

## User Interface - Feature #1785

### implement server-side redirected terminal support

10/30/2012 09:38 AM - Greg Shah

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Greg Shah	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	120.00 hours
<b>Target version:</b>	Runtime Support for Server Features	<b>vendor_id:</b>	GCD
<b>billable:</b>	No		
<b>Description</b>			
<b>Related issues:</b>			
Blocked by Base Language - Feature #1786: implement server-side support for n...			<b>Closed</b>

#### History

##### #1 - 10/31/2012 02:55 PM - Greg Shah

- Target version set to Milestone 7

##### #2 - 04/16/2013 10:14 AM - Greg Shah

- Status changed from New to Closed

- Assignee set to Greg Shah

This task is no longer needed.

The initial idea was that server-side applications (batch programs, appserver, webspeed...) would be able to leverage features normally designed for client-side usage, but that these features would be executed inside the server process. Upon further consideration, this idea won't work. The basic reasoning is that these client features all entail state that can be different for each process. In the 4GL, the appserver support is really implemented using multiple 4GL client processes and every batch program is likewise a separate client process. This means that these processes each have their own state. If that state was resident in the P2J application server and had a Java-backed representation, then it could be "virtualized" to separate each "process" from one-another. But each of these client-side features has state that exists at the operating system level (e.g. the current directory for a process, file descriptors or the environment) and/or in 3rd party code (e.g. DLL instance data). This state is outside of the control of the Java server and thus we must actually retain the multi-process client design of the 4GL. The processes will still run on the same server system, but for each appserver/batch process there must be a corresponding client process that can be used to execute/contain these features.

In the case of terminal redirection, there is no advantage to moving it to the server when we are already required to have a client process. In fact, that redirection would be potentially dependent upon the the current directory, the client process' psuedo-terminal and other operating-system level state like the ncurses environment which has process-specific state.

##### #3 - 11/16/2016 11:43 AM - Greg Shah

- Target version changed from Milestone 7 to Runtime Support for Server Features