

### Description

A distributed grid computer requires many processes to be run continuously in the background, such as the merge replication for DGP. Many of these processes are incremental and iterative, which means that they have their own state that must be constantly maintained globally (within a location) as well. DGP needs an automated processing subsystem designed to run all types of processes very reliably and securely, at a large scale with very good performance.

### Requirements

#### 1. Automated processes must be implemented as one-way API methods

What: *The automated processes must be implemented as one-way API methods performing automated work incrementally on small batches of records (for most automated processes).*

Why: *The one-way “fire and forget” methods provide the maximum scalability for the system by avoiding the blocking of synchronous method calls. Also, the web service API methods make the best use of the IO completion port functionality built into the Windows operating system. The fact that all automated process workloads are broken down into small batches of incremental work insures that the process executed in the background by the one-way worker thread is of a short duration.*

Testing: *The API Tester test harness can be used to test the execution of the automated processing API methods.*

#### 2. The execution of automated processes must be data-driven

What: *Complex process maintain their own state data between iterations. This state data records are used to schedule the execution time of iterations and also to store variables that keep track of the progress of the small batches of work when incrementally processing large amounts of data.*

Why: The processes themselves are built to read their state data from the schedule records, picking up where they left off from the previous iteration. When the iteration is complete, the processes store all of their updated state data and the time of their next scheduled execution in the same table record, ready for the next iteration of the process..

Testing: The configuration section of the DGPDive application has the functionality to configure and schedule new automated API methods, while the API Tester test harness application can be used to test those methods..

### 3. The execution of automated processes must use the full standard security of API methods

What: Automated processes, implemented as API methods, are subject to the same security as regular API methods for the system accounts that are used to execute them.

Why: The work performed by automated processes must be subject to the same security-in-depth safeguards as all other API's, eliminating the use of special privileges or admin account back doors and other security vulnerabilities.

Testing: The API Tester test harness tests the functionality of automated process API methods with full security, just like it does for all other API methods.

### 4. The automated processing subsystem must provide the capability to store a variable amount of data about each process iteration

What: Automated processes maintain their state data in reusable records of their schedule table, but also need to store a record of the results (both good and bad) of the work done during each process iteration..

Why: Observability of the automated processes running continuously in the background is a must.

Testing: A test method that deliberately creates many duplicate error messages is used to test the back-pressure logic.