

The count checks are the detection part of the in-place data repair processes. At the time the detection process is run, in the example above it starts by querying for the maximum `src_id` value for location 1 records in the location 2 database table. It then queries for the total count of records that are less or equal to than the maximum ID value, and the total count of active records.

It then runs all of these same count queries at the source, but uses the maximum `src_id` from the destination (not from the source) for the total rows and total active row counts. The difference between the maximum destination `src_id` and the maximum `row_id` of the source indicates the size of the replication lag. The row count values should be the same at the location and the destination. A lower count at the destination indicates missing replicated records, which is fixed by running a Verify scan. A higher count at the destination indicates some sort of duplication that must be investigated by admin staff.

The GeneralWork queue is used to manage the scheduling and execution of detection processes that are part of the In-Place Data Repair functionality in DGP systems. They consists of duplicate replica record checks, and count checks of the data in each source compared to the same count checks in each destination table.

Briefly, in-place data repair refers to the processes that first detect that a problem exists in the multiple copies of data distributed amongst the different locations, and then runs repair processes to correct those problems in-place and in the background, while people continue to use the system (as opposed to resetting the data with a snapshot or some other bulk operation).

The basic premise is that asynchronous data replication in a distributed system built using collections of unreliable parts contains an almost infinite number of things that could go wrong and end up compromising the accuracy and/or integrity of the replicated data. Rather than trying to prevent each and every one of those potential problems, the things that are easily prevented are prevented (low hanging fruit), after which the emphasis is shifted to 1) quickly detecting problems in the data and 2) fixing those problems incrementally, in-place, and in the background. As problems occur, they are analyzed to see if they would be likely to occur again, and if so, whether or not there is a reasonable way to prevent subsequent occurrences. Otherwise, the emphasis remains on quick detection and repair.

Configuring Detection Checks

To create configuration records for detection, some information from various sources needs to be collected.

- The schemaname.tablename combination
- The name of the source database, which is used for several of the queries in the process (the name of the destination database is not needed, and is determined by the destination URL and destination method name used).
- The URL of the web service used to check the source records
- The name of the method used to query for the counts in the source table
- The URL of the web service used to check the destination records
- The name of the method used to query for the counts in the destination table


DGPDrive Beta

Connect Help Storage User Security Configuration Testing

GeneralWork Queue | SrcDBName Search Clear New Page 1 of 22 < >

Network	SchemaTable	SrcDBName	ShardName	WorkType	SrcURL	SrcMethod	DestURL	DestMethod	StartID	FinalID	Interv
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS
LOCALHOST	SysWork.GeneralWork	TestDatabase		COUNTCHECK	TestSrcURL	TestSrcMethod	TestDestURL	TestDestMethod	0	0	MS

< Methods: 2 ClientMS: 54.22 ServerMS: 41.27 >


New GeneralWork Configuration

Global ID

SrcDBName*

Work Type*
COUNTCHECK

Source URL*

Source Method*

Dest URL*

Dest Method*

Input Param

StartID*
0

FinalID*
0

Interval Type*

Interval Val*
2000

Next Run
0

Run State*
READY

Logging*
OFF

Cancel
Delete
Save

When creating new configuration records, the schemaname.tablename combinations and the names of the API methods for replication, count checks, etc. are documented in the API section under DGPWork/SchemaTables. The URL's of the source and destination web services are determined by the network topology of the various locations of an environment for a given system.

**Important Note:* the URL's used in the configuration records must be able to be resolved from wherever the DGPWinSvc will be run.

Field Name	Field Values	Description
WorkType	DUPECHECK, COUNTCHECK used in GeneralWork	<ul style="list-style-type: none"> DUPECHECK calculates counts of any duplicate records in a table COUNTCHECK compares counts of records from a source table to a matching destination table

StartID	64-bit integer	This value is used as the starting value of a process placeholder. It will generally be set to zero for most processes. However, large sets of records can be segmented using the StartID and FinalID fields.
FinalID	64-bit integer	This value sets the upper limit of a process placeholder for some processes. It will generally be set to zero, and is ignored in most cases, but when paired with the StartID, it can be used to break up very large numbers of records into smaller segments.

Some of the other values include Interval Type and Interval Val, which are used to set the Next Run UNIX Time. The scheduling of the detection processes are different from the frequent iterations of replication. The Time Of Day (TOD) and Day Of Week (DOW) interval types in particular are different in that they run once per day at a fixed time, or one day a week at a fixed time. The value associated with the TOD type is the number of minutes offset from midnight on a 24 hour clock. The DOW value includes two comma-delimited parts. The first is an integer that represents the day of the week (with Sunday = 0) and the second is the same as a TOD value. The Next Run value is the Unix Time in milliseconds that the process is scheduled to be run – initially it can be set to zero. Setting the Run State value to READY is equivalent to turning a process on to begin execution.