Standard Action Elements

Action elements are responsible for performing some action and returning an indication whether the action was a success. A pre-built, configurable action element has already defined the actions to take and only requires a configuration to modify its behaviors. Standard action elements, however, are defined by the developer and have no configuration since they represent actions specific to an application.

A standard action element, in addition to the functionality provided all components, is allowed to create and modify element data. It can also act as a flag if desired.

Java API Use

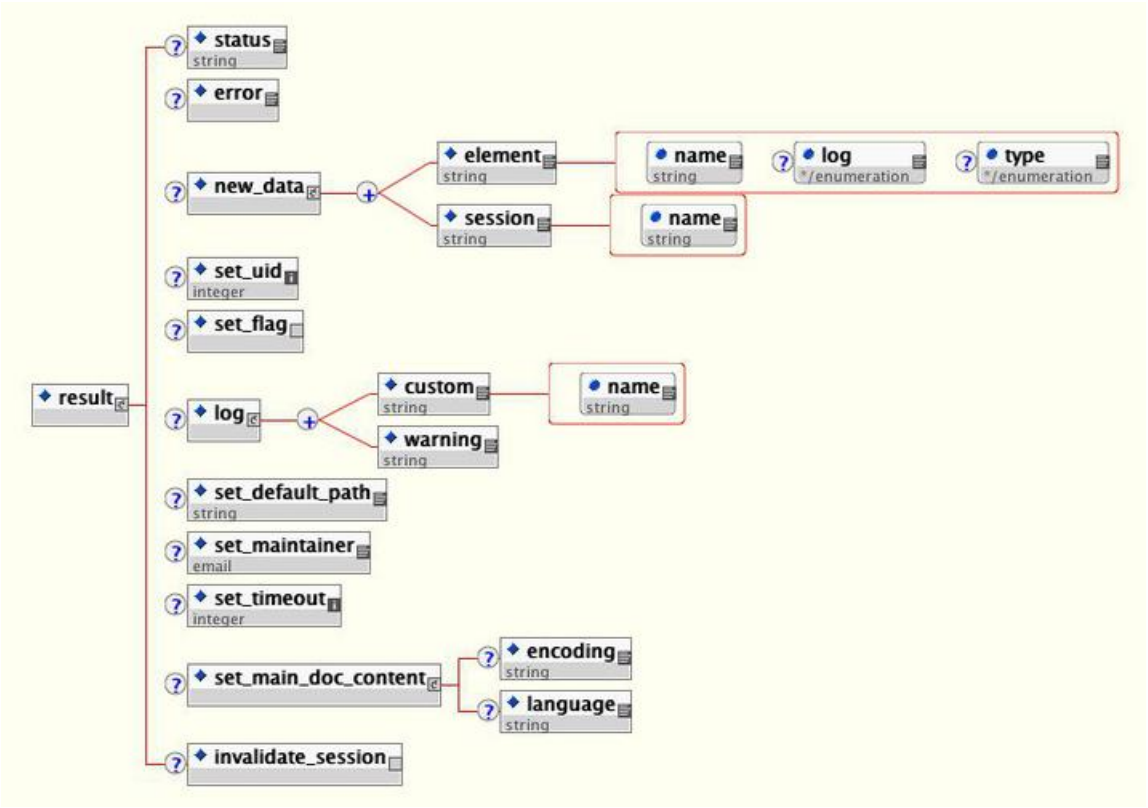
A standard action element is built in Java by extending the abstract base class `ActionElementBase` found in the `com.audium.server.voiceElement` package (this package's name is such due to backwards compatibility considerations). It contains a single abstract method named `doAction`, that acts as the method for the action element, and must be implemented by the developer. The method receives two arguments: the name of the action element (as a `String`) and an instance of `ActionElementData`. This class belongs to the Session API and is used to access session information (See Session API for more on this API). The method does not expect anything in return because all action elements have a single exit state (*done*). It is expected that should an unrecoverable error occur, an `AudiumException` is thrown.

The `ActionElementBase` class defines many methods in addition to `doAction`. These are used for configurable action elements, which also extend the class. The only method required for standard action elements is `doAction`, as it is the only abstract method in `ActionElementBase`.

XML API Use

As described in Session API, the standard *inputs* and *settings* XML documents are sent via POST to the standard action element URI. An additional parameter, called *name*, is sent containing the name of the action element. The following figure shows the DTD diagram of the XML document that must be sent in response. The DTD for the standard action element response is defined in the file `ActionResponse.dtd` found in the VXML Server `dtds` folder.

Figure 1: ActionElementsDTD



The elements in this XML document are:

- **status** – Since the XML API accesses a process that exists in context separate from VXML Server, there is no automatic way for an error that occurs during the creation of a response to be caught and handled properly by VXML Server. This tag exists to simulate that process by containing either the word *success* or a text message describing the error. When anything but *success* is returned, VXML Server throws an exception using the content of `<status>` as the error message. This way, from the perspective of VXML Server and the application logs, the result will be the same no matter whether the Java API or the XML API is used. See the description for the `<error>` tag below as there is some overlap in functionality.
- **error** – This tag reports to VXML Server that an error occurred while running the standard action. VXML Server will then throw an exception whose message is contained in the `<error>` tag. This tag acts almost exactly like the `<status>` tag and was introduced later to allow for consistency across all components. An error listed in this tag takes precedence over an error message listed in the `<status>` tag. The `<status>` tag must still be used to indicate that the standard action element was run without error by containing the word *success*.
- **new_data** – This tag holds the element and session data this standard action 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 action element act like a flag when visited. If it appears, a flag with the same name as the action element will be considered triggered and that fact will be noted in the activity log.
- **log** – This tag is used to trigger logger events when this standard action element is run. Any number of `<custom>` tags can appear, denoting the triggering of a custom event. The `name` attribute holds the name of the data, and the `<custom>` tag encapsulates the value. Any number of `<warning>` tags can appear, denoting the triggering of a warning event. 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 tab 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 application 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 action 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 Standard Action Elements, the following syntax is added 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`

where `remote://system` indicates that the configurations will be 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` will be 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.