

Runtime Infrastructure - Feature #2694

provide configurable control of STDERR logging for spawned clients

09/08/2015 04:21 PM - Greg Shah

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	Deployment and Management Improvements		
billable:	No	vendor_id:	GCD
Description			

History

#1 - 09/08/2015 04:38 PM - Greg Shah

When we spawn (using the server-directed spawn launcher) a web client, batch client or appserver agent, we always redirect STDERR.

For appserver agents (and batch processes, I think), we also automatically redirect STDOUT. Progress always redirects STDOUT for appserver agents and this is a duplication of that feature.

Web clients are not required to have their STDOUT redirected.

Although it is often convenient to redirect both STDOUT and STDERR to the same destination, we don't do that. This was done for compatibility purposes, to exclude STDERR from that appserver log. The idea was to write to the STDOUT log only data which would be actually written from 4GL. So we added a separate log for STDERR, to not change what data is added to this file.

Currently, the STDOUT log file is configured in the "clientConfig/outputToFile" directory node. If the format is like "client_%appserver%_%pid%_%timestamp%_%userid%.log", each token will be replaced accordingly.

The STDERR log file has its name hardcoded in ClientDriver.setClientLog, with the format: "client_%userid%_%pid%.log" - this is not configurable.

For this task we are going to add 2 features:

1. Although the default behavior will be to implement separate log files as we do today, we will add an optional flag in the directory to also redirect STDERR to the same destination log as STDOUT. The idea is that we default to a compatible mode but if the customer decides that the actual output is not a matter in which they care for compatibility, then we can optionally override the default and consolidate the log files. If this mode is active, then we need to consider excluding web clients from the STDOUT redirection.
2. In the separate log file case (the default), we will provide a directory node to configure the STDERR log file name, with the same possible string replacement options as can be used for the STDOUT log file name. Just as you can use either relative or absolute paths in "clientConfig/outputToFile", this new option should also support both relative and absolute paths.

#2 - 09/08/2015 04:40 PM - Greg Shah

Constantin: what was the original reason that we don't redirect STDOUT for web clients? The ChUI web client does not rely upon STDOUT in the same way as the native NCURSES client, but is there some other less obvious dependency that I'm forgetting?

#3 - 09/08/2015 05:12 PM - Constantin Asofiei

Greg Shah wrote:

Constantin: what was the original reason that we don't redirect STDOUT for web clients? The ChUI web client does not rely upon STDOUT in the same way as the native NCURSES client, but is there some other less obvious dependency that I'm forgetting?

For ChUI, STDOUT is used by default by 4GL. So regardless of web or ncurses terminal, we can't have data output to STDOUT by other means than 4GL legacy code... that's why we don't redirect STDOUT for ChUI web clients, as there is nothing to redirect - all 4GL output is sent to the terminal (regardless of how we emulate it, ncurses/STDOUT, swing, web, etc). And if we send other data to STDOUT (i.e. logging), then we will have unexpected results in ncurses vs. swing/web.

#4 - 11/16/2016 01:06 PM - Greg Shah

- Target version changed from *Deployment and Management Improvements* to *Deployment and Management Improvements*