Overview of Job Template Content

The topics in this section provide information about the XML syntax for job templates.

For a high-level introduction to job templates, see "Job Template" in SAS Event Stream Manager: Using SAS Event Stream Manager.

For example job templates, see SAS Event Stream Manager: Examples.

You cannot deploy job templates to an ESP server that is in a cluster. For more information, see "ESP Server in a Cluster" in SAS Event Stream Manager: Using SAS Event Stream Manager.

job-template

The job-template element is the top-level element in a job template.

Here is an example:

```xml
<job-template id="test1"
    localization-id="template-name"
    description-localization-id="template-description"/>

<!-- The rest of the job template contents are added here. -->

</job-template>
```
### Table 1 Properties of the job-template XML Element

#### Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Required</th>
<th>Regex Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>This attribute specified the unique identifier of this job template.</td>
<td>Yes</td>
<td><code>[_\w]+</code></td>
</tr>
<tr>
<td>localization-id</td>
<td>This attribute references a string value for the job template name used in SAS Event Stream Manager.</td>
<td>Yes</td>
<td><code>[0-9A-z][0-9A-Z_\-]+</code></td>
</tr>
<tr>
<td>description-localization-id</td>
<td>This attribute references a string value for a job template description used in SAS Event Stream Manager.</td>
<td>Yes</td>
<td><code>[0-9A-z][0-9A-Z_\-]+</code></td>
</tr>
</tbody>
</table>

#### Child Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Required</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>localization-strings</td>
<td>This element contains strings that display localized output.</td>
<td>Yes</td>
<td>The <code>localization-strings</code> element must appear as the first child element of the <code>job-template</code> element. For more information, see “localization-strings” on page 3.</td>
</tr>
<tr>
<td>parameters</td>
<td>This element contains user parameters that allow a user to enter data into the deployment and customize a job template when it is deployed.</td>
<td>No</td>
<td>The <code>parameters</code> element must appear before the initialization, instructions, server-filters, and failure-instructions elements. For more information, see “parameters” on page 5.</td>
</tr>
<tr>
<td>enumerations</td>
<td>This element contains definitions that restrict user input when used with user parameters.</td>
<td>No</td>
<td>For more information, see “enumerations” on page 12.</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
<td>Required</td>
<td>More Information</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>initialization</td>
<td>This element contains definitions to replace placeholders to alter a project when it is deployed.</td>
<td>No</td>
<td>“initialization” on page 26.</td>
</tr>
<tr>
<td>server-filters</td>
<td>This element specifies filters for ESP servers.</td>
<td>No</td>
<td>“server-filters” on page 13.</td>
</tr>
<tr>
<td>instructions</td>
<td>This element contains instructions that describe operations that must be performed to create or modify a deployment.</td>
<td>No</td>
<td>“instructions” on page 15.</td>
</tr>
<tr>
<td>failure-instructions</td>
<td>This element contains instructions that attempt to return the system to normal operation after an instruction has failed to execute.</td>
<td>No</td>
<td>“failure-instructions” on page 25.</td>
</tr>
</tbody>
</table>

**localization-strings**

The localization-strings element of the job template contains the string values for the labels displayed by SAS Event Stream Manager when executing that job template. Each string value is fully localizable within one or more language groups. The localization-strings element requires a default language.

Here is an example:

```xml
<job-template id="test1"
  localization-id="template-name"
  description-localization-id="template-description">
  <localization-strings default-language="en-us">
    <language id="en-us">
      <string id="template-name">test1</string>
      <string id="template-description"> Loads a project </string>
    </language>
  </localization-strings>
</job-template>
```
Each string element has an id attribute that is referred to by other elements of the job template when a string value is required. In the example here, template-name and template-description are referred to in the top-level job-template element.

In the SAS Event Stream Manager user interface, the string values are displayed in the following locations:

- on the **Job Templates** page
- in the window that appears when you deploy a job template from the **Job Templates** page
- on the **Log** page when the job created from the job template is executing

### Table 2  Properties of the localization-strings XML Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default-language</td>
<td>This attribute specifies the language to be used if the user’s current locale is not supported.</td>
</tr>
<tr>
<td></td>
<td>- Required: Yes.</td>
</tr>
<tr>
<td></td>
<td>- Valid values: This value must correspond to a defined language identifier. The value must be a valid ISO locale code. The locale code must be lowercase and in the format aa-bb (for example: en-us).</td>
</tr>
</tbody>
</table>

### Child Element

| language        | There must be one language element for each supported language and at least one that corresponds to the default-language attribute of the parent localization-strings element. |

### Table 3  Properties of the language XML Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>This attribute specifies the locale code for this language.</td>
</tr>
<tr>
<td></td>
<td>- Required: Yes.</td>
</tr>
<tr>
<td></td>
<td>- Valid values: A valid ISO locale code. The locale code must be lowercase and in the format aa-bb (for example: en-us).</td>
</tr>
</tbody>
</table>

### Child Element

| string    | There must be one string element for each localization string.                                                                          |
Overview of the parameters Element

The parameters element of the job template tells SAS Event Stream Manager what user input to capture and assign to internally stored variables that can be referenced throughout the job template. The parameters element includes child elements whose purpose is to guide user input. These selectors restrict the user’s text input or enable the user to select an option from a list. In the following example, the project-selector element enables the user to select from a list of projects stored in SAS Event Stream Manager’s repository:

```xml
<parameters>
  <project-selector id="projectSelector"
    localization-id="projectSelectorLabel"
    required="true"/>
</parameters>
```

You can nest some selectors inside other selectors. For example, information might be needed to populate available values for the selector in the user interface. In addition, you might need to show some parameters only upon selection of a valid value for another parameter. In such cases, the selectors should be nested inside each other, as shown in the following example:

```xml
<parameters>
  <project-selector id="projectSelector"
    localization-id="projectSelectorLabel"
    required="true">
    <query-selector id="querySelector"
      localization-id="checkQuery"
      required="true">
      <window-selector id="windowSelector"
        localization-id="checkWindow"
        required="true"/>
    </query-selector>
  </project-selector>
</parameters>
```

Table 4  Properties of the parameters XML Element

<table>
<thead>
<tr>
<th>Child Elements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project-selector</td>
<td>This element enables the user to select from a list of projects stored in SAS Event Stream Manager’s repository. For more information, see “project-selector” on page 7.</td>
</tr>
</tbody>
</table>
server-selector

This element enables the user to select from a list of ESP servers that are managed by SAS Event Stream Manager. For more information, see “server-selector” on page 8.

running-project-selector

This element enables the user to select a project that is either running on a specific ESP server or contained in a project that has been published in SAS Event Stream Processing Studio. For more information, see “running-project-selector” on page 9.

text-input-selector

This element enables the user to enter a text string that is checked against a regular expression. For more information, see “text-input-selector” on page 9.

query-selector

This element enables the user to select a currently running continuous query inside a project on a specific ESP server. For more information, see “query-selector” on page 10.

window-selector

This element enables the user to select a currently running window within a continuous query inside a project on a specific ESP server. For more information, see “window-selector” on page 11.

enum-selector

This element enables the user to select from a list of restricted values of an enumeration defined either globally or locally in the deployment. For more information, see “enum-selector” on page 11.

server-filter-selector

This element enables the user to select a previously created filter. For more information, see “server-filter-selector” on page 11.

Common Attributes of Selectors

Selectors have some common attributes, as shown in the following table:

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>This attribute specifies the unique identifier of the selector. This identifier is placed into the execution context.</td>
</tr>
<tr>
<td></td>
<td>- Required: Yes.</td>
</tr>
<tr>
<td></td>
<td>- A valid value is one that matches the regular expression: [0-9A-z] [0-9A-Z_] +</td>
</tr>
</tbody>
</table>
When a user selects the value of a selector, that value is stored internally by SAS Event Stream Manager. You can reference this value in the job template by the selector’s id value. To do this, enclose the id value in braces: { and }.

An id value can have one or more attributes. To access these attributes, use a dot (.) notation. For example, if the selector’s id value is alpha and this id value has the attribute beta, you can reference the selector by writing {alpha.beta}. Additional examples are provided in the following sections, where each selector is discussed in more detail.

**project-selector**

The project-selector element enables the user to select from a list of projects stored in SAS Event Stream Manager’s repository. This action produces a project object that has the following variables:

- **id**: the unique identifier of the project that has been selected by the user
- **version**: the major version number of the project that has been selected by the user
- **friendlyName**: the name of the project that has been selected by the user

The friendlyName project object can be particularly useful if you need to identify the project using its name rather than its identifier. For example, some SAS Event Stream Processing adapters need to reference the project name.

The project-selector element is a top-level selector and does not need to be nested inside any other selectors to function properly.

When deploying a job template, the user is prompted to select the project version. By default, the latest major version of the project is selected.

The project-selector element is useful for choosing a project that is then published to the ESP server. An example from a simplified job template is shown here:
server-selector

The server-selector element enables the user to select from a list of ESP servers that are associated with the selected deployment. This action produces an ESP server object that has the following variable:

id: the unique identifier of the ESP server that has been selected by the user

The server-selector element is a top-level selector and does not need to be nested inside any other selectors to function properly.

Alongside the project-selector element, the server-selector element is one of the most commonly used parameter types. The server-selector element is typically used to help define a server-filter element, as shown in the simplified example job template here:

```xml
<parameters>
  <project-selector id="projectSelector"
    localization-id="projectSelectorLabel"
    required="true" />
  <server-selector id="serverSelector"
    localization-id="serverLabel"
    required="true" />
</parameters>

<server-filters>
  <server-filter id="serverFilter"
    filter-expression="(objectId='{serverSelector.id}')" />
</server-filters>

<instructions>
  <load-project id="loadProjectInstruction"
    localization-id="loadProjectLabel"
    server-filter="serverFilter"
    project-id="{projectSelector.id}"
    project-version="{projectSelector.version}"
    running-project-name="{projectSelector.friendlyName}"
    start="true" />
</instructions>
```
running-project-selector

The running-project-selector element enables the user to select a project that is either running on a specific ESP server or contained in a project that has been published in SAS Event Stream Processing Studio. This action produces a project object that has the following variable:

id: the unique identifier of the project that has been selected by the user

The running-project-selector element must be nested inside a server-selector element to function properly.

Not all job templates relate to new installations. In some cases, you might need to make changes to existing projects or unload existing projects to complete the deployment successfully. The example here demonstrates how the following tasks are performed:

1. prompts the user to select an ESP server
2. prompts the user to select a project
3. stops the selected project running on that server

<parameters>
  <server-selector id="serverSelector"
      localization-id="serverLabel"
      required="true"
      <running-project-selector
      id="runningProjectSelector"
      localization-id="runningProjectLabel"
      required="true" />
    </server-selector>
  </parameters>
<server-filters>
  <server-filter id="serverFilter"
      filter-expression="(objectId='{serverSelector.id}')" />
</server-filters>
<instructions>
  <stop-project id="stopProjectInstruction"
      localization-id="stopProjectLabel"
      server-filter="serverFilter"
      running-project-name="{runningProjectSelector.friendlyName}" />
</instructions>

Note: The stop-project instruction stops the project. To entirely unload a project, use the unload-project instruction. For more information, see "unload-project" on page 23.

---

text-input-selector

The text-input-selector element enables the user to enter a text string that is checked against a regular expression. This action produces a string object that can be used throughout the job template.
The text-input-selector element is a top-level selector and does not need to be nested inside any other selectors to function properly.

In addition to properties common to all selectors (for more information, see "Common Attributes of Selectors" on page 6), the text-input-selector element has the XML attribute shown in the following table:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>validation-expression</td>
<td>This attribute defines the regular expression that is used to perform validation.</td>
</tr>
<tr>
<td></td>
<td>- Required: Yes.</td>
</tr>
<tr>
<td></td>
<td>- Valid values: The valid regular expression.</td>
</tr>
</tbody>
</table>

In the following example, the user's choice of a project name is restricted so that the name must begin with a letter. The remainder of the string must contain letters, numbers, or underscores. If validation fails, the localized string defined in project-not-valid is displayed to the user.

```xml
<parameters>
  <project-selector id="projectSelector"
    localization-id="projectSelectorLabel"
    required="true" />
  <server-selector id="serverSelector"
    localization-id="serverLabel"
    required="true" />
  <text-input-selector id="projectName"
    localization-id="invalidProjectLabel"
    required="true"
    validation-expression="^[A-z][A-z0-9_]+$"
    validation-localization-id="project-not-valid" />
</parameters>
```

query-selector

The query-selector element enables the user to select a currently running continuous query inside a project on a particular ESP server. This action produces a query object with the following variable:

- id: the unique identifier of the query that has been selected by the user
The query-selector element must be nested inside a project-selector element or a running-project-selector element to function properly.

**window-selector**

The window-selector element enables the user to select a currently running window within a continuous query inside a project on a particular ESP server. This action produces a query object with the following variable:

- **id**: the unique identifier of the window that has been selected by the user

The window-selector element must be nested inside a query-selector element to function properly.

**enum-selector**

The enum-selector element enables the user to select from a list of restricted values of an enumeration defined either globally or locally in the deployment. This action produces a string that can then be used anywhere in the job template.

The enum-selector element is a top-level selector and does not need to be nested inside any other selectors to function properly.

In addition to properties common to all selectors (for more information, see “Common Attributes of Selectors” on page 6), the enum-selector element has the XML attribute shown in the following table:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>This attribute provides additional information about the source of the data.</td>
</tr>
<tr>
<td></td>
<td>- Required: Yes.</td>
</tr>
<tr>
<td></td>
<td>- Valid values: The name of a defined enumeration.</td>
</tr>
</tbody>
</table>

For more information about enumerations, see “enumerations” on page 12.

**server-filter-selector**

The server-filter-selector element enables the user to select a filter. A separate window is displayed to the user. The window lists filters available only to the specific deployment as well as filters available for use with all deployments. The window also provides additional details (such as the filter expression) for each filter to help the user select an appropriate filter.

For more information, see “Filters” in SAS Event Stream Manager: Using SAS Event Stream Manager.
The server-filter-selector element does not need to be nested inside any other selectors to function properly.

Here is an example:

```xml
<parameters>
  <server-filter-selector id="failedServerFilter"
    required="true"
    localization-id="serverFilterLabel" />
</parameters>
<server-filters>
  <server-filter id="filter1"
    filter-expression="{failedServerFilter.filterExpression}" />
</server-filters>
<instructions>
  <group id="group1" localization-id="groupLabel"
    server-filter="filter1">
    <!-- appropriate instructions are added here -->
  </group>
</instructions>
```

This simplified example shows a server-filter-selector element called serverFilter. This element is then referenced in a server-filter element called filter1. This means that the set of ESP servers included in filter1 is the set of ESP servers returned by whichever filter a user selects for the serverFilter selector. Within the instructions element, the group element called group1 then refers to filter1. In this way, you can specify a group of instructions to execute against a set of ESP servers that a user selects. The strings shown in this example must be referenced in the localization-strings element, which is not shown in the example code here.

### enumerations

An enumeration is a finite list of options that is presented to the user to restrict input.

Each job template can support the declaration of local localized enumerations. Such declarations are optional. The enumerations element of the job template specifies a set of enumerations that can be used in the deployment. Each enumeration is fully localized and produces output to the user in the user’s specified locale (subject to the job template supporting the locale). Here is an example enumerations element that defines a single enumeration:

```xml
<enumerations>
  <enumeration id="country">
    <enumeration-value id="USA" localization-id="united-states" />
    <enumeration-value id="UK" localization-id="united-kingdom" />
  </enumeration>
</enumerations>
```

The enumeration with the id value of country can be referenced in the job template from the parameters element when you are limiting the input options the user can select from. In this example, the user is limited to the values USA and UK.

<table>
<thead>
<tr>
<th>Child Element</th>
<th>Properties of the enumerations XML Element</th>
</tr>
</thead>
</table>

Table 8
There must be one enumeration element for each enumeration.

**Table 9  Properties of the enumeration XML Element**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>This attribute specifies the unique identifier of the enumeration. If a global enumeration with the same identifier is defined, this one takes precedence.</td>
</tr>
<tr>
<td></td>
<td>- Required: Yes.</td>
</tr>
<tr>
<td></td>
<td>- A valid value is one that matches the regular expression: <code>[0-9A-z][0-9A-Z_\-]+</code></td>
</tr>
</tbody>
</table>

**Child Element**

| enumeration-value | Each distinct value in the enumeration must be listed as a child element. |

---

**server-filters**

The server-filters element specifies filters for ESP servers. A filter resolves to a set of ESP servers to which the project will be deployed or with which the project is associated. For more information about how filters work, see “Filters” in SAS Event Stream Manager: Using SAS Event Stream Manager.

**Table 10  Properties of the server-filters XML Element**

<table>
<thead>
<tr>
<th>Child Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server-filter</td>
<td>This element specifies a filter for ESP servers.</td>
</tr>
</tbody>
</table>

**Table 11  Properties of the server-filter XML Element**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>This attribute specifies the unique identifier of the server-filter element.</td>
</tr>
<tr>
<td></td>
<td>- Required: Yes.</td>
</tr>
<tr>
<td></td>
<td>- A valid value is one that matches the regular expression: <code>[0-9A-z][0-9A-Z_\-]+</code></td>
</tr>
</tbody>
</table>
localization-id

This attribute specifies the localization string ID. If a job fails, SAS Event Stream Manager enables the user to create a filter that references the failed ESP servers (for more information, see “Rerun a Job” in SAS Event Stream Manager). The job template that was used to create the failed job can include multiple server-filter elements. In this case, SAS Event Stream Manager creates a separate filter for each ESP server group on which the job failed. (For more information about the group element, see “group” on page 23.) The id attribute of the relevant server-filter element is added to the filter name that is entered by the user.

If the server-filter element includes a localization-id attribute, this attribute’s value is used instead of the id attribute to provide a friendly, localizable name. If a translated string is available in the user’s language, that translation is used. If no translated string is available, the value of the id attribute is used instead.

- Required: No.
- Valid values: This value must exist in the default language information within the localization-strings element.

filter-expression

This attribute contains a filter expression or references a previously saved filter (which contains a filter expression).

- Required: Yes.
- Valid values: A valid filter expression or the name of a previously saved filter. For more information, see “Filter Expression Syntax” in SAS Event Stream Manager: Using SAS Event Stream Manager.

Here is an example where the filter-expression attribute references an existing filter called failedserverfilter1:

```
<server-filters>
  <server-filter id="filter1"
    filter-expression="{failedserverfilter1.filterExpression}" />
</server-filters>
```

Here is a second example, where the filter-expression attribute includes a filter expression:

```
<server-filters>
  <server-filter id="filter1"
    filter-expression="{objectId='{server.id}'}" />
</server-filters>
```

In this second example, the filter expression contains a {server.id} variable. This code extract is from the Stock Trade example, where the {server.id} variable references the server-selector element (within the parameters element). The filter resolves to the ESP server selected by the user when the job template is deployed. For more information, see “The Stock Trade Job Template” in SAS Event Stream Manager: Examples.
The server-filters element can also be used together with the server-filter-selector element (within the parameters element). In addition, a server-filter element can be referenced from a group element (within the instructions element). For more information, including additional example code, see “server-filter-selector” on page 11.

instructions

The instructions element is at the core of the job template. It contains the commands that SAS Event Stream Manager executes. The instructions often contain identifiers for selectors used with the parameters element, as well as string identifiers from the localization-strings element.

The instructions element is the parent element that contains specific instructions.

Here is an example that shows the outline of the instructions element. The details of the stop-project and unload-project instructions have been removed.

```xml
<instructions>
  <stop-project <!-- The details for the stop-project instruction are added here. --> />
  <unload-project <!-- The details for the unload-project instruction are added here. --> />
</instructions>
```

The instructions element can also contain a group element that groups instructions together.

Here is an example. The details of the elements have been removed.

```xml
<instructions>
  <group <!-- The attributes of the first group element are added here. --> >
    <load-project <!-- The details for the load-project instruction are added here. --> />
    <start-project <!-- The details for the start-project instruction are added here. --> />
    <start-connectors <!-- The details for the start-connectors instruction are added here. --> />
  </group>
  <group <!-- The attributes of the second group element are added here. --> >
    <load-project <!-- The details for the load-project instruction are added here. --> />
    <start-project <!-- The details for the start-project instruction are added here. --> />
    <start-connectors <!-- The details for the start-connectors instruction are added here. --> />
  </group>
</instructions>
```
### Table 12  Properties of the instructions XML Element

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
</table>
| server-filter               | This attribute specifies a filter for ESP servers. The filter resolves to a set of ESP servers to which the project will be deployed or with which the project is associated.  
  - **Required**: Each instruction must have a `server-filter` attribute. This attribute can be included in the instruction itself or in the enclosing `group` element or `instructions` element.  
  - A `server-filter` attribute on a child element overrides a `server-filter` attribute on a parent element. For example, a `server-filter` attribute on a specific instruction overrides a `server-filter` attribute on the parent `instructions` element.  
  - **Valid values**: The ID of a `server-filter` element. For more information, see “server-filters” on page 13. |
| on-failure                  | This attribute specifies how SAS Event Stream Manager handles an instruction that fails to execute. By default, if an instruction fails to execute, any instructions that depend on the failed instruction are canceled. You can use the `on-failure` attribute to specify different behavior.  
  - **Required**: No. This attribute can be included in a specific instruction or in the enclosing `group` element or `instructions` element.  
  - An `on-failure` attribute on a child element overrides an `on-failure` attribute on a parent element. For example, an `on-failure` attribute on a specific instruction overrides an `on-failure` attribute on the parent `instructions` element.  
  - **Valid values**: `continue`. Setting the `on-failure` attribute to `continue` means that dependent instructions will be executed even if the original instruction fails. If you do not want this to happen, remove the `on-failure` attribute. |

<table>
<thead>
<tr>
<th>Child Elements</th>
<th>Description</th>
</tr>
</thead>
</table>
| load-project                | This instruction takes a specific project that is available in SAS Event Stream Manager and loads it to an active ESP server.  
  - **Required**: No.  
  - For more information, see “load-project” on page 19. |
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
<th>Required</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>start-project</td>
<td>This instruction starts an already loaded SAS Event Stream Processing project.</td>
<td>No</td>
<td>For more information, see “start-project” on page 20.</td>
</tr>
<tr>
<td>start-connectors</td>
<td>This instruction starts any connectors associated with a published SAS Event Stream Processing project.</td>
<td>No</td>
<td>For more information, see “start-connectors” on page 21.</td>
</tr>
<tr>
<td>modify-project</td>
<td>This instruction updates an existing running project, subject to the rules imposed by SAS Event Stream Processing.</td>
<td>No</td>
<td>For more information, see “modify-project” on page 21.</td>
</tr>
<tr>
<td>stop-project</td>
<td>This instruction stops a published SAS Event Stream Processing project. Stopping a project does not remove the project, but it stops any new connections to it.</td>
<td>No</td>
<td>For more information, see “stop-project” on page 22.</td>
</tr>
<tr>
<td>unload-project</td>
<td>This instruction unloads a published SAS Event Stream Processing project.</td>
<td>No</td>
<td>For more information, see “unload-project” on page 23.</td>
</tr>
<tr>
<td>group</td>
<td>This element groups instructions together.</td>
<td>No</td>
<td>For more information, see “group” on page 23.</td>
</tr>
</tbody>
</table>

Common Attributes of Instructions and the group Element

Instructions and the group element have some common attributes, as shown in the following table:

<table>
<thead>
<tr>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td><strong>id</strong></td>
</tr>
<tr>
<td>Required: Yes.</td>
</tr>
<tr>
<td>A valid value is one that matches the regular expression: [0-9A-z][0-9A-Z_-]+</td>
</tr>
<tr>
<td><strong>localization-id</strong></td>
</tr>
<tr>
<td>Required: Yes.</td>
</tr>
<tr>
<td>Valid values: This value must exist in the default language information within the localization-strings element.</td>
</tr>
<tr>
<td><strong>server-filter</strong></td>
</tr>
<tr>
<td>Required: Each instruction must have a server-filter attribute. This attribute can be included in the instruction itself or in the enclosing group element or instructions element.</td>
</tr>
<tr>
<td>A server-filter attribute on a child element overrides a server-filter attribute on a parent element. For example, a server-filter attribute on a specific instruction overrides a server-filter attribute on the parent instructions element.</td>
</tr>
<tr>
<td>Valid values: The ID of a server-filter element. For more information, see “server-filters” on page 13.</td>
</tr>
<tr>
<td><strong>on-failure</strong></td>
</tr>
<tr>
<td>Required: No. This attribute can be included in a specific instruction or in the enclosing group element or instructions element.</td>
</tr>
<tr>
<td>An on-failure attribute on a child element overrides an on-failure attribute on a parent element. For example, an on-failure attribute on a specific instruction overrides an on-failure attribute on the parent instructions element.</td>
</tr>
<tr>
<td>Valid values: continue. Setting the on-failure attribute to continue means that dependent instructions will be executed even if the original instruction fails. If you do not want this to happen, remove the on-failure attribute.</td>
</tr>
</tbody>
</table>
depends-on

This attribute lists the instructions that this instruction or group depends on.

- **Required**: No.
- **Valid values**: The values of id attributes of other instructions in the same job template.

The common attributes have been omitted from subsequent sections, which describe additional XML properties for each instruction and the `group` element.

---

**load-project**

This instruction takes a specific project that has previously been published in SAS Event Stream Processing Studio and loads it to an active ESP server.

In addition to properties common to all instructions (for more information, see "Common Attributes of Instructions and the group Element" on page 17), the `load-project` instruction has the XML attributes shown in the following table:

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>project-id</strong></td>
<td>This attribute specifies the ID of the project to be loaded to the SAS Event Stream Processing factory server. The project ID is generated by SAS Event Stream Processing Studio when a project is published. The ID is available using the <code>project-selector</code> element, or the literal value can be obtained from the application when viewing the project metadata.</td>
</tr>
<tr>
<td></td>
<td><strong>Required</strong>: No. Your job template must contain either the <code>project-id</code> element or the <code>project-name</code> element, but not both.</td>
</tr>
<tr>
<td></td>
<td><strong>Valid values</strong>: A valid project ID stored internally by SAS Event Stream Manager.</td>
</tr>
<tr>
<td><strong>project-name</strong></td>
<td>This attribute specifies the literal name of the project to be loaded to the SAS Event Stream Processing factory server. This attribute can be useful if you do not have access to project IDs and cannot use the <code>project-id</code> attribute.</td>
</tr>
<tr>
<td></td>
<td><strong>Required</strong>: No. Your job template must contain either the <code>project-id</code> element or the <code>project-name</code> element, but not both.</td>
</tr>
<tr>
<td></td>
<td><strong>Valid value</strong>: A valid value is one that matches the regular expression: <code>[\w]+</code></td>
</tr>
</tbody>
</table>
This attribute specifies the major version of the project to be published to the SAS Event Stream Processing factory server.

- Required: No.
- Valid values: A valid project version, which is specified by a number (for example, 2) or by using the {project.version} placeholder.
- latest. If the attribute is set to latest, the latest version of the project is used.

Project XML files created in SAS Event Stream Processing Studio 5.2 and later versions contain separate variables that correspond to a project’s major version number and a project’s minor version number. This attribute specifies a project’s major version number as follows: project-version="{project.version}".

You cannot specify the minor version number. Instead, SAS Event Stream Processing always uses the latest minor version of the specified major version.

This attribute specifies the name of the SAS Event Stream Processing project that you want to use to publish the model.

- Required: Yes.
- A valid value is one that matches the regular expression: [A-z0-9_]+.

This attribute specifies whether the project starts automatically.

If set to true, the project automatically starts upon publication. Otherwise, a manual start-project instruction is required. The default value is true.

- Required: No.
- Valid values: true or false.

This instruction starts an already loaded SAS Event Stream Processing project.

In addition to properties common to all instructions (for more information, see “Common Attributes of Instructions and the group Element” on page 17), the start-project instruction has the XML attributes shown in the following table:

Table 15 Properties of the start-project XML Element

<table>
<thead>
<tr>
<th>Attributes</th>
</tr>
</thead>
</table>
running-project-name

This attribute specifies the name of the SAS Event Stream Processing project that you want to use to publish the model.

- Required: Yes.
- A valid value is one that matches the regular expression: `[A-z0-9_]`+

---

start-connectors

This instruction starts any connectors associated with a published SAS Event Stream Processing project.

In addition to properties common to all instructions (for more information, see "Common Attributes of Instructions and the group Element" on page 17), the start-connectors instruction has the XML attributes shown in the following table:

<table>
<thead>
<tr>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>running-project-name</td>
</tr>
</tbody>
</table>

Table 16 Properties of the start-connectors XML Element

---

modify-project

This instruction updates an existing running project, subject to the rules imposed by SAS Event Stream Processing.

The modify-project instruction uses the following SAS Event Stream Processing command to set the state of the project to modified:

```plaintext
PUT http://<server>:/SASESP/projects/<projectId>/state?value=modified
```

In addition to properties common to all instructions (for more information, see "Common Attributes of Instructions and the group Element" on page 17), the modify-project instruction has the XML attributes shown in the following table:

<table>
<thead>
<tr>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>running-project-name</td>
</tr>
</tbody>
</table>
**project-id**

This attribute specifies the ID of the project to be loaded to the SAS Event Stream Processing factory server. The project ID is generated by SAS Event Stream Processing Studio when a project is published.

The ID is available using the `project-selector` element, or the literal value can be obtained from the application when viewing the project metadata.

- **Required:** Yes.
- **Valid values:** A valid project ID stored internally by SAS Event Stream Manager.

**project-version**

This attribute specifies the version of the project to be published to the SAS Event Stream Processing factory server.

- **Required:** Yes.
- **Valid values:** A valid project version, which is specified by using the `{project.version}` placeholder.

Project XML files created in SAS Event Stream Processing Studio 5.2 and later versions contain separate variables that correspond to a project's major version number and a project's minor version number. This attribute specifies a project's major version number as follows: `project-version="{project.version}"`.

You cannot specify the minor version number. Instead, SAS Event Stream Processing always uses the latest minor version of the specified major version.

**running-project-name**

This attribute specifies the name of the SAS Event Stream Processing project that you want to use to publish the model.

- **Required:** Yes.
- **Valid values:** One that matches the regular expression: `[A-z0-9_]`.

**stop-project**

This instruction stops a published SAS Event Stream Processing project. Stopping a project does not remove the project, but it stops any new connections to it.

In addition to properties common to all instructions (for more information, see "Common Attributes of Instructions and the group Element" on page 17), the `stop-project` instruction has the XML attributes shown in the following table:

**Table 18** Properties of the `stop-project` XML Element

<table>
<thead>
<tr>
<th>Attributes</th>
</tr>
</thead>
</table>
running-project-name

This attribute specifies the name of the SAS Event Stream Processing project to stop.

- Required: Yes.
- A valid value is one that matches the regular expression: \[A-z0-9_]+\n
For an example of how to use the `stop-project` instruction, see "A Job Template for Stopping a Project" in SAS Event Stream Manager: Examples.

unload-project

This instruction unloads a published SAS Event Stream Processing project.

In addition to properties common to all instructions (for more information, see "Common Attributes of Instructions and the group Element" on page 17), the `unload-project` instruction has the XML attributes shown in the following table:

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>running-project-name</td>
<td>This attribute specifies the name of the SAS Event Stream Processing project to unload.</td>
</tr>
<tr>
<td></td>
<td>- Required: Yes.</td>
</tr>
<tr>
<td></td>
<td>- A valid value is one that matches the regular expression: [A-z0-9_]+\</td>
</tr>
</tbody>
</table>

For an example of how to use the `unload-project` instruction, see "A Job Template for Stopping a Project" in SAS Event Stream Manager: Examples.

If a project is reported as missing (🚫) in the Running Projects tile, you might not be able to unload it using the `unload-project` instruction. For more information, see “Stop a Running Job” in SAS Event Stream Manager: Using SAS Event Stream Manager.

group

The `group` element enables you to group instructions together. Grouping instructions can make it easier to manage common dependencies. Grouping instructions also enables you to execute a job on several ESP servers.

In addition to properties common to instructions (for more information, see “Common Attributes of Instructions and the group Element” on page 17), the `group` element has the XML properties shown in the following table:
Here is an example in which two projects are deployed to two different sets of ESP servers:

```xml
<instructions>
  <group id="startgroup1" localization-id="startgroup1"
    server-filter="filter1">
    <load-project id="load-project1"
      localization-id="load-project"
      project-id="{project1.id}"
      project-version="{project1.version}"
      running-project-name="{project1display}" start="false" />
    <start-project id="start-project1"
      localization-id="start-project"
      running-project-name="{project1}"
      depends-on="load-project1" />
    <start-connectors id="start-connectors1"
      localization-id="start-connectors"
      running-project-name="{project1}"
      depends-on="start-project1" />
  </group>
  <group id="startgroup2" localization-id="startgroup2"
    server-filter="filter2">
    <load-project id="load-project2"
      localization-id="load-project"
      project-id="{project2.id}"
      project-version="{project2.version}"
      running-project-name="{project2display}" start="false" />
    <start-project id="start-project2"
      localization-id="start-project"
      running-project-name="{project2}"
      depends-on="load-project2" />
    <start-connectors id="start-connectors2"
      localization-id="start-connectors"
      running-project-name="{project2}"
      depends-on="start-project2" />
  </group>
</instructions>
```

The first `group` element provides instructions for loading and starting a project and for starting connectors. When a user deploys the template, the user selects the project to deploy (`project1`) and enters a display name for the project (`project1display`). These strings must be referenced in the `localization-strings` element (these are not shown in the example code here). The project is deployed to a set of ESP servers that match a filter (`filter1`). This filter must be referenced in the `server-filters` element (this is not shown in the example code here).

Similarly, the second `group` element enables the user to select a second project to deploy to another set of ESP servers.
failure-instructions

An element called failure-instructions is activated if any of the commands in the instructions element of the job template fail.

The failure-instructions element has the same child elements as the instructions element. That is, the failure-instructions element can contain any instructions as well as the group element. For more information, see “instructions” on page 15.

Table 21 Properties of the failure-instructions XML Element

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>localization-id</td>
<td>This attribute specifies the localization string ID.</td>
</tr>
<tr>
<td></td>
<td>■ Required: Yes.</td>
</tr>
<tr>
<td></td>
<td>■ Valid values: This value must exist in the default language information within the localization-strings element.</td>
</tr>
<tr>
<td>server-filter</td>
<td>This attribute specifies a filter for ESP servers. The filter resolves to a set of ESP servers to which the project will be deployed or with which the project is associated.</td>
</tr>
<tr>
<td></td>
<td>■ Required: Each instruction must have a server-filter attribute. This attribute can be included in the instruction itself or in the enclosing group element or failure-instructions element.</td>
</tr>
<tr>
<td></td>
<td>■ A server-filter attribute on a child element overrides a server-filter attribute on a parent element. For example, a server-filter attribute on a specific instruction overrides a server-filter attribute on the parent failure-instructions element.</td>
</tr>
<tr>
<td></td>
<td>■ Valid values: The ID of a server-filter element. For more information, see “server-filters” on page 13.</td>
</tr>
</tbody>
</table>
on-failure

This attribute specifies how SAS Event Stream Manager handles a failure instruction that fails to execute. By default, if an instruction fails to execute, any instructions that depend on the failed instruction are canceled. You can use the on-failure attribute to specify different behavior.

- Required: No. This attribute can be included in a specific instruction or in the enclosing group element or failure-instructions element.

An on-failure attribute on a child element overrides an on-failure attribute on a parent element. For example, an on-failure attribute on a specific instruction overrides an on-failure attribute on the parent failure-instructions element.

- Valid values: continue. Setting the on-failure attribute to continue means that dependent instructions will be executed even if the original instruction fails. If you do not want this to happen, remove the on-failure attribute.

Child Elements

The failure-instructions element has the same child elements as the instructions element. That is, the failure-instructions element can contain any instructions as well as the group element. For more information, see “instructions” on page 15.

Here is an example of a failure-instructions element:

```xml
<failure-instructions localization-id="recovery-steps">
  <unload-project id="unload" localization-id="unload"
    server-filter="filter1" project="example_proj" />
</failure-instructions>
```

initialization

You can use the initialization element of the job template to alter SAS Event Stream Processing projects as the projects are deployed by using the project-transformation instruction. The transform looks for a placeholder variable in the project and replaces it with the contents of the XML element. Here is an example:

```xml
<initialization>
  <project>
    <project-transformation id="transformed-project"
      project-id="{project.id}"
      project-version="{project.version}"
    >
      <placeholder id="lang">{language}</placeholder>
    </project-transformation>
  </project>
</initialization>
```
Variables used for {project} are resolved at execution when the user selects a project. This is defined in the parameters element of the job template. The placeholder element within the initialization element has an attribute id that is set to the value of the placeholder in the project. In the example here, the project for deployment uses the SAS Event Stream Processing Twitter adapter. The placeholder allows the user to set the language to filter on when deploying:

<expression>tw_Lang == '{lang}'</expression>

Table 22 Properties of the project-transformation XML Element

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
</table>
| **id**          | This attribute specifies the unique identifier of this instruction.  
|                 | - Required: Yes.  
|                 | - A valid value is one that matches the regular expression: [0-9A-z][0-9A-z_-]+                                                               |
| **project-id**  | This attribute specifies the ID of the project to be transformed. The ID is available using the project-selector element (for more information, see “project-selector” on page 7), or the literal value can be obtained from the application when viewing the project metadata.  
|                 | - Required: Yes.  
|                 | - Valid values: A valid project ID stored internally by SAS Event Stream Manager.                                                            |
| **project-version** | This attribute specifies the version of the project to be transformed.  
|                  | - Required: Yes.  
|                  | - Valid values: A valid project version. Project XML files created in SAS Event Stream Processing Studio 5.2 and later versions contain separate variables that correspond to a project’s major version number and a project’s minor version number. This attribute specifies a project’s major version number as follows: project-version="{project.version}".  
|                  | You cannot specify the minor version number. Instead, SAS Event Stream Processing always uses the latest minor version of the specified major version. |

Child Elements

| placeholder     | One or more placeholder elements can be defined to replace placeholders in the project. |

Table 23 Properties of the placeholder XML Element

Attributes
<table>
<thead>
<tr>
<th><strong>id</strong></th>
<th>This attribute specifies the placeholder as it appears in the project.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Required:</strong> Yes.</td>
</tr>
<tr>
<td></td>
<td><strong>Valid values:</strong> This value must match the value of the placeholder in the project.</td>
</tr>
</tbody>
</table>

**Content**

| **placeholder** | This content specifies the value that the placeholder resolves to. The content can be a job template variable enclosed in braces: `{ and }`. |