



Standard Decision Elements

Decision elements apply business logic to decide which exit state to return. A pre-built, configurable decision element has already defined the business logic and only requires a configuration to modify its behavior. Standard decision elements, however, are defined by the developer and have no configuration since they represent decisions specific to an application. For simple to moderately complex decisions, Unified CVP provides a means of defining decisions without programming by constructing an XML document (the Unified CVP XML decision format is described in the [User Guide for Cisco Unified CVP VXML Server and Unified Call Studio](#)). Should this format prove insufficient, Java or XML APIs are provided to allow the developer to build the business logic programmatically.

A standard decision element, in addition to the functionality provided all components, is allowed to create and modify element data.

- [Java API Use, on page 1](#)
- [XML API Use, on page 1](#)
- [Remote Execution, on page 3](#)

Java API Use

A standard decision element is built in Java by extending the abstract base class `DecisionElementBase` found in the `com.audium.server.voiceElement` package (this package's name is such due to backwards compatibility considerations). It contains a single abstract method named `doDecision` that acts as the method for the decision element, and must be implemented by the developer. The method receives two arguments: the name of the decision element (as a *String*) and an instance of `DecisionElementData`. This class belongs to the Session API and is used to access session information (See [Session API](#) for more on this API). The method expects a *String* object in return containing the exit state in the exact format specified in Builder for Call Studio when the standard decision element was first defined.

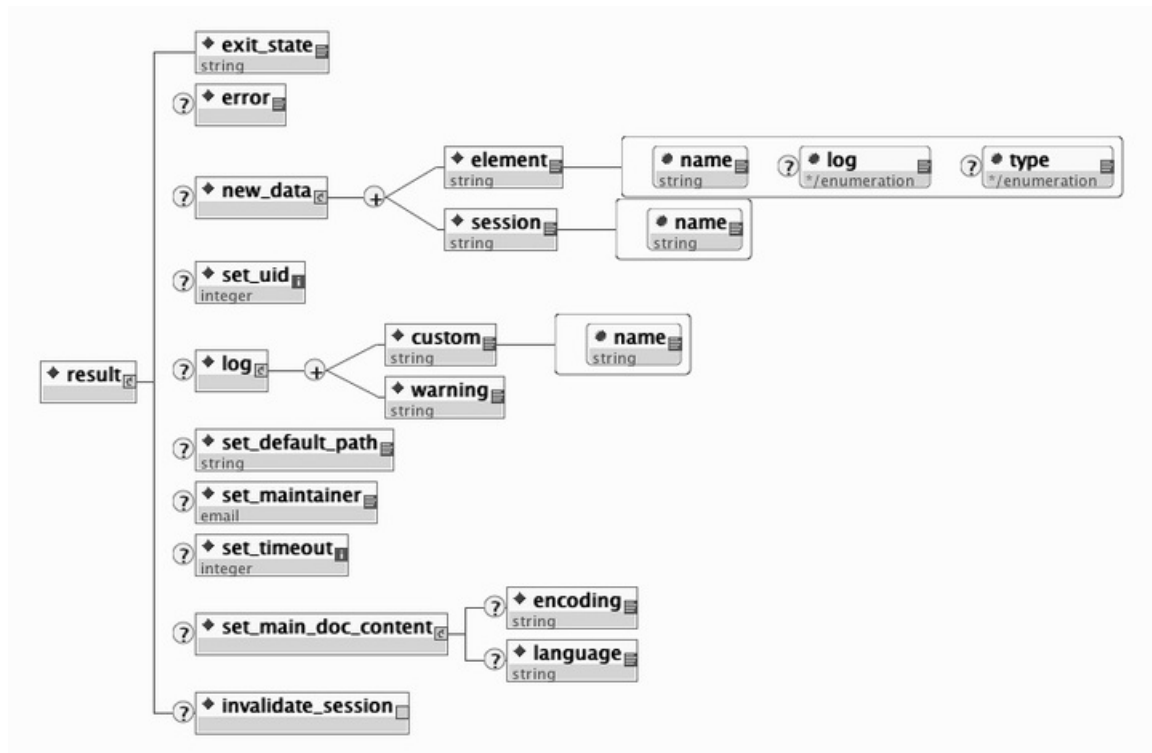
The `DecisionElementBase` class defines many methods in addition to `doDecision`. These are used for configurable decision elements, which also extend the class. The only method required for generic decision elements is `doDecision`, as it is the only abstract method in `DecisionElementBase`.

XML API Use

As described in [Session API](#), the standard *inputs* and *settings* XML documents are sent via POST to the decision element URI. An additional parameter, called *name*, is sent containing the name of the decision element. The following figure shows the DTD diagram of the XML document that must be sent in response.

The DTD for the generic action element response is defined in the file `DecisionResponse.dtd` found in the VXML Server `dtDs` folder.

Figure 1: Standard Decision Elements DTD Diagram



The elements in this XML document are:

- **exit_state** – This tag contains the string value representing the exit state in the exact format specified in Builder for Call Studio when the standard decision element was first defined.
- **error** – This tag reports to VXML Server that an error occurred while running the standard decision element. VXML Server will then throw an exception whose message is contained in the `<error>` tag. This allows the XML API to throw exceptions just as the Java API does.
- **new_data** – This tag holds the element and session data this standard decision element is to create. Any number of `<set_element>` and `<set_session>` tags can appear, one for each element and session data variable to be created. The `log` attribute of `<set_element>` sets whether the value of the variable is stored in the activity log. The optional `type` attribute is used to specify the data type of the variable and can be *string*, *int*, *float*, or *boolean*. The `create` attribute found in both tags determines when the variable is created, before the element is entered (*before_enter*), or after the element exits (*after_exit*).
- **set_uid** – This tag is used to associate the call with a UID in the user management system. The content of the tag should be the integer UID.
- **set_flag** – This tag is used to make the decision element act like a flag when visited. If it appears, a flag with the same name as the decision element will be considered triggered and that fact will be noted in the activity log.
- **log** – This tag is used to store information in log files for this application when this standard decision element is run. Any number of `<custom>` tags can appear, denoting the information to insert in the

application's activity log. The `name` attribute holds the name of the data, and the `<custom>` tag encapsulates the value. Any number of `<warning>` tags can appear, denoting warnings to be placed in the application's error log. The `<warning>` tag encapsulates the warning message.

- **set_default_path** – This tag is used to change the default audio path from this point onwards for this call.
- **set_maintainer** – This tag is used to change the maintainer e-mail address from this point onwards for this call.
- **set_timeout** – This tag allows the timeout length set for this session to be changed. The contents of the tag must be an integer representing the number of minutes in the timeout.
- **set_main_doc_content** – This tag allows the encoding and language settings for the main VoiceXML documents to be changed from this point onwards for this call. The `<language>` tag content is formatted according to the specification for using languages in VoiceXML (for example, *en-US*). The `<encoding>` tag content is formatted according to the specification for encoding XML pages (for example, *UTF-8*).
- **invalidate_session** – This tag, if included in the XML, will prompt VXML Server to invalidate the call session it retains in memory, call the end of call class or URI (if defined), and free up the VXML Server port utilized by the call. The session is invalidated only after the method of the standard decision element is completed. This tag is rarely used and would be needed in a few circumstances where some external process takes the call away from VXML Server such as when using a CTI system to transfer the call to an agent.

Remote Execution

For remote execution of the Standard Decision Elements, the following syntax for URI is used in the source tab of General Settings:

For HTTP and RPC call:

```
remote://system/?classurl=<fully_qualified_java_class_path>
```

For example:

```
remote://system/?classurl=com.cisco.cvp.callstudio.Action.CustomAction
```

`remote://system` indicates that the configurations are fetched from the **Remote Url Settings** property tab which is application-specific.



Note If a direct remote server URI is provided, then that **IP:Port** is used and not fetched from the Remote Url Setting property tab.

For example:

```
http://<IP>:<Port>/<target_path>/?classurl=<fully_qualified_java_class_path>
```

Add Events in the Element Configuration to handle any particular or general exception gracefully.

