect, with the value at the operation level given precedence. **Note:** Operation timeout is applicable for a reference only. For a service, add a thread policy on a component service and set timeout on the thread policy.
Reply Message				
Message Type	Y	Y	Y	Message type used for reply messages. The type is one of the following: • XML-Text - A text message carrying XML payload that confirms to specified schema. • Bytes - Binary data • Text - A text message carrying a payload of type `xsd:string`. • XML Bytes - XML content sent as bytes. (JMS resource treat this type as bytes but JMS bindings expect content in XML.) **Default:** XML-Text
Delivery Mode	Y	Y	Y	(Configurable only in JMS binding on Promoted Service.) The delivery mode of the message.

Field	Required?	Editable?	Accepts Svar?	Description
				The mode is one of the following: • Persistent - Messages are stored and forwarded. • Non-Persistent - Messages are not stored and could be lost due to failures in transmission. • TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic, enabling higher message rates. **Default:** Non-Persistent
Message Priority	N	Y	Y	(Configurable only in JMS binding on Promoted Service.) Priority of the message. **Valid value range:** 0-9
Message Expiration	N	Y	Y	(Configurable only in JMS binding on Promoted Service.) The time, in milliseconds, for which reply messages are retained by the JMS provider.
Fault Message: This section is visible only in JMS Binding on Promoted Service and if operation has defined faults. It is applicable only for In-Out-Fault MEP.				

Field	**Required?**	**Editable?**	**Accepts Svar?**	**Description**
Override Reply Message	N	Y	N	Configuration from Reply Message is INHERITED by default. To Override Reply Message configuration in the Interface Settings for Fault Message, select `Override Reply Message`.
Message Type	Y	Y	Y	Message type used for reply messages. The type is one of the following: • XML-Text - A text message carrying XML payload that confirms to specified schema. • Bytes - Binary dataText - A text message carrying a payload of type `xsd:string`. • xmlBytes - XML content sent as bytes. (JMS resource treat this type as bytes but JMS bindings expect content in XML.) **Default:** XML-Text
Delivery Mode	Y	Y	Y	The delivery mode of the message. The mode is one of the following: • Persistent - Messages are stored and forwarded. • Non-Persistent - Messages are not stored and could be lost due to failures in transmission. • TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a

Field	Required?	Editable?	Accepts Svar?	Description
				receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic enabling higher message rates. **Default:** Non-Persistent
Message Priority	Y	Y	Y	Priority of the message. **Valid value range:** 0-9
Message Expiration	Y	Y	Y	The time, in milliseconds, for which reply message is retained by the JMS provider.

Operation Configuration

Field	Editable?	Accepts Svar?	Description
Operation Settings			
Name	N	N	Operation name.
Description	Y	N	Notes for operation name.
Operation Selection **Note:** Configurable only in JMS Binding on Promoted Service.			
Message Selector	Y	Y	A JMS message selector allows a client to specify, by message header, the messages it is interested in. Message Selector is displayed only when the Operation Selection Type is Custom or the Operation Selection Error Action is `Retain Message in Service Destination` and is used as a operation selector for

Field	Editable?	Accepts Svar?	Description
			the selected operation.
Request Message			
Override Request Message	Y	N	Override INHERITED Request Message configuration from the Interface Settings for this operation only. If selected, Message Type can be overridden.
Message Type	Y	Y	Message type used for request messages. The type is one of the following: - XML Text - A text message carrying XML payload that confirms to specified schema. - Bytes - Binary data - Text - A text message carrying a payload of type `xsd:string`. - XML Bytes - XML content sent as bytes. (JMS resource treat this type as bytes but JMS bindings expect content in XML.) **Default:** XML Text
Durable Subscription	Y	Y	Specifies a durable subscription. You must specify a name in the Durable Subscription field which gets registered with the JMS application as the durable subscription name. Durable subscription is displayed only when the Request Message Destination Type is Topic and Operation Selection Type is Custom or the Operation Selection Error Action is `Retain Message in Service Destination`.
Subscription Name	Y	Y	The subscription name registered with the JMS application for durable subscriptions.

Field	Editable?	Accepts Svar?	Description
			This field is only available when the Durable field is selected, the Request Message Destination Type is Topic, and the Operation Selection Type is Custom or the Operation Selection Error Action is `Retain Message in Service Destination`.
Delivery Mode	Y	Y	(Configurable only in JMS binding on Promoted Reference.) The delivery mode of the message. The mode is one of the following: • Persistent - Messages are stored and forwarded. • Non-Persistent - Messages are not stored and could be lost due to failures in transmission. • TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic enabling higher message rates. **Default:** Non-Persistent
Message Priority	Y	Y	(Configurable only in JMS binding on Promoted Reference.) Priority of the message. **Valid value range:** 0-9
Message Expiration	Y	Y	(Configurable only in JMS binding on Promoted Reference.) The time, in milliseconds, for which reply messages are retained by the JMS provider.

Field	Editable?	Accepts Svar?	Description
Operation Timeout	Y	Y	(Configurable only in JMS binding on Promoted Reference.) The period that the JMS binding waits for the response to arrive. **Default:** If the MEP is In-Out, the defaults are 6000 ms at the port type and operation levels. If other values (non-default values) are specified, these values take effect, with the value at the operation level given precedence. **Note:** Operation Timeout is applicable for a reference only. For a service, add a thread policy on a component service and set timeout on the thread policy.
Reply Message			
Override Reply Message	Y	N	Override INHERITED Reply Message configuration from Interface Settings for this operation only.
Message Type	Y	Y	Message type used for reply messages. The type is one of the following: • XML Text - A text message carrying XML payload that confirms to specified schema. • Bytes - Binary data • Text - A text message carrying a payload of type `xsd:string`. • XML Bytes - XML content sent as bytes. (JMS resource treat this type as bytes but JMS bindings expect content in XML.) **Default:** XML Text
Delivery	Y	Y	(Configurable only in JMS binding on Promoted

Field	Editable?	Accepts Svar?	Description
Mode			Service.) The delivery mode of the message. The mode is one of the following: • Persistent - Messages are stored and forwarded. • Non-Persistent - Messages are not stored and could be lost due to failures in transmission. • TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic enabling higher message rates. **Default:** Non-Persistent
Message Priority	Y	Y	(Configurable only in JMS binding on Promoted Service.) Priority of the message. **Valid value range:** 0-9
Message Expiration	Y	Y	(Configurable only in JMS binding on Promoted Reference.) The time, in milliseconds, for which reply messages are retained by the JMS provider.
Fault Message **Note:** This section is visible only if faults are configured.			
Override Fault Message	Y	N	Override INHERITED fault message configuration from the Interface Settings.

Field	Editable?	Accepts Svar?	Description
Fault Name	N	N	Name of the fault.
Message Type	Y	Y	Message type used for reply messages. The type is one of the following: • XML Text - A text message carrying XML payload that confirms to specified schema. • Bytes - Binary data • Text - A text message carrying a payload of type `xsd:string`. • XML Bytes - XML content sent as bytes. (JMS resource treat this type as bytes but JMS bindings expect content in XML.) **Default:** XML Text
Delivery Mode	Y	Y	The delivery mode of the message. The mode is one of the following: • Persistent - Messages are stored and forwarded. • Non-Persistent - Messages are not stored and could be lost due to failures in transmission. • TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic enabling higher message rates. **Default:** Non-Persistent
Message Priority	Y	Y	Priority of the message. **Valid value range:** 0-9

Field	Editable?	Accepts Svar?	Description
Message Expiry	Y	Y	(Configurable only in JMS binding on Promoted Service.) The time, in milliseconds, for which reply messages are retained by the JMS provider.

REST Bindings

REST bindings allow you to integrate your SCA services with clients that use REST instead of SOAP, over HTTP, to invoke services. Services can be exposed as REST services that can consume Badgerfish JSON, Standard JSON, or XML.

See the following topic for more information about the fields that you can configure in the REST Binding:

- REST Binding Reference

REST Binding Reference

REST Binding - Service

Field	Required?	Editable?	Accepts SVARs?	Description
Name	Y	Y	N	The name of the binding.

Transport Configuration

Field	Required?	Editable?	Accepts SVARs?	Description
Context root	Y	Y	N	The context in which the application is invoked. That is, the base path for the URLs exposed by the REST binding. Default: None **Note:** Context Root and Connector Name

Field	Required?	Editable?	Accepts SVARs?	Description
				together define the URL that is used at runtime. You can define and name the HTTP Connector at design time. At runtime, you must create a resource of type HTTP Connector and assign it the name you used at design time. See also HTTP Connector.
Connector Name	Y	Y	Y	The name of the HTTP Connector resource that provides incoming transport services. Both HTTP and HTTPS are supported. **Default:** None **Note:** Connector Name and Context Root together define the URL that is used at runtime. You can define and name the HTTP Connector at design time. At runtime, you must create a resource of type HTTP Connector and assign it the name you used at design time. Click **New** to create the new resource template. See also HTTP Connector.

REST Default Configuration

Field	Required?	Editable?	Accepts SVARs?	Description
Operation Name	Y	N	N	The name of the operation.
HTTP Method	Y	Y	N	The HTTP method to indicate the action to be performed for a given resource.

Field	Required?	Editable?	Accepts SVARs?	Description
				Supported methods are: • GET • POST • PUT • DELETE **Default:** GET
Media Type	Y	Y	N	The media type for the request or response message. Supported formats are: • STANDARD_JSON • XML • BADGERFISH_JSON
Path	Y	Y	N	Path can be any URI on which a given operation can be exposed.
Exclude namespace from response	N	Y	N	Excludes namespaces from the response message. This option is displayed only when the **Media Type** is set to BADGERFISH_JSON. **Default:** Unchecked

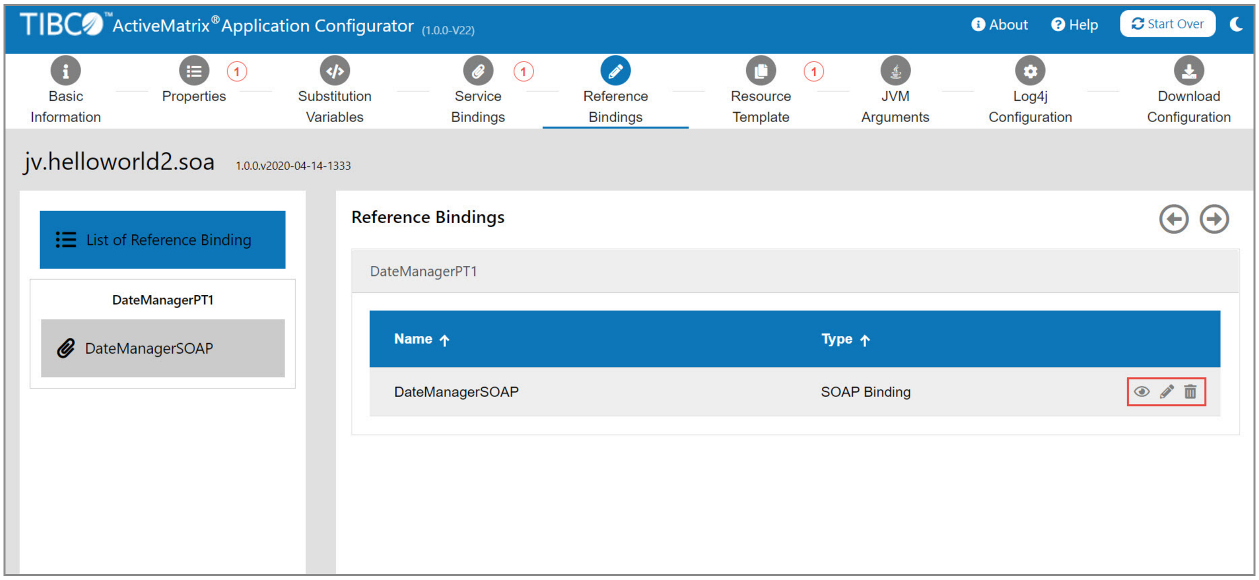
Viewing and Managing Reference Bindings

A *Reference* identifies the service consumed by a component or composite. Component references consume services provided by other components in the same composite or services provided outside the composite. A composite reference has an interface and one binding.

Viewing Bindings for a Reference

You can view the bindings for a reference on the **Reference Bindings** tab. The **Reference Bindings** tab lists the references and bindings configured in a reference. The left pane lists the references and respective bindings. Click the binding name to update or view the configuration. The right pane of this page displays the references and respective bindings. You can add, update, or delete the binding.

The following image shows the bindings for a reference:



Name ↑	Type ↑
DateManagerSOAP	SOAP Binding

You can view YAML configuration for a Reference Binding by clicking the **View YAML** next to the binding as highlighted in the preceding image.

Adding a Reference Binding

You can add a reference binding on the **Reference Bindings** tab.

1. In the Application Configurator, click the **Reference Bindings** tab.
2. Select the type from the drop-down list for the particular reference in the right pane.

> **Note:** In ActiveMatrix Service Grid - Container Edition 1.0.0, you can add or update only SOAP reference binding. Adding or updating JMS and REST reference binding is not supported.

3. Click **Add Reference Binding**. Reference can have only one binding, therefore if there

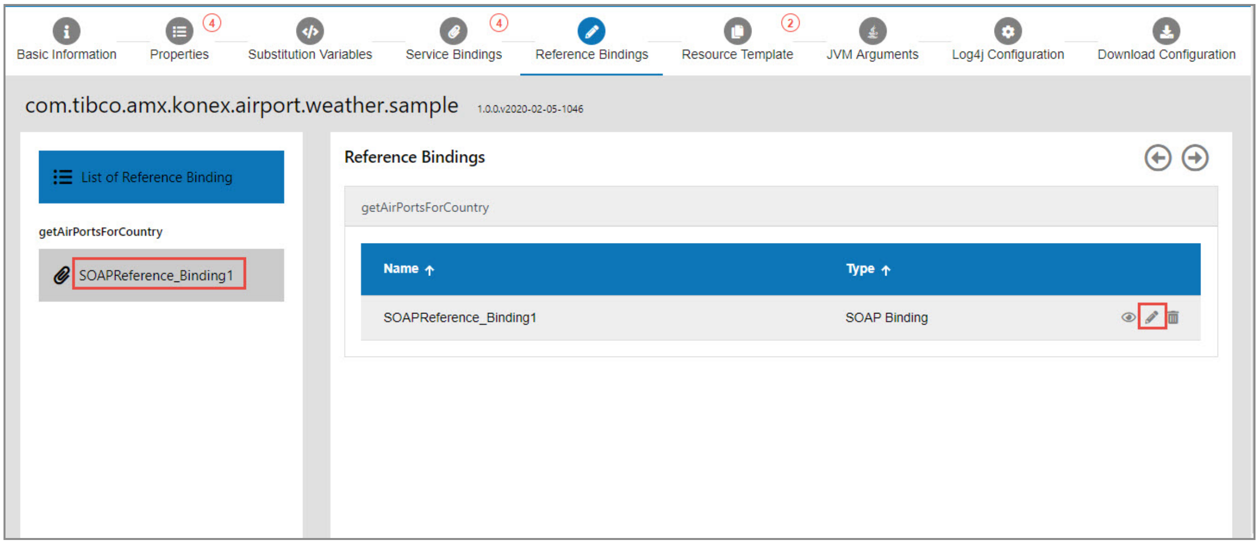
is already one Binding, **Add Reference Binding** button is not visible.

4. Enter the name of the binding.
5. Specify binding properties. For more information, see SOAP Reference Binding Properties.
6. Click **Save** ✓.

Editing a Reference Binding

You can edit a binding on the **Reference Bindings** tab.

1. In the Application Configurator, click the **Reference Bindings** tab.
2. Click the name of the binding in the left pane or click **Edit** ✎ next to the binding in the right pane.



com.tibco.amx.konex.airport.weather.sample 1.0.0.v2020-02-05-1046

Name ↑	Type ↑
SOAPReference_Binding1	SOAP Binding

3. Edit the configuration information and click **Save** ✓. For information about the fields, see SOAP Reference Binding Properties.

Deleting a Reference Binding

You can delete a binding on the **Reference Bindings** tab.

1. In the Application Configurator, click the **Reference Bindings** tab.
2. Click **Delete** 🗑 next to the binding that you want to delete.

SOAP Reference Binding Properties

When you configure a binding for a reference, you are prompted for information about the binding.

Field	Required?	Editable?	Description
Name	Y	Y	Name of the binding.
Type	Y	Y	Type of binding. **Default:** SOAP binding
Transport Type	Y	Y	Type of transport supported by the binding: HTTP or JMS.
SOAP Version	N	N	Version of the SOAP specification: 1.1 or 1.2
Enable Request Message Validation	Y	Y	Schema validations are enabled for the SOAP binding by using this field. The validations are enabled or disabled by selecting true or false values from the drop-down list respectively. The field can also be configured as substitution variables by selecting the SVAR option. Selecting the SVAR option enables a text box where you can enter the name of the substitution variable.
Enable Response Message Validation	Y	Y	Schema validations are enabled for the SOAP binding by using this field. The validations are enabled or disabled by selecting true or false values from the drop-down list respectively. The field can also be configured as substitution variable by selecting the **Substitution Variable** option in the drop-down list. Selecting the **Substitution Variable** option enables a text box where you can enter the name of the substitution variable.

Field	Required?	Editable?	Description
HTTP			
HTTP Client Configuration	Y	Y	The HTTP Client resource template represents an outgoing HTTP connection.
Enable WS-Addressing	N	Y	Indicates whether to enable the WS-Addressing headers. When selected, the Connector Name field is displayed.
Filespec or Endpoint URL	Y	Y	The endpoint URL. This field is populated from the SOAP address element of the WSDL port associated with the SOAP-HTTP reference binding. This value can be edited by typing the new value or by using the substitution variables picker to select a substitution variable that points to a valid endpoint URL value.
Override HTTP Client Timeout	N	Y	Select the check box to override HTTP Client connection timeout value. When the check box is selected, the **Binding Socket Timeout** field is enabled. Specify the new value in the Binding Socket Timeout field.
Binding Socket Timeout(ms)	N	Y	Defines the socket timeout (SO_TIMEOUT), which is the timeout for waiting for the data or a maximum period of inactivity between consecutive data packets. This must be changed when connecting to very slow external services. **Default:** 0 ms (that is, infinite timeout)
JMS			
Binding Specification	Y	N	Binding specification supported: TIBCO or W3C SOAP-JMS.

Field	Required?	Editable?	Description
			Default: TIBCO
JMS - Inbound			
Acknowledge Mode	Y	N	Acknowledgment mode for incoming messages. Set to Auto, meaning that the message is automatically acknowledged when it is received.
Reply Destination	Y	Y	A JMS Destination specifies destination objects, which represent virtual channels (topics and queues) in JMS.
JMS-Outbound			
JMS Connection Factory	Y	Y	A JMS Connection Factory resource.
JMS Destination	Y	Y	A JMS Destination specifies destination objects, which represent virtual channels (topics and queues) in JMS. **Note:** Only queues are supported for SOAP/JMS. Topics are not supported.
Delivery Mode	Y	Y	The delivery mode of messages: • Persistent Messages are stored and forwarded. • Non-Persistent Messages are not stored and may be lost due to failure. **Default:** Persistent
Message Priority	Y	Y	The priority of the message. **Valid value range:** 0-9

Field	Required?	Editable?	Description
			Higher numbers signify a higher priority (that is, 9 is a higher priority than 8). **Default:** 0
Message Expiration	N	Y	The time for which a message can remain active. **Default:** 0 (that is, the message does not expire)
Correlation Scheme	Y	Y	Scheme which identifies the correlation scheme used when sending reply messages. • MessageID to CorrelationID: Message ID of the request message is copied to the Correlation ID of the response message. • CorrelationID to CorrelationID: Correlation ID of the request message is copied to the Correlation ID of the response message. **Default:** MessageID to CorrelationID

JMS Reference Binding

The JMS reference binding is not supported in ActiveMatrix Service Grid - Container Edition 1.0.0.

REST Reference Binding

You cannot add or update REST reference binding by using the Application Configurator.

If an application DAA contains REST reference binding already configured in it, it is displayed in the list of bindings for particular references. Adding or updating REST reference binding is not supported.

Managing Resource Templates

A resource template specifies configuration details for resources such as JMS connections and JDBC connections. Resources help lose the coupling between business logic and infrastructure support, such as messaging and database access. They also enable better management of the infrastructure (for example: caching, connection pooling, timeout, security, and so on.)

Resources eliminate the need to provide the infrastructure details in the services, component implementations, and references. Instead, you can specify a property of the type of required resource in the service, component, or reference. When deploying a configured ActiveMatrix Service Grid - Container Edition application to a container, the resource template is mapped to the application property; the resource is created from the resource template and is used by the application.

Creating a Resource Template

You can create a resource template on the **Resource Template** tab in the Application Configurator.

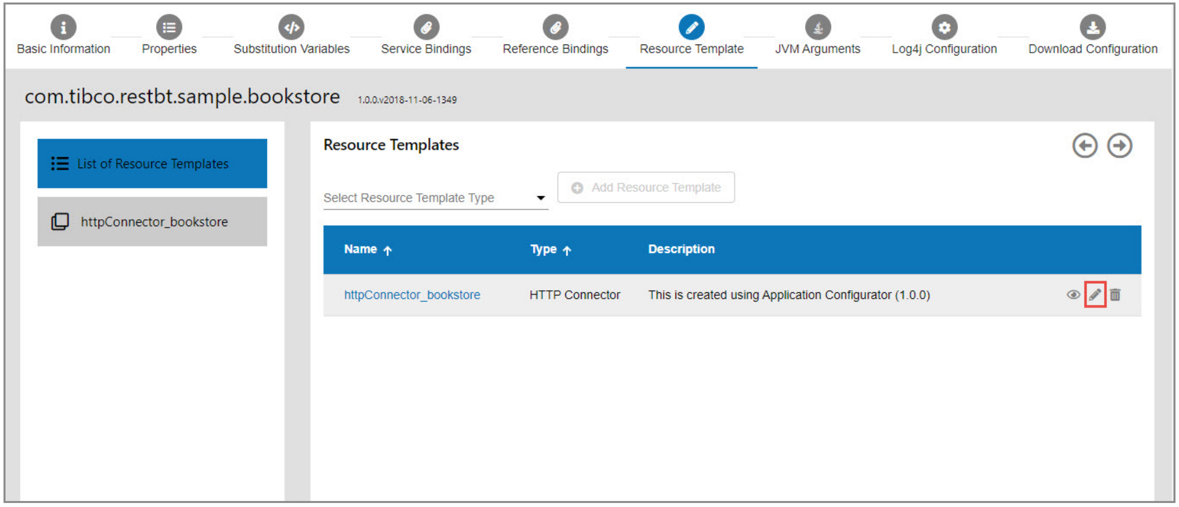
1. Click the **Resource Template** tab.
2. Select a resource type from the **Select Resource Template Type** drop-down list.
3. Click **Add Resource Template**.
4. Edit the template configuration fields. The name of the resource template must not contain colon (:) or ampersand (&) characters. For more information, see Resource Templates Reference.
5. Click **Save** ✓.

Result
The resource template is added to the list of resource templates.

Editing a Resource Template

You can edit a resource template from the resource templates list in the Application Configurator.

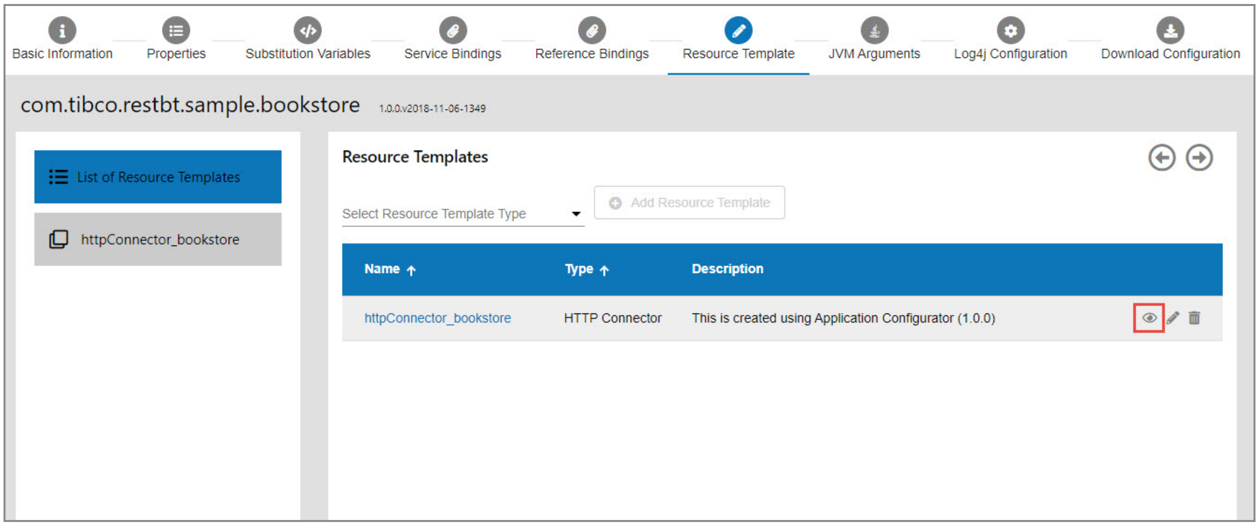
1. Click the **Resource Template** tab.
2. To edit a resource template, click the name of the resource template from the list of resource templates in the left pane. Alternatively, in the list of resource templates in the right pane, click **Edit** ✎ next to the resource template that you want to edit.



Name ↑	Type ↑	Description
httpConnector_bookstore	HTTP Connector	This is created using Application Configurator (1.0.0)

3. Edit the configuration fields as required. For more information, see Resource Templates Reference.
4. Click **Update** ✓ .

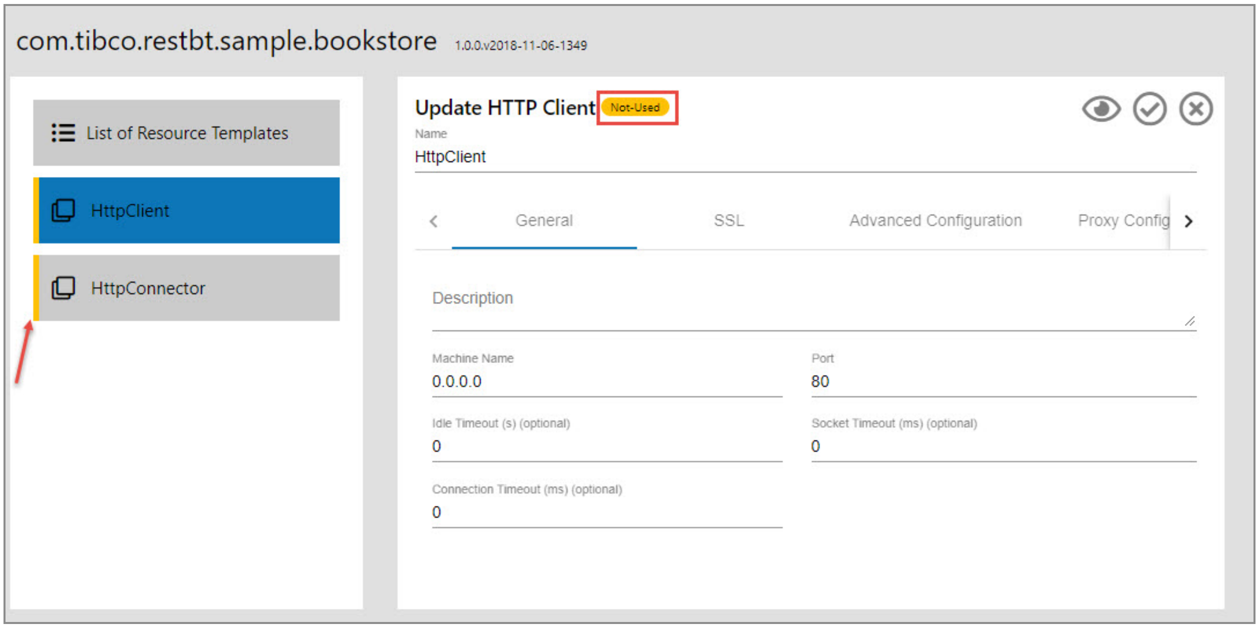
You can view YAML configuration for a resource template by clicking the **View YAML** 👁 next to the resource template as highlighted in the following image:



Name ↑	Type ↑	Description
httpConnector_bookstore	HTTP Connector	This is created using Application Configurator (1.0.0)

If there is a **Not-Used** label beside the name of the resource template, it indicates that the resource template is not used in the current application configuration. In the **List of**

Resource Templates pane, you can see a yellow bar for the resource template which is not used in the configuration.



Name: HttpClient

Machine Name: 0.0.0.0

Port: 80

Idle Timeout (s) (optional): 0

Socket Timeout (ms) (optional): 0

Connection Timeout (ms) (optional): 0

Renaming a Resource Template

You can rename a resource template when editing the resource template. For more information, see Editing a Resource Template.

Deleting a Resource Template

You can delete a resource template from the resource list in the UI.

1. Click the **Resource Template** tab.
2. From the list of resource templates in the right pane, click **Delete** 🗑 next to the resource template that you want to delete.

Result

The resource template is deleted from the list of resource templates.

Testing a Connection to a JDBC Resource

You can test the connection to the configured database when you create or update the JDBC resource template from the Application Configurator. If any issues are found in the configuration, you will see a message as JDBC Test Connection FAILED. Use this feature to verify database configuration errors before the application attempts the JDBC connection.

Before you begin

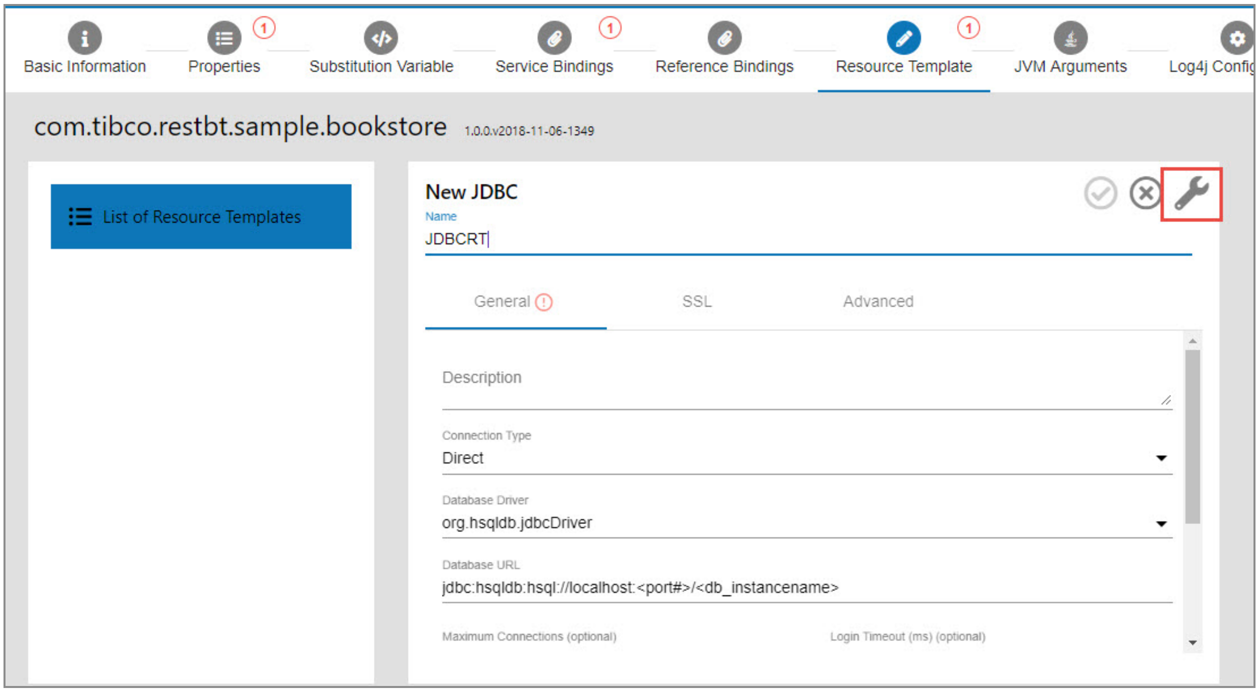
- Before creating Application Configurator image, copy custom features(JDBC driver DAA) to `amsgce-runtime-<version>/applicationConfigurator/application/`.
- Create Application Configurator image, so these custom features will be enabled for Application configurator.
- Alternatively, if you are running Application Configurator in Docker, mount a local directory containing the database drivers with the Application Configurator container directory `thirdpartyjdbcdrivers` by using the following Docker command:

```
docker run -p 8087:8087  -v
/home/tibco/jars/:/opt/tibco/tibco.home/thirdpartyjdbcdrivers --name
amxceac amxce_ac:1.0
```

Note: TIBCO recommends that you mount only one database driver version JAR in the `thirdpartyjdbcdrivers` directory for each database that you want to use. If there are more than one driver JAR files, the first in the list is used to make the database test connection.

Procedure

1. (Optional) To check enabled database drivers, click **About** in the upper-right corner.
2. In the Application Configurator, click the **Resource Template** tab.
3. When creating or updating a JDBC resource template, click **Test Connection** 🔧 in the upper-right corner.



com.tibco.restbt.sample.bookstore 1.0.0.v2018-11-06-1349

New JDBC

Name
JDBCRT

Description

Connection Type
Direct

Database Driver
org.hsqldb.jdbcDriver

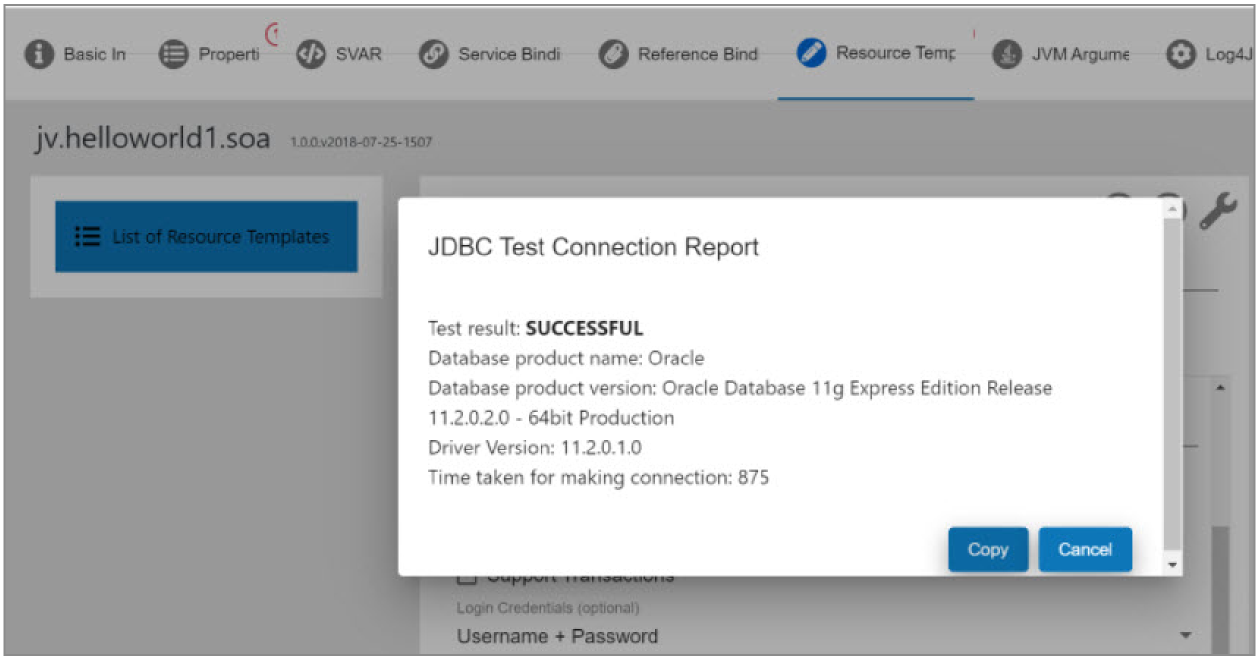
Database URL
jdbc:hsqldb:hsql://localhost:<port#>/<db_instancename>

Maximum Connections (optional)

Login Timeout (ms) (optional)

Result

If the test connection is successful, you can see the test result as **SUCCESSFUL**.



jv.helloworld1.soa 1.0.0.v2018-07-25-1507

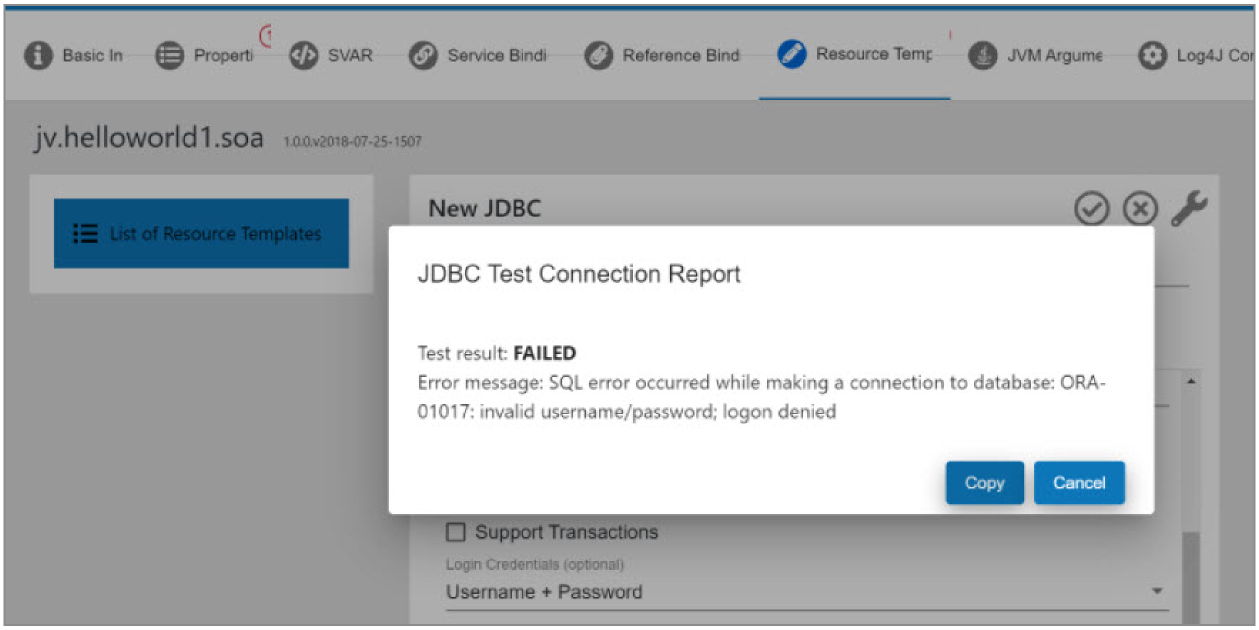
JDBC Test Connection Report

Test result: **SUCCESSFUL**
Database product name: Oracle
Database product version: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
Driver Version: 11.2.0.1.0
Time taken for making connection: 875

Support Transactions

Login Credentials (optional)
Username + Password

If the test connection fails, test result is displayed as **FAILED** with the error message. You can copy the test results by using the **Copy** button.



jv.helloworld1.soa 1.0.0.v2018-07-25-1507

New JDBC

JDBC Test Connection Report

Test result: **FAILED**
Error message: SQL error occurred while making a connection to database: ORA-01017: invalid username/password; logon denied

□ Support Transactions

Login Credentials (optional)
Username + Password

Resource Templates Reference

The topics in this section provide detailed information about the properties of resource templates in the UI.

> **Note:** Unlike ActiveMatrix Service Grid, in Application Configurator, you cannot set a blank password for all the user credential fields on the UI.

HTTP Client

The HTTP Client resource template represents an outgoing HTTP connection. HTTP clients are used by a reference's SOAP binding.

General

Field	Required?	Editable?	Accepts SVars?	Description
Machine Name	Y	Y	Y	The name of the host that accepts the incoming requests.

Field	Required?	Editable?	Accepts SVars?	Description
				- For machines that have only one network card, the default value localhost specifies the current machine. - For machines that have more than one network card, this field specifies the host name of the card that is used to accept incoming HTTP requests. **Default:** localhost
Port	Y	Y	Y	The port number on which to invoke outgoing HTTP requests. **Default:** 80
Idle Timeout (s)	N	Y	Y	The time to wait before closing an inactive connection. - If it is more than zero and data transmission has not finished, a call to close the connection blocks the calling program until the data is transmitted, or until the specified timeout occurs. - If it is 0, a call to close the connection returns without blocking the caller, and an attempt is made to send the data. Normally, this transfer is successful but it cannot be guaranteed. **Note:** Change this value only on the advice from TIBCO support. **Default:** 0 s

Field	Required?	Editable?	Accepts SVars?	Description
Socket Timeout (ms)	N	Y	Y	Defines the socket timeout (`SO_TIMEOUT`), which is the timeout for waiting for data or a maximum period of inactivity between consecutive data packets. This must be changed when connecting to very slow external services. **Default:** 0 ms (that is, infinite timeout)
Connection Timeout (ms)	N	Y	Y	Determines the timeout until a connection is established. This must be changed when connecting to very slow external services. **Default:** 0 ms (that is, infinite timeout)

SSL

Field	Required?	Editable?	Accepts SVars?	Description
Enable SSL	Y	Y	N	Enables SSL connections. When selected, the SSL properties are displayed. **Default:** Unchecked
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource. Default: None

Advanced Configuration

Field	Required?	Editable?	Accepts SVars?	Description
Accept Redirect	N	N	N	Indicates whether the HTTP method should automatically follow HTTP redirects.

Field	Required?	Editable?	Accepts SVars?	Description
				This option is used when client connection receives redirect responses from the server like Moved Permanently, Moved Temporarily, Temporary Redirect, and so on. **Default:** Unchecked
Reuse Address	N	N	N	When a TCP connection is closed, the connection might remain in a timeout state for a period of time after the connection is closed (typically known as the TIME_WAIT state or 2MSL wait state). For applications that use a well-known socket address or port, sometimes you cannot bind a socket to the required SocketAddress if a connection involving the socket address or the port times out. **Default:** Unchecked
Disable Connection Pooling	N	N	N	Indicates whether to use the single or multi-threaded connection manager. **Default:** Unchecked
Suppress TCP Delay	N	N	N	Determines whether the Nagle algorithm is used. The Nagle algorithm tries to conserve bandwidth by minimizing the number of segments that are sent. When applications wish to decrease network latency and increase performance, the applications can disable Nagle's algorithm by enabling Suppress TCP Delay. Data is sent earlier at the cost of an increase in bandwidth

Field	Required?	Editable?	Accepts SVars?	Description
				consumption and the number of packets. **Default:** Checked
Stale Check	N	N	N	Determines whether the stale connection check is to be used. Disabling the stale connection check can result in slight performance improvement. However, you might see an I/O error when executing a request over a connection that is already closed at the server side. **Default:** Unchecked
Buffer Size (B)	N	Y	N	Socket buffer size, in bytes. This is the recommended buffer size to be used for data transfer over the socket. **Default:** -1 (allow the runtime to determine the buffer size)
Connection Retrieval Timeout (ms)	N	Y	Y	The timeout, in milliseconds, until a connection is established. **Default:** 0
Local Socket Address	N	Y	N	Local host address to be used for creating the socket. **Default:** None
Maximum Total Connections	N	Y	Y	Controls the maximum number of simultaneous active connection that this resource allows. Increase the value for applications that create a lot of long-lived connections.

Field	Required?	Editable?	Accepts SVars?	Description
				Default: 20
Maximum Total Connections per Host	N	Y	Y	Controls the maximum number of simultaneous active connections to the same host that this resource allows. **Note:** This number cannot be greater than the Maximum Total Connections. **Default:** 2

HTTP Proxy

Field	Required?	Editable?	Accepts SVars?	Description
Configure Proxy	N	N	N	Select the check box to configure the HTTP Proxy options described in this table. **Default:** Unchecked
Proxy Type	Y	N	N	Type of proxy server: HTTP or SOCKS V4 / V5. **Default:** HTTP
Proxy Host	Y	Y	Y	Address of the proxy host. **Default:** localhost
Proxy Port	Y	Y	Y	Port of the proxy host. **Default:** 8080
Configure BASIC authentication	N	Y	N	Select the check the box to configure access to proxy server with a user name and password. **Default:** Unchecked

Field	Required?	Editable?	Accepts SVars?	Description
				When you select this check box, the fields for specifying the user name and password are enabled. **Default user name:** None **Default password:** None Both user name and password do not accepts SVARs. If you try to update the existing encrypted password then the existing password will be removed. You can add a new password.

HTTP Connector

The HTTP Connector resource template represents an incoming HTTP connection. HTTP connectors are used by a service's SOAP binding, REST Binding and also by the WebApp component.

General

Field	Required?	Editable?	Accepts SVars?	Description
Machine Name	Y	N	N	The name of the machine that accepts the incoming requests. **Note:** In the Docker container, the IP address of the HTTP Connector is different and points to the hostname of the container. By default, the container is assigned an IP address for every Docker network it connects to. Therefore, the default value 0.0.0.0 is not editable. **Default:** 0.0.0.0

Field	Required?	Editable?	Accepts SVars?	Description
Port	Y	Y	Y	The port number on which to listen for incoming requests. **Default:** 7777
Accept Queue Size	N	Y	Y	The number of incoming requests that can be queued before additional requests are rejected. **Default:** 0, which indicates that the JVM must use the default value for the Accept Queue Size property. **Oracle JVM default value:** 50
Acceptor Threads	N	Y	Y	The number of threads dedicated to processing incoming connection requests. Ideally, you should have enough acceptor threads so that one thread is always available when needed, but the thread count does not cause a burden on the system. The threads are started when the HTTP Connector resource is installed on a node. After an acceptor thread accepts the connection, the request to the work thread pool is queued and the next connection request is processed. In general, the number of acceptor threads should be kept low. A good rule of thumb is that the number of acceptor threads should not be greater than twice the number of processors. To enable the "Acceptor Threads", you need to uncheck the "Use Non-Blocking IO Sockets" on the **Advanced** Tab.

Field	Required?	Editable?	Accepts SVars?	Description
				Default: 1 and 20

SSL

Field	Required?	Editable?	Accepts SVars?	Description
Enable SSL	N	N	N	Indicates that SSL connections must be enabled. When selected, the SSL Certificate Source field is enabled. **Default:** Cleared
SSL Server Provider	N	Y	Y	The name of an SSL Server Provider resource. **Default:** <*None*>

Advanced

Field	Required?	Editable?	Accepts SVars?	Description
Low Resources Max Idle Time (ms)	N	Y	Y	The period that a connection is allowed to be idle when there are more than (the number of) Low Resources Connections. **Default:** -1. There is no timeout.
Idle Timeout (ms)	N	Y	Y	(New property in Jetty 9) The idle timeout, in milliseconds, for I/O operations during the handling of a HTTP request. The max idle time is applied to a HTTP request for I/O operations and delayed dispatch. **Default:** 200000 ms
Request Header	N	Y	Y	(New property in Jetty 9) The maximum size of a request header.

Field	Required?	Editable?	Accepts SVars?	Description
Size (B)				Larger headers allow for more and/or larger cookies plus larger form content encoded in a URL. However, larger headers consume more memory and can make a server more vulnerable to denial of service attacks. **Default:** 4096 bytes
Response Header Size (B)	N	Y	Y	(New property in Jetty 9) The maximum size of a response header. Larger headers allow for more and/or larger cookies and longer HTTP headers (for example, for redirection). However, larger headers also consume more memory. **Default:** 4096 bytes
Output Buffer Size (B)	N	Y	Y	(New property in Jetty 9) The size of the buffer into which response content is aggregated before being sent to the client. A larger buffer can improve performance by allowing a content producer to run without blocking. However, larger buffers consume more memory and may induce some latency before a client starts processing the content. **Default:** 24576 bytes
Linger Time (ms)	N	Y	Y	The time to delay before a socket resets. Before a socket terminates a connection, it can linger, allowing unsent data to be transmitted or it can reset, which means that all unsent data is lost. **Default:** -1 (that is, there is no delay before resetting.)

Field	Required?	Editable?	Accepts SVars?	Description
Use Non-Blocking IO Sockets	N	N	N	Indicates whether to use non-blocking or blocking IO. • In the non-blocking IO, the thread reads whatever data is available and returns to perform other tasks. • In the blocking IO, the thread does not perform a read operation until all the data is available. **Default:** Checked **Restriction:** Due to a bug in the Oracle JVM version 1.6, non-blocking IO does not work when the Machine Name field contains an iPV6 address or when the machine name resolves only to an IPv6 address. If the machine name resolves to an IPv4 and an IPv6 address, the IPv4 address is used and non-blocking IO works correctly. Because of this limitation, you should either use blocking IO or use IPv4 addresses for connectors with non-blocking IO.
Use Direct Buffers	N	N	N	Indicates whether to use direct buffers with non-blocking IO. Some JVMs have memory management issues with direct buffers. **Default:** Checked
Worker Thread Pool	N	Y	Y	The name of a Thread Pool resource containing the threads used to handle the HTTP request. **Note:** When undefined, a thread pool with Max Pool Size set to 250 is created. **Default:** None

Identity Provider

The Identity Provider resource template provides access to a user name and password credential stored in a keystore.

General

Field	Required?	Editable?	Accepts SVars?	Description
Keystore Provider to Supply Identity	Y	Y	Y	Name of a Keystore Provider resource that maintains a keystore used to assert an identity.
Enable Access to Credential Store Containing Identity (optional)	N	N	N	Enables access to an identity keystore. To establish SSL connections, certain third-party systems, such as MySQL, require access to a keystore file location. In such situations, the Administrator provides a copy of credentials in a keystore, which are then written to the disk and used by the third party as the SSL credential store. To prevent the Administrator from providing credentials, clear the check box. **Default:** Checked
WSS Enable Protect Token	N	Y	N	Specifies whether to enable WSS security token or not. **Default:** Checked
Key Alias to Access Identity	Y	Y	Y	Name of the alias used to access the identity.
Key Alias Password	Y	Y	Y	Password for the alias. **Note:** If you try to update the existing encrypted password then the existing

Field	Required?	Editable?	Accepts SVars?	Description
				password will be removed. You can add a new password.
Max Pool Size	N	Y	Y	The maximum number of connections per connection identity that can be maintained concurrently. **Default:** 20

JDBC

The JDBC resource template represents a JDBC connection that is used by component implementations to access databases.

General

Field	Editable?	Accept SVars?	Description
Connection Type (Required)	N	N	The type of the JDBC connection: • **Direct**: The connection to the database is through a vendor-specific driver. When selected, the Database Driver and Database URL fields are displayed. • **XA**: The connection to the database is through a vendor-specific data source. When selected, the Data Source field is displayed. A component implementation that uses a JDBC connection of connection type XA typically executes within a global transaction and consequently may not explicitly commit transactions. To ensure that such implementations always behave correctly, the TIBCO ActiveMatrix platform detects when such a resource is used outside of a global transaction and enables the JDBC autocommit feature, so that all

Field	Editable?	Accept SVars?	Description
			database access by the component is committed. **Default Login Timeout:** 60000 ms (60s) **Default:** Direct

Direct

Field	Editable?	Accepts SVars?	Description
Database Driver (Required)	Y	Y	The name of the JDBC driver class. Select from a drop-down list of supported drivers or type the name of a custom driver: • `org.hsqldb.jdbcDriver` • `com.microsoft.sqlserver.jdbc.SQLServerDriver` • `com.mysql.jdbc.Driver` • `oracle.jdbc.OracleDriver` • `com.ibm.db2.jcc.DB2Driver` • `org.postgresql.Driver` Additional drivers available when using TIBCO Business Studio: • `com.ibm.as400.access.AS400JDBCDriver` • `com.informix.jdbc.lfxDriver` • `ca.edbc.jdbc.EdbcDriver` When you select a driver, the Database URL field is populated with a template for the URL of the driver. **Default:** `org.hsqldb.jdbcDriver`
Database	Y	Y	The URL to connect to the database.

Field	Editable?	Accepts SVars?	Description
URL (Required)			Format of the URL is supplied for the driver defined in the **Database Driver** field or you can type the name of a URL: - `jdbc:hsqldb:hsql://localhost:<port>/<db_instancename>` - `jdbc:sqlserver://<server Name>:<port>;databaseName=<dbname>;` - `jdbc:mysql://<localhost>:<port>/<DBName>` - `jdbc:oracle:thin:@<machine_name>:<port>:<instance_name>` - `jdbc:db2://<host>:<port default is 50000>/<database name>` - `jdbc:postgresql://<servername>:<port>/<dbname>` Available when using TIBCO Business Studio: - `jdbc:as400://server<server_ip>;libraries=<lib>` - `jdbc:informix-sqli://<host>:<port>/<database>:informixserver=<server>` - `jdbc:edbc://<host>:<port>/<database>` **Default:** `jdbc:hsqldb:hsql://localhost:<port>/<db_instance name>`

XA

Field	Editable?	Accept SVars?	Description
Data Source	Y	Y	The fully qualified name of the `javax.sql.XADataSourceimplementation` class.

Field	Editable?	Accept SVars?	Description
(Required)			The supported classes are: • com.ibm.db2.jcc.DB2XADataSource • com.mysql.jdbc.jdbc2.optional.MysqlXADataSource • oracle.jdbc.xa.client.OracleXADataSource • com.microsoft.sqlserver.jdbc.SQLServerXADataSource • org.postgresql.xa.PGXADataSource **Default:** oracle.jdbc.xa.client.OracleXADataSource

Field	Editable?	Accepts SVars?	Description
Maximum Connections (Optional)	Y	Y	The maximum number of database connections to allocate. The minimum value that can be specified is 0. **Default:** 10
Login Timeout (in ms) (Optional)	Y	Y	Time to wait for a successful database connection. If the JDBC driver does not support connection timeouts, the value of this field is ignored. Only JDBC drivers that support connection timeouts use this configuration field. Most JDBC drivers support connection timeouts. **Default:** 60000 (60 seconds)
Supports Transactions (Optional)	Y	Y	Indicates whether the application demarcates transaction boundaries. • If not selected, the application does not demarcate transaction boundaries and all SQL

Field	Editable?	Accepts SVars?	Description
			statements are autocommitted. - If selected, the application demarcates transaction boundaries. **Default:** Unchecked

Login Credentials

Field	Editable?	Accepts SVars?	Description
Login Credentials (Required)	Y	N	Indicates how the credentials required to authenticate to a server are provided: - **Identity Provider:** Provide user name and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. - **User name + Password:** Provide inline user name and password credentials. When selected, the Username and Password fields are activated. **Default:** Identity Provider
Identity Provider (Optional)	Y	Y	Name of the Identity Provider resource used to authenticate the user.
Username (Required)	Y	N	User name used to authenticate connections to the server.
Password (Required)	Y	N	User's password used to authenticate connections to the server. **Note:** If you try to update the existing encrypted password then the existing password will be removed. You can add a new password.

SSL

GUI	Editable?	Accepts SVars?	Description
Enable SSL (Optional)	N	N	Enables SSL connections. When selected, the SSL properties are displayed. **Default:** Unchecked
SSL Client Provider (Required)	N	Y	The name of an SSL Client Provider resource.

Advanced

GUI	Editable?	Accepts SVars?	Description
Host Type Properties (Optional)	Y	N	Properties to configure the connection between the JDBC resource and a specific type of host.
Commit Before Auto Commit (Optional)	Y	N	Indicates whether the driver requires a commit to be performed before enabling auto-commit on a connection. This must be (and is, by default) set to false for compliant drivers to avoid extraneous commits to the database. **Default:** false
Exception Sorter Class (Optional)	Y	N	The class used by the resource adapter to judge if an exception is fatal to the connection. That is, whether the connection pool should discard the connection from the pool, since it is no longer reusable. As the name implies, the default `SQLState08ExceptionsAreFatalSorter` treats SQL State 8 exceptions as fatal (connection errors). All other exceptions do not result in any connection pool action (but are passed up to the application for it to react). The

GUI	Editable?	Accepts SVars?	Description
			class must implement `org.tranql.connector.ExceptionSorter`. **Default:** `com.tibco.amf.sharedresource.runtime.tibcohost.jdbc.SQLState08ExceptionsAreFatalSorter`
POOL_MIN_SIZE (Optional)	Y	N	Minimum number of connections in the pool. **Default:** 5
POOL_BLOCKING_TIMEOUT (in ms) (Optional)	Y	N	Time for which a requestor waits for a connection when the pool is at maximum. **Default:** 60000 ms
POOL_IDLE_TIMEOUT (in min) (Optional)	Y	N	Time after which idle connections are closed. **Default:** 5 min
Prepared Statement Cache Size (Optional)	Y	N	The size of the cache containing prepared statements. The size must correspond to the number of JDBC statements you expect your application to reuse. **Default:** 0 (that is, the cache is disabled)

Direct

Field	Editable?	Accepts SVars?	Description
Connection Properties (Optional)	Y	N	Properties to configure connections to a database driver. The properties are vendor specific.

XA

Field	Editable?	Accept SVars?	Description
Connection Properties (Optional)	Y	N	Properties to configure connections to a data source. The properties are vendor specific.

JMS Resource Templates

The JMS resource templates enable applications to access objects maintained in JMS servers.

The JMS resource templates are:

- JNDI Connection Configuration — Provides a JNDI connection to look up a JMS server.
- JMS Connection Factory — Used to create an outbound connection to a JMS server.
- JMS Destination — Used for Request/Reply messages.

  JMS Destination specifies destination objects, which represent virtual channels (topics and queues) in JMS. When a message is sent, it is addressed to a destination, not to a specific application. Any application that subscribes to that destination can receive that message. Depending on the JMS messaging model used, the destination is called a topic or a queue. In the publish-subscribe model, a message is published for many subscribers to a topic (destination). In the point-to-point model, one message is sent to one potential receiver using a queue (destination).

JMS Resource Template Relationships

The JMS resource templates are used in different combinations to accomplish the tasks involved in setting up JMS enterprise messaging:

- Identifying the JMS server to connect to.
- Establishing request communication.
- Establishing reply communication.

Identifying the JMS server is accomplished through the JNDI Connection Configuration resource template. All the other JMS resource templates contain a link for the JNDI Connection that assists them in determining which JMS server to look up.

Additionally, before the connection to the JNDI server is made, the JNDI might require authentication. Authentication can take the form of a user name and password, or supplying credential information stored in a keystore using an identity provider. If the JNDI server is SSL-enabled, you must provide the required SSL configuration.

To establish request or reply communication, you need these resource templates: JMS Connection Factory, JMS Destination, and JNDI Connection Configuration.

Note: If direct destinations are used, only the JMS Connection Factory resource template is needed.

JMS Connection Factory

A JMS Connection Factory creates an outbound connection to a JMS server.

General

Field	Editable?	Required?	Accepts SVars?	Description
Connection Factory JNDI Name	Y	Y	Y	JNDI name of the JMS Connection Factory that points to a particular queue or topic.
Maximum Pool Size	Y	N	Y	(Optional) This property is available when creating a new JMS Connection Factory resource template. It can also be updated for an existing JMS Connection Factory resource template. **Default:** 20
JNDI Connection Configuration	Y	Y	Y	The name of a JNDI Connection Configuration resource. **Note:** You can use a substitution variable for JNDI connection

Field	Editable?	Required?	Accepts SVars?	Description
				configuration in the JMS Connection Factory resource template in the Application Configurator.

Security

Field	Editable?	Required?	Accepts SVars?	Description
Enable Authentication	Y	Y	N	Enables server authentication. When selected, the authentication properties (Login Credentials, Username, and Password) are displayed. **Default:** Unchecked
Login Credentials	Y	Y	N	Indicates how the credentials required to authenticate to a server are provided: • None • User name + Password - Provide inline user name and password credentials. When selected, the Username and Password fields are activated. • Identity Provider - Provide user name and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. **Default:** None
Username	Y	Y	N	User name used to authenticate connections to the server.

Field	Editable?	Required?	Accepts SVars?	Description
Password	Y	Y	N	User's password used to authenticate connections to the server. Default: None **Note:** If you try to update the existing encrypted password then the existing password will be removed. You can add a new password.
Identity Provider	Y	N	N	Name of the Identity Provider resource used to authenticate the user. Default: None
Enable SSL	Y	Y	N	Enables SSL connections. When selected, the SSL properties are displayed. **Default:** Unchecked
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource. Default: None

JMS Destination

A JMS Destination resource template specifies destination objects, which represent virtual channels (topics and queues) in JMS. It is used for Request/Reply messages.

When a message is sent, it is addressed to a destination, not to a specific application. Any application that subscribes or registers an interest in that destination can receive that message. Depending on the JMS messaging model used, the destination is called a topic or a queue. In the publish-subscribe model, a message is published for many subscribers to a topic (destination). In the point-to-point model, one message is sent to one potential receiver using a queue (destination).

Field	Editable?	Required?	Accepts SVars?	Description
Destination JNDI Name	Y	Y	Y	A JNDI name of a JMS destination that points to a particular queue or topic.
JNDI Connection Configuration	Y	Y	Y	The name of a JNDI Connection Configuration.

JNDI Connection Configuration

A JNDI Connection Configuration resource template provides a JNDI connection to look up a JMS server.

General

Field	Editable?	Required?	Accepts SVars?	Description
JNDI Provider	Y	Y	N	The provider to use for JNDI lookup: - TIBCO EMS - Progress SonicMQ - IBM MQ - Custom - Used for custom JNDI providers. The Initial Context Factory field is populated based on the JNDI provider selected. SSL lookup is only available for the TIBCO EMS provider. **Default:** TIBCO EMS
Initial Context Factor	Y	Y	Y	Initial context factory to be used for the JNDI lookup. The value for Initial Context Factory is set based on the JNDI provider selected: - TIBCO EMS - The following value is populated:

Field	Editable?	Required?	Accepts SVars?	Description

y				`com.tibco.tibjms.naming.TibjmsInitialContextFactory`
				- Progress SonicMQ - The following value is populated:
				`com.sonicsw.jndi.mfcontext.MFContextFactory`
				- IBM MQ -
				◦ `com.sun.jndi.ldap.LdapCtxFactory` for the JNDI lookup in LDAP. Pair this value with the Naming Provider URL: `ldap://<ldap_url>`
				◦ `com.sun.jndi.fscontext.RefFSContextFactory` for the JNDI lookup in a file system. Pair this value with the Naming Provider URL: `file:<url_ of_ bindings_file>`
				- Custom provider - Specify the custom initial context factory value.
				Default:
				`com.tibco.tibjms.naming.TibjmsInitialContextFactory`

Provider URL	Y	Y	Y	Provider URL of the JNDI server. The value for Naming Provider URL is set based on the JNDI provider selected: - TIBCO EMS - `tibjmsnaming://<host>:<port>` - Progress SonicMQ - `tcp://<host>:<port>` - IBM MQ - ◦ `ldap://<ldap_url>` for the JNDI lookup

Field	Editable?	Required?	Accepts SVars?	Description
				in LDAP. **Example:** `ldap://mymachine.tibco.com:2076/dc=tibco,dc=com` Pair this value with the Initial Context Factory: `com.sun.jndi.ldap.LdapCtxFactory`. ○ `file:<url_ of_bindings_file>` for the JNDI lookup in a file system. **Example:** `file: /D:/Program Files/IBM/fileBinding` Pair this value with the Initial Context Factory: `com.sun.jndi.fscontext.RefFSContextFactory` • Custom - specify the custom provider URL. **Default:** `tibjmsnaming://<host>:<port>` The Naming Provider URL is validated using recommendation of the "Uniform Resource Identifiers (URI): Generic Syntax" [RFC2396] standard for the TIBCO EMS, Progress SonicMQ and IBM MQ JNDI provider.

Security

Field	Required?	Editable?	Accepts SVars?	Description
Enable Authentication	N	Y	N	Enables server authentication. When selected, the authentication properties (Login Credentials, Username, and

Field	Required?	Editable?	Accepts SVars?	Description
				Password) are displayed. **Default:** Unchecked
Login Credentials	Y	Y	N	Indicates how the credentials required to authenticate to a server are provided: • None • Username + Password - Provide inline user name and password credentials. When selected, the Username and Password fields are activated. • Identity Provider - Provide user name and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. **Default:** Username + Password
Username	N	Y	N	User name used to authenticate connections to the server. Default: None
Password	N	Y	N	User's password used to authenticate connections to the server. **Note:** If you try to update the existing encrypted password then the existing password will be removed. You can add a new password. Default: None

Field	Required?	Editable?	Accepts SVars?	Description
Identity Provider	N	Y	N	Name of the Identity Provider resource used to authenticate the user. Default: None
Enable SSL	Y	Y	N	Enables SSL connections. When selected, the SSL properties are displayed. **Default:** Unchecked
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource. Default: None

Application Properties: A list of properties used for JNDI lookup.

Field	Description
Name	Name of the property. Default: None
Type	Type of the property. Default: String The type is one of: string, boolean, byte, short, char, int, long, float, or double.
Value	Property value. **Default:** Depends on value of Type. You can set a property value to a literal or a substitution variable.

Keystore Provider

The Keystore Provider resource template provides access to a keystore.

General

Field	Required?	Editable?	Accepts SVars?	Description
Keystore Location	Y	Y	Y	Location of the keystore. • Upload the keystore file from your local machine • External URL of the keystore. **Note:** If the uploaded application DAA contains already configured keystore resource template and the location of the keystore is from your local machine then you must again upload the keystore file in the Application Configurator.
Password	Y	Y	Y	Password for the keystore. **Note:** If you try to update the existing encrypted password then the existing password will be removed. You can add a new password.
Provider	N	Y	Y	Name of the keystore provider: • IBMJCE (IBM JREs) • SUN (JKS format) **Default:** Empty. The first matching provider supporting the format will be chosen.
Type	Y	Y	Y	**Type of the keystore:** JCEKS, JKS, PKCS12 **Default:** JCEKS
Refresh Interval (ms)	Y	Y	Y	Refresh interval, greater than 0. If the keystore provider is accessed after the refresh interval has expired:

Field	Required?	Editable?	Accepts SVars?	Description
				1. The keystore provider is refreshed from its backing keystore. 2. The refresh timer is reset to zero. 3. Operations on the keystore provider are performed on the refreshed copy. **Default:** 360000 ms
Max Pool Size	N	Y	Y	Specifies the maximum number of connections in the connection pool. **Default:** 20

Keystores

If you set up your environment for SSL, you have to set up a keystore. As part of the process, you configure a keystore provider.

SSL uses keys and certificates when it establishes the secure connection. A keystore is a database of keys and certificates. A keystore password is required to access or modify the keystore.

Keystore Entries

A keystore has two types of entries:

- Private key - holds a cryptographic private key, which is optionally stored in a protected format to prevent unauthorized access. The private key is accompanied by a certificate chain for the corresponding public key. Private keys and certificate chains are used by a given entity for self-authentication.
- Trusted certificate - contains a single public key certificate. It is called a trusted certificate because the keystore owner trusts that the public key in the certificate belongs to the identity identified by the subject (owner) of the certificate. This type of entry is used to authenticate other parties.

Certificates of trusted entities are typically imported into a keystore as trusted certificates.

Keystore Entries and Aliases

Each entry in a keystore is identified by an alias. In the case of private keys and their associated certificate chains, these aliases are the different ways in which the entity may authenticate itself. For example, the entity may authenticate itself using different certificate authorities, or using different public key algorithms. An alias might be named after the role in which the keystore owner uses the associated key, or might identify the purpose of the key.

Keystore Passwords and Private Key Passwords

The private keys in a keystore are encrypted with a keystore password, which should be several words long.

You can also protect each private key with its individual password, which may or may not be the same as the keystore password.

⚠ **Warning:** If a password is lost, the associated keys cannot be recovered.

LDAP Authentication

The LDAP Authentication resource template represents an LDAP server providing authentication services.

LDAP authentication is done in one of the following ways:

- Bind mode — The bind mode authenticates (binds) each user's Distinguished Name (DN) and password to the LDAP server. In this case, you can use the DN Template field so that users do not have to provide their whole DN.

  For example, a DN Template of uid={0}, OU=Department, DC=company, and DC=com allows users to type in only their uid and the resource will use the template to create the DN.

- Search mode — In the search mode, a connection binds as the administrative user. It then searches for the given users and authenticates their found DNs and passwords with the LDAP server. In this case, you must provide the credentials of such an administrative user by selecting **Log in as Administrator**.

General

Field	Required?	Editable?	Accepts SVars?	Description

Server URLs	Y	Y	Y	A space-separated list of URLs for an LDAP server. To achieve fault tolerance, you can specify URLs.
				For example:
				`ldap://server1.example.com:686`
				`ldap://server2.example.com:1686`
				Default: `ldap://localhost:389`

User Attribute with User Name	N	Y	Y	The name of the LDAP attribute from which the user display name can be obtained. Always specify an Attribute Name even though this field is labeled optional. You must use an attribute that is part of the LDAP schema. Otherwise, any attribute not defined by the schema can result in an error. **Default:** None
Search Entire Subtree Starting at Base DN	N	N	N	Determines whether the authentication should search sub-branches of the LDAP directory. Always select Yes. **Default:** Checked
Log in as Administrator	Y	N	N	If you select the check box **Log in as Administrator**, you must provide the DN of the administrative user to connect to the LDAP server. If selected, the following fields are displayed:

Field	Required?	Editable?	Accepts SVars?	Description
				- User Search Base DN - Login Type with Username + Password option shown - Username - Password **Default:** Unchecked
User DN Template	Y	Y	Y	The template by which the User DN used to bind to the LDAP server is generated. Because the full DN is always supplied, the template should always contain {0} which gets replaced with the actual user name. **Default:** {0}
User Search Base DN	Y	Y	Y	Base distinguished name from which the search starts. Example: ou=department, dc=company, dc=com.
User Search Expression	N	Y	Y	The expression used for searching a user. An example for this expression is (CN={0}). '{0}' is replaced by the username being searched for. You can define any complex filter like (&(cn={0})(objectClass=account)). **Default:** &(objectClass=person)(uid={0})
Login Credentials	Y	Y	N	Method to identify the administrative user: - Username + Password - Activates

Field	Required?	Editable?	Accepts SVars?	Description
				the Username and Password fields. - Identity Provider that Supplies Credentials - Activates the Identity Provider field. **Default:** Username + Password
Username	Y	Y	N	Full Distinguished Name (DN) of an administrative user in the LDAP server.
Password	Y	Y	N	Password for the user. **Note:** If you try to update the existing encrypted password then the existing password will be removed. You can add a new password.
Identity Provider	Y	Y	Y	The name of an Identity Provider.
Keystore Provider to Supply Identity	Y	Y	Y	Name of a Keystore Provider resource that maintains a keystore used to assert an identity.
Max Pool Size	N	Y	Y	The maximum number of connections per connection identity that can be maintained concurrently. **Default:** 20

Group Attributes

Field	Required?	Editable?	Accepts SVars?	Description
Group Indication	N	Y	N	Specifies how a user's group memberships are found. Group information is used by

Field	Required?	Editable?	Accepts SVars?	Description
				Administrator when a user, once authenticated, performs other activities in the system. Options: • Group has users: A list of users that belong to the group. • User has groups: A list of groups to which the user belongs. • User DN has groups: The DN with a list of groups to which the user belongs. • No Group Info: Group memberships are not handled. If the selected value is User has groups or User DN has groups, the Users Attribute with Group Names field is displayed. If the selected value is Group has users, the following fields are displayed: • Group Search Base DN • Group Search Expression • Group Attribute with User Names • Group Attribute with Group Name • Group Attribute with Subgroup Names • Group Search Scope Subtree **Default:** No Group Info
User Attribute	Y	Y	Y	The name of the attribute in each user object that lists the groups to which the

Field	Required?	Editable?	Accepts SVars?	Description
with Group Names				user belongs. **Default:** None
Group Search Base DN	N	Y	Y	Searches for groups beginning at this base distinguished name (DN). **Default:** None
Group Search Expression	Y	Y	Y	Searches by matching this expression against potential groups. **Default:** None
Group Attribute with User Names	Y	Y	Y	The name of the attribute in the group object that contains its users. For example: For OpenLDAP: `uniqueMember` For ActiveDirectory: `member` **Default:** None
Group Attribute with Group Name	Y	Y	Y	The name of the attribute in the group object that contains the name of the group. For example: For OpenLDAP: `cn` For ActiveDirectory: `AccountName` **Default:** None
Group Attribute with Subgroup	N	Y	Y	The name of the attribute in the group object that contains its subgroups. For example:

Field	Required?	Editable?	Accepts SVars?	Description
Names				For OpenLDAP: `uniqueMember` For ActiveDirectory: `member` **Default:** None
Group Search Scope Subtree	N	N	N	Search the entire subtree starting at the base DN for groups (default). Otherwise, search only the nodes one level below the base DN. **Default:** Checked

SAML Options

SAML assertions are accessed from a security context and can be propagated between components to achieve single sign-on.

Field	Required?	Editable?	Accepts SVars?	Description
Validity of SAML Tokens (s)	N	Y	Y	The duration of the validity of the SAML tokens. **Default:** 600 s
Signer of SAML Tokens	N	Y	Y	The name of an Identity Provider resource that identifies the signer of the SAML tokens.

Advanced

Field	Required?	Editable?	Accepts SVars?	Description
Context Factory	N	Y	Y	The factory object that provides the starting point for resolution of names within the LDAP server.

Field	Required?	Editable?	Accepts SVars?	Description
				Default: `com.sun.jndi.ldap.LdapCtxFactory`
Maximum Connections (disabled in non-Admin mode)	N	Y	Y	The maximum number of connections to keep active in the pool. (Enabled only when **Log in as Administrator** is selected in the General tab) **Default:** 10
Security Authentication	N	Y	Y	Value of Simple Authentication and Security Layer (SASL) authentication protocol to use. Values are implementation-dependent. Some possible values are simple, none, and md-5. **Default:** Blank
Search Timeout (ms)	N	Y	Y	The time to wait for a response from the LDAP directory server. **Default:** -1, which means to wait forever.
Follow Referrals	N	Y	N	Indicates whether the client should follow referrals returned by the LDAP server. **Default:** Unchecked
User Attributes Extra	N	Y	Y	Optional list of user attributes to retrieve from the LDAP directory during authentication. **Default:** None

SSL

Field	Required?	Editable?	Accepts SVars?	Description
Enable SSL	Y	Y	N	Enables SSL connections. When selected, the SSL properties are displayed. **Default:** Unchecked
SSL Client Provider	Y	Y	Y	The name of an SSL Client Provider resource. Default: None

LDAP Connection

An LDAP connection resource template represents a connection to an LDAP server. It is used by component implementations to look up names in an LDAP directory server.

General

Field	Required?	Editable?	Accepts SVars?	Description
Connection Factory	Y	Y	Y	The factory object that provides the starting point for the resolution of names within the LDAP server. **Default:** `com.sun.jndi.ldap.LdapCtxFactory`
Provider URL	Y	Y	Y	This URL contains the host and port number on which the LDAP server is listening for connections. It can also include a Base DN, the DN of an entry in the directory.

Field	Required?	Editable?	Accepts SVars?	Description
				Default: `ldap://localhost:389`
Connection Timeout (ms)	N	Y	Y	The time to wait for a response from the LDAP directory server. **Default:** 0
Login Credentials	Y	Y	N	Indicates how the credentials required to authenticate to a server are provided: • Identity Provider - Provide user name and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. • Username + Password - Provide inline user name and password credentials. When selected, the Username and Password fields are activated. **Default:** Username + Password
Identity Provider	Y	Y	Y	Name of the Identity Provider resource used to authenticate the user.
Username	Y	Y	N	User name used to authenticate connections to the server.
Password	Y	Y	N	User's password used to authenticate connections to the server. **Note:** If you try to update the existing encrypted password then the existing password will be removed. You can add a new password.

SSL

Field	Required?	Editable?	Accepts SVars?	Description
Enable SSL	Y	Y	N	Enables SSL connections. When selected, the SSL properties are displayed. **Default:** Unchecked
SSL Client Provider	Y	Y	Y	The name of an SSL Client Provider resource.

Advanced

Field	Required?	Editable?	Accept SVars?	Description
Pool Size	N	Y	Y	The preferred number of connections per connection identity that must be maintained concurrently. **Default:** 10
Pool Maximum	N	Y	Y	The maximum number of connections per connection identity that can be maintained concurrently. **Default:** 15
Pool Initial	N	Y	Y	The number of connections per connection identity to create when initially creating a connection for the identity. **Default:** 5
Pool Timeout (ms)	N	Y	Y	The time that an idle connection may remain in the pool without being closed and removed from the pool. **Default:** 300000

Field	Required?	Editable?	Accept SVars?	Description
Follow Referrals	N	N	Y	Indicates whether an LDAP server should return a reference (a referral) to another LDAP server which may contain further information instead of returning a result. **Default:** Unchecked

Mutual Identity Provider

The Mutual Identity Provider resource template provides access to a user name and password credential stored in a keystore.

General

Field	Required?	Editable?	Accepts SVars?	Description
Keystore Provider to Supply Identity	Y	Y	Y	Name of a Keystore Provider resource that maintains a keystore used to assert an identity. Default: None
Enable Access to Credential Store Containing Identity (optional)	N	N	N	Enables access to an identity keystore. To establish SSL connections, certain third-party systems, such as MySQL, require access to the keystore file location. **Default:** Checked
Key Alias to Access Identity	Y	Y	Y	Name of the alias used to access the identity.
Key Alias Password	Y	Y	Y	Password for the alias.

Field	Required?	Editable?	Accepts SVars?	Description
				Note: If you try to update the existing encrypted password then the existing password will be removed. You can add a new password.
Max Pool Size	N	Y	Y	The maximum number of connections per connection identity that can be maintained concurrently. **Default:** 20

Open ID Authentication

You can configure OpenID Authentication policy by using Open ID Authentication resource template.

Field/Button	Description
Description (optional)	A description for the OpenID resource.
Access token URI	The REST OpenID token service URI, which is used to obtain an ID Token for the authenticated user. **Note:** Using the OpenID Access Token is not currently supported in ActiveMatrix Service Grid - Container Edition. The OpenID ID Token is used to identify the user. This is unique to the IdP and can be obtained from the IdP's website on which they describe how to register an application with the IdP. **Examples:** Google: `https://www.googleapis.com/oauth2/v3/token`

Field/Button	Description
	Microsoft AD FS: https://*host*:*port*/adfs/oauth2/token Here, - *host* is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, - *port* is the port used by the application.
Client ID	The ID that identifies the client at the Identify Provider (IdP). This and the Client Secret (see below) are obtained from the IdP when the client registers an application with the IdP for the purpose of providing authentication for users.
Client Secret	The password for the Client ID account. See the description above.
Redirect URI	The URI to which the IdP redirects the user after authenticating the user and generating an ID Token. http://*host*:*port*/*appPath* Here, - *host* is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, - *port* is the port used by the application, and - *appPath* is the path to your Web application. **Note:** This URI must exactly match the Redirect URI that was specified when registering the application with the IdP.
Authorization URI	The REST Open ID user claims/information service URI, which is used to obtain user profile information. This URI can be obtained from the IdP's website on which they describe how to register an application with the IdP.

Field/Button	Description
	Examples: Google: `https://accounts.google.com/o/oauth2/auth` Microsoft AD FS: `https://`*`host`*`:`*`port`*`/adfs/oauth2/authorize` Here, - *host* is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, - *port* is the port used by the application.
Auth Scope (optional)	Defines the claims to be returned by the IdP when the IdP authenticates the user and issues anID Token. These claims are user attributes and are intended to provide the application with user details. The openid scope is included by default (even though it does not appear in the **Auth Scope** field by default). (The openid scope causes the sub claim to be returned, which uniquely identifies the user.) However, if any scope is entered in the **Auth Scope** field, it overrides the default value of openid. Therefore, if your IdP requires the openid scope, plus another scope, you must also specify openid. Specify the scopes required by your IdP. **Examples:** Google and Microsoft AD FS: openid, email Multiple scopes can be comma- or space-separated in the **Auth Scope** field.
User Key (optional)	From the list of claims that are returned from the IdP (based on the scope), this specifies the field that is used as a User ID. **For example:** Google and Microsoft AD FS: email
JSON Web Key Set URI	The URI to the JSON Web Key Set (JWKS), which is a JSON data structure that represents a set of public keys used to verify the signature of the JSON Web Token (JWT)ID issued by the IdP.

Field/Button	Description
	This is unique to the IdP and can be obtained from the IdP's website on which they describe how to register an application with the IdP. **Examples:** Google: `https://www.googleapis.com/oauth2/v3/certs` Microsoft AD FS: `https://`*`host`*`:`*`port`*`/adfs/discovery/keys` Here, • *host* is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, • *port* is the port used by the application.
Logout Path	When a user logs out of a TIBCO ActiveMatrix Web application, the browser sends this value to the TIBCO ActiveMatrix server. This property must be set to: `/logout` This value indicates to the server that it needs to send a request to the IdP to log the user out, using the value specified in the SignOutURL property (see below).
Signout URL	Upon receiving `/logout` in the LogOutPathproperty, the server uses this URL to send the IdP a request to log the user out of the IdP. The Signout URL is specific to the IdP. **Examples:** Google: `https://www.google.com/accounts/Logout?continue=https://appengine.google.com/_ah/logout?continue=http://`*`host`*`:`*`port`*`/`*`appPath`*`/logout` Here, • *host* is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, • *port* is the port used by the application, and • *appPath* is the path to the application's landing page.

Field/Button	Description
	Microsoft AD FS: `https://`*`host`*`:`*`port`*`/adfs/ls/?wa=wsignout1.0&wreply=http://`*`host`*`:`*`port`*`/`*`appPath`* Here, - *host* is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, - *port* is the port used by the application, and - *appPath* is the path to the application's landing page.
Unauthorized Redirect Requests (optional)	Specifies whether it is the responsibility of the application to handle unauthorized redirect requests. Select this option if you want the application to handle unauthorized requests and forward them to the appropriate location. For example, TIBCO ActiveMatrix BPM handles unauthorized requests. Therefore, for TIBCO ActiveMatrix BPM applications, this option must be selected.

SAML SSO Web Profile Authentication

The SAML SSO Web Profile Authentication resource template provides configuration fields for SAML SSO Web Profile Authentication.

General

Field	Description
Entity Id (Required)	Unique identifier for the service provider. This must be the same as that configured in the IdP. Example: `https://host:port/saml/saml/metadata`
Authentication Successful URL (Required)	URL for authentication successful landing page. Example: `/landing`

Field	Description
IDP Metadata Source (Required)	SAML metadata describes service provider or identity provider. Select one of the options from following: • IDP HTTP Metadata URL. • IDP String Metadata: Select this option if you have metadata source file present at local computer (For Google IdP).
IDP Metadata URL (Required)	Location of IdP metadata source file (if IDP String Metadata option is selected) or HTTP URL of IdP metadata (if IDP HTTP Metadata URL option is selected). Example: Google: `D:\SAML\GoogleIDPMetadata.xml` AD FS: `https://idp-alias/Metadata.xml`
IDP Login URL (Required)	URL to initiate SAML login. Example: `/login`
IDP Logout URL (Required)	URL to initiate SAML logout. Example: `/logout`
IDP SSO URL (Required)	URL where SAML assertions are posted back by IdP. Example: `/SSO`
IDP Single Logout URL (Required)	URL where logout response is sent back by IdP. Example: `/SingleLogout`
Logout Successful URL (Required)	URL for logout successful landing page. Example: `/loggedOut`
Authentication Failure URL	URL for authentication failure landing page.

Field	Description
(Required)	Example: `/error`
Response Skew Time (seconds) (Required)	Duration for which response from IdP is valid. Example: 60 seconds
Unauthorize Redirect Requests (Optional)	For TIBCO ActiveMatrix BPM applications, this check box must be selected. **Default:** Unchecked, for SOA applications
Max Authentication Age (seconds) (Optional)	You can configure this field to ensure that the existing SAML assertion returned by the IdP is not older than the value specified in this field. **Default:** 7200 seconds
Local Logout (Optional)	Select the check box if you are using Google IdP.

Advanced Tab

You can sign or encrypt SAML requests and responses for advanced security. The **Advanced** tab provides configuration fields for signing or encrypting SAML requests and responses. You must provide valid public key or certificate to the IdP so that it can identify signed requests. For more information about keystore, see Keystores.

Field	Description
Keystore Provider (Required)	The name of a Keystore Provider resource.
Sign Authentication Request (Optional)	If you select this check box, authentication request by service provider must be signed. You must provide valid public key or certificate to the IdP so that it can identify signed requests.
Sign Logout Request (Optional)	Select the check box to sign logout request.

Field	Description
Sign Logout Response (Optional)	If you select this check box, the IdP must sign the logout response before returning it to the service provider.
Sign Assertions (Optional)	Select the check box to sign SAML assertions.
Sign Metadata (Optional)	Select the check box to sign SAML metadata.
Encrypt Assertion (Optional)	Select the check box to encrypt SAML assertion.
Key Alias to Encrypt and Key Alias Password (Optional)	Name of the key alias used for encryption and password for the alias.
Key Alias to Sign and Key Alias Password (Optional)	Name of the key alias used to sign and password for the alias.
Default Key Alias and Key Alias Password (Required)	Name of the default key alias and password for the alias.
Use Load Balancer	Select the check box if you are using the Load Balancer for an application.
Entity Base URL	This is the URL where the IdP will send and receive SAML requests and responses.
Scheme (http/https)	Name of the scheme (http or https).
Server Name	Name of the server.
Server Port	Port number of the server.
Include Server Port	Select the check box to include server port number in the URL.

Field	Description
in Request URL	
Context Path	The prefix of a URL path that is used to select the contexts to which an incoming request is passed. For example, the path is displayed as `http://hostname.com/contextPath/`.

Copying ADFS Encryption Certificate to Application Docker Image

If you want to use OpenID or SAML SSO with ADFS, you must have cacert containing ADFS encryption certificate for authentication purpose.

Procedure

1. Copy default cacert from container to the host machine.
2. Import ADFS encryption certificates to cacert by using Keystore Explorer or any other tool.
3. Save the certificate in the `certs` folder inside the application folder that you pass to `--app_location` argument when building the Docker image.
4. Create an application Docker image.

SMTP

An SMTP resource template represents a connection to an SMTP server. Component implementations use the SMTP resource template to send and receive messages to and from an SMTP mail server.

General

Field	Required?	Editable?	Accepts SVars?	Description
Machine Name	Y	Y	Y	The name of the host that accepts incoming requests. **Default:** localhost

Field	**Required?**	**Editable?**	**Accepts SVars?**	**Description**
Port	Y	Y	Y	The port number on which to listen for SMTP requests. **Default:** 25
Timeout (ms)	N	Y	Y	The time to wait for a response from the server. The timeout must be greater than 0. **Default**: 0 (that is, infinite timeout)
Login Credentials	Y	Y	Y	Indicates how the credentials required to authenticate to a server are provided: • **Identity Provider** - Provide user name and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. • **Username + Password** - Provide inline user name and password credentials. When selected, the Username and Password fields are activated. **Default:** Identity Provider
Identity Provider	N	Y	N	Name of the Identity Provider resource to authenticate the user.
Username	N	Y	N	User name to authenticate connections to the server.
Password	N	Y	N	User's password to authenticate connections to the server. **Note:** If you try to update the existing encrypted password then the existing password will be removed. You can add a new password.

SSL

Field	Required?	Editable?	Accepts SVars?	Description
Enable SSL	Y	Y	N	Enables SSL connections. When selected, the SSL properties are displayed. **Default:** Unchecked
SSL Client Provider	Y	Y	Y	The name of an SSL Client Provider resource.

SSL Client Provider

The SSL Client Provider resource template maintains the credentials required by an SSL client.

General

Field	Required?	Editable?	Accepts SVars?	Description
Keystore Provider as Trust Store	Y	Y	Y	The name of a Keystore Provider resource that maintains the keystore that confirms an identity.
Enable Access to Trust Store	N	N	N	Enables access to a trust credential store. To establish SSL connections, certain third-party systems, such as MySQL, require access to a keystore file location. In such situations, the Administrator provides a copy of the credentials in a keystore. The credentials are written to the disk and used by the

Field	Required?	Editable?	Accepts SVars?	Description
				third party as the SSL credential store. **Default:** Checked
Enable Mutual Authentication	N	Y	N	Indicates whether the client in the SSL connection authenticates with the server. When selected, the identity fields are enabled. **Default:** Unchecked
Keystore Provider Having Identity	Y	Y	Y	Name of Keystore Provider resource that maintains a keystore used to assert an identity.
Enable Access to Credential Store Containing Identity	N	N	N	Enables access to an identity keystore. To establish SSL connections, certain third-party systems, such as MySQL, require access to a keystore file location. In such situations, the Administrator provides a copy of credentials in a keystore which are written to the disk and used by the third party as the SSL credential store. To prevent the Administrator from providing credentials, clear the check box. **Default:** Unchecked
Key Alias To Access Identity	Y	Y	Y	Name of the alias used to access the identity. **Default:** None

Field	Required?	Editable?	Accepts SVars?	Description
Key Alias Password	Y	Y	Y	Password for the alias. **Note:** If you try to update the existing encrypted password then the existing password will be removed. You can add a new password.
Max Pool Size	N	Y	Y	Specifies the maximum number of connections in the connection pool. **Default:** 20

Advanced

Field	Required?	Editable?	Accepts SVars?	Description
SSL Security Provider	N	Y	Y	(Optional) The SSL security provider.
SSL Protocol	N	Y	N	The SSL protocol to use in the SSL connection: • SSLv3 • TLSv1 • TLSv1.1 • TLSv1.2 **Default:** TLSv1.2
SSL Cipher Class	N	Y	N	The number of bits in the key used to encrypt data: • No Exportable Ciphers

Field	Required?	Editable?	Accepts SVars?	Description
				• At Least 128 Bit • More Than 128 Bit • At Least 256 Bit • FIPS Ciphers • All Ciphers • Explicit Ciphers The greater the number of bits in the key (cipher strength), the more possible key combinations and the longer it takes to break the encryption. **Default:** At Least 128 Bit
Explicit Cipher List	N	Y	Y	A list of ciphers. Enabled when SSL Cipher Class is set to Explicit Ciphers. Use the JSSE format for ciphers names. **Default:** None
Verify Remote Hostname	N	N	N	Indicates whether the name on the server's certificate must be verified against the server's hostname. If the server's hostname is different than the name on the certificate, the SSL connection fails. The name on the certificate can be verified against another name by specifying Expected Remote Hostname. When selected, the Expected Remote Hostname field is enabled. **Default:** Unchecked
Expected Remote Hostname	N	Y	Y	The expected name of the remote host. **Default:** None

SSL Server Provider

An SSL Server Provider resource template maintains the credentials required by an SSL server.

General

Field	Required?	Editable?	Accepts SVars?	Description
Keystore Provider Having Identity	Y	Y	Y	Name of the Keystore Provider resource that maintains a keystore used to assert an identity.
Enable Access to Credential Store containing Identity	N	N	N	Enables access to an identity keystore. To establish SSL connections, certain third-party systems, such as MySQL, require access to a keystore file location. In such situations Administrator provides a copy of credentials in a keystore, which are then written to disk and used by the third party as the SSL credential store. To prevent Administrator from providing credentials, uncheck the checkbox. **Default:** Checked
Key Alias to Access Identity	Y	Y	Y	Name of the alias used to access the identity. **Default:** None
Key Alias Password	Y	Y	N	Password for the alias. **Note:** If you try to update the existing encrypted password then the existing password will be removed. You can add a new password.

Field	Required?	Editable?	Accepts SVars?	Description
Enable Mutual Authentication	N	Y	N	Indicates whether mutual authentication is enabled. When selected, the Make Client Authentication As, Keystore Provider as Trust Store, and Enable Access to Trust Store fields are displayed. **Default:** Unchecked
Make Client Authentication As	N	Y	N	Indicates whether it is optional or required for an SSL client to authenticate to the SSL server. **Default:** Optional
Keystore Provider as Trust Store	Y	Y	Y	The name of the Keystore Provider resource that maintains a keystore that confirms an identity.
Enable Access to Trust Store	N	N	N	Enables access to a trust credential store. To establish SSL connections, certain third-party systems, such as MySQL, require access to a keystore file location. In such situations, the Administrator provides a copy of the credentials in a keystore which are written to disk and used by the third party as the SSL credential store. **Default:** Checked
Maximum Pool Size	N	Y	Y	Specifies the maximum number of connections in the connection pool. Default: 20

Advanced

Field	Required?	Editable?	Accepts SVars?	Description
SSL Security Provider	N	Y	Y	The SSL security provider.
SSL Protocol	N	Y	N	The SSL protocol to use in the SSL connection: • SSLv3 • TLSv1 • TLSv1.1 • TLSv1.2 **Default:** TLSv1.2
SSL Cipher Class	N	Y	N	The number of bits in the key used to encrypt data: • No Exportable Ciphers • At Least 128 Bit • More Than 128 Bit • At Least 256 Bit • FIPS Ciphers • All Ciphers • Explicit Ciphers The greater the number of bits in the key (cipher strength), the more possible key combinations and the longer it takes to break the encryption. **Default:** At Least 128 Bit
Explicit	N	Y	Y	A list of ciphers. Enabled when SSL Cipher

Field	Required?	Editable?	Accepts SVars?	Description
Cipher List				Class is set to Explicit Ciphers. Use the JSSE format for ciphers names. **Default:** None
Verify Remote Hostname	N	N	N	Indicates whether the name on the server's certificate must be verified against the server's hostname. If the server's hostname is different than the name on the certificate, the SSL connection fails. The name on the certificate can be verified against another name by specifying Expected Remote Hostname. When checked, the Expected Remote Hostname field is enabled. **Default:** Unchecked
Expected Remote Hostname	N	Y	Y	(Optional) The expected name of the remote host. **Default:** None

Thread Pool

A thread pool is a queue of threads available to run a queue of tasks. Thread pools are used to improve performance when executing large numbers of asynchronous tasks by reducing per-task invocation overhead and provide a means of bounding and managing the resources consumed when executing a collection of tasks.

A thread pool is created with zero threads.

General

Field	Required?	Editable?	Accepts SVars?	Description
Note: If no value is specified for Core pool size, Max Pool Size, and, Keep Alive Time fields, then the default values are used.				
Core Pool Size	N	Y	Y	When a new task is submitted and fewer than Core Pool Size threads are running, a new thread is created to handle the request, even if other threads are idle. If there are greater than Core Pool Size but fewer than Max Pool Size threads running, a new thread is created only if no threads are idle. Must be greater than or equal to zero. **Note:** When a Java or Spring component service is configured with a threading policy with a non-zero timeout value and is promoted to a composite service using a SOAP or JMS binding, concurrency is halved because two threads are used per request. To achieve the desired concurrency, specify double the number of threads for the thread pool size. **Default:** 2 (that is, two threads are used to service one request: one for receiving the request and one for receiving the response)
Max Pool Size	N	Y	Y	The maximum number of threads in the pool. Must be greater than zero and greater than or equal to Core Pool Size. **Default:** 20
Keep Alive	N	Y	Y	The time for which an idle thread remains in

Field	**Required?**	**Editable?**	**Accepts SVars?**	**Description**
Time (s)				the pool before being reclaimed if the number of threads in pool is more than Core Pool Size. **Default:** 30 Seconds
Autostart Core Threads	N	N	N	Indicates that the Core Pool Size threads must be created and started when the thread pool is created. Normally core threads are created and started only when new tasks arrive. **Default:** false
Thread Pool Name Prefix	N	Y	Y	A string prepended to the name of each thread. **Default:** `<pool-poolnumber-thread-threadnumber>`
Priority	Y	N	N	The default priority of the threads in the pool. **Default:** 5
Rejection Policy	Y	N	N	The policy applied when no thread is available to run a task: • Abort - The task is aborted and an exception is thrown. • Blocking - The task is blocked until a thread from thread pool picks up this task. • Caller Runs - The task is run in the calling thread. **Default:** Blocking

Field	Required?	Editable?	Accepts SVars?	Description
Daemon	N	N	N	Indicates whether the threads can be started as daemon or user. **Default:** Unchecked

WSS Authentication

A Web Services Security (WSS) Authentication resource template resource template enables a connection to Web Services Security authentication services.

General

Property	Editable?	Required?	Accepts SVars?	Description
Security Token	Y	Y	N	Security Token is an online security credential that adds an extra layer of identity protection. Default: X.509
X.509	N	N	N	X.509 is an ITU-T standard for a public key infrastructure (PKI) and Privilege Management Infrastructure (PMI).
Enable Signature Verification	N	Y	N	Indicates whether to verify the signatures. If selected, the Trust Provider field is activated. **Default:** Unchecked **Note:** Enabling signature verification requires Trust Provider resource template to be configured. Trust Provider resource template is not supported in ActiveMatrix Service Grid - Container Edition 1.0.0. This field can only be selected when the

Property	Editable?	Required?	Accepts SVars?	Description
				Enable Decryption field is selected, and will activate the **Mutual Identity Provider** field.
Enable Decryption	N	Y	N	Indicates whether to enable decryption. If selected, the Identity Provider field is activated. **Default:** Unchecked
Identity Provider	Y	N	Y	Name of the Identity Provider resource that provides the credential used to decrypt messages. Activated if **Enable Decryption** field is selected.
Mutual Identity Provider	Y	N	Y	Name of a Mutual Identity Provider resource. Activated if the **Enable Decryption** and **Enable Signature Verification** fields are selected.
Maximum Pool Size	N	Y	Y	Specifies the maximum number of connections in the connection pool. Default: 20

Username Authentication

Property	Editable?	Required?	Accepts SVars?	Description
Enable Username authentication	N	N	N	Indicates whether to verify the user name. If selected, the Authentication Provider field is activated. **Default:** Unchecked

Property	Editable?	Required?	Accepts SVars?	Description
Authentication Provider	Y	Y	Y	Name of an LDAP Authentication resource that provides authentication services.

Using Substitution Variable in the Password Field

For the following shared resources, the password is part of the inline credentials, which do not accept substitution variables.

- JDBC
- JMS Connection
- LDAP Connection
- LDAP Authentication
- HTTP Client
- SMTP

For the following shared resources, the password is a keystore password or key alias password. The passwords are stored in resource instance configuration files (runtime) in obfuscated format. These fields accept substitution variables.

- Keystore Provider
- Identity Provider
- Mutual Identity Provider
- SSL Client Provider
- SSL Server Provider

All the password fields have substitution variable validation. If a valid substitution variable in the format `%%svar-name%%` is provided in the password field, which does not accept substitution variable, then the validation error is displayed while saving shared resource in the Application Configurator, while validating configuration, and while generating runtime image.

If the invalid substitution variable is provided (for example, %%pass) in the password field, which does not accept the substitution variable, the validation error is not displayed.

Use of characters '%%' is allowed in the password field, provided that the password does not match the valid substitution variable format.

You can use container environment variables as alternatives to substitution variables to update inline credentials. For more information, see "Docker Run Command Reference" topic in *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*.

For example:

```
docker run -d -p 8931:8931 -e sr.NewSMTPResource.username=username -e
sr.NewSMTPResource.password=obfuscatedPassword --name bookstore
bookstore:v1
```

For more information about creating an obfuscated password from ActiveMatrix Administrator CLI, see "Creating an Obfuscated Password" in *TIBCO ActiveMatrix Service Grid Administration Guide*.

Also, you can update substitution variable for keystore password or key alias password by using container environment variables.

For example:

```
docker run -d -p 10104:10104 -e svar.sr.SVAR1.4_keyAlias=amx  -e
svar.sr.SVAR1.4_keyPassword=test123 --name bookstore bookstore:v1
```

If you are running application in Kubernetes, you can use ConfigMaps to pass substitution variables to the applications. You can also use Kubernetes Secrets to store passwords. For more information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*.

JVM Arguments

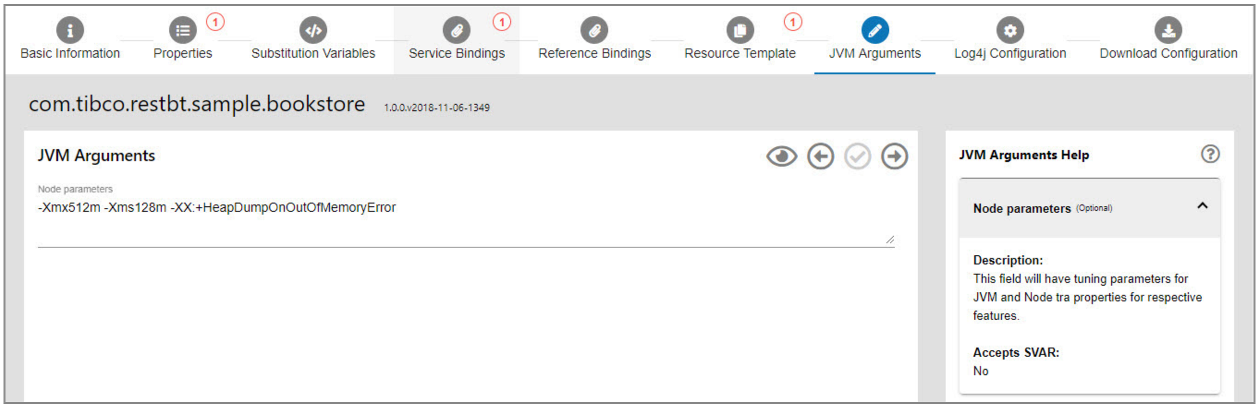
You can configure JVM parameters and node properties on the **JVM Arguments** tab. The properties must be added in the following format: -D*name*=*value*

For example:

```
-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -
Damx.securitymanager.enabled=false -
Dcom.tibco.tibjms.connect.attempt.timeout=3000
```

The default value (General arguments, without node properties) is `-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError`. If unspecified, then the default value is used.

Note: JVM arguments values are not validated. You must specify correct values of JVM arguments.



com.tibco.restbt.sample.bookstore 1.0.0.v2018-11-06-1349

JVM Arguments

Node parameters
-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError

JVM Arguments Help

Node parameters (Optional)

Description:
This field will have tuning parameters for JVM and Node tra properties for respective features.

Accepts SVAR:
No

Logging Configurations (Log4j)

In Log4j, a *logger* associates a runtime object with an appender, specifies the types of events to be logged, and whether to pass messages to a parent logger. A logger is an ancestor of another logger if its name (followed by a dot) is a prefix of the descendant logger name. A logger is a parent of a child logger if there are no ancestors between itself and the descendant logger.

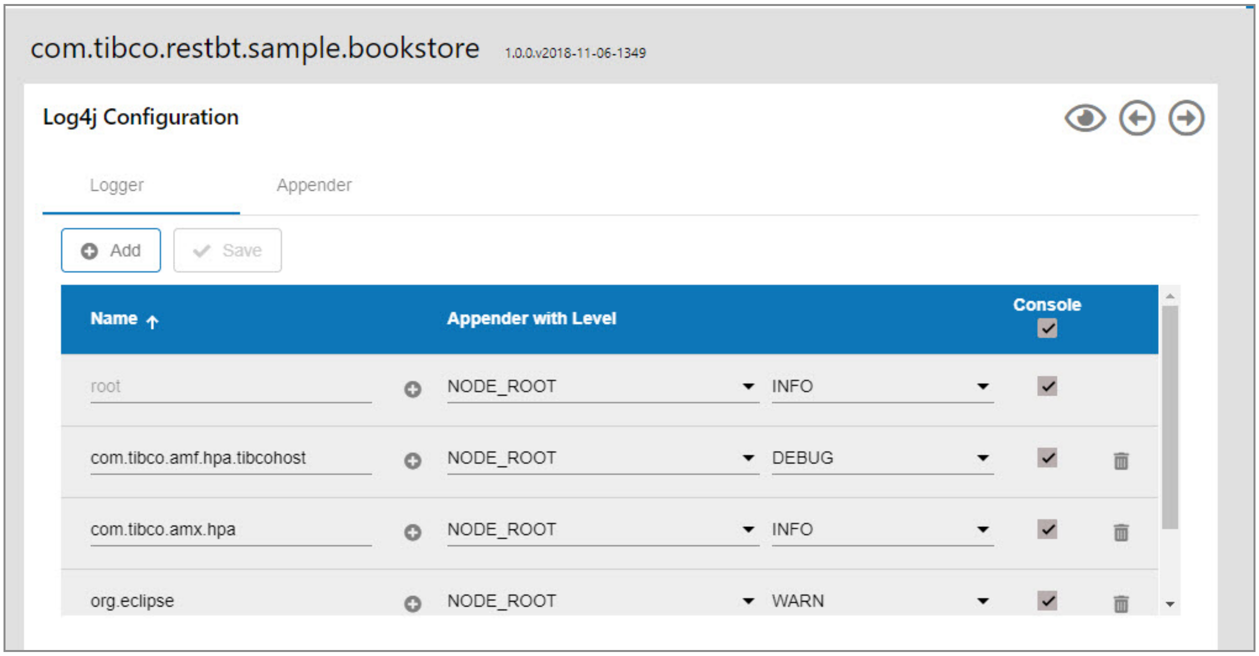
For example, `com.tibco` is a parent of `com.tibco.silver`. The logging level is specified for each appender that belongs to a logger. This lets a logger send logs to a different destination with a different level. You can use the Application Configurator interface to create loggers and appenders and to add appenders to existing loggers.

You can upload the same log configuration file from TIBCO ActiveMatrix Service Grid 3.x in Application Configurator. TIBCO recommends that you use the Application Configurator to edit the `log4j.xml` file instead of using a text editor.

Creating a Logging Configuration for an Application

On the **Log4j Configuration** tab in the Application Configurator, you can create a logging configuration for an application.

1. Click the **Log4j Configuration** tab.



Name ↑		Appender with Level		Console ☑	
root	⊕	NODE_ROOT	INFO	☑	
com.tibco.amf.hpa.tibcohost	⊕	NODE_ROOT	DEBUG	☑	🗑
com.tibco.amx.hpa	⊕	NODE_ROOT	INFO	☑	🗑
org.eclipse	⊕	NODE_ROOT	WARN	☑	🗑

2. In the **Logger** tab, click **Add**. A row is added to the list.
3. In the Logger Name column, type a logger name. The logger name must consist of alphabet, numbers, and characters (such as period [.], underscore [_], and hyphen [-]). The name must start with a letter of the alphabet.
4. In the Appender column, select an appender from the list. You can assign multiple appenders to the logger by clicking **Add New Appender** ⊕.
5. Select a logging level. If there are multiple appenders, the logging level is the highest level of the assigned appenders to that particular logger.
6. (Optional) Select the check box in the **Console** column to enable console appender for the logger. In the **Console** column, you can select all the loggers by selecting the check box at the top.
7. Click anywhere outside the fields and click **Save**.

Updating a Logging Configuration

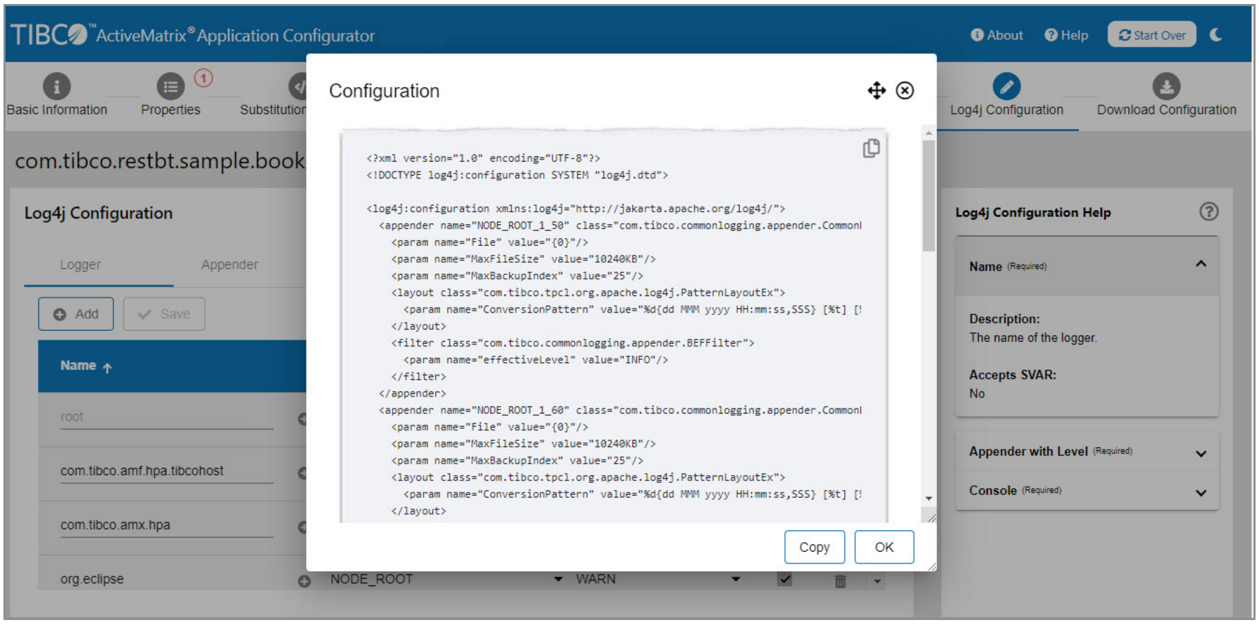
To update the logging configuration, edit the fields of a logger and click **Save**.

Deleting a Logging Configuration

To delete a logger, click **Delete** 🗑 next to the logger. You cannot delete or edit the root logger and assign appender. But you can change the logging level (INFO, DEBUG, etc.) of the "NODE_ROOT" log appender for the root logger.

Viewing XML Configuration

To view the XML log Configuration, click **Show XML** 👁 in the upper-right corner.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
  <appender name="NODE_ROOT_1_50" class="com.tibco.commonlogging.appender.Commonl
    <param name="File" value="{0}"/>
    <param name="MaxFileSize" value="10240KB"/>
    <param name="MaxBackupIndex" value="25"/>
    <layout class="com.tibco.tpcl.org.apache.log4j.PatternLayoutEx">
      <param name="ConversionPattern" value="%d{dd MMM yyyy HH:mm:ss,SSS} [%t] [
    </layout>
    <filter class="com.tibco.commonlogging.appender.BEFFilter">
      <param name="effectiveLevel" value="INFO"/>
    </filter>
  </appender>
  <appender name="NODE_ROOT_1_60" class="com.tibco.commonlogging.appender.Commonl
    <param name="File" value="{0}"/>
    <param name="MaxFileSize" value="10240KB"/>
    <param name="MaxBackupIndex" value="25"/>
    <layout class="com.tibco.tpcl.org.apache.log4j.PatternLayoutEx">
      <param name="ConversionPattern" value="%d{dd MMM yyyy HH:mm:ss,SSS} [%t] [
    </layout>
```

Name (Required)

Description:
The name of the logger.

Accepts SVAR:
No

Appender with Level (Required)

Console (Required)

Example:

The following example shows a sample Log4j XML configuration.

The number appended to the appender name is according to the level of the appender set for particular logger. For example, NODE_ROOT_1_50 is NODE_ROOT logger with applied level at INFO. NODE_ROOT_1_60 is NODE_ROOT logger with applied level at DEBUG.

The level of the logger is decided based on the highest level of the assigned appenders to that particular logger.

In the following example, the logger `com.tibco.amf.hpa.tibcohost` has only one FileAppender 'NODE_ROOT_1_50' assigned. Therefore, the logger level is INFO.

In the case of the logger `com.tibco.amf.admin.api.amx.application.impl`, two appenders are assigned to it, which are 'NODE_ROOT_1_50' and 'NODE_ROOT_1_60'. In this case, the higher level is DEBUG. Therefore, logger level is DEBUG.

```
<appender name="NODE_ROOT_1_50"
class="com.tibco.commonlogging.appender.CommonRollingFileAppender">
```

```
    <param name="File" value="{0}"/>
    <param name="MaxFileSize" value="10240KB"/>
    <param name="MaxBackupIndex" value="25"/>
    <layout class="com.tibco.tpcl.org.apache.log4j.PatternLayoutEx">
      <param name="ConversionPattern" value="%d{dd MMM yyyy HH:mm:ss,SSS}
[%t] [%-5p] %c - %m%n"/>
    </layout>
    <filter class="com.tibco.commonlogging.appender.BEFFilter">
      <param name="effectiveLevel" value="INFO"/>
    </filter>
  </appender>
  <appender name="NODE_ROOT_1_60"
class="com.tibco.commonlogging.appender.CommonRollingFileAppender">
    <param name="File" value="{0}"/>
    <param name="MaxFileSize" value="10240KB"/>
    <param name="MaxBackupIndex" value="25"/>
    <layout class="com.tibco.tpcl.org.apache.log4j.PatternLayoutEx">
      <param name="ConversionPattern" value="%d{dd MMM yyyy HH:mm:ss,SSS}
[%t] [%-5p] %c - %m%n"/>
    </layout>
    <filter class="com.tibco.commonlogging.appender.BEFFilter">
      <param name="effectiveLevel" value="DEBUG"/>
    </filter>
  </appender>
```

```
<logger name="com.tibco.amf.hpa.tibcohost" additivity="false">
    <level value="INFO"/>
    <appender-ref ref="NODE_ROOT_1_50"/>
    <appender-ref ref="CONSOLE"/>
  </logger>
<logger name="com.tibco.amf.admin.api.amx.application.impl"
additivity="false">
    <level value="DEBUG"/>
    <appender-ref ref="NODE_ROOT_1_50"/>
   <appender-ref ref="NODE_ROOT_1_60"/>
   <appender-ref ref="CONSOLE"/>
</logger>
```

Note: If the same appender is assigned to the logger with the different level, the one with the highest level is only assigned to the logger.

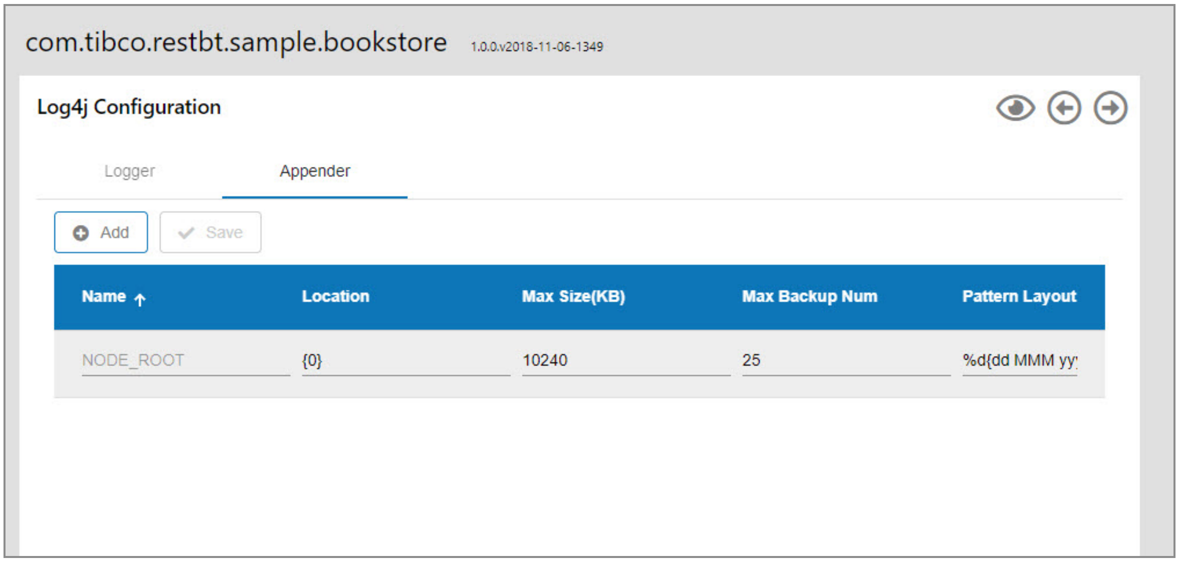
Note: You cannot create a logger with the name 'root' (or any variant of this by case sensitivity). If you try to add such a logger, a warning message is displayed. You cannot rename the existing root logger.

Configuring a Logging Appender

You can create, update, or delete a logging appender from the Application Configurator.

Creating a Logging Appender

1. Click the **Log4j Configuration** tab.
2. Click the **Appender** tab.



Name ↑	Location	Max Size(KB)	Max Backup Num	Pattern Layout
NODE_ROOT	{0}	10240	25	%d{dd MMM yy

3. Click **Add**. A row is added to the list.
4. Type a name for the appender in the Name column. Appender name must consist of alphabet, numbers, and characters (period [.], underscore [_], and hyphen [-]). The name must start with a letter of the alphabet. Enter the remaining fields as per your requirement. For information about the fields, see Logging Configuration Reference.
5. Click **Save**.

Updating a Logging Appender

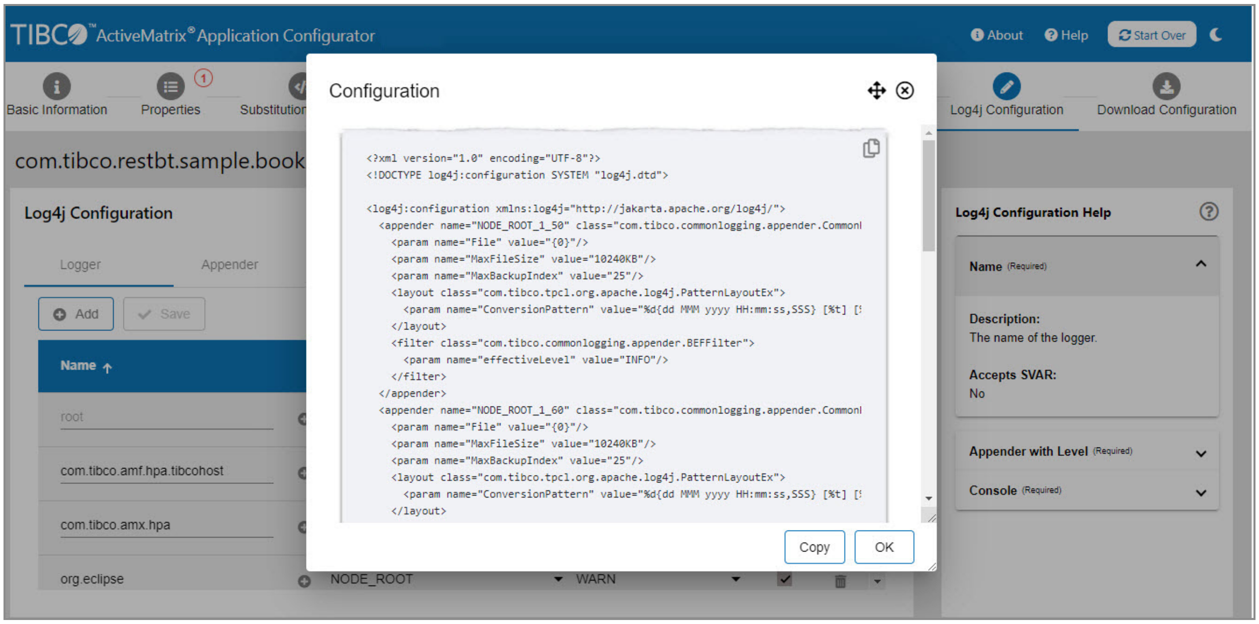
You can update the fields of a logging appender and click **Save**. Appenders name cannot be updated if it is assigned to any Logger.

Deleting a Logging Appender

To delete a logging appender, click **Delete** 🗑 next to the logging appender. An appender cannot be deleted if it is assigned to any Logger.

Viewing XML Configuration

To view the XML Log Configuration, click **Show XML** 👁 in the upper-right corner.



Description:
The name of the logger.

Accepts SVAR:
No

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
  <appender name="NODE_ROOT_1_50" class="com.tibco.commonlogging.appender.Commonl
    <param name="File" value="{0}"/>
    <param name="MaxFileSize" value="10240KB"/>
    <param name="MaxBackupIndex" value="25"/>
    <layout class="com.tibco.tpcl.org.apache.log4j.PatternLayoutEx">
      <param name="ConversionPattern" value="%d{dd MMM yyyy HH:mm:ss,SSS} [%t] [
    </layout>
    <filter class="com.tibco.commonlogging.appender.BEFFilter">
      <param name="effectiveLevel" value="INFO"/>
    </filter>
  </appender>
  <appender name="NODE_ROOT_1_60" class="com.tibco.commonlogging.appender.Commonl
    <param name="File" value="{0}"/>
    <param name="MaxFileSize" value="10240KB"/>
    <param name="MaxBackupIndex" value="25"/>
    <layout class="com.tibco.tpcl.org.apache.log4j.PatternLayoutEx">
      <param name="ConversionPattern" value="%d{dd MMM yyyy HH:mm:ss,SSS} [%t] [
    </layout>
```

Logging Configuration Reference

Log4j Configuration

Field	Required?	Editable?	Accepts SVars?	Description
Logger Name	Y	Y	N	The name of the Logger. Logger name must consist of alphabet, numeric and characters (period [.], underscore [_], hyphen [-]). The name must start with a letter of the alphabet.
Appender with Level	Y	Y	N	The destination to which log events are appended.

Log4j Configuration(Continued)

Field	Required?	Editable?	Accepts SVars?	Description
				All events of a level equal to or lower than the specified level are logged. For the Info level, Info, Warn, Error and Fatal events are logged. • TRACE: All events. • DEBUG: Fine-grained informational events used for debugging an application. • INFO: Coarse-grained informational messages that highlight the progress of the application. • WARN: Potentially harmful events. • ERROR: Errors that allow the application to continue running. • FATAL: Errors that cause the application to fail. • OFF: Blocks passing messages to a parent.
Console	Y	Y	N	Select the check box in the **Console** column to enable console appender for the logger.

Logging Appender Reference

Field	Description
Name	Name of the logging appender. Appender name must consist of characters, numeric (such as period [.], underscore [_], and hyphen [-]). The name must start with characters.
Location	The fully-qualified path to the log file. **Default:** {0}, represents the path of the default location.
Max Size(KB)	The maximum size of each log file, in kilobytes. Default: 10240
Max Backup Num	The number of log files to keep. When a log file reaches the maximum size, a new log file is created. After the number of files matches the number specified, the oldest file is deleted when a new file is created. Each file is appended with a number. **Default:** 25

Pattern Layout	Controls the format of the log entries for a clear text file appender. Conforms to the log4j pattern layout.
	Default:
	`"%d{dd MMM yyyy HH:mm:ss,SSS} [%t]`
	`[%-5p] %c %X{_cl.correlationId} - %m%n"`
	This string prints the date, the name of the thread that generated the event, the level of the logged event, the category of the logged event, a correlation ID (an enrichment field), a message, and a line separator. For example:
	`17 Dec 2009 16:43:41,250`

Logging Appender Reference(Continued)

Field	Description

	`[Job_Executor2] [INFO ] com.tibco.amf.hpa.tibcohost.node.TibcoHostNode. - Successfully finished processing of RDA rda6705267566599374829.zip`
	In addition to the default format, ActiveMatrix Service Grid - Container Edition also supports the pattern layouts extended with enrichment fields.
	`%R{_cl.physicalCompId.matrix.host} %d'{dd MMM yyyy HH:mm:ss,SSS}' [%t] [%-5p] %c - %m%n`
	Note: Pattern Layout is not validated in Application Configurator. Ensure that pattern layout conforms to Log4j pattern layout.

TIBCO Documentation and Support Services

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit https://docs.tibco.com.

Product-Specific Documentation

The following documentation for TIBCO ActiveMatrix® Service Grid - Container Edition is available on the TIBCO ActiveMatrix® Service Grid - Container Edition Product Documentation page:

- *TIBCO ActiveMatrix® Service Grid - Container Edition Release Notes*
- *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*
- *TIBCO ActiveMatrix® Service Grid - Container Edition Quick Start*
- *TIBCO ActiveMatrix® Service Grid - Container Edition Administration*
- *TIBCO ActiveMatrix® Service Grid - Container Edition Monitoring*

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit http://www.tibco.com/services/support.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at https://support.tibco.com.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to https://support.tibco.com. If you do not have a user name, you can request one by clicking Register on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to https://community.tibco.com.

Legal and Third-Party Notices

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (https://www.tibco.com/patents) for details.