# Element Specifications

# Introduction

Every element included with Call Studio and VXML Server must be configured before it can be used. This reference file contains a detailed specification for each of the core Cisco Unified Customer Voice Portal (Unified CVP) elements, listing all the options available in the configuration. The specifications must be followed, or the element may complain with an error message or behave erratically.

Each element specification in this reference file presents information on some or all of the following topics:

- **Overview** – Each specification starts with a brief description of the element's behavior including what it does, how it reacts to various settings and audio groups, and other miscellaneous behavior. This information should help the developer decide whether to use these elements in an application or to rely on custom elements.

- **Settings** – Settings contain information that affects how the element behaves. Each setting has the following attributes:

    - **Type** – The type of data accepted such as a string, text, boolean, integer, or enumeration.

    - **Required** – This defines whether the setting is required to have a value *if the setting is active* (available to be configured in Builder for Studio).

> **Note**   The definition of required in this case is that the setting must have an appropriate value for Builder for Studio to validate the voice element configuration.

    - **Single setting value** – This defines whether the setting can have multiple values. If set to `true`, then the setting may have only a single configuration value. Multiple value settings are created in Builder for Studio by right clicking on the setting and choosing the *add setting name* option.

    - **Substitution allowed** – This setting attribute determines if the setting value can include substitution.

    - **Default** – The initial value of an element setting when a new element is dragged to the workspace.

- **Events** – Event and exceptions occurring in a Cisco Unified Call Studio application can be handled by event handlers defined in the applications. When event handlers are configured for elements, the corresponding exit states are created. Following events types are supported:

    - Custom Exception- User defined application specific exception.

    - Java Exception - Java Exception occurring on a VXML server when running applications.

    - VXML Event - VXML events thrown by the Voice XML browser.

    - Hotlink - Local Hotlinks defined for voice elements.

  Event handlers can be configured with the following attributes:

    - **Name** - The Event Handler name can be changed according to the requirement and the Event Handler name will be added as an exit state.

    - **Event Type** - You can select the event handler type depending on the element, the applicable event types are listed in the drop-down list.

    - **Event List** - You can select from a pre-defined list of VXML and Java exceptions using the drop-down list. In addition, you can also enter a comma separated list of VXML events, Java exceptions, or user defined custom exceptions based on the Event Type selected.

      Example 1, for VXML event you can enter error.badfetch to catch a VXML event named error.badfetch

      Example 2, for Java Exception event you can enter "java.io.FileNotFoundException" to catch a Java exception named java.io.FileNotFoundException

      Example 3, for Custom Exception event you can enter "com.cisco.CustomException" to catch a user defined exception named com.cisco.CustomException

> **Note** You can enter *.* to handle all the events and exceptions. * is allowed only at the end of the event name followed by "."(dot).

    - **DTMF** - A digit which activates the hotlink. This attribute is applicable to Hotlink event handlers.

    - **Speech** - A spoken keyword or keywords which activate the hotlink. This attribute is applicable to Hotlink event handlers.

    - **Throw Hotevent** - The Voice XML event to be thrown when Hotlink is activated. When choosing the option to throw an event, the full name of the VoiceXML event must be entered in the provided text box.

- **Element Data** – Some elements capture data or yield information that may be useful to other elements, or for logging purposes. The variables created by each element are listed here.

- **Exit States** – Each element may have one or more exit states that indicate the dialog status when the element was run successfully. These are pre-defined Exit states that do not appear in an element configuration and cannot be changed. However, when an Event handler is associated to the elements, the corresponding Exit state (<event handler type>-<event handler name> is added along with the pre-defined state.

- **Audio Groups** – Voice elements define audio groups that define the different places within the element that audio can be played. Application designers configure the contents of audio groups as a list of audio items that are played one after the other. Audio items may be pre-recorded audio files, text-to-speech (TTS) phrases, and Say It Smart types (playback of formatted data such as dates, currency amounts, and so on). Each audio group can be required or optional and can also define multiple counts. Audio groups with multiple counts are used to define different audio to play each time a certain VoiceXML event occurs (often known as tapered prompts).

**Note** You can create your custom elements or use additional Java classes in the Cisco Call Studio. If you need support in developing or troubleshooting it, you must have a developer support services contract or work with a Cisco partner/Cisco Advanced Services who has a developer support services contract.