Monitoring a Flux Instance using Enterprise Monitoring Tools

Some Flux customers have IT monitoring suites that support capabilities such as:

- Telnet
- Ping
- Database queries
- File and Directory Monitoring
- Log file parsing and text searching
- Script execution
- HTTP URL executions (a simulated browser session)
- Web service execution
- ….

If using such a tool to monitor a Flux installation, consider the following guidance for issue identification.

If you are trying to:	Then use the tool capabilities to:
Determine if there are general network connectivity issues	Ping the server where Flux is executing Ping the database server
Determine if Flux engine is down	Test to see if the Flux service is running by: - Telneting to the Flux Engine port to see if the Flux engine is listening to the port, and therefore running. - Determining if the Flux logs are being written to. If they are not increasing in size, the engine may not be running. - Perform a text search in the Flux logs for the term "out of memory". This indicates that the engine has not been properly configured with sufficient memory and has shutdown. - Execute an HTTP GET web service call to the Flux engine via the Flux REST API, e.g., http://FluxEngine:7520/engines.
Determine if workflows are failing	Text search in the Flux logs for the term '*threw an exception. The exception is being handled using standard flow chart behavior*'. This statement indicates a flowchart has failed and is attempting to recover.
Determine if the database server is down	Telnet to the port the database is configured to listen on. If the Telnet connection is successful, it is likely the database is available and operating correctly.
Insufficient number of database connections for Flux to properly execute	Text search in the Flux logs for the terms "*the engine was unable to get a connection*" + "*sleeping for X time*". This is indicative of there being an insufficient number of database connections being available.
Poor performing Flux engine	Text search in the Flux logs for the term "*deadlock*". This is indicative of database deadlocks which may be hampering Flux performance. Confirm that Flux has sufficient memory allocated to it. While a Flux engine will start with 512K, most customers allocate 1 to 1.2 GB to their Flux instances (some significantly more if they are running large or memory-intense Java actions.