APLNext Desktop Server Hints

Overview

The APLNext Desktop Server is designed for an APL+Win programmer to develop and test APL+Win functions exposed as Microsoft Windows web services. Its use is a precursor to the deployment of a production web service using APLNext Application or APLNext Commerce server.

APLNext web services software provides for a programmer-defined, xml-format configuration file editable by the APLNext web services Admin tool. Based upon this configuration, APL+Win ActiveX instances are spawned by APLNext web services and these instances load APL+Win workspaces with APL+Win functions which run in response to client requests submitted to the web server. Such client requests can originate from a client browser or running application on a workstation remote from the web server.

Development vs Production Environment

APLNext Desktop Server is suitable for a development environment because it runs as a Windows application under the credentials of the APL+Win programmer. This enables the APL+Win programmer to view the APL+Win ActiveX instances spawned by the APLNext and debug the APL+Win functions exposed as a web service in real time.

APLNext Desktop Server is not suitable for use in a production environment because the credentials of a browser-based client should be much more limited than those of the APL+Win programmer and configuration settings appropriate for design, testing and debugging are not appropriate in a production environment.

Configuration Settings for Debugging

The ‘visible’ configuration setting should be set to ‘1’ so that the APL+Win ActiveX instance spawned by APLNext Desktop Server will be visible to the APL+Win programmer.

The ‘debug’ configuration setting should be set to ‘1’ so that the ‘timeout’ configuration setting will be ignored, giving the APL+Win programmer sufficient time to examine the running APL+Win function and workspace environment.

For initial testing of an APL+Win function exposed as a web service, the ‘minpool’ and ‘maxpool’ configuration settings should be set to ‘1’, so that it is clear which APL+Win ActiveX instance is associated with client request being tested.

The 'minpool' value in a production environment can be increased so that when client browser requests are received by the server, any time necessary to instantiate APL+Win ActiveX sessions and load the application workspaces would be avoided.

Web Services Logs

APL+Win Desktop Server utilizes three logs:

- Exceptions in the underlying APLNext .Net C# software are appended to the Windows Control Panel > Administrative Tools > Event Viewer log. The APL+Win programmer should set the Event Viewer log parameters appropriately so that such notifications are properly captured.

- Notifications associated with the web services communication portion of the APLNext Desktop Server are appended to the ...\APLNext\APLNextDesktopServer\APLNCommSVCccyy-mm-dd.log files. The location of these log files is determined when the APLNext Desktop Server software is installed to the target machine.

- Notifications associated with a particular web service configured on the APLNext Desktop Server are appended to the …\APLNCommSVCwebSvcNameCCYY-MM-DD.log file where 'webSvcName' is the programmer-assigned name of the web service in APLNext Desktop Server configuration file. The location of these log files is programmer-assigned in the APLNext Desktop Server configuration file.

  The notifications intended for this file are 'buffered' in memory to reduce the effect on server performance. Periodically, when the buffer is full or when the server is idle, the notifications are appended to the log file. The APLNext Desktop Server Admin tool 'Action > Flush Log' can also trigger the append of buffered notifications. This is an important step before examining this log file.

Starting APLNext Desktop Server

The 'Action > Start' menu item of the APLNext Desktop Server Admin tool applies separately to the 'Web Servers' node and each of the web server nodes, so that it is possible to keep the overall APLNext Desktop Server running and individually start a particular web service among several web services configured on the target machine.

Starting the 'Web Servers' node will cause APLNext Desktop Server to spawn the APL+Win ActiveX instances specified by the value of the 'minpool' configuration setting.

If the 'maxpool' configuration setting value exceeds the 'minpool' configuration setting value, as additional requests arrive at the server, additional APL+Win ActiveX instances will be spawned by APLNext Desktop Server if all previously-spawned instances are currently busy satisfying prior requests.

Stopping the 'Web Servers' node will cause APLNext Next Desktop Server to close any APL+Win ActiveX instance that has been spawned.

It is important to stop the 'Web Servers' node before closing the APLNext Desktop Server Admin tool dialog. If this is not done, the configuration-specified port will remain bound to the associated instance of APL+Win ActiveX server spawned by APLNext Desktop Server, but that instance will not be visible or accessible to the programmer. This condition can result in an unresponsive web server once the 'maxpool' instances of APL+Win have been spawned. Checking this condition with the Windows Task Manager will reveal this condition if it is present. Use the Windows Task Manager > End Task to end the applicable APL+Win ActiveX instance to resolve this condition.

By design the APLNext Desktop Server Admin tool must remain open when testing an APL+Win function exposed as a web service, since it runs as a Windows application. The APLNext Application Server and APLNext Commerce Server Admin tools may be closed and the web server will continue to run, since they run as Windows services with separate credentials from the user of the Admin tool.

If a web service is unresponsive in testing

If a web service is unresponsive, check that there are no 'inaccessable' instances of APL+Win as an ActiveX server [described above] and that both the 'Web Server' and applicable web server nodes are 'Started' using the APLNext Desktop Server Admin tool.

Web Server Port

The APLNext web services configuration requires a programmer-specified TCP port number for each web service configured in a web server. For security reasons most ports are closed by default in Windows Firewall, so it is necessary to open the port for incoming and outgoing traffic using the Advanced dialog of the Windows Firewall software. Other anti-malware software on the target machine may also affect the availability of a port.