SOFTWARE RELEASE DOCUMENT (SOFTDOC)

Product: Shadowbase

Release: Gravic Version 6.320
                    HP NonStop Shadowbase:
                            T1122-AAG (SB Repl/Guardian)

Release Date: March 15, 2017

Copyright Notice: Copyright Gravic, Inc. 1995 – 2017 (www.gravic.com)

File Name: IPM6320.pdf

NOTE: This softdoc covers new features and corrected problems for Shadowbase
            for H, J, and L series Guardian NonStop operating system versions. This
            softdoc is available in an Adobe PDF file (.PDF). Softdoc files are named
            IPMnnnn (where nnnn is the Shadowbase version number).

NOTE: This softdoc applies to the HPE NonStop Shadowbase Replication for
            Guardian component. Other softdocs document the releases of the other
            components in T1122H06, including:
            • HPE Shadowbase Audit Reader,
            • HPE Shadowbase Enterprise Manager, and
            • HPE Shadowbase Compare for SQL.

            You are advised to reference those other softdocs for the changes related
            to those specific components of T1122H06.

NOTE: **Supported Release Version Updates (RVUS)**: This softdoc applies to all
            currently supported database and operating system versions as described
            by the Shadowbase Supported Versions Cross-Reference. Alternatively, visit the
            http://www.ShadowbaseSoftware.com web site for the list of supported platforms
            and databases.

Table of Contents

Disclaimer

We are distributing this communication in an effort to bring important information to the attention of users of the affected products. We recommend that all users determine the applicability of this information to their individual situations and take appropriate action. We do not represent or warrant that this information is necessarily accurate or complete for all user situations and, consequently, we will not be responsible for any damages resulting from the user's use or disregard of the information provided. To the extent permitted by law, we disclaim all representations and warranties, whether express, implied, statutory, or otherwise, including the warranties of the merchantability, fitness for a particular purpose, title, and non-infringement.

Special Notes for Version 6.320

1)   Due to a known problem outstanding, you should not configure a QMGR process for a Collector if it has one or more DBS's that are processing DDL operations with coordination and one or more Consumers doing TRACKTX processing. TIDFILECONSTANT ON may be used in place of TRACKTX processing.

2)   As part of the changes in this release, the behavior of the Queue Manager ABNORMALSHUTDOWN parameter and the default value has changed. As a result, when upgrading from an older of Shadowbase in an environment with Queue Manager objects configured, the associated consumer processes will need to have a TIDFILE configured (if not already), or the ABNORMALSHUTDOWN parameters will have to be set to STOP for the Queue Manager objects. For more information on the changes to the ABNORMALSHUTDOWN parameter, and the new related parameters, DRAINTXTIMER, HUNGTX, and PARTIALTX, see the NEW AND MODIFIED SET QMGR PARAMETERS section.

3)   If you utilize the SOLV loader related to a DBS object that has ALLPARTITIONS set ON, you should set the SOLV TARGETFNAME parameter to the DBS SOURCEFILE value. TARGETFNAME consists of the node name, volume and subvolume (no filename) of the SOURCEFILE.

4)   As of version 4.091, the CONTROLFILE record size, associated with the CONS CONTROLFILE parameter, has been increased. You must purge any existing CONTROLFILE's before restarting Shadowbase if you are upgrading from a pre-4.091 version of Shadowbase.

5)   Enscribe and SQL/MP CREATE, PURGE, PURGEDATA, and ALTER events are now processed directly by the Consumer. If you have the DBS setting for these events enabled (ON) and have user exit code that processes these events, you must change the code to call the SBSETIGNORE API function before returning. Otherwise, the Consumer will attempt to process the events again and this could result in errors.

6)   For those that have user exits, you must use the 6.230 USRXLIBO (non-native version), or USRXLIBN or USRXLIBX (native versions) API library included in the distribution subvolume. If you have copies of the USRXLIBO, USRXLIBN, or USRXLIBX file in other subvolumes, make sure you replace them with the current version from the distribution subvolume. You should not use earlier versions of USRXLIBO, USRXLIBN, or USRXLIBX with Shadowbase version 6.230.

7)   Control file changes made in Version 5.001 require that you perform an initial START AUD COLD if you are upgrading from a release prior to 5.001.

8)   For bi-directional replication environments, due to enhancements made to the internal algorithms within the COLL and CONS, it is recommended that both (or all) nodes be upgraded at the same time. Shut down both, do your upgrade, and then bring each node's

Shadowbase environment back up. Contact Shadowbase Support if you have bi-directional environments and you cannot upgrade them both at the same time.

9) For SQL/MX IEEE 32-bit floats, 8 digits of precision in scientific notation is the SQL/MX Consumer (new AUDCONXN and AUDCONXR programs delivered in 5.000) assigned value in the generated SQL statement data. The Tandem format 32-bit float will continue to use 6 digits of precision in the non-IEEE Consumer (AUDCONS, AUDCONSN, and AUDCONSR programs).

Additionally, for DDL related to Enscribe files, the Consumer assumes that all floats are in Tandem format and not IEEE format. A user exit is required if they are actually in IEEE format. Otherwise, a generated SQL statement could have an incorrect number.

10) If you are developing user exit code to use with the SQL/MX Consumer, make certain that you compile your code for IEEE floating point format. Only native code can be used with the SQL/MX Consumer. Non-native user exits are not supported.

11) You must install Shadowbase on a physical volume if SQL/MX tables will be replicated. You should not install Shadowbase on an SMF virtual drive, otherwise errors will result during the install process.

12) If SQL/MX tables will be replicated with Shadowbase, NonStop Shadowbase must be installed on the source system(s) and Shadowbase Open Server must be installed on the target system(s) as Shadowbase's SQL/MX target support is implemented in the OSS environment. Shadowbase Open Server for OSS, version 5.000 or higher, is required.

13) Support for SQL/MX has the following limitations:

- Tables with Relational Integrity (RI) constraints. Ordering of I/Os within a transaction for multiple tables cannot be guaranteed in the audit trail due to parallel nature of transaction processing on NonStop systems. For example, consider the case where there are parent and child tables, with a constraint to force a corresponding parent row to exist when a child row is added. If you add both the parent and child row under the same transaction when a child is added with a new parent, and the parent and child tables are on separate data disks, TMF may store them in reverse order in the audit trail and Shadowbase may not be able to add those rows on the target. Depending on which disk flushed to the audit trail first, the audit trail may return the rows with the child row insert first and the parent row insert second. In this case, Shadowbase will fail to insert either row due to the RI constraint being violated on replay.

- Tables with large rows and Cascading Deletes configured. We have seen that TMF may save updates of large rows as a delete event of the old row followed by an insert of the revised row. Shadowbase will replay this event as a delete followed by an insert as well. If cascading deletes are configured,

the delete may cause all child rows to be deleted on the target. Note that this behavior may also run afoul of RI constraints noted above, if a child row is inserted between the delete and insert of the parent row.

14) If you are upgrading from a release prior to Version 6.100, note that the Shadowbase EMS subsystem owner and version (the EMS SSID token) changed in Version 6.100. The subsystem owner changed from ITI to GRAVIC, and the subsystem version is now composed of VNN, where NN are the two most significant digits of the Shadowbase release version id. As a result, the subsystem version has changed from D42. Any EMS filters using these attributes for filtering must likewise be updated. Alternatively, the SBEMSOLDSSID TACL parameter can be enabled to have Shadowbase use the old values for the subsystem owner and version.

15) The EMS subsystem version has been updated with this version from V61 to V63 to match the two most significant digits of the current Shadowbase release version id.

16) Errors 8401, 8405, 8406, 8410, 8411, 8412, 8417, 8418, 8423, 8424, 8425, 8426, 8427, and 8428 will no longer be logged and ignored by the Consumer by default. The user exit stubs included with Shadowbase, USRXC, USRXCOB, and USRXCOBA, have been modified to remove the call to SBSETEXCEPTIONCODES. The code to setup the structure and make the call remains as a comment in the stub files. To use the structure and make the call to SBSETEXCEPTIONCODES, uncomment the code from the stub file being used.

17) Some customers experienced a rare issue which caused the ARLIB2 library to hang when there was no activity on one of the auxiliary audit trails. The Collector now links with the ALV version of the ARLIB2 library which incorporates a fix for this issue.

18) When using a multi-ported DOC Writer, it is recommended as a best practice not to use Queue Managers for the related threads in the Shadowbase NonStop configuration. Furthermore, it is recommended not to use the SUSPENDUPD/RESUMEUPD commands when replicating to a multi-ported DOC Writer.

Changes in Release 6.320 (since Version 6.230)

New Features

1) The first release of Shadowbase Synchronous Replication, Shadowbase NonStop ZDL®, is now available for homogenous, unidirectional Enscribe and SQL/MP replication over Expand. Shadowbase NonStop ZDL® is available as a separately licensable product released under Controlled Availability. For further details, see the HPE NonStop Shadowbase ZDL® Manual or contact Shadowbase Product Management.

2) Extended statistics are now available when using SBCOLL. Previously, extended statistics were only available when using AUDCOLLN. This includes use of STATS AUD, ESTATS AUD, and ESTATS DBS. See SBCOLL SUPPORT FOR EXTENDED STATISTICS for further details.

3) Support for the ALTER command has been added to the QMGR object. For a listing of the alterable parameters and other information, see ALTER QMGR COMMAND.

4) Measure counters have been implemented for SBCOLL, AUDQMGR, SBCOOR, and SBMETRO. See NEW MEASURE COUNTERS for more information.

5) Enhancements have been made to allow for managing the behavior of uncommitted transactions in the event of an abnormal shutdown. When synchronous replication is enabled, these enhancements leverage additional transaction state knowledge provided by the subsystem. See UNCOMMITTED TRANSACTION MANAGEMENT ENHANCEMENTS for more information.

6) The TACL configuration macros for configuring replication for SQL/MX have been modified to allow configuration of the new DOC blocking and writing modes. See SBUNDROS AND SBBIDROS ENHANCEMENTS for more information.

7) Reading of audit from up to 64 TMF audit trails (master audit trail plus up to 63 auxiliary audit trails) is now supported. See SUPPORT FOR 64 TMF AUDIT Trails for further details.

8) Enhanced support for Enscribe queue file replication has been added. With the new ALLOWQUEUEFILE DBS parameter REPLICATEKEY option, Enscribe queue file replication is available without the need for a user exit. In this case, Shadowbase will preserve the source primary key SYSKEY value when it inserts the target record into the target Enscribe queue file. See NEW AND MODIFIED SET DBS PARAMETERS for details on the updated ALLOWQUEUEFILE parameter.

Problems Fixed

1) When replicating SQL/MX data, the Consumer process could be configured on the target node. This caused problems in some circumstances, such as with partitioned tables, as the Consumer needs to retrieve schema information, but it is not possible to do so from a remote node for SQL/MX (this is an SQL/MX limitation). To prevent this from occurring, validation has been added to prevent configuration of the Consumer process on the target node for SQL/MX replication. We are working on this issue at this time with HPE SQL/MX product management to determine if there is a reasonable alternative.

2) For Enscribe replication to an Other Servers target with a target DDL record definition (via the TARGETREC DBS parameter) specified but no target file defined (via the TARGETFILE DBS parameter), the Consumer process used the key offset and length from the source file to attempt to determine the key fields in the target record definition. For a target record definition excluding fields before the key, and thus having a different key offset in the target definition, this resulted in an incorrect determination of the key fields. Because of this, it was required to specify a target file with the correct key offset and key length of the target record definition in this case.

   This behavior has been changed, so that when a target DDL record definition is specified without specifying a target file, the target key fields are determined based upon which fields make up the key of the source file. In other words, if the key of the record consists of the fields A, B, and C, and the target record definition contains the fields A, B, and C, these three fields will be treated as the key fields for the target record. If the target record only contains fields A and C, but not field B, only A and C will be used as key fields for the target record. Please note that this may result in unintended consequences for UPDATE and DELETE statements. Hence, always define a target file to explicitly control the key used.

   In the case where a target file is defined, the key fields are instead determined based upon the target file key offset and key length, as before.

3) When replicating purgedata events with the MAPTOPRIPTN option enabled, the "cut off" logic was not working properly (the "cut off" logic only replicates one purgedata event rather than all of them for each partition). When set, the first purgedata event should be mapped to the primary partition and processed, and then subsequent purgedata events for the other partitions should be skipped. Instead, all of the purgedata events were being mapped to the primary partition and processed.

4) SOLV would hang while loading with a standalone client Consumer in a NonStop to NonStop TCP/IP environment.

5) Issuing a stats qmgr command in an environment with a lag duration of greater than roughly 600 hours caused the audmon process to trap with a math fault.

6) Setting an =_EMS_COLLECTOR define to an EMS collector on a remote node caused AUDCOLLN to fail to start with an ARSTART error -7.

7) When running the INSTALL script with the OWNER option specified, a handful of files were not given to the specified owner as expected.

8) Corrected an intermittent issue uncovered in in-house testing that caused the QMGR to trap with a memory error on a restart. Subsequent restarts succeeded.

9) The QMGR will now save the SUSPEND / RESUME state of the consumer in its restart file and will restart in the last saved state. Previously, the QMGR would restart with the CONS in the RESUME state, meaning that the replication flow would start after a QMGR restart.

10) When using a TRACKTX file with a QMGR, the QMGR did not properly handle transactions with negative transaction identifiers during TRACKTX processing on a restart. Negative transaction identifiers occur on systems with a node number greater than 127.

11) A memory leak in the QMGR related to the new sync (Shadowbase ZDL) functionality was found and corrected. The leak would eventually cause the QMGR to run out of memory and trap.

12) Corrected an issue that caused the Consumer to trap with an arithmetic overflow fault when doing processing for Base24 replication and the FILECLOSEDELAY parameter is set higher than 30.

13) Corrected an issue that caused the new Collector, SBCOLL, to trap with an arithmetic fault after processing approximately 2.1 billion events when throttling was disabled.

14) Corrected an issue that caused the new Collector, SBCOLL, to trap with an arithmetic fault when more than 2.1 billion events were processed without the statistics being reset.

15) Corrected an issue that caused the new Collector, SBCOLL, to trap when compressed audit records for SQL/MP tables and Enscribe files were read.

16) Corrected a performance issue in the new Collector, SBCOLL. This issue was introduced in a previously released restricted availability release and caused SBCOLL to use significantly more CPU time to process the same audit than in prior releases.

New Features Description

SBCOLL Support for Extended Statistics

Support for extended statistics has been added for environments using SBCOLL. STATS AUD, ESTATS AUD, and ESTATS DBS commands are available with all of the existing options supported previously by AUDCOLLN. This includes both collector level statistics, and consumer level statistics previously unavailable for SQL/MX replication.

For SQL/MX tables, both the guardian file name and the SQL/MX table ansi name are displayed in the extended statistics output.

An example of the output for a SQL/MX table named MYCAT.MYSCH.MYTAB from the collector portion of the extended statistics is as follows:

```
\GRAVIC1.$DATA4.ZSDZN5D7.LD3K1T00     ADT: BB  LE: 08-02 17:14:51.483   INCL
ANSINAME: MYCAT.MYSCH.MYTAB
  INSERT                3               3               3               0
  UPDATE                1               1               1               0
  DELETE                8               8               8               0
  CREATE                0               0               0               0
  ALTER                 0               0               0               0
  PURGE                 0               0               0               0
  PURGEDATA             0               0               0               0
    TOTALS             12              12              12               0
```

In the consumer file statistics, both names are similarly output for the source. For the target, the associated guardian file name is not known by the consumer, and it is instead marked as SQLMX along with the DBS name:

```
CONS CONS-OPN-JMZC1
DBS:    DBS1
SOURCE: \GRAVIC1.$DATA4.ZSDZN5D7.LD3K1T00       LEV: 08-02 17:14:51.483
S-ANSI: MYCAT.MYSCH.MYTAB
TARGET: [SQLMX:DBS1]                             LIO: 08-02 17:14:56.500
T-ANSI: MYCAT.MYSCH2.MYTAB
        EVENT TYPE         DO       UNDO TRANSFORMED    IGNORED
    -------------- ----------- ----------- ----------- -----------
      INSERT              3           0           0           0
      UPDATE              1           0           0           0
      U->I INSERT         0           0           0           0
      I->U UPDATE         0           0           0           0
      DELETE              8           0           0           0
      CREATE              0           0           0           0
      ALTER               0           0           0           0
      PURGE               0           0           0           0
      PURGEDATA           0           0           0           0
        TOTALS           12           0           0           0
```

New Measure Counters

Additional measure counters have been added for AUDQMGR, SBCOLL, SBCOOR, and SBMETRO. To perform a measurement using these counters, first add a MEASURE userdef for the process and add the desired measure counters in MEASCOM, then start the measurement. Then, start Shadowbase with the MEASURE parameter configured ON

for the corresponding objects. The counters specified in MEASCOM for the measurement will be initialized and incremented/recorded accordingly.

NOTE: For busy and queue counters, it is recommended to start Shadowbase with the measure parameters enabled after starting the measurement rather than altering measure on. The reason for this is that these types of counters are only updated when the state or when the queue length changes (increments or decrements). As a result, the current state is not reflected when altering the parameters on. Similarly, best results can be attained for these types of counters when no restarts occur during the measurement, and when the measurement is stopped before Shadowbase is stopped or the measure attributes are altered to off.

The new statically named measure counters are described in the following table:

Process	Counter Name	Type	Description
SBCOLL	ADTRECSREAD	Accum	Number of audit records read
	ADTINSREAD	Accum	Number of inserts read
	ADTINSUSED	Accum	Number of inserts selected
	ADTINSBUSY	Busy	Time spent in auditreading procedures retrieving insert data
	ADTDELREAD	Accum	Number of deletes read
	ADTDELUSED	Accum	Number of deletes selected
	ADTDELBUSY	Busy	Time spent in auditreading procedures retrieving delete data
	ADTUPDREAD	Accum	Number of updates read
	ADTUPDUSED	Accum	Number of updates selected
	ADTUPDBUSY	Busy	Time spent in auditreading procedures retrieving update data
	ADTEFLDUPDREAD	Accum	Number of Enscribe compressed updates read
	ADTEFLDUPDUSED	Accum	Number of Enscribe compressed updates selected
	ADTEFLDUPDBUSY	Busy	Time spent in auditreading procedures retrieving Enscribe compressed update data
	ADTSFLDUPDREAD	Accum	Number of SQL/MP compressed updates read
	ADTSFLDUPDUSED	Accum	Number of SQL/MP compressed updates selected
	ADTSFLDUPDBUSY	Busy	Time spent in auditreading procedures retrieving SQL/MP compressed update data
	ADTBEGINSREAD	Accum	Number of implicit begins (on detection of a new transaction)
	ADTCOMMITSREAD	Accum	Number of commits read
	ADTOTHERSREAD	Accum	Number of other audit trail record types read
	ADTPREPBUSY	Busy	Time spent reading audit trail and preparing the buffers for processing
	ADTPREPIDLE	Busy	Time not reading audit trail preparing buffers for processing. This includes processing time for sending messages to Consumers and all other processes
	ADTCONSMSGS	Accum	Number of messages sent to Consumers
	ADTCONSQTIME	Queue	Consumer process time (from start of writeread until reply is received)
	ADTCONSOPS	Accum	Number of inserts, updates and deletes sent to Consumers

Process	Counter Name	Type	Description
SBCOLL	ADTFLDFETCH	Accum	Number of auditreading procedure field fetches performed
	ADTNETABORTREAD	Accum	Number of network aborts read
	ADTNETCOMMITREAD	Accum	Number of network commits read
	ADTCRTREAD	Accum	Number of creates read
	ADTCRTUSED	Accum	Number of creates used
	ADTPRGREAD	Accum	Number of purges read
	ADTPRGUSED	Accum	Number of purges used
	ADTABORTSREAD	Accum	Number of aborts read
	ADTAUXPTRSREAD	Accum	Number of aux pointers read
	ADTFLDFETCHSAME	Accum	Number of unchanged fields fetched when using the old fetch method with ALLUPDATES OFF
	ADTFLDFETCHDIFF	Accum	Number of changed fields fetched when using the old fetch method with ALLUPDATES OFF
	ADTALTREAD	Accum	Number of alters read
	ADTALTUSED	Accum	Number of alters selected
	ADTPRGDATAREAD	Accum	Number of purgedatas read
	ADTPRGDATAUSED	Accum	Number of purgedatas selected
	ADTMXFLDUPDREAD	Accum	Number of SQL/MX compressed updates read
	ADTMXFLDUPDUSED	Accum	Number of SQL/MX compressed updates selected
	ADTMXFLDUPDBUSY	Busy	Time spent in auditreading procedures retrieving SQL/MX compressed update data
	ADTREADLAG	FAccum	Total read lag time measured on all commit events as the difference between the audit trail timestamp and the read timestamp (Max 1 hour per event)
	ADTREADLAGSAMPLE	FAccum	Number of read lag samples included in ADTREADLAG number
	ADTRLAGMAX	FAccum	Accumulation of maximum read lag time taken over each minute
	ADTRLAGMAXSAMPLE	FAccum	Number of samples in ADTRLAGMAX
	ADTCOMMITSUSED	Accum	Number of commits selected
	ADTABORTSUSED	Accum	Number of aborts selected
	ADTPRESCREENHIT	FAccum	Number of prescreen hits
	ADTPRESCREENMISS	FAccum	Number of prescreen misses
	ADTPRESCREENBUSY	Busy	Time spent performing prescreening
	ADTSCREENBUSY	Busy	Time spent performing event screening from file lists

Process	Counter Name	Type	Description
SBCOLL	ADTDBSCHECKBUSY	Busy	Time spent checking event files against configured DBSes
	ADTSCREENINCLUDE	FAccum	Number of included file events matched by screening
	ADTSCREENEXCLUDE	FAccum	Number of excluded file events matched by screening
	ADTSCREENMISS	FAccum	Number of file event not found by screening, requiring DBS check
	SYNCTXCOUNT	FAccum	Number of sync transactions processed
	RTSPITCHED	FAccum	Number of RTS requests that were discarded (not used)
	ROLLBACKPITCHED	FAccum	Number of rollback notifications that were discarded (not used)
	FORGETPITCHED	FAccum	Number of forget notifications that were discarded (not used)
	ADTTRANSQCOUNT	Queue	Queue of active transactions
	SYNCRTSQCOUNT	Queue	Queue of active RTS requests
	PREP2RTSREPTIME	FAccum	Time in ms from prepare request to RTS reply
	PREP2RTSREPSAMP	FAccum	Number of samples in PREP2RTSREPTIME
	LE2ALLREADTIME	FAccum	Time in ms from when last event read to when all data read known
	LE2ALLREADSAMP	FAccum	Number of samples in LE2ALLREADTIME
	RTSREQ2REPTIME	FAccum	Time in ms from RTS request received to RTS reply sent
	RTSREQ2REPSAMP	FAccum	Number of samples in RTSREQ2REPTIME
	RTSREP2ENDTIME	FAccum	Time in ms from RTS reply sent to end audit event read
	RTSREP2ENDSAMP	FAccum	Number of samples in RTSREP2ENDTIME
	END2RTSREPTIME	FAccum	Time in ms from end audit event read to RTS reply sent
	END2RTSREPSAMP	FAccum	Number of samples in END2RTSREPTIME
	FORGET2ENDTIME	FAccum	Time in ms from forget notification received to commit audit event read
	FORGET2ENDSAMP	FAccum	Number of samples in FORGET2ENDTIME
	END2FORGETTIME	FAccum	Time in ms from commit read to forget notification received

Process	Counter Name	Type	Description
SBCOLL	END2FORGETSAMP	FAccum	Number of samples in END2FORGETTIME
	ROLLBACK2ENDTIME	FAccum	Time in ms from rollback notification received to abort audit event read
	ROLLBACK2ENDSAMP	FAccum	Number of samples in ROLLBACK2ENDTIME
	END2ROLLBACKTIME	FAccum	Time in ms from abort event read to rollback notification received
	END2ROLLBACKSAMP	FAccum	Number of samples in END2ROLLBACKTIME
SBCOOR	SYNCTXCOUNT	FAccum	Number of synchronous transactions processed
	ASYNCTXCOUNT	FAccum	Number of asynchronous transactions processed
	METRONOMETXCOUNT	FAccum	Number of metronome transactions processed
	SLATXCOUNT	FAccum	Number of synchronous transactions exceeding the first SLA duration
	MAXCOMMITTXCOUNT	FAccum	Number of synchronous transactions committed early due to reaching second SLA duration
	APPSYNCABORTS	FAccum	Number of application synchronous transaction aborts
	APPASYNCABORTS	FAccum	Number of application asynchronous transaction aborts
	SBSYNCABORTS	FAccum	Number of coordinator synchronous transaction aborts
	SYNCTXPREP1QCNT	Queue	Number of synchronous transactions at a time that have not started commit processing
	SYNCTXP1QCNT	Queue	Number of synchronous transactions at a time in phase 1 of the commit sequence
	SYNCTXP2QCNT	Queue	Number of synchronous transactions at a time in phase 2 of the commit sequence
	ASYNCTXPREP1QCNT	Queue	Number of asynchronous transactions at a time that have not started commit processing
	ASYNCTXP1QCNT	Queue	Number of asynchronous transactions at a time in phase 1 of the commit sequence

Process	Counter Name	Type	Description
SBCOOR	ASYNCTXP2QCNT	Queue	Number of asynchronous transactions at a time in phase 2 of the commit sequence
	PREP2RTSSENTTIME	FAccum	Total time in ms from prepare received to RTS sent
	PREP2RTSSENTSAMP	FAccum	Number of samples in PREP2RTSSENTTIME
	PREP2TGTSYNCTIME	FAccum	Total time in ms from prepare received to target sync'd
	PREP2TGTSYNCSAMP	FAccum	Number of samples in PREP2TGTSYNCTIME
	PREP2RDYSENTTIME	FAccum	Total time in ms from prepare received to READY signal sent
	PREP2RDYSENTSAMP	FAccum	Number of samples in PREP2RDYSENTTIME
	PREP2COMMITTIME	FAccum	Total time in ms from prepare received to commit (forget) signal received
	PREP2COMMITSAMP	FAccum	Number of samples in PREP2COMMITTIME
SBMETRO	METROTICKCOUNT	FAccum	Number of metronome ticks
	MISSEDTICKCOUNT	FAccum	Number of missed metronome ticks
	COORBEGINBUSY	Busy	Time spent starting coordinating transaction
	COORWAITREG	Busy	Time spent waiting for coordinating transaction registration to complete
	COORIDLE	Busy	Time spent idle in coordinating transaction waiting for commit
	COORCOMMITBUSY	Busy	Time spent committing coordinating transaction
	MONBEGINBUSY	Busy	Time spent starting monitoring transaction
	MONWAITREG	Busy	Time spent waiting for monitoring transaction registration to complete
	MONIDLE	Busy	Time spent idle in monitoring transaction waiting for commit
	MONCOMMITBUSY	Busy	Time spent committing monitoring transaction
	MONWRITEBUSY	Busy	Time spent writing to disks for monitoring transaction
	TICKWAITBUSY	Busy	Time spent waiting after committing for one tick before starting the tick.
AUDQMGR	ADTBEGINRECV	FAccum	Number of begins received

Process	Counter Name	Type	Description
AUDQMGR	ADTBEGINSENT	FAccum	Number of begins sent
	ADTCOMMITRECV	FAccum	Number of commits received
	ADTCOMMITSENT	FAccum	Number of commits sent
	ADTABORTRECV	FAccum	Number of aborts received
	ADTABORTSENT	FAccum	Number of aborts sent
	ADTNETCOMMITRECV	FAccum	Number of network commits received
	ADTNETCOMMITSENT	FAccum	Number of network commits sent
	ADTNETABORTRECV	FAccum	Number of network aborts received
	ADTNETABORTSENT	FAccum	Number of network aborts sent
	ADTALTRECV	FAccum	Number of alters received
	ADTALTSENT	FAccum	Number of alters sent
	ADTCRTRECV	FAccum	Number of creates received
	ADTCRTSENT	FAccum	Number of creates sent
	ADTPRGRECV	FAccum	Number of purges received
	ADTPRGSENT	FAccum	Number of purges sent
	ADTPRGDATARECV	FAccum	Number of purgedatas received
	ADTPRGDATASENT	FAccum	Number of purgedatas sent
	ADTINSRECV	FAccum	Number of inserts received
	ADTINSSENT	FAccum	Number of inserts sent
	ADTUPDRECV	FAccum	Number of updates received
	ADTUPDSENT	FAccum	Number of updates sent
	ADTDELRECV	FAccum	Number of deletes received
	ADTDELSENT	FAccum	Number of deletes sent
	TXSENDLAG	FAccum	Total lag in seconds from commit record audit time to send time
	TXSENDSAMPLE	FAccum	Number of samples in TXSENDLAG
	QUEUEREADBUSY	Busy	Portion of time sending from cache (Note this is not processing time)
	CACHEREADBUSY	Busy	Portion of time sending from queue (Note this is not processing time)
	QFILESINUSEQCNT	Queue	Number of queue files in use at a time
	IPCSINUSEQCOUNT	Queue	Number of ipcs in use at a time
	TXTRACKQCOUNT	Queue	Number of transactions tracked at a time
	TXABORTQCOUNT	Queue	Number of transactions aborting at a time

In addition to the statically name measure counters, one set of dynamically named measure counters have been added for SBCOLL. For each attached thread, there is one recognized QUEUE type measure counter that measures the number of messages queued for sending at a time. Note that messages that are in the process of being sent are tracked separately, and not included in this count. The following naming convention is used for

these counters, based off of the process name of the QMGR or CONS directly attached to
SBCOLL:

```
<process-node>_<process-name>_Q
```

process-node
The name of the node the CONS or QMGR process is running on, with the leading "\"
(SLASH) removed.

process-name
The CONS or QMGR process name, with the leading "$" (DOLLAR-SIGN) removed.

For example, to add the counter for collector $COLL1 for a consumer thread with a
process name of \NODE1.$CONS1 without a QMGR configured, the MEASCOM add
command for the measure counter would be:

```
add counter NODE1_CONS1_Q    ,process $coll1, queue
```

If a QMGR object was configured for the thread, the process name of the QMGR would
be used instead of the CONS process name.

Uncommitted Transaction Management Enhancements

Several enhancements were made to support improved control over the management of
uncommitted transactions. Previously, the ABNORMALSHUTDOWN parameter was
added to the QMGR to allow draining of the queue in the event of an AUDMON failure,
or loss of communication with the AUDMON process. Upon reaching the end of the
queue in the event of a drain, the state of all outstanding transactions was unknown, as it
was unknown whether or not there was more data for the transactions.

When replicating synchronously with Shadowbase ZDL enabled, however, there is now
additional knowledge of some transactions. To accommodate this, the
ABNORMALSHUTDOWN parameter has been reworked, and now defaults to DRAIN.
In addition, several new QMGR parameters, DRAINTXTIMER, HUNGTX and
PARTIALTX have been added. These parameters: 1) control the time to wait before
taking the configured default actions for uncommitted transactions, 2) specify the default
action for hung transactions, and 3) specify the default action for partial transactions,
respectively. For more information on these parameters, see NEW AND MODIFIED SET
QMGR PARAMETERS. Similarly, changes have been made to the DRAIN QMGR
command. For details, see DRAIN QMGR COMMAND.

Outstanding transactions are now broken into four types:
   1. Transactions for which a FORGOT signal has been received, but the commit
      event has not been received (this means the source application intended to commit

the transaction and TMF was obliging that request). These transactions will be committed by default.

2. Transactions for which a ROLLBACK signal has been received, but the abort event has not yet been read. These transactions will be aborted by default.

3. Transactions for which a request to sync has been received, and all data is known to have been received (this means the source application intended to commit the transaction, but we do not know whether TMF was planning to commit or abort the transaction). The default behavior for these transactions is controlled by the HUNGTX parameter.

4. All other outstanding transactions, the default behavior for which is controlled by the PARTAILTX parameter.

When replicating asynchronously, all transactions will fall into the fourth category, and be treated as partial transactions.

SBUNDROS and SBBIDROS Enhancements

The ADD CONSOS command for the SBUNDROS and SBBIDROS TACL configuration macros, has been enhanced to allow the configuration of the new DOC blocking and writing options introduced in the Version 6.230 DOC Writer.

Version 6.230 of the DOC Writer introduced a new method of writing and reading the DOC files that eliminated the need for one of the tracking files, the TRANSDOC file, significantly reducing the DOC writing/reading overhead, particularly for BEGIN and COMMIT events. The new mode is configured by disabling transaction processing (Transaction process='N') when adding or editing a DOC writer object with SBMON.

That release also introduced a new method of DOC blocking that combined the BEGIN and COMMIT events with the Insert, Update, and Delete DML events. That mode is enabled by specifying DOC_BLOCKING=2 in the SHADPARM.INI file.

When configured with both transaction processing disabled and with DOC_BLOCKING mode 2, the DOC writing performance is 5 to 10 times faster than in previous releases.

> **Note: Because of the significant performance increase, Gravic suggests that all customers consider upgrading and running in this new mode for DOC/TRS environments.**

The IPM6230_Other_Servers_OSS document has more details on the changes to DOC writing, including some important limitations as to when the new modes can be used.

Previous versions of the SBUNDROS and SBBIDROS macros could not configure the new modes (even with overrides) and required the DOC objects to be edited after they were added by the macros. The new versions allow the new modes to be configured directly.

These macros will generate configurations that use the old method of DOC writing compatible across all versions of the OSS software when used with existing, unmodified configuration files. You can continue to use these macros without modifying your configuration files or upgrading the OSS release to Version 6.230 or later.

As mentioned above, the SB_ADD CONSOS command has changed to support the new DOC writing options. The command documentation is provided below.

SB_ADD CONSOS (SBUNDROS Macro, DOCW/TRS Configuration)

Two optional parameters have been added to the end of the SB_ADD CONSOS command to specify transaction processing and the DOC blocking mode. Both should be specified or omitted.

The new syntax is:

```
sb_add consos <src_cons_pname> <src_cons_cpu_p> <src_cons_cpu_b> <src_cons_pri>
              <src_ip_pname_p> <src_ip_pname_b> <tgt_ip_addr_p>
              <tgt ip addr b> <port> <tgt ip pname p> <tgt ip pname b>
              <tgt_pname_p> <tgt_pname_b> <tgt_cpu_p> <tgt_cpu_b> <tgt_pri>
              <db_catalog> <db_schema> <db_user> <db_password> <db_dsn>
              <trs_pname> <trs_cpu> <trs_pri> <dcl_pname_p>
              <dcl pname b> <dcl cpu p> <dcl cpu b> <dcl pri> <trans proc> <docblk ver>


        <src_cons_pname>  is a unique process name used for the source consumer
                          (audconxn).
        <src cons cpu p>  is a valid CPU number for the source consumer's primary
                          CPU.
        <src_cons_cpu_b>  is a valid CPU number for the source consumer's backup
                          CPU (use -1 for no backup).
        <src cons pri>    is the priority number of the source consumer.
        <src_ip_pname_p>  is the source consumer's primary tcpip process name.
        <src ip pname b>  is the source consumer's backup tcpip process name (use - for no
                          backup).
        <tgt_ip_addr_p>   is the target DOC server primary ip address.
        <tgt_ip_addr_b>   is the target DOC server backup ip address (use - for no
                          backup).
        <port>            target ip port address.
        <tgt ip pname p>  is the DOC server primary tcpip process name.
        <tgt_ip_pname_b>  is the DOC server backup tcpip process name.
        <tgt_pname_p>     is the primary DOC server process name.
        <tgt_pname_b>     is the backup DOC server process name.
        <tgt cpu p>       is the DOC server primary cpu.
        <tgt_cpu_b>       is the DOC server backup cpu.
        <tgt pri>         is the DOC server priority.
        <db_catalog>      default catalog name used by the DOC server.
        <db_schema>       default schema used by the DOC server.
        <db_user>         database logon user name
        <db_password>     database user password
        <db dsn>          ODBC data source name.
        <trs_pname>       TRS process name
        <trs_cpu>         TRS cpu
        <trs_pri>         TRS priority.
        <dcl pname p>     is the primary DOC cleaner process name.
        <dcl_pname_b>     is the backup DOC cleaner process name.
        <dcl cpu p>       is the DOC cleaner primary cpu.
        <dcl_cpu_b>       is the DOC cleaner backup cpu.
        <dcl_pri>         is the DOC cleaner priority.
        <trans_proc>      (opt) is the transaction processing prompt setting. Should be 'Y'
                          to enable transdoc writing (the old algorithm) or 'N' to disable
                          transaction processing. Defaults to 'Y' (old algorithm).
        <docblk ver>      (opt) is the DOC Blocking algorithm version to use.
                                0: No DOC blocking
                                1: DOC blocking version 1 (separate records for BEGIN,
                                   COMMIT, ABORT events).
                                2: DOC blocking version 2 (BEGIN, COMMIT, ABORT events
                                   combined with IUD events).


Example:
> sb_add consos $gpjc1 0 1 131 $Zb03b $Zb02b 10.1.50.70 10.1.50.60 30100 &
$zb03b $Zb02b $GPDCP $GPDCB 1 0 132 catalog schema db.user pass &
TDM_Default_DataSource $GPJTR 1 133 $GPCLP $GPCLB 0 1 130 N 2
```

Note: If you wish to configure for a single $TCP/IP stack, then the SB_ADD CONSOS
line should look like so:

```
sb_add consos $gpjc1 0 -1 131 $Zb03b - 10.1.50.70 - 30100 &
$zb03b $Zb02b $GPDCP $GPDCB 1 0 132 catalog schema db.user pass &
TDM_Default_DataSource $GPJTR 1 133 $GPCLP $GPCLB 0 1 130 N 2
```

The dashes represent the source's backup TCP/IP process name and the target's backup
IP address respectively. The -1 represents the source system's backup CPU to use for the
consumer process (the second is -1 since no alternate path is being configured).

SB_ADD CONSOS (SBBIDROS Macro)

Two optional parameters have been added to the end of the SB_ADD CONSOS
command to specify transaction processing and the DOC blocking mode. Both should be
specified or omitted.

The new syntax is:

```
 sb_add consos consos-id cpu-p cpu-b doc-cpu-p doc-cpu-b base-doc-port trs-cpu trs-cat
             trs-sch trs-user trs-pass trs-dsn shadtrs-tab-vol shadtrs-idx-vol
             trans-proc docblk-ver

      <consos-id>     is two characters long and must be unique
      <cpu-p>         primary cpu of the consumer
      <cpu-b>         backup cpu of the consumer (use -1 for no backup).
      <doc-cpu-p>     primary cpu of the doc
      <doc-cpu-b>     backup cpu of the doc (use -1 for no backup).
      <base-doc-port> is the base port of the doc.  The port used for each DOC beyond
                      the first will be incremented from this number.
      <trs-cpu>       primary cpu of the trs
      <trs-cat>       catalog supplied to the trs configuration (database server
                      name prompt)
      <trs-sch>       schema supplied to the trs configuration (target database
                      prompt)
      <trs-user>      login user name for the trs
      <trs-pass>      login password for the trs
      <trs-dsn>       ODBC data source name
      <shadtrs-tab-vol> volume where the base shadtrs table will be created
      <shadtrs-idx-vol> volume where the shadtrs index will be created
      <trans-proc>    (opt) is the transaction processing prompt setting. Should be 'Y'
                      to enable transdoc writing (the old algorithm) or 'N' to disable
                      transaction processing. Defaults to 'Y' (old algorithm).
      <docblk-ver>    (opt) is the DOC Blocking algorithm version to use.
                             0: No DOC blocking
                             1: DOC blocking version 1 (separate records for BEGIN,
                                COMMIT, ABORT events).
                             2: DOC blocking version 2 (BEGIN, COMMIT, ABORT events
                                combined with IUD events).
```

Note: If you wish to configure for a single $TCP/IP stack, then the SB_ADD CONSOS line should look like so:

```
SB_ADD CONSOS C1 0 -1 0 -1 12345 1 SVFECAT FETEST SUPER.QA QA DS1 $QA $QA N 2
```

The -1's represent the source's backup CPU to use for the consumer process and the target's backup CPU to use for the DOC process (the second is -1 since no alternate path is being configured).

Support for 64 TMF Audit Trails

Support for up to 64 TMF audit trails has been added to HPE NonStop Shadowbase for L series systems with a supporting version of TMF. The ability to configure up to 64 audit trails was added with the AMG SPR release version of TMF. Upgrading to the AMG release (or newer) of TMF does not require upgrading the version of Shadowbase unless more than 16 audit trails are configured. This corresponds to up to one master audit trail plus up to fifteen auxiliary audit trails, the previous limit. If more than 16 TMF audit trails are configured on the system, then HPE NonStop Shadowbase must be upgraded to version 6.320 or newer to run on that system.

To accommodate a configuration that reads from the additional audit trails, the ADTAUXMASK parameter has been enhanced. It now allows for a list of comma separated values and ranges of auxiliary audit trail ids to be specified. If selecting only auxiliary audit trails in the range of 1 through 15, either the old parameter specification method or the new one may be used. The new parameter specification method must be used to select auxiliary audit trails in the range of 16 through 63. For details, see the ADTAUXMASK entry in the NEW AND MODIFIED SET COLL PARAMETERS section.

A current limitation of the support for 64 audit trails is that ADTSCRATCHVOLnn parameter functionality has not been expanded. Reading from audit scratch volumes is still supported for AUDCOLLN only, and reading from audit scratch volumes may only be configured for the master audit trail and auxiliary audit trails 1 through 15 via the corresponding ADTSCRATCHVOL parameters.

New and Modified Commands

ALTER QMGR Command

Support has been added to the QMGR for the ALTER command. The ALTER QMGR command changes the attributes of a previously defined QMGR process.

In the SET QMGR command parameter descriptions, there is a version box, as in the samples below, with a column titled 'Alter' which is set to 'Yes' for those parameters that can be altered while running. If the value under 'Alter' is 'No', then these parameters cannot be altered once AUDCOLL is started.

ABNORMALSHUTDOWN {DRAINSTOP}
                                      {DRAINPAUSE}
                                      {STOP}

Initial Version	Changed Version	Values Default	Value Specifications	Alter
4.092	6.320	DRAINSTOP	Valid Action	YES

The syntax is:

```
ALTER  [ QMGR ] [<audmon_name>.]<qmgr_name> { , qmgr_param } [
,qmgr_param ]...
```

audmon_name
Optionally specifies the Guardian process name for the Shadowbase monitor process
controlling the replication environment. If omitted, the currently opened replication
environment will be used.

qmgr_name
Is the logical name of a previously defined Queue Manager.

qmgr_param
Is any attribute to be changed. A qmgr_param consists of a parameter keyword and a
value or set of values for that parameter. Available parameters can be found in the SET
QMGR section.

If you receive an error during an ALTER QMGR command, do an INFO QMGR
command to see if any values have changed.

See the SET QMGR command description for more information on the qmgr_parameters.

The following is the list of existing parameters for the QMGR which changed in 5.200 to
become alterable while running:

ABNORMALSHUTDOWN
BACKUPCPU
CHKPTDELAY
CHKPTTIME
CHKPTWARNRATE
CHKPTWARNTDIFF
CPU
CPULIST
DRAINTXTIMER
FAILMAX
FAILRETRYDELAY
FAILSPAN
HUNGTX

MEASURE
PARTIALTX
QFILEEXTENT
QFILESPERVOLUME
QFILESECURITY
QFLUSHDELAY
QFLUSHLIMIT
QVOLUMES
TRACE
TRACEFILE

For some parameters, such as those related to the CPU the QMGR will run in, alters will not take effect until the next time the QMGR object is started. Additionally, some parameters have further considerations for their behavior when altered. These are noted in updated parameter descriptions in the New and Modified SET QMGR Parameters section.

DRAIN QMGR Command

The DRAIN QMGR command has been changed, and the new syntax is as follows:

The DRAIN QMGR command dynamically specifies the behavior of the QMGR and its attached Consumer following an AUDMON failure. This command may only be used for QMGRs with ABNORMALSHUTDOWN set to DRAIN, which also requires the configuration of a TIDFILE for the attached CONS.

The syntax for the DRAIN command is as follows:

```
DRAIN [QMGR] <qmgr_process_name> { , CANCELTIMER }
                                  { , SHUTDOWN [ , COMMIT {ALL      }
                                                          {HUNGTX   }
                                                          {PARTIALTX}]
                                            [ , ABORT  {ALL      }
                                                          {HUNGTX   }
                                                          {PARTIALTX}] }
                                  { , STATUS }
                                  { , STOP }
```

`<qmgr_process_name>`
The process name of the QMGR to which the DRAIN command is to be directed.

The CANCELTIMER option disables the DRAINTXTIMER, causing the QMGR to wait until a subsequent DRAIN command, as if DRAINTXTIMER was set to -1.

The SHUTDOWN option, performs uncommitted transaction processing as soon as it reaches the end of the queue, or immediately if it has already reached the end. When the ABORT option is used, the specified types of transactions will be aborted instead of the

configured default action for HUNGTX and/or PARTIALTX. Similarly, when the
COMMIT option is used, the specified types of transactions will be committed instead of
the configured default action for HUNGTX and/or PARTIALTX, using the same pre-
commit logic as when those parameters are set to COMMIT.

When the STATUS option is specified, the QMGR is queried for the current state of
drain operations. The current state is output, and if a drain has completed, the count of
hung transactions and partial transactions detected will be output. Additionally, the
numbers to be committed and aborted based on HUNGTX and PARTIALTX settings will
also be output.

When the STOP option is specified while the QMGR is draining its queue, the QMGR
and CONS stop draining and shutdown immediately.

Example output following completion of drain but before expiration of
DRAINTXTIMER:

```
+drain qmgr $JMZQ1, status
QMGR DRAIN STATUS AT 2016-09-07:17:49:20 :
  QMGR PROCESS: $JMZQ1  DRAIN STATE: COMPLETE
  OUTSTANDING TRANSACTION COUNTS BY TYPE:
      PARTIAL          HUNG COMMITTING   ABORTING
   ---------- ---------- ---------- ----------
            1          0          1          0
```

New and Modified SET COLL Parameters

A number of new SET COLL parameters have been added for use with synchronous
replication, and Shadowbase ZDL in particular. For information on these parameters, see
the Shadowbase Replication AUDCOM Collector Parameter Reference section of the
Shadowbase ZDL Manual.

ADTAUXMASK   {mask}
                          {(auxrange [, auxrange]…)}

Initial Version	Changed Version	Values Default	Value Specifications	Alter
3.700	6.320	111111111111111	mask, or aux range list	No

This is used to identify the auxiliary audit trail files, which are to be read by the
Collector. The mask form of the parameter is 15 bytes long and each byte is entered as a
zero (0) or one (1) to respectively disable or enable access to each of the original 15
auxiliary audit trails. By default all 15 bytes are set to 1 meaning that the Collector will
read all of the 15 original auxiliary audit trail files (i.e., aux01 through aux15) if present
in the TMF configuration. When using this form, access to any auxiliary audit trails
beyond aux15 will be disabled.

As an example, suppose you have aux01 and aux02 configured in TMF, but only want a
Collector to read aux02 because there are tables and files on data volumes associated with

aux02 that you want replicated by HPE Shadowbase. You would enter the following
command so that the Collector only reads aux02:

    SET COLL ADTAUXMASK 010000000000000

As of version 6.240, the ADTAUXMASK parameter also supports a list of auxiliary
audit trail numbers or ranges for selection. Valid numbers are 1 through 63,
corresponding to aux01 through aux63 respectively (if supported by your version of
TMF). Ranges are specified by the lower and upper numbers of the range, separated by a
dash. In the simple example above where you only want to read aux02, the following
command would be used:

    SET COLL ADTAUXMASK (2)

If you have upgraded to a version of TMF supporting 64 audit trails, and want to read
from all of the auxiliary audit trails configured on your system, the following command
would be used:

    SET COLL ADTAUXMASK (1-63)

As a more complex example with the new parameter format, suppose that you only want
to read from aux02, aux05 through aux08, aux14, aux15, and aux63. You could configure
this using the following command:

    SET COLL ADTAUXMASK (2, 5-8, 14-15, 63)

Restricting the reading of auxiliary audit trail files can significantly reduce the amount of
data that a Collector needs to read and process, and can dramatically "speed up" its
processing of master audit trail events. However, you need to be careful that restricting
the reading of these AUX trails won't skip data that you want to replicate (e.g., if
$DATA is saved by TMF into auxiliary audit trail file AUX02, and you restrict reading
AUX02, you will never be able to replicate this data). Similarly, you need to be careful
to place certain HPE Shadowbase configuration files, such as the TRANSLOGs in a bi-
directional environment, onto audit trails that the Collector will read (i.e., do not place the
TRANSLOGs on auxiliary audit trail files that you skip).

Note: All Collectors read the master audit trail (this contains the pointers to the auxiliary
audit trail files, as well as the transaction termination records (such as commit or
abort)).

Note: When using AUDCOLL or AUDCOLLN, the number of existing audit trails
configured for collection determines the number of cursor opens which will be
requested in the call to ARSTART, which impacts the amount of memory
required. With larger numbers of audit trails, the LABELCACHEDEPTH
parameter may need to be set to a higher value than the default. If

LABELCACHEDEPTH is not set high enough, the collector will abend with an
ARE_INVALID_PARAM error on the call to ARSTART at startup.

New and Modified SET CONS Parameters

No SET CONS parameters were added or modified since version 6.230.

New and Modified SET DBS Parameters

ALLOWQUEUEFILE {REPLICATEKEY}
                            {GENERATEKEY}
                            {OFF}

Initial Version	Changed Version	Values Default	Value Specifications	Alter
5.002	6.320	OFF	REPLICATEKEY GENERATEKEY OFF	No
	Client ☒	Server ☐	Extended ☐ Open Target ☒	

Special processing is likely in order to replicate Enscribe QUEUE files. By default, the
Consumer will abend if it receives an event associated with a queuefile.
ALLOWQUEUEFILE can be set to GENERATEKEY or REPLICATEKEY to allow the
Consumer to process the event.

If you need to process a queuefile, special user exit logic may be (and often is)
required. By default, Enscribe QUEUE files operate as FIFO objects, meaning that the
first record enqueued (enqueue operations are replicated as INSERT events) is the first
record removed by a dequeue operation (dequeue operations are replicated as DELETE
events). This is because the default Enscribe QUEUE ordering scheme is via an internal
8-byte Julian timestamp as the high order part of the primary key (this 8-byte Julian
timestamp is referred to as a "SYSKEY"). The SYSKEY is guaranteed by the file system
to always be increasing for each successive enqueue operation. The SYSKEY is normally
file system assigned when the event is enqueued. This is the case with HPE Shadowbase
replication when set to GENERATEKEY; this part of the key for the target file record is
generated by the file system. Hence, the "primary key" between the source and target
QUEUE files will not match. This means that, without user exit support, HPE
Shadowbase does not know the correct key of the record to remove when a dequeue
operation is replicated. Note that this is equivalent to the behavior of the ON value in
prior Shadowbase versions.

When set to REPLICATEKEY, however, HPE Shadowbase will also specify the Julian
timestamp high order part of the primary key when performing an enqueue operation.
The timestamp value from the source file record is used in this case. As a result, the
"primary key" between the source and target QUEUE files will match, and HPE
Shadowbase will be able to apply dequeuer operations to the correct record without
needing a user exit. In most cases, this should be the value selected unless you use a user
exit that handles the SYSKEY mismatch issue.

To use this option, T9050J01^AYP with T9053J03^BAQ for J series RVUs (included with J06.20), or T9050L02-AYT/T9053L02-BAS for L series RVUs (included with L16.05) is required. This option is not supported for H series RVUs.

MAPTOFILE dummy_fname

Initial Version	Changed Version	Values Default	Value Specifications	Alter
3.600	6.320	None	**Valid file name Include NODE**	No

This parameter indicates the name of a dummy file or table to be used for mapping schema/DDL Records from all of the sourcefile partitions, even when the partitions do not exist at the start of replication. The Dummy_fname must exist at start, and must be of identical schema/DDL Record representation as SOURCEFILE.

Indicate a fileset in the designation of the filename by using the following wildcard characters for volume, subvolume and/or filename:

*   (asterisk) matches from 0 to 8 characters in the position where it appears. The corresponding portion of the source filename is then used to determine the For example,

       \NODE1.$VOL.*.*

       Indicates all files matching the DBS SOURCEFILE will be mapped to the corresponding file in the corresponding subvolume on volume $VOL of node \NODE1.

       Note: When an asterisk is specified as part of the volume, subvolume, or filename, that entire entity will be replaced with the corresponding entity from the source file. In other words, $VOL.SVOL.ABC* is equivalent to $VOL.SVOL.* for the replacement that occurs.

?   cannot be used.

NOTE: MAPTOFILE functionality is not supported for DDL replication when a fileset (i.e. wildcarded) value is specified for the MAPTOFILE parameter.

New and Modified SET QMGR Parameters

ABNORMALSHUTDOWN   { DRAIN }
                                { STOP }

Initial Version	Changed Version	Values Default	Value Specifications	Alter
4.092	6.320	DRAIN	DRAIN STOP	No

This parameter is used to control the behavior of the QMGR and its attached Consumer in the event of AUDMON failure. When set to DRAIN, the QMGR and attached CONS drains as much of the queue as possible. After reaching the end of the queue, the QMGR can be configured to shutdown, or to wait an amount of time configured by the DRAINTXTIMER QMGR param. At the end of the waiting period, any transactinos still in progress with will committed or aborted based on the settings of the HUNGTX and PARTIALTX QMGR params. While waiting, a DRAIN QMGR command can be issued to alter the wait period, initiate the end transaction actions, or alter the configured commit and abort behavior. When ABNORMALSHUTDOWN is configured to DRAIN (the default), a TIDFILE must be configured for the attached CONS. When set to STOP, the QMGR and its attached CONS immediately shutdown upon detecting a failure of AUDMON. The default value is DRAIN.

DRAINTXTIMER   number_of_seconds

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.320		0	-1 disable	32767	Yes

The amount of time the Queue Manager will wait after losing connection with the AUDMON process before performing ABNORMALSHUTDOWN processing. The actions that will be taken are based upon the configured values for HUNGTX and PARTIALTX. A value of -1 will disable this processing. A value of 0 means to perform processing immediately following a failure.

HUNGTX   { ABORT }
                         { COMMIT }

Initial Version	Changed Version	Values Default	Value Specifications		Alter
6.320		COMMIT	ABORT	COMMIT	Yes

Determines the action that is to be taken for "hung" transactions when the DRAINTXTIMER expires. For these purposes, a hung transaction is one for which a Ready To Commit? (RTC) has been received and acknowledged (all data has been received), but neither the FORGET signal nor the commit have been seen.

ABORT   hung transactions are aborted.
COMMIT   hung transactions are pre-commited.

When the COMMIT option is specified, pre-commit logic will be used similar to that used for RESOLVELOCKS processing. If the transaction aborted on the source system, the pre-committed events will be undone when a connection from the source is reestablished and the abort event is replicated. Similarly, when ABORT is specified, if a commit occurred on the source system, the events will be replayed and then committed following a reconnect.

PARTIALTX   { ABORT }
                             { COMMIT }

Initial Version	Changed Version	Values Default	Value Specifications	Alter
6.320		ABORT	ABORT COMMIT	Yes

Determines the action that is to be taken for partial (incomplete) transactions when the DRAINTXTIMER expires. For these purposes, an incomplete transaction is one for which events have been received, but no Ready To Commit? (RTC), FORGET, or end event has been seen. For asynchronous replication, this will include all outstanding transactions.

ABORT   incomplete transactions are aborted.
COMMIT   incomplete transactions are pre-committed.

When the COMMIT option is specified, pre-commit logic will be used similar to that used for RESOLVELOCKS processing. If the transaction aborted on the source system, the pre-committed events will be undone when a connection from the source is reestablished and the abort event is replicated. Similarly, when ABORT is specified, if a commit occurred on the source system, the events will be replayed and then committed following a reconnect.

QVOLUMES {volume}
          {(volume [,volume...])}

Initial Version	Changed Version	Values Default	Value Specifications	Alter
4.090	6.320	None	A disk volume or list of disk volumes for storing the queue data.	Yes

This parameter is the volume or list of volumes to store the queue data on. This is a required parameter. Up to 16 disks can be specified.

Note: if possible, the disks should be dedicated to the disk queue to avoid having the writes impact your application. If the disks cannot be dedicated, consider setting the QBUFFEREDIO parameter to off.

When altering this parameter, if a disk is removed from the list, any existing queue files on that disk will remain in use until the data in the file has been replayed successfully.

New and Modified TACL Parameters

No commands were added or modified since version 6.230.

New or Modified EMS Messages

4526	UNKNOWN MONITOR STATE DETECTED, DRAINING EVENTS FROM QUEUE

Cause: The AUDMON process failed, or communication with the process is otherwise unavailable from the Queue Manager. The Queue Manager ABNORMALSHUTDOWN parameter is set to DRAIN.

Effect: The Queue Manager begins draining the remaining events from the queue by continuing to send then to the associated Consumer process. When the drain completes, behavior of the Queue Manager is dependent on the configured values of the HUNGTX, PARTIALTX, and DRAINTXTIMER parameters.

Recovery: If a change of the drain behavior is desired, or to force Queue Manager to stop immediately, the DRAIN command can be used.

4526	ABNORMALSHUTDOWN DRAIN REACHED END OF QUEUE WITH <num_tx> OUTSTANDING TRANSACTIONS; <num_partial> PARTIAL, <num_hung> HUNG, <num_forgot> COMMITTING, <num_rollback> ABORTING

Cause: The drain of the queued events from the Queue Manager has completed, with the transaction counts noted for hung and partial transactions, as well as the number to be committed or aborted.

Effect: Informational only.

Recovery: None needed.

4526	ABNORMALSHUTDOWN DRAIN COMPLETE; <num_precommit> TRANSACTIONS WERE PRECOMMITTED

Cause: The drain of the Queue Manger's queue has completed following a detected abnormal shutdown condition. The count of transactions displayed were pre-committed based upon the configured actions for partial and hung transactions upon reaching the end of the queue.

Effect: Informational only.

Recovery: The Queue Manager may be restarted as normal.

4527	DETECTED DIFFERENCE IN <attribute> PARAMETER BETWEEN CONFIGURED VALUE: <configured-value>, AND RESTART FILE VALUE:<restart-value>, USING CONFIGURED VALUE

Cause: A change in a configured parameter value was detected since the last restart of the Queue Manager.

Effect: Informational only. The current and previous values are output, and the newly configured value is put into effect.

Recovery: None needed.

New and Modified User Information Messages

The following message was modified to reflect a change of the QMGR ABNORMALSHUTDOWN parameter values:

ERR 2130:	TIDFILE IS REQUIRED WHEN ATTACHED QMGR ABNORMALSHUTDOWN PARAMETER IS SET TO DRAIN

Cause: An ADD CONS command was entered for a CONS with a QMGRNAME set to a QMGR with ABNORMALSHUTDOWN set to DRAIN. No TIDFILE was configured for the CONS that is being added.

Effect: The command is not executed, and the CONS object is not added.

Recovery: Configure a TIDFILE for the CONS object or SET the ABNORMALSHUTDOWN QMGR parameter to a different value for the QMGR the CONS is feeding from.

The new user information messages are:

ERR 2168:	CAN'T ALTER QMGR PARAMETER -> \<param>

Cause: An ALTER QMGR command was issued for a parameter that is not alterable.

Effect: The attempt to alter the QMGR parameter fails.

Recovery: Issue an ALTER QMGR command for an alterable parameter. To change the value of a parameter that is not alterable, a change of the configuration file and restart of Shadowbase will be required.

New or Modified SOLVCOM User Messages

No SOLVCOM messages have been modified since version 6.230.

Installation Instructions (HPE NonStop Shadowbase)

Shadowbase is now built for HPE NonStop installations using the HPE PAK and UNPAK utilities – refer to the G06.16 and above operating system documentation (or contact Shadowbase Support) for more information. The SBPAK file described below is delivered as a self-extracting file.

1) Binary FTP the SBPAK from your PC (if you obtained the files from a Gravic FTP site) or DVD. If FTP was used, enter the following commands to alter the file code.

>   For G-based systems:
>   FUP ALTER SBPAK, CODE 700

>   For H-based and J-based systems:
>   FUP ALTER SBPAK, CODE 800

2) To "unpak" the files from the self-extracting SBPAK, enter the following command:

RUN SBPAK,   *.*.*, VOL <restore volume and subvolume>, LISTALL, MYID

Where <restore volume and subvolume> is the location you want the Shadowbase files installed into, e.g. $DATA.SHADBASE

3) Follow instructions in the Installing Shadowbase on a NonStop System section of the *HPE Shadowbase NonStop Installation and Planning Manual*. You must now run either the INSTALL or ENSINST script.

IMPORTANT NOTES ABOUT USER EXITS:

IF YOU HAVE USER EXITS DEVELOPED WITH COBOL, YOU MUST RECOMPILE THE USER EXIT "PROGRAMS" SO THAT THE LATEST VERSION OF USRXLIBO OR USRXLIBN (the native library) IS ADDED TO THE RESULTING OBJECT FILE THAT WILL SUBSEQUENTLY BE INCLUDED IN THE FINAL BIND OR LINK.

AFTER COMPILING THE USER EXITS (IF NECESSARY AS DISCUSSED ABOVE), RUN BINDUSRX, BINDXYPR OR LINKUSRX TO GENERATE THE CUSTOM CONSUMER (AUDCONS OR AUDCONSN). NOTE THAT THIS SHOULD BE DONE AFTER THE STANDARD INSTALL IS RUN.

Known Problems Remaining

1) The DOC Roll command is not supported in configurations that use a QMGR out to either the OSS SQL/MX target or an Other Servers target.

2) For SQL/MX, the documented maximum value for the 32-bit float is 3.402823<mark>47e</mark>+38. However, the HPE Shadowbase for Other Servers ODBC Server gets a range error starting with 3.402823<mark>37e</mark>+38 (<mark>difference highlighted and underlined</mark>). The cause of this issue remains under investigation.

3) With a NonStop AUDMON configured for auto-restart and backup CPUs are configured for the Collectors and Consumers, you may have to issue an AUDCOM RUN command for replication to continue after a CPU failure if you have previously entered a SUSPEND COLL command followed by a RESUME COLL command (prior to the failover of AUDMON). A workaround for this is to enter an AUDCOM run command immediately after the RESUME COLL command.

4) During SOLV testing at Gravic, the consumer received an SQL error -1031 for an insert to one of its complex test tables. HP's SQL error description for the error is:

> *A supplied numeric value exceeds the declared precision of the column, some of the supplied values for DECIMAL or VARCHAR columns are invalid, or the supplied row is too long. Also, the disk process might have encountered a bad column in a stored row or a value in an update on a row that would change the length of a VARCHAR column in an entry-sequenced table.*

After investigation of the problem, the error was subsequently reproduced with SQLCI and submitted to HP for further investigation. At this time, it appears that this is caused by a problem in SQL/MP. A fix by HP has yet to be released. HP development is working on the problem.

It has not been determined as to the exact cause of the error -1031. It could be related to the order of certain data types in the schema definition for the SQL table, resulting in an alignment issue that is not handled properly in SQL/MP. The error appeared to be related to SQL processing on the column named C63 in the following series of columns in the schema:

```
,c62      SMALLINT    DEFAULT NULL
,c63      varchar(1)      default NULL
,c64      DATETIME    year to fraction
```

HP has indicated that a workaround could resolve the problem. Add the following define before starting AUDMON.

```
ADD DEFINE = _SQL_CMP_FORCE_EXE_EXPR,
FILE <any Shadowbase TARGETFILE file name>
```

Contact Shadowbase support if the SQL error -1031 continues after you have attempted the workaround.

5)   The new SOLVMGR process will abend if configured with an INACTIVITYDELAY greater than 2147 seconds (approximately 35 minutes).

6)   In certain conditions, the Consumer may continuously fail (with the Collector restarting at the same audit position) replaying audit and not be able to advance. This problem may occur under the following conditions:

- The Consumer is replicating to a file (table) with a unique alternate key (index).
- The Consumer receives a Guardian error 40 (timeout) on an insert operation due to a lock collision between two active transactions on the alternate key (index).
- The Consumer receives a Guardian error 59 (disk file is bad) on its subsequent delete of the record (row) to 'reverse' the failed insert.

If this issue occurs, either set the Consumer parameter TRACKTX to enable the TRACKTX processing or set the Consumer parameter TIDFILECONSTANT to ON to enable TIDFILE processing to allow the Consumer to continue after the initial failure.

7)   A failed insert into a file (table) with alternate keys (index) may leave the record inserted in the base file (table) with no corresponding alternate key records (index entries) under the following conditions:

- The maximum number of active transactions in a Consumer has been reached so the Consumer is doing the "RESOLVETRANS" processing (pre-committing transactions to allow additional simultaneous transactions).
- The Consumer receives a Guardian error 40 (timeout) on an insert operation due to a timeout due to a lock collision on the alternate key (index).

In this case, the record (row) inserted into the base file (table) may not be properly removed.

This condition can be avoided either by turning RESOLVETRANS off or by ensuring that enough Consumers are configured to avoid RESOLVETRANS processing.

8)   SOLV has the potential of missing records at the end of a file in a sequence when used with SOLVMGR (solv manager) for 'file chasing'. This problem is more likely to occur with low values of SOLVMGRDELAY, SOLVEOFDELAY, INACTIVITYDELAY and ACTIVITYDELAY are set. By default these values are set to 60 seconds. The minimum values for these configurations are 1 second, however, Gravic does not recommend using values lower than the default to avoid the potential for data loss.

9) For SQL/MX replication, running the Consumer process on the target system is problematical. In several cases, the Consumer needs to access file / schema information from the catalog that is only available on the source system. To prevent issues, the Consumer process should be run on the source system. This is the normal configuration, and will be configured by default when using the SBUNDROS and SBBIDROS scripts.

10) The maximum key length supported by SOLV, SOLVMX, and SOLVNV remains 255 bytes.

11) SOLVMX cannot handle fully qualified SQL/MX table names more than 48 characters in length.

12) For SQL/MX replication, the maximum column name length that can be successfully replicated is 76 characters.

13) When performing a SOLV load of a partitioned file or table from NonStop to NonStop, no warning message is output if the TARGETFNAME is not specified. A warning is output when replicating to a HPE Shadowbase for Other Servers target when configured with the basefile as the source with the ALLPARTITIONS DBS parameter ON. This warning indicates the number of events for which no associated DBS was found in the Consumer during the SOLV load. This warning message should be output in this case if performing a SOLV load to a NonStop target as well.

14) The Consumer does not properly handle statement text that is generated in excess of 64KB for a single statement. This can occur in certain extreme cases with SQL/MX replication or with excessive user exit "put" function calls. To generate statement text exceeding this limit requires the equivalent of a table with approximately 1000 columns with an average column name length in excess of 60 bytes.

15) For DDL replication, certain DDL events require multi-stage processing by HPE Shadowbase replication. Create events, for example, are processed by creating the base file, then performing an alter of the file, and then creating alternate key files (if any). In the event of a failure in the middle of such multi-stage processing, the remaining stages may not be properly applied. If the base file was created, but the Consumer was stopped before the alternate key files were created, for example, on restart the Consumer will attempt to reprocess the event. Since the base file already exists, however, attempting to recreate it fails with an error 10. After receiving this error, the remaining stages are not attempted, and the alternate key files are not created.

16) When replicating from an Enscribe source file using a DDL definition out to a HPE Shadowbase for Other Servers target, the proper where clause is not generated if no dummy TARGETFILE is specified. This results in an empty where clause, causing a SQL error in the Consumer process. The workaround is to specify a dummy TARGETFILE.

17) SQL/MX tables may not be replicated if the SQL/MX catalog they are located in has multiple definition schema versions defined. This can occur if a new schema is added to a catalog following an upgrade to a newer version of SQL/MX.

18) After a QMGR restart with the CONS in a suspended (SUSPEND CONS command issued) state, you will have to issue another SUSPEND CONS command prior to the RESUME CONS command. Issuing just a RESUME CONS command will result in an erroneous "ERR 1244: DIDN'T FIND ANY OBJECTS TO RESUME" message to be issued. Issuing another SUSPEND CONS command followed by the RESUME CONS command will resume the replication flow.

19) There is an issue when replicating from HPE NonStop Enscribe files out to SQL that may cause Shadowbase to erroneously drop some columns from replication in certain configurations. This may occur if all of the following are true:

- You are replicating from Enscribe files to any SQL database except SQL/MP.
- You have periodic cleanup of inactive files and schemas enabled in the Consumer. Periodic cleanup is enabled and runs every 24 hours by default.
- You are replicating from files that may become candidates for being removed from memory due to lack of activity over the cleanup period.
- The Consumer that is replicating the files has very few different schemas. This problem was uncovered in a Consumer that had two DBS's attached. The likelihood of the issue occurring drops as the number of different DBS/files increase.

A fix to this issue will be provided in the next release. In the meantime, Gravic recommends that you disable the memory cleanup when replicating from Enscribe to SQL tables (other than SQL/MP). You can disable the cleanup by setting the TACL parameter SBSRCCLEANUPRECUR to 0.

20) There is an issue with the QMGR that may cause it to stop with the message:

```
LOGIC ERROR IN C:\Release\nsb\audqmgr\TNS-E\../src/qmgr.cpp AT LINE 607:
ASSERTION  cons_seqno > last_cons_position_.get_queue_sequence_number()
FAILED. PLEASE CONTACT SHADOWBASE SUPPORT AT SBSUPPORT@GRAVIC.COM
```

This may occur if the attached Consumer goes down with no data outstanding and no data is flowing from the Collector. The QMGR will fail to restart with the message:

```
QMGR  QMGR-SRC-CJCQ1(\GRAVIC1.$CJCQ1): UNABLE TO FIND QUEUE FILE SEQUENCE
NUMBER (1) MATCHING $*.CJCQ1.QF* REQUIRED BY CONSUMER FOR RESTART
```

This can only occur if there is no data queued for the Consumer. If this error does occur, you can purge the QMGR restart file to allow it to successfully restart without data loss.

[*** End of Document *****]**