APL+Win as an APLNext Application Server Client
Using APLNext WebTransfer

Overview

A local installation of APL+Win can be an effective and powerful client of a remote APLNext Application Server application. Such a system configuration:

- Can protect proprietary application code by putting it on the server-side
- Make it easy to transition from a fully-standalone application to a client-side GUI and server-side application service, creating a multi-tiers application architecture for increased flexibility
- Reduce the size of the installed portion of the application system to just the GUI portion
- Centralize the maintenance of the business rules and algorithms of the application system

Summary Application System Configuration

APLNext Application Server supports the business rules, algorithms and possibly the data persistence of the application. This portion of the application system remains based on APL+Win, but the application functions and workspaces are maintained on the server-side

Locally-installed APL+Win support the client-side GUI by collecting the required input data, e.g. using an APL+Win □wi form, submitting a client request to the server and receiving the server response using APLNext WebTransfer and presenting the response to the client, e.g. using an APL+Win □wi form.

This example illustrates the use of binary and SOAP serialization to send to the server and receive from the server any APL+Win data value. Using this technology an entire client-provided application data set can be efficiently transmitted to the server in a single operation even if the APL+Win data item is heterogenous, nested, etc.

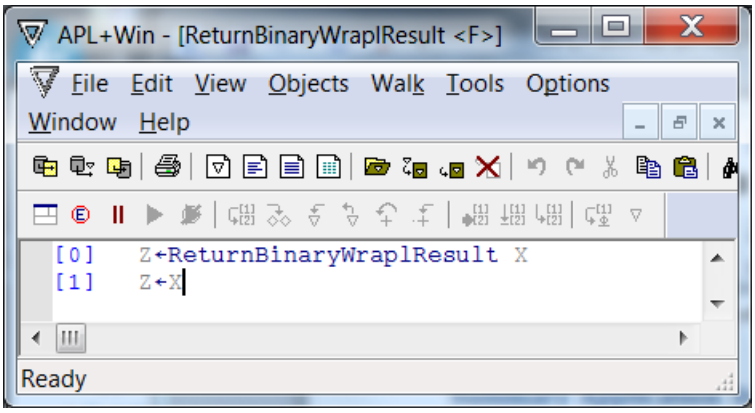
APLNext Application Server performs the programmer-selected serialization when transmitting data between the client and the server. The serialization process is analogous to the proprietary APL+Win wrapl serialization, however APLNext Application Server uses industry-standard serialization technology included in the Microsoft .Net Framework.

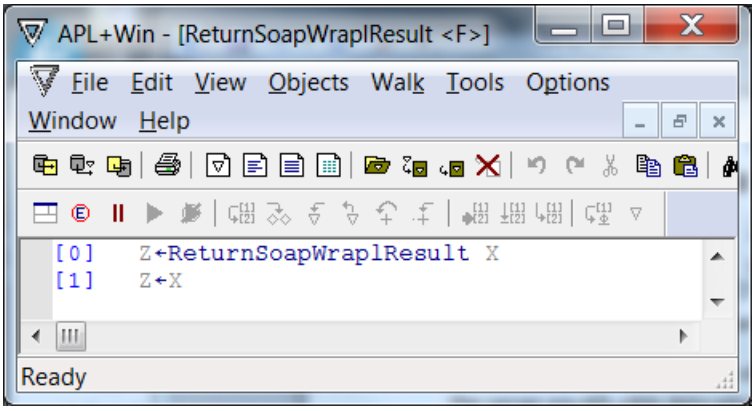
The client-side GUI can create complex data sets and these can be easily transmitted to the server and used as input to a processing request made to the server. The server-side processing can create a complex response and this data can be easily transmitted back to client.

This example includes the APL+Win 'returnresult.w3' workspace containing the client- and server-side APL+Win function. This workspace and the 'TestSerializationConfig.txt' APLNext Application Server configuration file should be placed in the 'C:\APLNEXT APPLICATION SERVER TEST SERIALIZATION\' folder on the target machine.

Server-side APL+Win

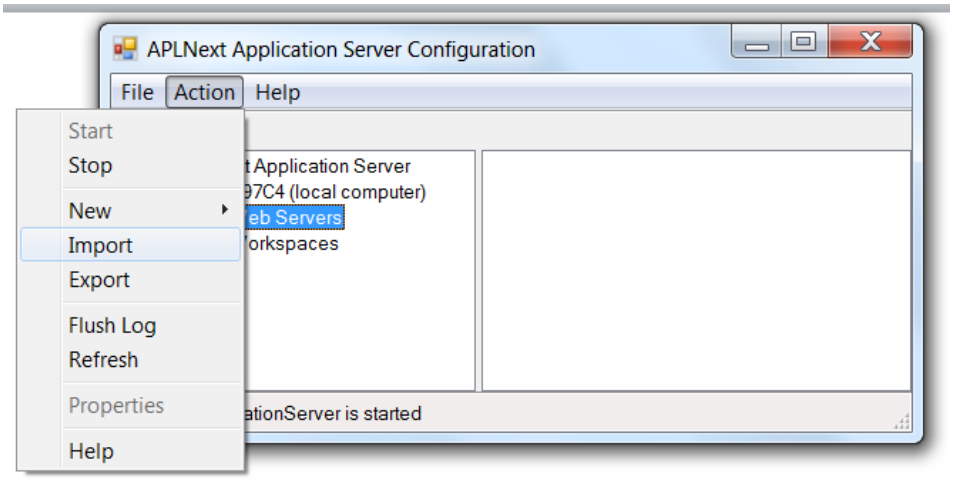
On the server side this example is very simple. The server-side APL+Win functions create a response which is the value of the data provided by the client and transmitted to the server. In a production application system, the server-side APL+Win would support the application system's business rules and algorithms.

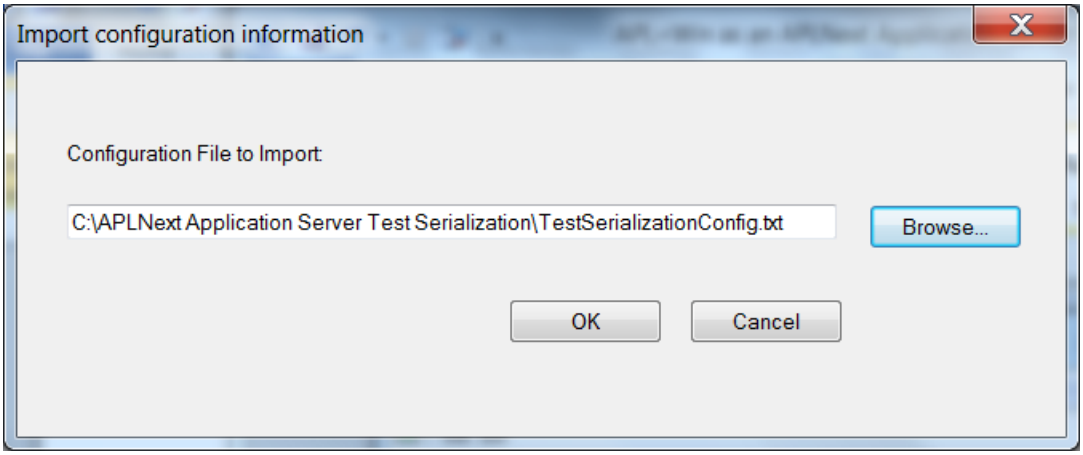




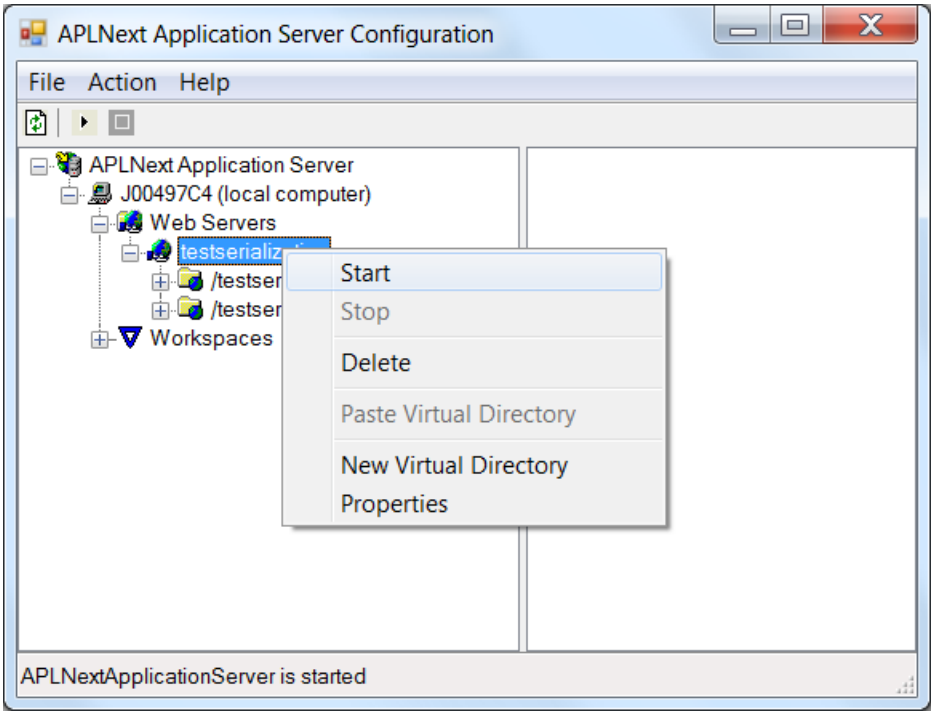
APLNext Application Server Configuration

This example includes a pre-defined configuration which can be imported using the APLNext Application Server Administration tool.

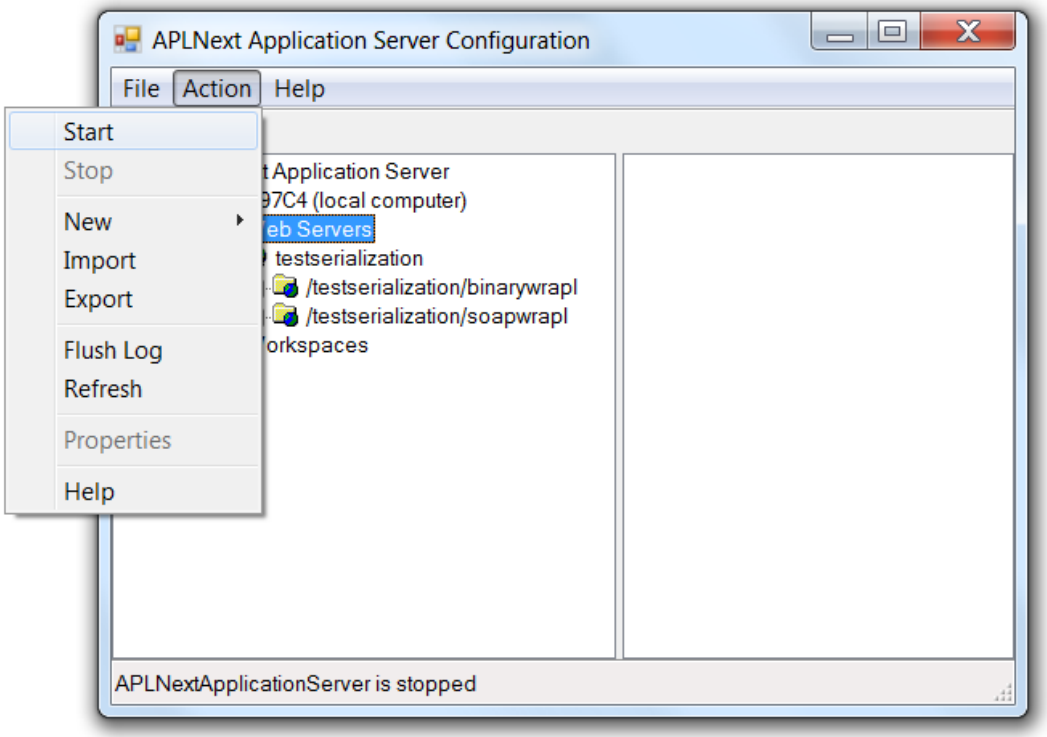




After importing the example configuration, start the 'testserialization' web server:



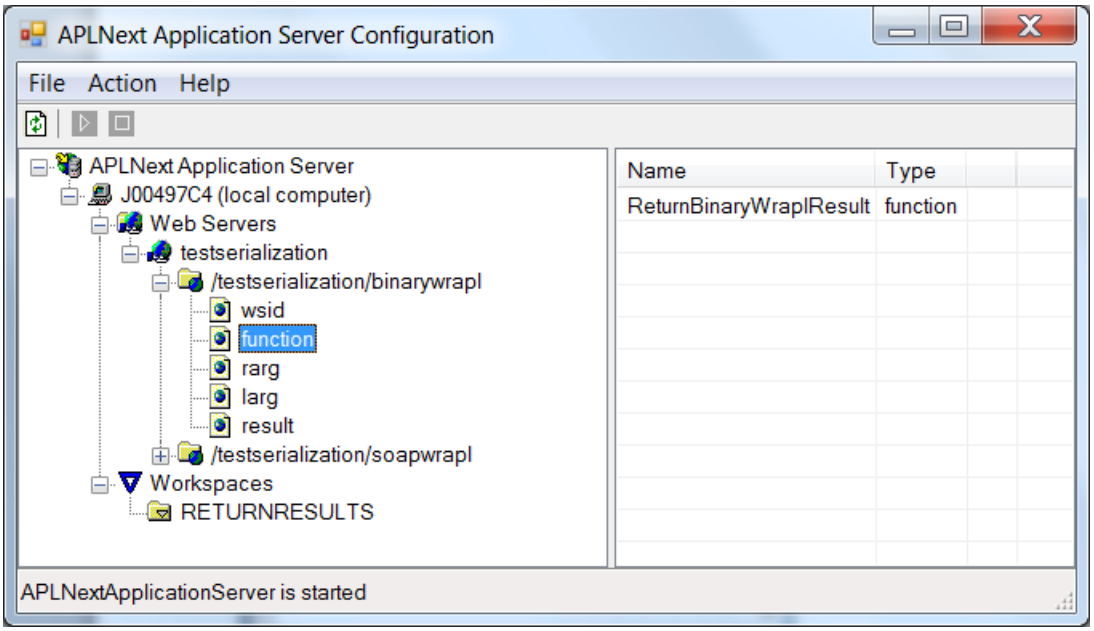
If necessary start the APLNext Application Server Windows service:

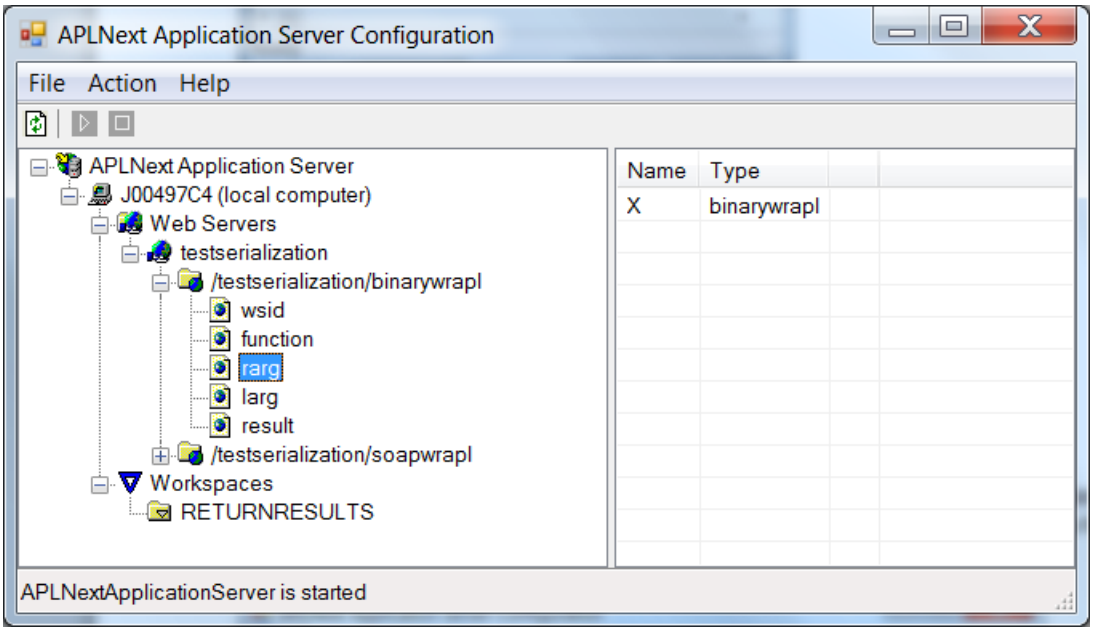


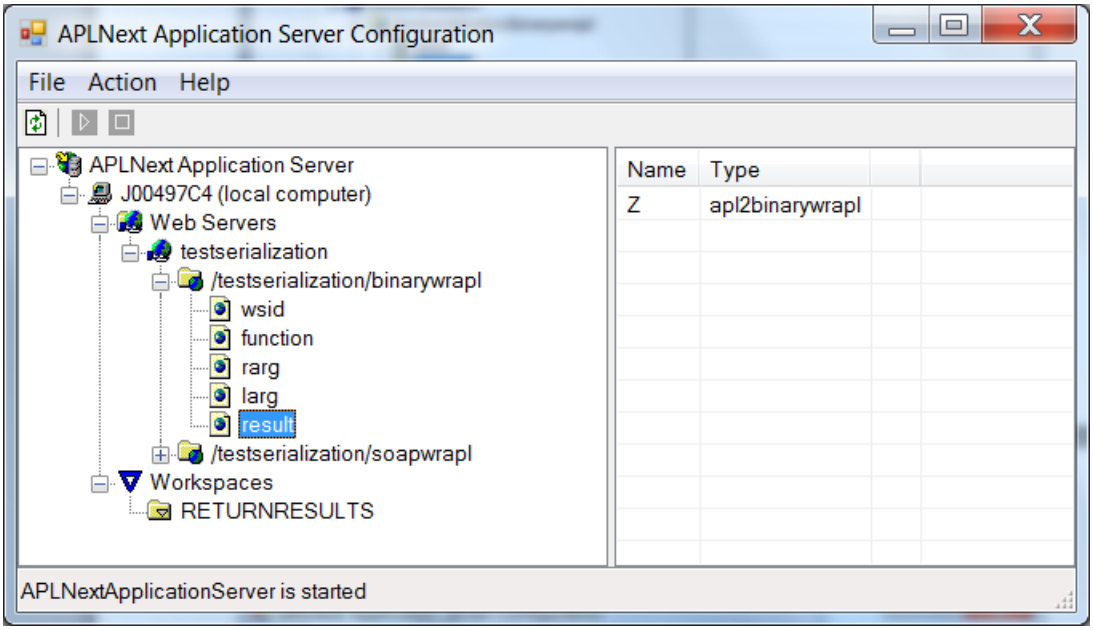
The key elements of this configuration are the specified data types for argument and result of the server-side APL+Win functions.

For example for the server-side ReturnBinaryWraplResult function the data type for the right [and only argument] is specified as ‘binarywrapl’. This indicates that the data provided to the server is assumed to be a binary serialization and APLNext Application Server will de-serialize this data and provide it to the APL+Win server-side function.

The data type for the result of the ‘ReturnBinaryWraplResult’ is ‘apl2binarywrapl’ which indicates that the result of the APL+Win server-side function will be binary serialized by APLNext Application Server and transmitted back to the client.



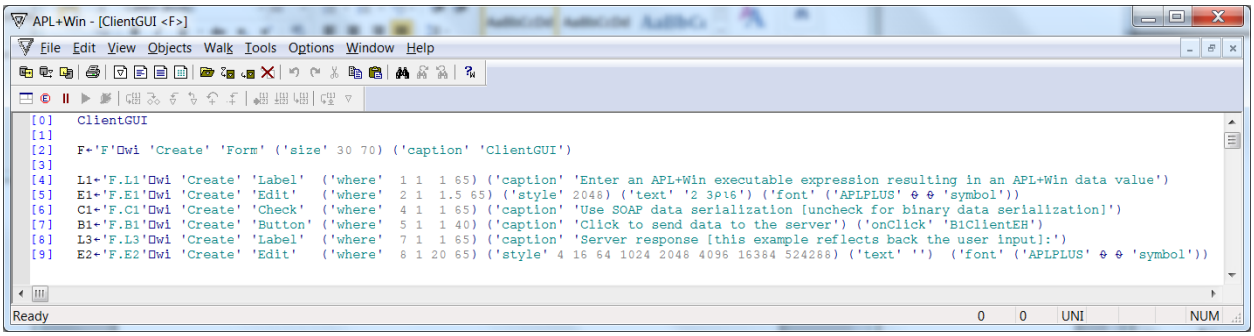




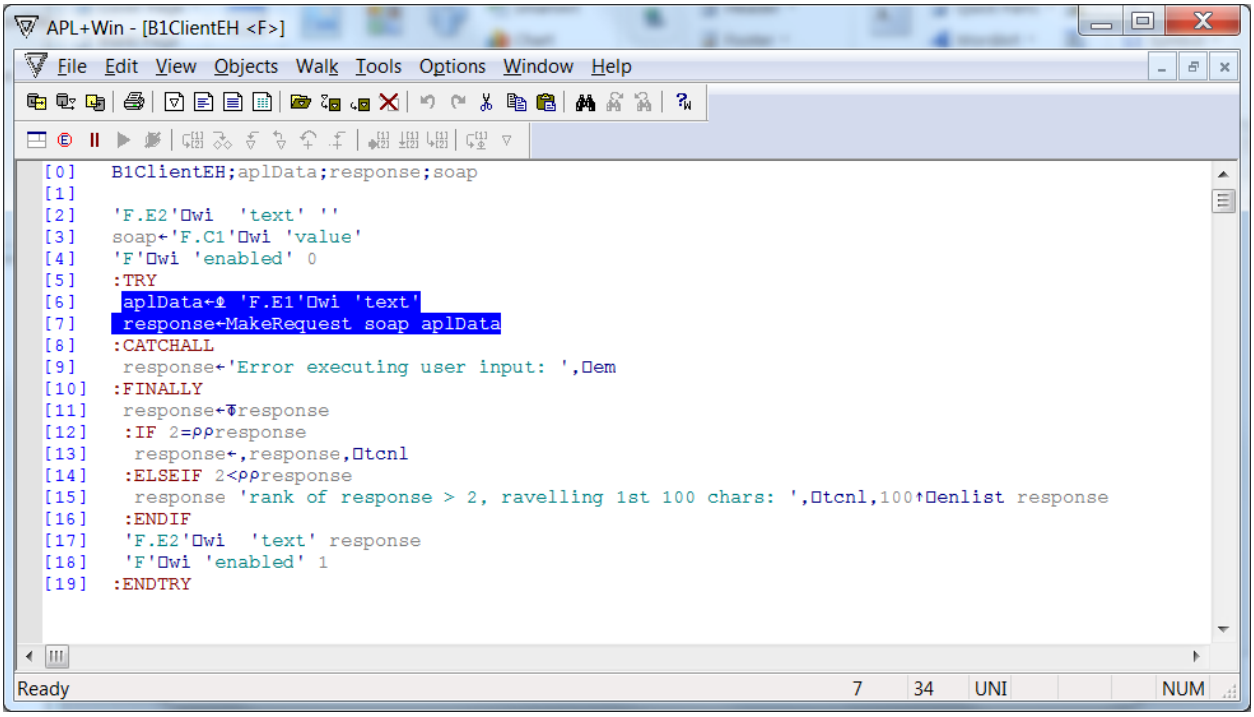
Client-side APL+Win

For simplicity in this example the client-side workspace is the same workspace as the server-side workspace.

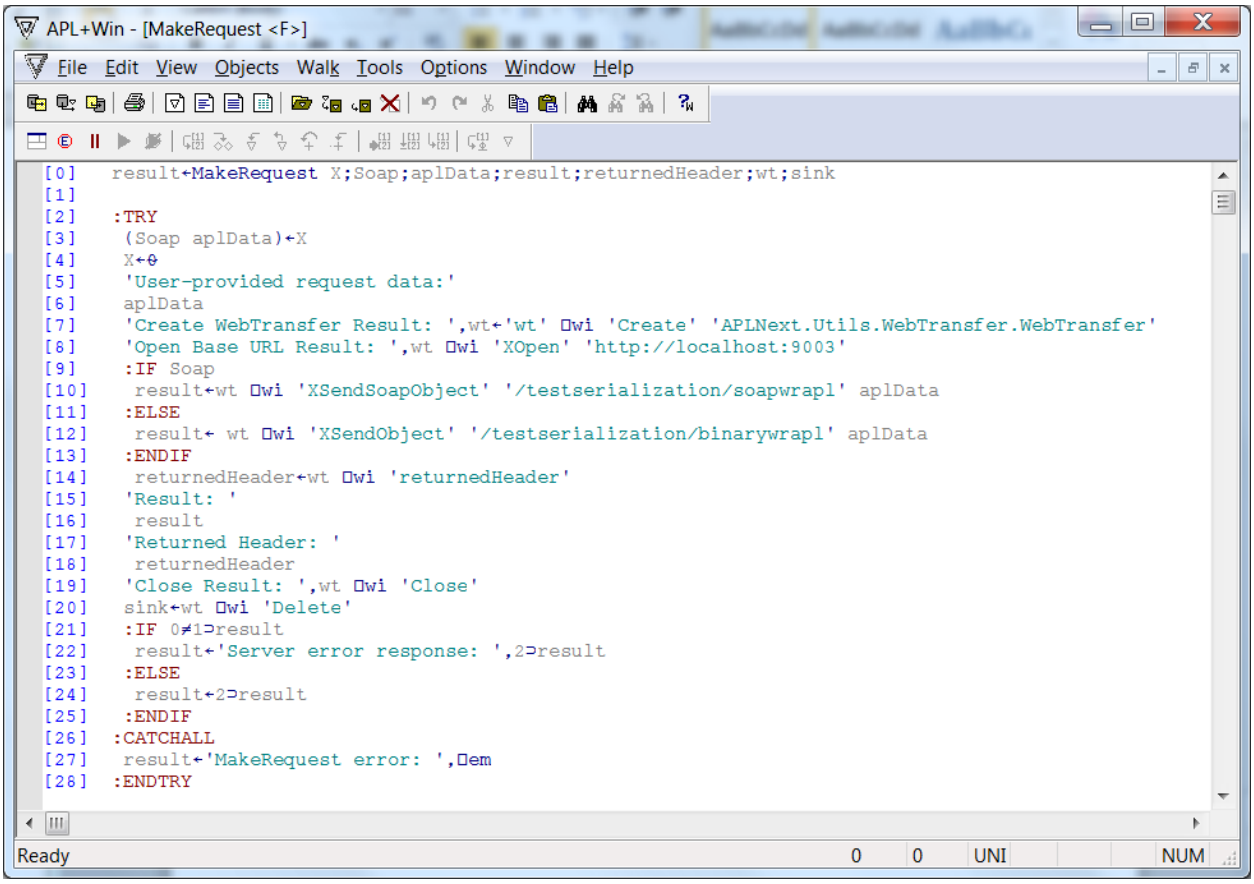
The 'ClientGUI' APL+Win function creates and displays a simple APL+Win ⎕wi-based graphical user interface for this example:



The 'B1ClickEH' APL+Win client-side button-click event handler function captures the user input, calls the 'MakeRequest' APL+Win function and displays the server response in the GUI.



The 'MakeRequest' APL+Win client-side function uses the APLNext WebTransfer ActiveX component to transmit the request from the client to the server and receives the server response.



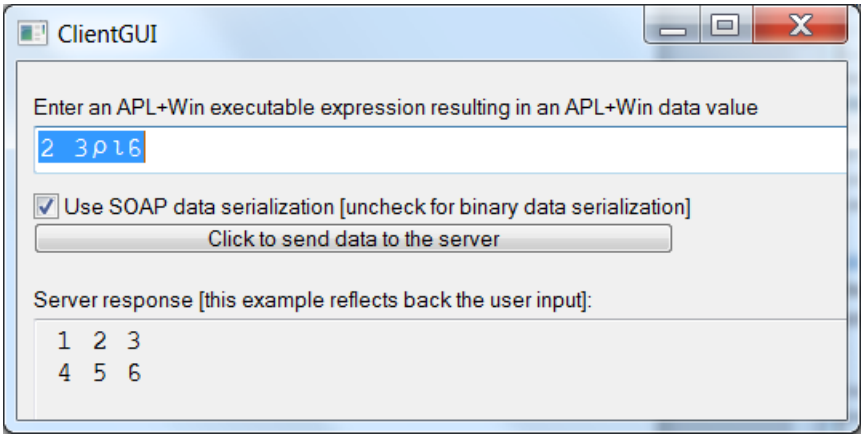
For purposes of this example, this function displays its actions in the APL+Win developer session window in addition to returning the server response as the function result:

The 'Base URL' for this web service application is 'http://localhost:9003' because this example uses the APL+Win programmer's workstation as both the client and the server. The virtual paths '/testserialization/soapwrapl' and '/testserialization/binarywrapl' are associated in the APLNext Application Server configuration with the APL+Win server-side functions 'ReturnSoapWraplResult' and 'ReturnBinaryWraplResult' respectively.

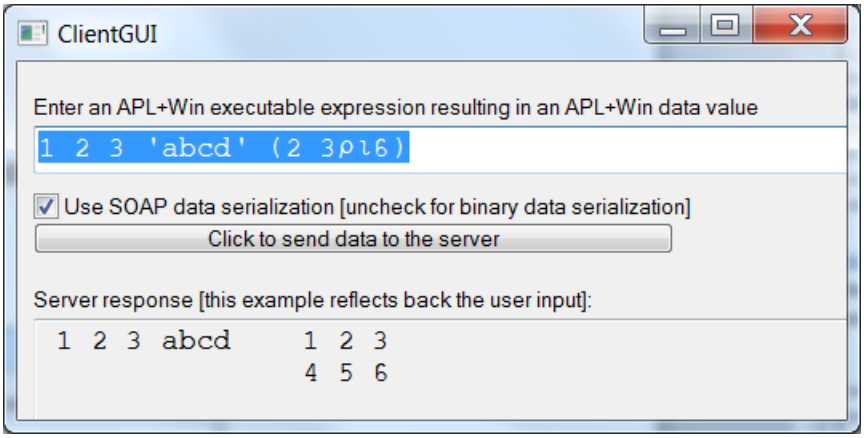
The APLNext WebTransfer ActiveX component has an extensive object model which supports synchronous and asynchronous web service interactions and many other capabilities related to web services. APLNext WebTransfer is part of the APLNext Application Server software license.

Using this Example

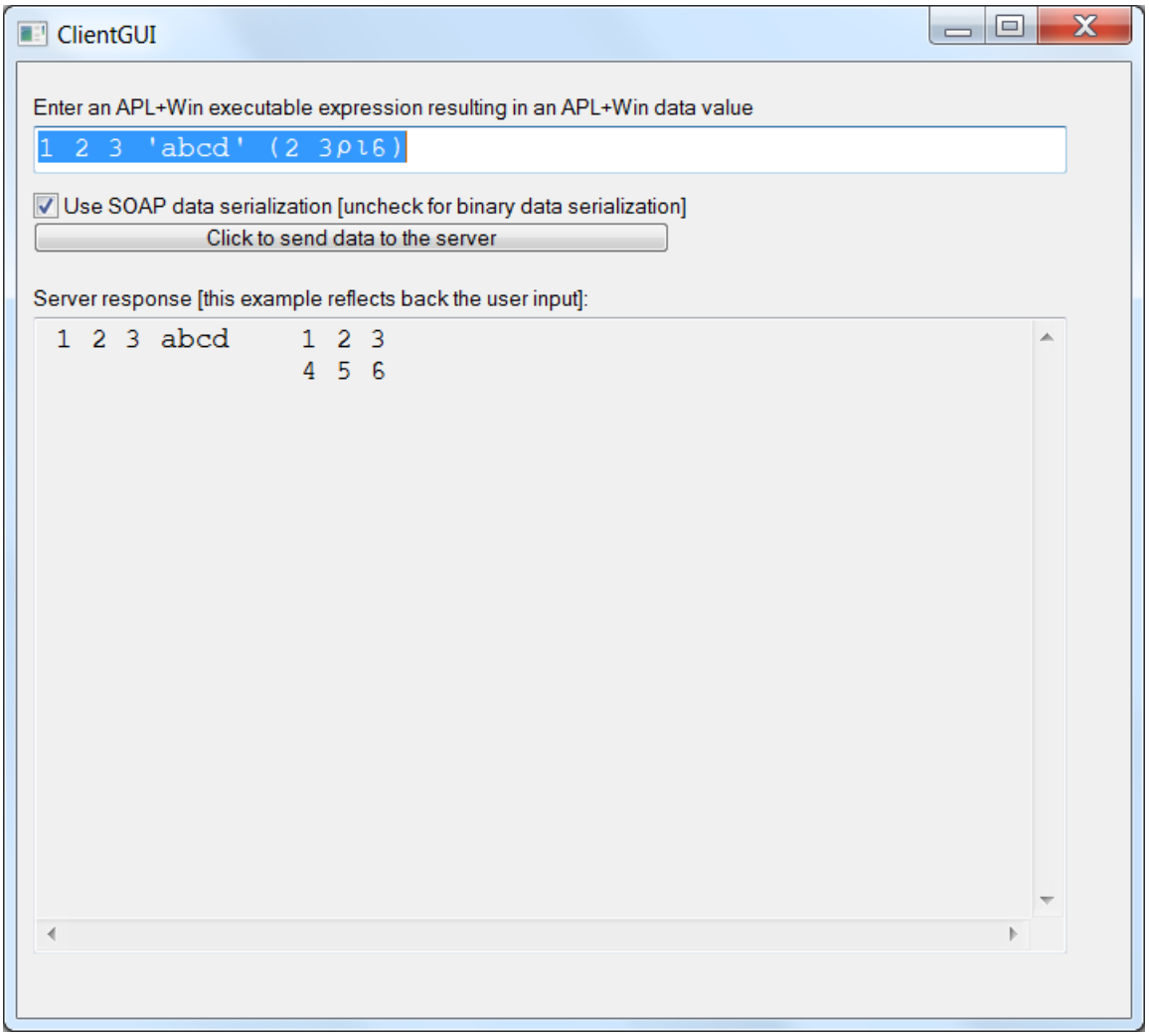
When the client-side GUI is presented, a sample APL+Win executable expression which results in an APL+Win data value is provided. Clicking the GUI button transmits the request to the server which responds, for purposes of this example, with the user input which has made a round trip from the client, to the server and back to the client.



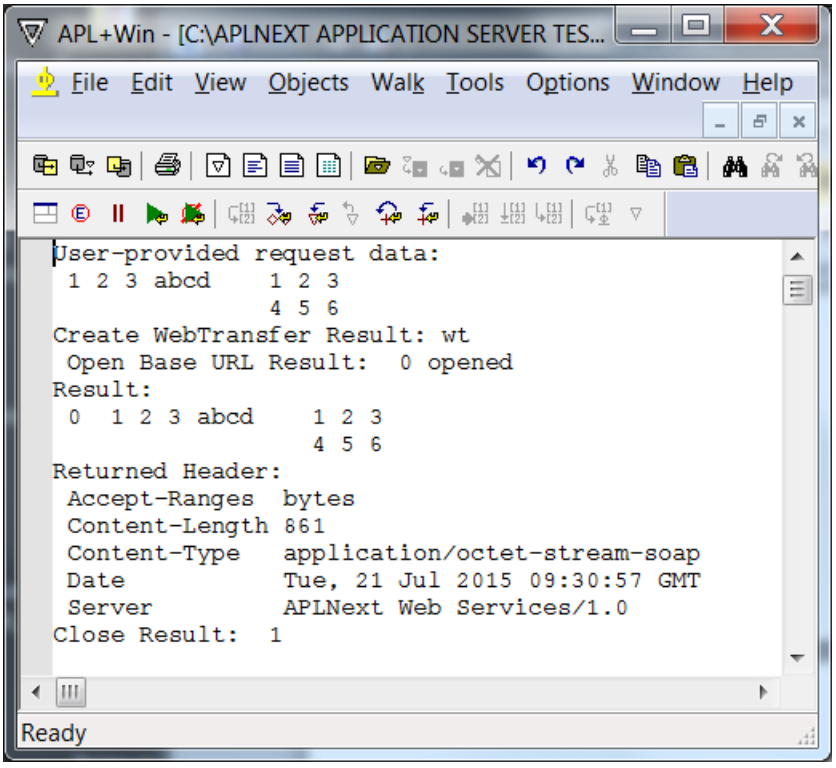
The APL+Win functions in this example incorporate exception handling to trap and report invalid user input:



The following example illustrates a more complex APL+Win data value being transmitted to and returned from the server:



APL+Win developer session output for this more complex example shows additional APLNext WebTransfer object model properties:



More Information

Please contact APL2000 for more information about APLNext Application Server or APLNext WebTransfer. APL2000 customers may access the APL2000 Web Services forum for additional information.