Application End Classes

Application End Classes

Application end classes are unlike most components in that they are not related to a call session. Application end classes are instead associated with a particular application and are run when the application itself is taken down or updated. Currently, only the Java API can construct code to run when an application starts.

An application can define in its application settings any number of application end classes. VXML Server will run the classes sequentially in the order they appear in the application's settings. By conforming to this order, a developer can create an application end class that stores information that can then be referenced by subsequent end of application classes.

Unlike application start classes, an error in one application end class does not cancel the application release, once started the application will be released no matter what occurs. Should an error occur in an application end class, an error event will be thrown for any error loggers to report.

There are four situations where the application end class is run:

- The application server is shut down. VXML Server is configured to shut down all of its own operations as well as shut down each individual application by running their application end classes.
- The VXML Server web application is restarted. Most application servers provide the ability to restart just a certain web application running within it rather than restarting the entire application server. The act of restarting the web application that defines VXML Server will prompt it to initiate the application unloading process, including calling its application end classes just like an application server restart.
- An application is released after the VXML Server has started. Using the release administration scripts, an application can be released while the system is actively handling calls to other applications. VXML Server will first run all application end classes for the application and then release it from memory.
- An application is updated. The process of updating an application prompts VXML Server to create a new instance of the application in memory, while keeping the old instance in memory long enough for all existing callers to complete their calls. Once all calls have completed with the old instance, that instance's application end classes are run.

An application end class only has access to the Global API, which allows for the creation of global and application data. It does not have access to the Session API because it is not run within a call session and is associated only with an application, not a call.

The main purpose for an application end class would be to perform cleanup operations for an application. Typically this would involve closing database connections or files that were opened by an application start class or by code run within calls to the application. Application end classes have the ability to create application or global data, though the concept of creating data right before the application is released would seem pointless. The one situation where this would be useful would be to set data that a subsequent application end class could use. For example, an application start class could open up a database connection and store it in global data so that all applications deployed on VXML Server could utilize the connection. This connection would need to be closed, but if there are multiple applications with multiple application end classes, the desire would be to close the connection by the last application to be released, in case the application end classes need to use the connection. Each application start classes could increment an application count value stored in global data that the application end classes would decrement. The application end class that yielded a zero would know that it was the last application released and so close the database connection.

The application end class action is built by implementing the Unified CVP class `EndApplicationInterface` found in the `com.audium.server.proxy` package. It contains a single method named `onEndApplication` that is the method to run for the application end class. This method receives a single argument, an instance of `ApplicationEndAPI`. This class belongs to the Global API and is used to access and create application data and global data (see the User Guide for Cisco Unified CVP VXML Server and Unified Call Studio for details on application and global data). The method does not have a return value. It is expected that should an unrecoverable error occur, the application end class will throw an `AudiumException`.

Remote Execution

For remote execution of the Application End Classes, the following syntax for URI is to be used:

For HTTP and RPC call:
`remote://system/?classurl=<fully_qualified_java_class_path>`

For example:
`remote://system/?classurl=com.cisco.cvp.callstudio.Action.TestEndApplicationClass`

`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 Settings** property tab.

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