Server Process Reference

Servers

A server provides its services in the Altium Designer environment (the client side). The Client module of the Altium Designer interprets the tasks in terms of processes and then delegates these processes to the appropriate servers. For example when a user is clicking on the Schematic menu to place a wire, the Client interprets this action as a **'PlaceWire'** process and delegates the process to the Schematic Editor server. The Schematic server responds by executing the process. The functionality of a server that is installed in the Altium Designer is exposed by that server's processes and its exposed functions.
Generally a process is executed by selecting a packaged process launcher (such as clicking on a toolbar button, or pressing a hot key or selecting a menu item) called as a command in Altium Designer, however you may wish to manually run a process: Up to three different types of process launchers can be used to launch the same process.

Server Processes

Each server process has a process identifier. The process identifier is made up of two parts separated by a colon. The first part of the process identifier indicates the server that defines the process, and the second part is the process name.
For example, the process Sch:ZoomIn is provided by the Schematic Editor server. When this process is launched, either by selecting a menu item, pressing a hot key or activating a toolbar button (which are all defined as process launchers in the Altium Designer), it will perform the task of zooming in on the currently active schematic sheet.
A server process is implemented as a server name:server process string. Processes are stored in a command launcher table maintained by the server. Every time you execute a process via the user interface in Altium Designer, it consults the appropriate server's command table to fetch the process string and then sends this string over to the server for the server to determine which process to execute. These processes are stored in corresponding server install files (*.ins).

Parametric Server Processes

A parametric server process allows the information, a process needs, to be passed when the process is called. This ability to be able to pass process parameters allows direct control over the operation of a process. For parametric processes, each parameter has a value assigned and this parameter / value block is represented as Parameter = Name.

- For example FileName = C:\Program Files\TestFile.Txt.

To concatenate several parameters as a whole string, each parameter / value block is separated by the pipe | symbol.

- For example Parameter1 = Name1 | Parameter2 = Name 2 etc.

There are two ways you can execute a process in a script
To execute a server process in a script, you need to use commands such as ResetParameters and RunProcess procedures or invoke the Client.SendMessage function.
RunProcess Example

Error rendering macro 'code' : org/lobobrowser/html/gui/HtmlPanel

Client.SendMessage Example

Error rendering macro 'code' : org/lobobrowser/html/gui/HtmlPanel

Assignment of Parameters for a Server Process in a Script

For parametric processes, each parameter has a value assigned and this parameter name / parameter value block is represented as Parameter = Name.
For example FileName = C:\Program Files\TestFile.Txt.
To concatenate several parameters as a whole string, each parameter name / value block is separated by the pipe | symbol.
For example Parameter1 = Name1 | Parameter2 = Name 2 etc.
The TParameterList object helps you deal with parameter strings that consist of multiple parameters (names and values) in a script.

Example

Error rendering macro 'code' : org/lobobrowser/html/gui/HtmlPanel

Consult the System API Reference for more information on the TParameterList object and its methods.

Various Server Processes Links

- Server Process Routines
- Client (System) Processes
- Integrated Library Processes
- PCB Processes
  - FPGAFlow Processes
  - Schematic Processes
  - WorkSpace Manager Processes