OpenDSS COM Documentation

CtrlQueue Interface

The CtrlQueue interface contains properties and methods for interacting between internal DSS control objects and control algorithms written in external programs.

DSS Control Queue

The OpenDSS program has an internal control dispatching process implemented as a queue of action requests at specified times. The Control Queue gets populated after a converged solution is achieved. Each of the control objects currently in the circuit are polled. If they determine that they need to make a change, they will push a control action onto the Control Queue using codes that are unique to each class of control object. When it comes time to execute the action, the DSS pops the appropriate control actions off the Control Queue and dispatches them to the appropriate control handler (a function called DoPendingAction).

The COM interface instantiates a proxy that provides a DoPendingAction function to the DSS and, thus imitates an internal DSS control object.

COM Interface Control Proxy

The COM interface contains a proxy for any control objects that might be implemented in some program external to the OpenDSS and driving the OpenDSS through the COM interface. The COM Control Proxy contains an Action List that contains a list of actions that have been popped off the DSS Control Queue (see Figure 1). User-written control algorithms should check this List and execute the requested action, if it is still appropriate.

The Action List dispatches control requests using a Device Handle and an Action Code. Controls implemented in external languages should be associated with a particular Device Handle for dispatching by the COM Control Proxy object. At this time, Device Handles and Action Codes are defined in the external control code, but a future version may also employ handles and codes managed by the simulator.

Typically, an external control would step through the solution one step at a time. After achieving a converged circuit solution, the control would sample the quantities of interest. It could then Push any control actions onto the DSS Control Queue and rely on the DSS to dispatch the control. The DSS would know the action request came from the COM Control Proxy and dispatch any actions back to the Proxy where they are accumulated in the Action Queue. The user-written control would then be responsible for checking the Action List, setting the Active Action (directly or by popping in sequence), and then dispatching the action code to the proper control algorithm.

All external controls should employ the DSS Control Queue for proper synchronization with other active controllers in the circuit. When the time comes to execute an action, external controls may act directly upon circuit objects through the COM interface. For example, if a user-written relay control determines that a device terminal needs to be opened, it can do so through either the text interface or the CktElement interface.

Note that accessing most of these properties and methods without an active circuit defined will result in no action.

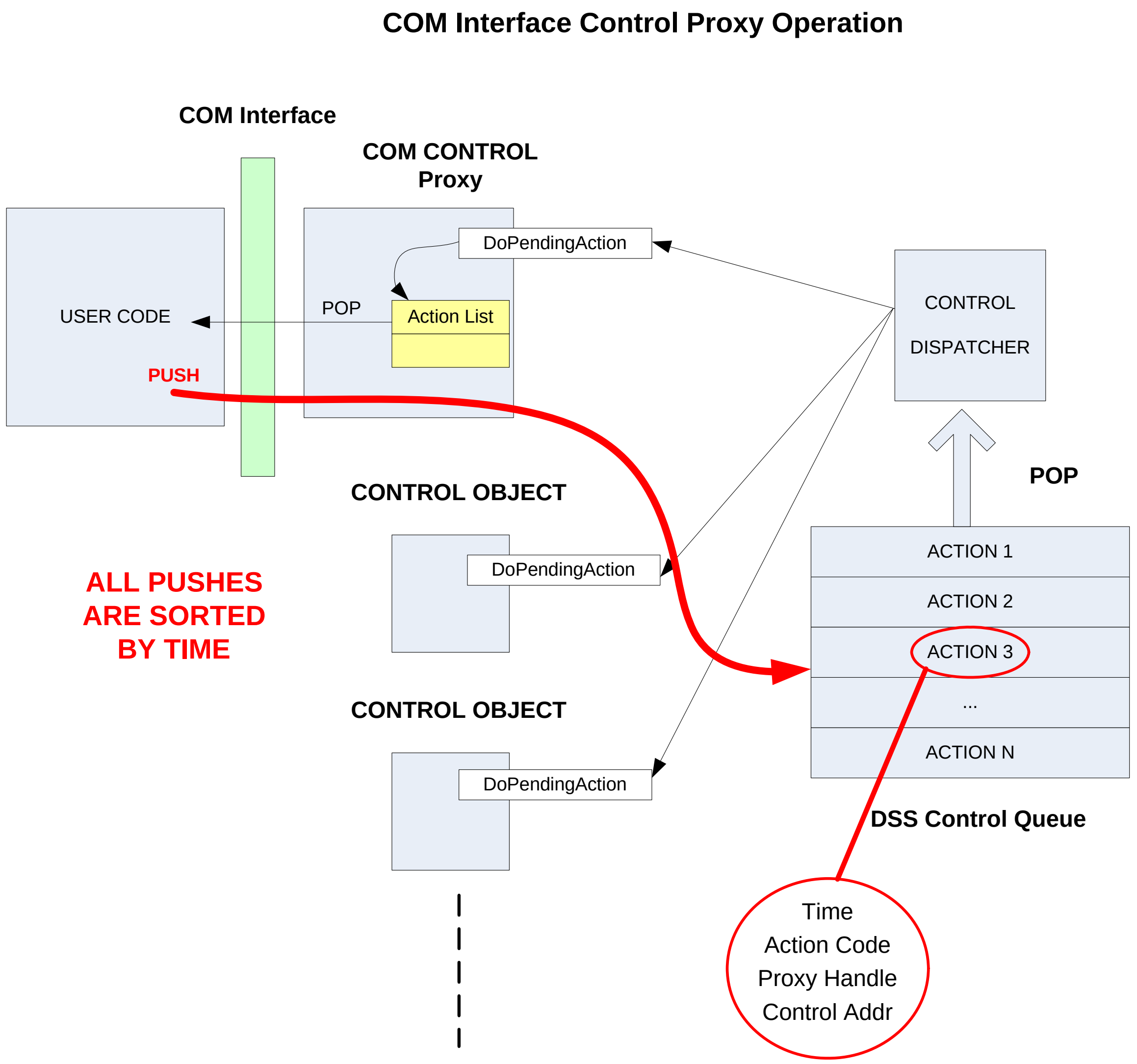

Figure 1. Illustration of how the COM Control Proxy interacts with the DSS Control Queue

Properties

NumActions:Integer (Read only)

Returns the number of actions in the COM Control Proxy Action List.

Action(Index: Integer): (Write Only)

One way to set the *Active Action*. Sets it by index value after checking to make sure that Index is within the valid range (based on NumActions).

ActionCode: Integer: (Read Only)

Action Code for the Active Action in the COM Control Proxy Action List. Use this to determine what user-defined controls are supposed to do. Can be any long integer of the programmer's choosing.

DeviceHandle:Integer (Read only)

Returns a handle to the user's control device from the Active Action.

PopAction: Integer (Read Only)

Sets the Active Action by popping the next action off the COM Control Proxy action list.

Methods

Function ClearQueue: HResult;

This clears all actions from the DSS Control Queue. The return value is always zero.

Function Delete(ActionHandle: Integer): HResult

Deletes an Action from the DSS Control Queue by the handle that is returned when the action is added. (The Push function returns the handle.) The return value is always zero.

Function Push(Hour: Integer; Seconds: Double; ActionCode: Integer; DeviceHandle: Integer): HResult;

Pushes an Action onto the DSS Control Queue. The return value is the handle to the action on the queue. Specify the time of the requested action in Hour, Seconds. Push an ActionCode that is meaningful to the user-written control object. DeviceHandle is a user-defined handle to the control object that will be used to dispatch the control action when the Action is popped off the Control Queue. The DSS will return this handle to the control proxy in the COM interface when the action is popped.

Function Show: HResult;

This function executes the procedure inside the OpenDSS to show a text file in CSV form containing the present contents of the DSS Control Queue. The return value is always zero.

When this command is complete the Result property of the Text interface contains the name of the file, which you may use for further processing. (The Result property must be accessed before doing anything else.)

Function ClearActions: HResult;

This clears all actions from the Control Proxy’s Action List (they are popped off the list). The return value is always zero.