APLNext Application Server – Simple Sample

Contents

Overview

The APL2000 Web Services Forum article What Is APLNext WebServices is essential reading before continuing with this sample.

The workflow outlined in this document will illustrate exposing an APL+Win function as a web service:

- A web service supported by APLNext Application Server or APLNext Desktop Server
- An html-based client GUI to make a request to the web service and receive the web service response
- An APL+Win workspace and function which will provide the algorithm and response for the web service

Create the Client Request GUI

Generally a web site has a GUI and many tools are available for this purpose, but underlying these tools is the basic programming language of the web which is html. This sample uses elementary html source code to define the web site GUI. The purpose of a web site GUI is to create a human interface to the underlying web service.

The underlying web service provides the algorithms and other functionality to receive client request, process that request and respond to the client appropriately. Web services are remote from the client to protect the web service and increase its deployment potential.

Some web services require no GUI component because they are going to be accessed by other servers or applications. APLNext Application Server includes the APLNext WebTransfer component to enable accessing any web service, including those supported by APLNext Application Server or APLNext Desktop Server as an ActiveX or .Net assembly.

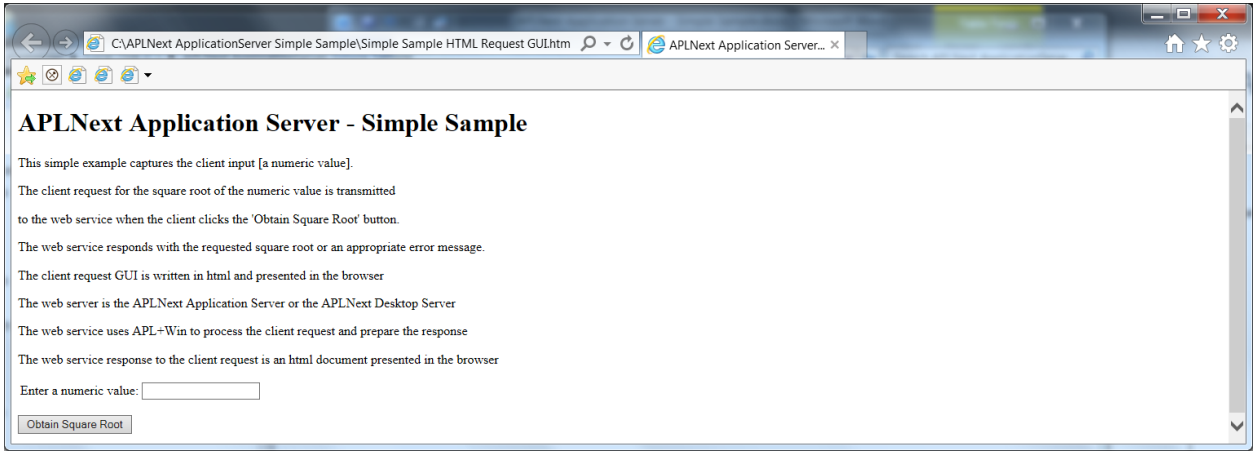
- Using the Microsoft Windows Explorer create a folder 'c:\ APLNext ApplicationServer Simple Sample\'
- Using Microsoft Windows Explorer create a new text file called 'Simple Sample HTML Request GUI.htm' in the folder. Be sure to use the file extension '.htm' and not '.txt'.
- Open the file using Microsoft Notepad
- Copy the following html program source code and paste it into this file.
- Save the file being sure to preserve the file extension '.htm'.

```
<!DOCTYPE html>
<html>
	<head>
		<title>APLNext Application Server - Simple Sample Request GUI</title>
	</head>
	<body>
		<h1>APLNext Application Server - Simple Sample</h1>
		<p>This simple example captures the client input [a numeric value].</p>
		<p>The client request for the square root of the numeric value is transmitted</p>
		<p>to the web service when the client clicks the 'Obtain Square Root' button.</p>
			<p>The web service responds with the requested square root or an appropriate error message.</p>
		<p></p>
		<p>The client request GUI is written in html and presented in the browser</p>
		<p>The web server is the APLNext Application Server or the APLNext Desktop Server</p>
```

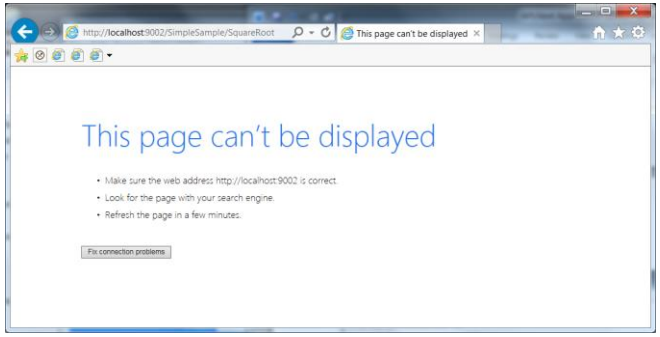
```
                <p>The web service uses APL+Win to process the client request and prepare the response</p>
                <p>The web service response to the client request is an html document presented in the browser</p>
                <p></p>
                <p></p>
                <form method="POST" action="http://localhost:9002/SimpleSample/SquareRoot">
                        <TABLE>
                                <TR>
                                        <TD>Enter a numeric value: </TD>
                                         <TD><input type="text" name="NumericValue"></TD>
                                </TR>

                        </TABLE>
                        <p><input type="submit" value="Obtain Square Root" name="Square_Root"></p>
                </form>
        </body>
</html>
```

Test the client request GUI by double [left] clicking the file in Microsoft Windows Explorer. The client request GUI should be presented in the browser as:



Even though APLNext Application Server is not yet installed or started and the Simple Sample web service not started, a value may be entered in the client request GUI and the 'Obtain Square Root' button clicked, but a standard Internet error message will be received. The format of the message will depend on the client-selected browser settings:



Create the APL+Win workspace and function

APL+Win is a superb tool to implement the algorithms of a web service, access/update data stores accessible to the server and prepare information for the response to the client. In this example the algorithm is trivial, but it illustrates how to format the APL+Win function for use by a web service. Note that the APL+Win function has an explicit argument [which is optional] and an explicit result [which is required]. In a production application system, the top-level APL+Win function since it is directly exposed as a web service will need this structure, but subordinate APL+Win functions not directly exposed as a web service can use any valid APL+Win structure.

Since a web service client cannot directly access or view the server, the APL+Win function should incorporate exception handling to provide the client with appropriate information even if the client-provided source information is not valid.

- Start a developer session of APL+Win
- Create the APL+Win function 'SquareRoot' following the source code below.

```
Z←SQUAREROOT X
:TRY *
X←'Square root of ',(⍕X),' is: ',⍕X*0.5
⍝↑Application algorithms/business rules go here
:CATCHALL
```

```
      X←'Input Error: ',⎕em
      ⍝↑Robust exception handling goes here
      :FINALLY
      Z←'<!DOCTYPE html>'
      Z←Z,'<html>'
      Z←Z,'<head>'
      Z←Z,'<title>APLNext Application Server - Simple Sample Response GUI</title>'
      Z←Z,'</head>'
      Z←Z,'<body>'
      Z←Z,'<h1>APLNext Application Server - Simple Sample Response</h1>'
      Z←Z,'<p></p>'
      Z←Z,'<p>',X,'</p>'
      Z←Z,'</body>'
      Z←Z,'</html>'
      ⍝↑Response formatted in html
      :ENDTRY
```

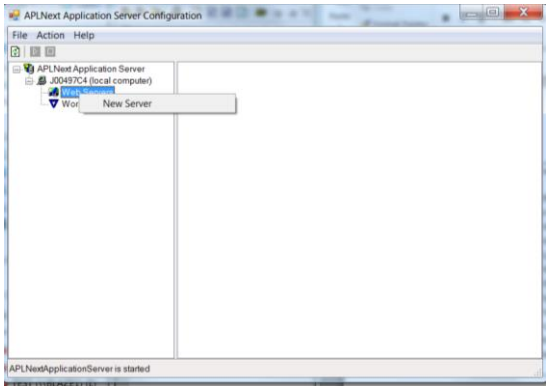
- Save the workspace as 'C:\APLNEXT APPLICATIONSERVER SIMPLE SAMPLE\SIMPLESAMPLE' to the folder

Install the APLNext Application Server or APLNext Desktop Server

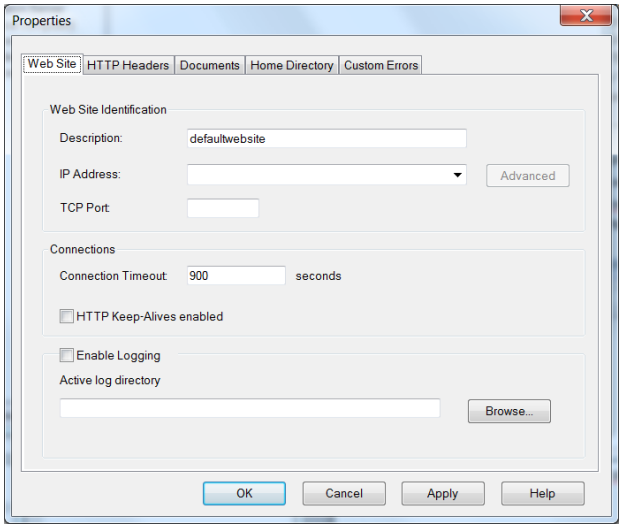
Refer to the APL2000 Web Service Forum installation documentation. A demonstration and evaluation version of APLNext Application Server is available as an APL2000 Web Services Forum download.

Configure the APLNext server software

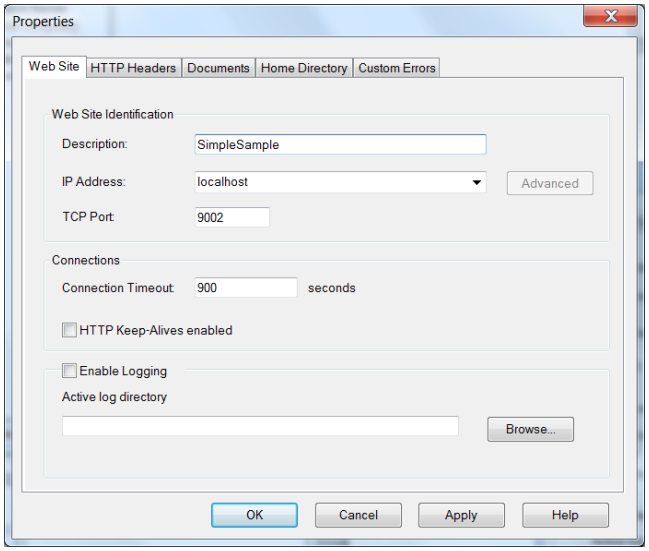
- Start the APLNext Application Server or APLNext Desktop Server administration tool. These should be run as administrator on the target workstation.
- Right click the Web Servers node and select the context menu 'New Server' option to create a new web service:



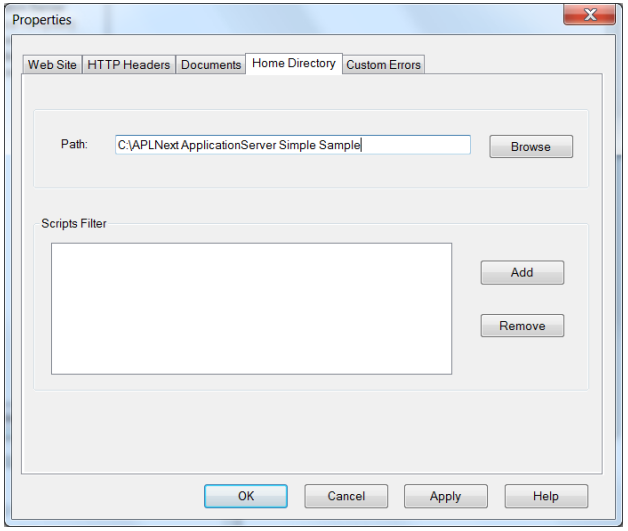
- Right click the ‘defaultwebsite’ and select the context menu ‘Properties’ option to display the ‘Properties’ dialog:



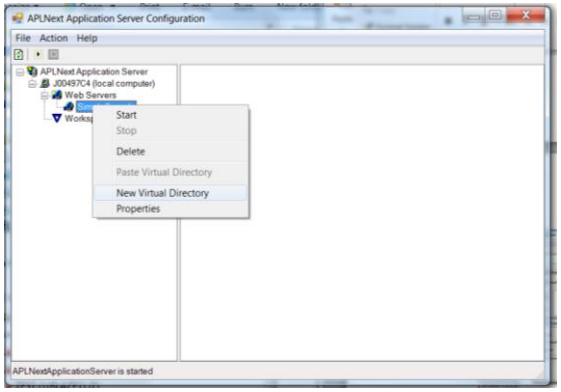
- On the ‘Web Site’ tab:
- Modify the Description field to: ‘SimpleSample’
- Modify the IP Address to ‘localhost’ to test this sample using the current workstation as the machine to support the web server. In a production application system the web server will have an IP address which is publicly exposed to the Internet, or intranet.
- Modify the TCP Port to ‘9002’. Any positive port number may be used, however by convention many port numbers have traditionally been used on the Internet for specific purposes.



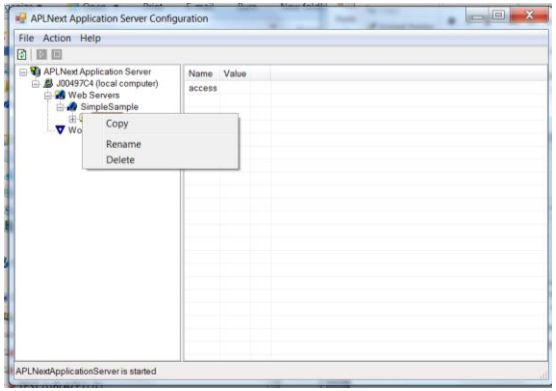
- On the 'Home Directory' tab modify the Path field to 'C:\APLNext ApplicationServer Simple Sample'

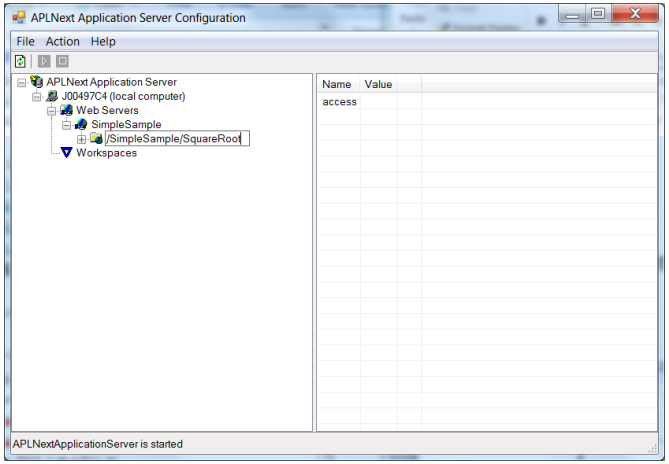


- Click the 'OK' button to return to the main form of the APLNext Application Server or APLNext Desktop Server administration tool.
- Right click the 'SimpleSample' web server node and use the context menu 'New Virtual Directory' option to create a new virtual directory:

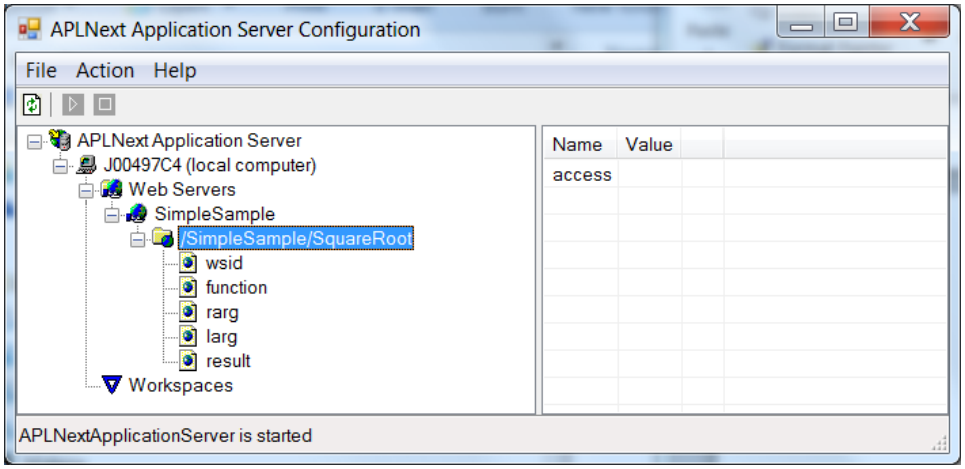


- Open the tree to the '\default' virtual directory
- Right click the '\default\ virtual directory and use the context menu 'Rename' option to rename the new virtual directory to '/SimpleSample/SquareRoot'. The virtual directory name will become part of the url for the web service. That url will be used by the client to access the web service, e.g. send input and a request to the web service and receive a response from the web service.

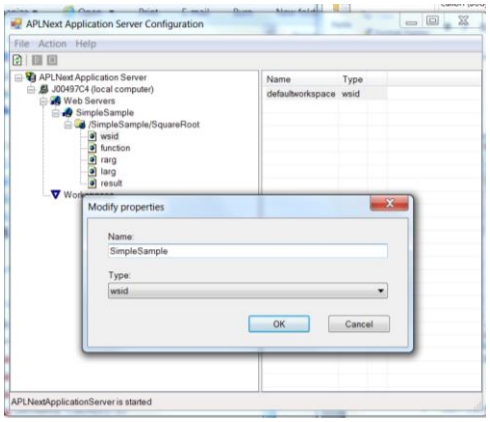




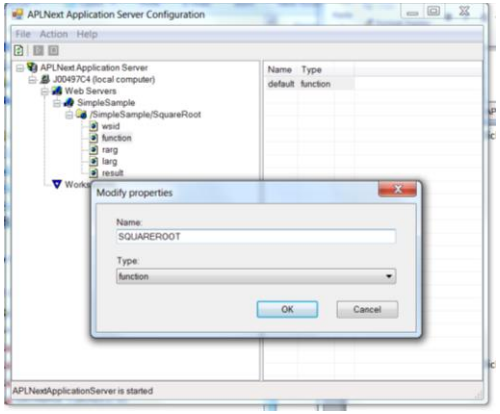
- Open the '\SimpleSample\SquareRoot' node to expose the configuration detail.



  - Left click the 'wsid' node
    - Right click the 'defaultworkspace' field and use the context menu 'Modify' option to rename it to 'SimpleSample' and click the 'OK' button. The 'wsid' text is not case sensitive, but must be selected as a workspace named in the 'Workspaces' node of the web services configuration.

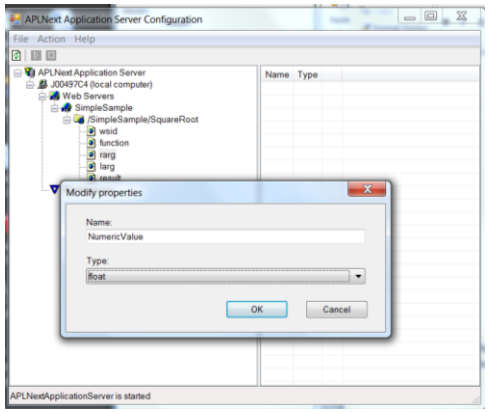


- Left click the 'function' node
  - Right click the 'default function' field and use the context menu 'Modify' option to rename it to 'SQUAREROOT', which is case sensitive, and click the 'OK' button. The function name must exactly match the name of the APL+Win function which will be associated with the web service. That function name should avoid APL+Win glyphs which are not easily used in a url.

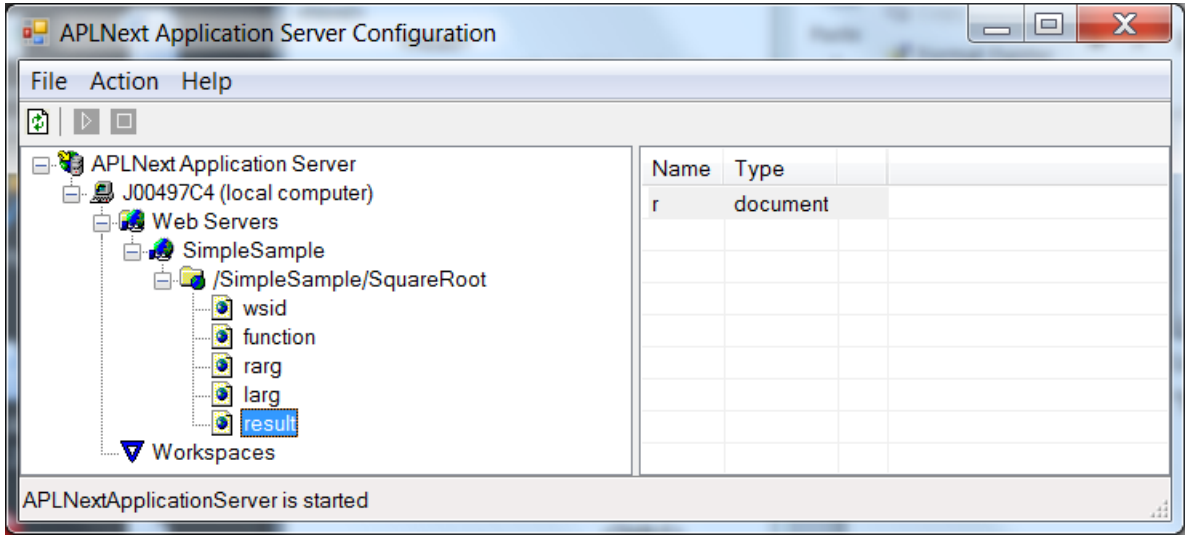


- Right click the 'rarg' node and use the context menu 'New Value' option to define it as 'Float' with name 'NumericValue' and click the 'OK' button. It is possible to define multiple arguments to an APL+Win function which is exposed as a web service. When multiple arguments are defined, APLNext Application Server provides them to the APL+Win function as an array. In this sample the client request GUI is an html-format document. When that document is submitted to the web server, the user input will be

embedded within it in association with the html name for the input value. The APLNext Application server will extract that input and assign it to the argument of the APL+Win function that is exposed as a web service.

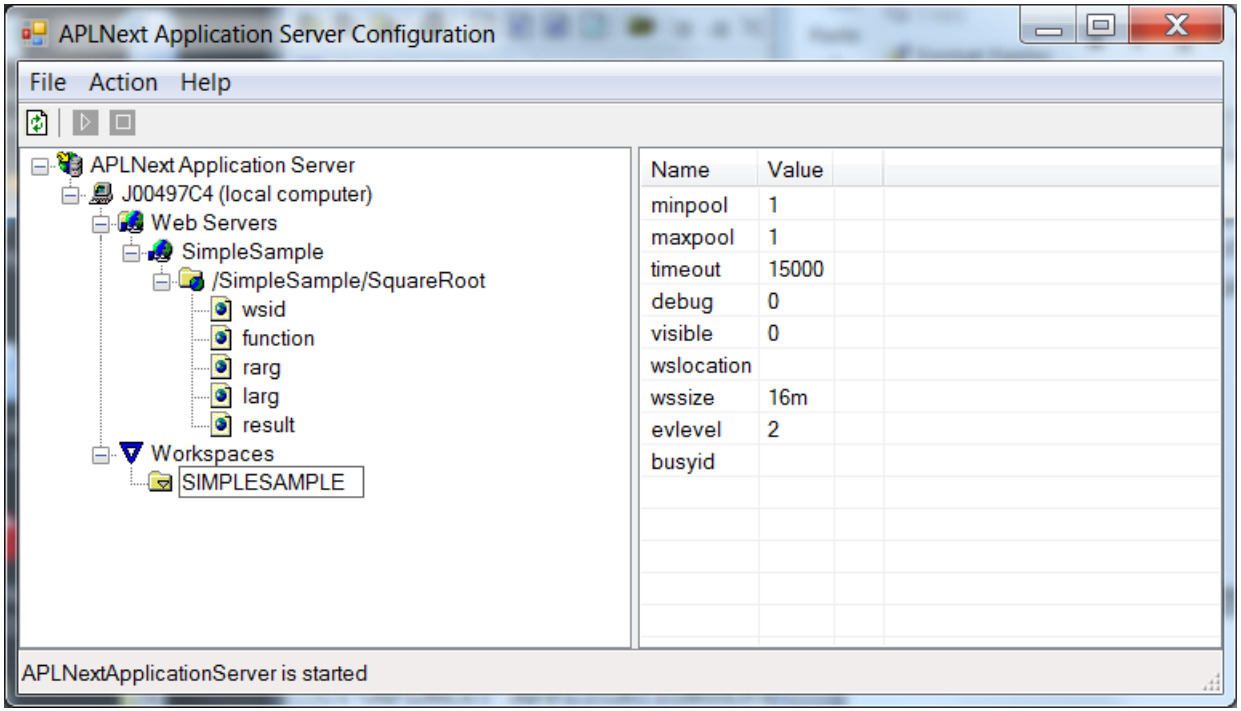


- Left click the 'result' node to observe that by default the result is a 'document' with name 'r'. Since the response of the APL+Win 'SQUAREROOT' function is an html-format document this result type is appropriate. It is possible to define additional result values of an APL+Win function exposed as a web service. In that case the APL+Win function result must provide all the defined arguments values in an array result.

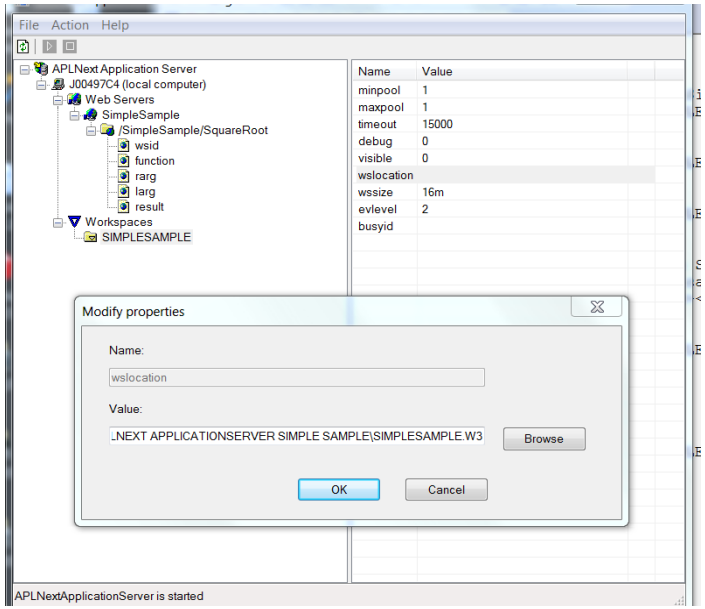


Additional argument and result data types are supported by APLNext Application Server and APLNext Desktop Server.

- Right click the 'Workspaces' node and use the context menu 'New Workspace' option to create a new workspace. The 'Workspaces' node defines the workspace names used by web services. This workspace names used in APLNext Application Server and APLNext Desktop Server configurations do not have to match the physical name of the associated APL+Win workspaces.
- Right click the 'defaultworkspace' node and use the context menu 'Modify' option to rename it to 'SIMPLESAMPLE'. The name selected here is used by the web services when associating an APL+Win function with a virtual path.

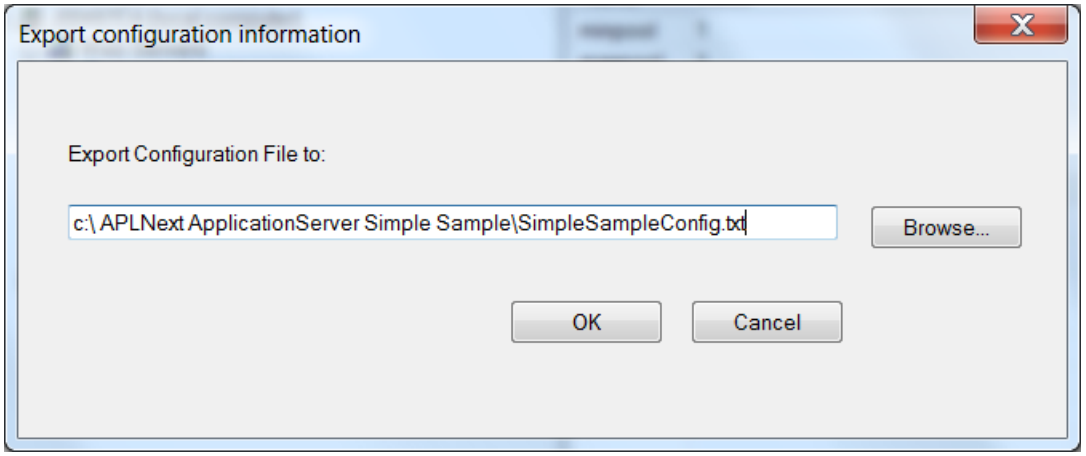


- Right click the 'wslocation' field and use the context menu 'Modify' to set the address to 'C:\APLNEXT APPLICATIONSERVER SIMPLE SAMPLE\SIMPLESAMPLE.w3'. The workspace name used in the APLNext Application Server and APLNext Desktop Server configuration must be associated, via the 'wslocation' field with a physical APL+Win workspace containing the APL+Win function which will be exposed as a web service.



Verify the Web Service Configuration

- Export the new configuration using the 'Action > Export' menu option to the file 'C:\APLNext ApplicationServer Simple Sample\SimpleSampleConfig.txt'



- Use Windows NotePad to open the configuration file 'C:\APLNext ApplicationServer Simple Sample\SimpleSampleConfig.txt' and verify that it is the same as:

```
<?xml version="1.0"?>
<config>
 <servers>
  <server id="simplesample">
   <host>localhost</host>
   <port>9002</port>
   <publichttpdir>C:\APLNext ApplicationServer Simple Sample</publichttpdir>
   <state>started</state>
   <enable-default-file>False</enable-default-file>
   <connection-timeout>900</connection-timeout>
   <http-keep-alives>False</http-keep-alives>
   <enable-logging>False</enable-logging>
   <logfile-directory />
   <enable-content-expiration>False</enable-content-expiration>
   <expire-content>after;1;Hour(s)</expire-content>
   <custom-error name="400">Default</custom-error>
   <custom-error name="403">Default</custom-error>
   <custom-error name="404">Default</custom-error>
   <custom-error name="500">Default</custom-error>
```

```
    <custom-error name="501">Default</custom-error>
    <custom-error name="503">Default</custom-error>
    <custom-error name="504">Default</custom-error>
    <virtualpath name="/simplesample/squareroot">
     <access />
     <wsid>SimpleSample</wsid>
     <function>SQUAREROOT</function>
     <rarg type="float">NumericValue</rarg>
     <result type="document">r</result>
    </virtualpath>
   </server>
  </servers>
  <workspaces>
   <workspace id="SIMPLESAMPLE">
    <minpool>1</minpool>
    <maxpool>1</maxpool>
    <timeout>15000</timeout>
    <debug>0</debug>
    <visible>0</visible>
    <wslocation>C:\APLNEXT APPLICATIONSERVER SIMPLE SAMPLE\SIMPLESAMPLE.W3</wslocation>
    <wssize>16m</wssize>
    <evlevel>2</evlevel>
    <busyid />
   </workspace>
  </workspaces>
</config>
```

An exported APLNext Application Server or APLNext Desktop Server configuration can be imported to the software installed on other server hardware, making deployment of multiple servers easy.

Only a few of the configuration options have been discussed in this document. Refer to the APL2000 Web Services Forum for additional configuration documentation.

Start the APLNext server software

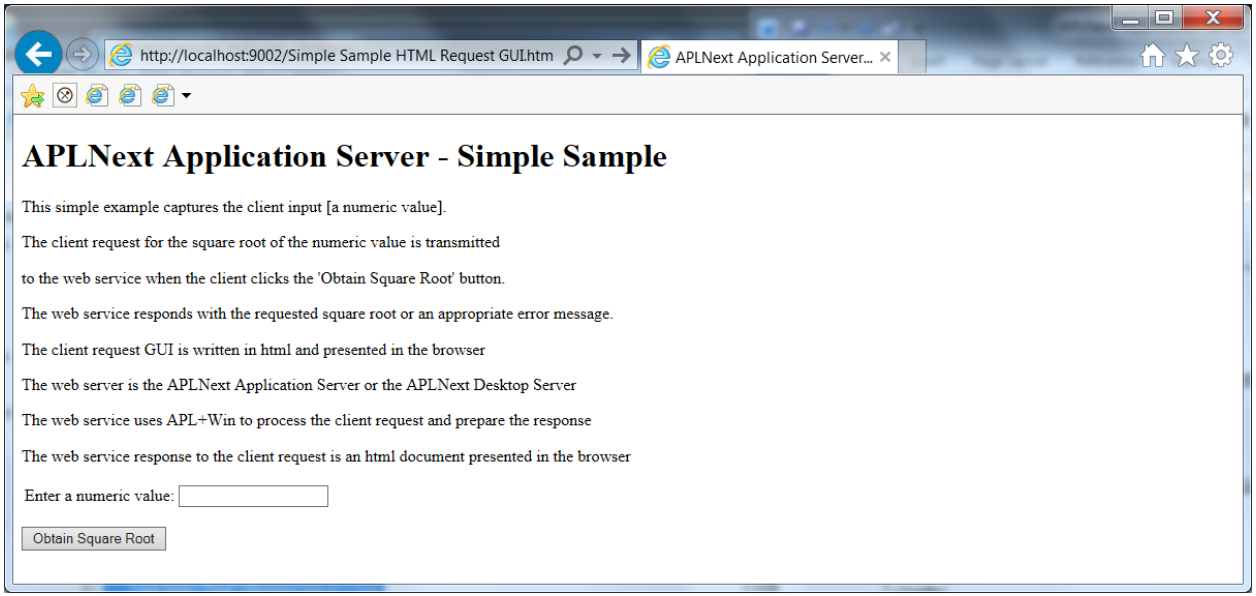
- Left click the 'Web Servers node and use the 'Action' menu item to start it if not already started.

Start the SimpleSample web service

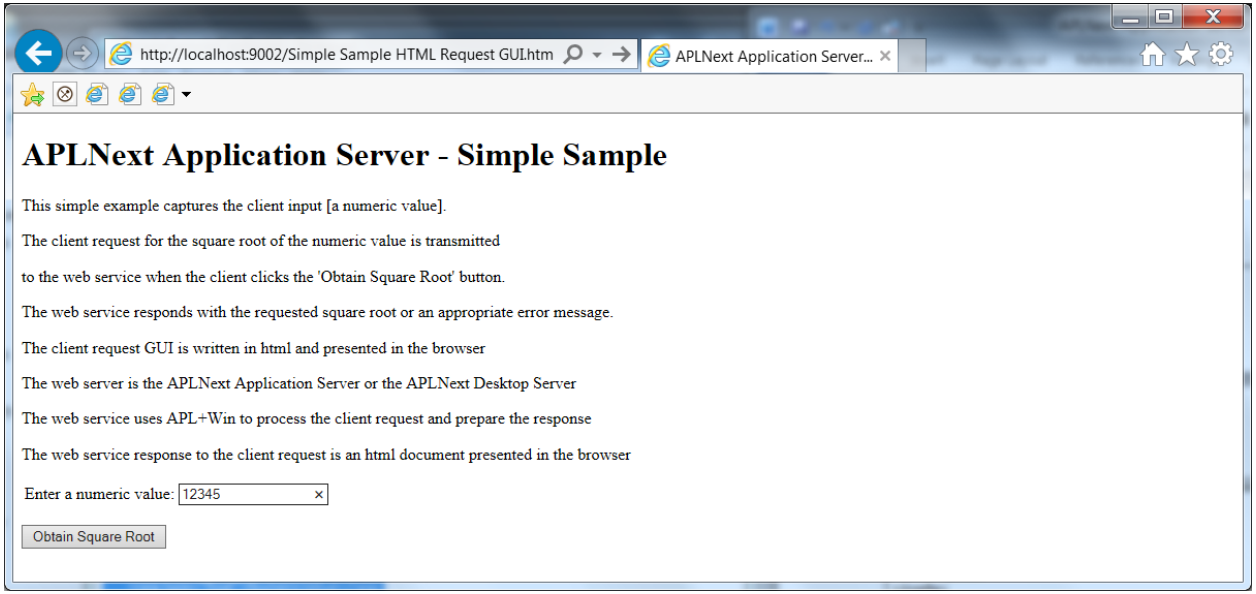
- Left click the 'simplesample' node and use the 'Action' menu item to start it if not already started.

Use the Client Request GUI

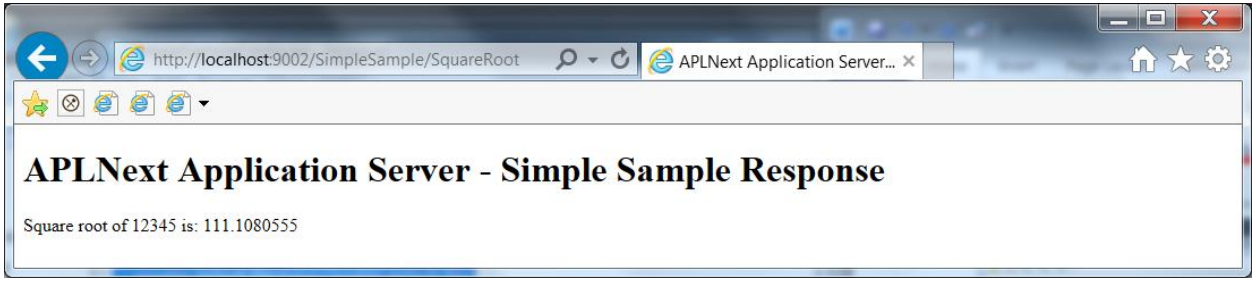
- Point the browser at the SimpleSample url: 'http://localhost:9002/Simple Sample HTML Request GUI.htm'. A universal resource locator [URL] is not case sensitive.



- Enter the desired numeric value



- Click the 'Obtain Square Root' button to receive the receive the response from the 'http://localhost:9002/SimpleSample.SquareRoot' web service



- Again point the browser at the SimpleSample url: 'http://localhost:9002/Simple Sample HTML Request GUI.htm', but enter a negative numeric value and click the 'Obtain Square Root' to receive the response. In a production environment, the response to the client in case of an exception will probably be more detailed and friendly.

Summary

The resulting sample application is highly simplified to illustrate the main concepts. APLNext Application Server and APLNext Desktop Server provide numerous other options and features which are beyond the scope of this document. Web services created using APL+Win and APLNext Application Server comply with Internet standards so they can be used from any client environment with access to the web. Advanced options, such as exporting a 'wsdl' definition, timeout and debug, are also possible. The APLNext Application Server supports multi-threading, load balancing, scaling for user load and state-less and state-full modes.

The APL2000 Web Services Forum contains significant additional documentation for the APLNext Application Server. Other samples are included with the installation of the product.

Contact sales@apl2000.com for more information about licensing APLNext Application Server.