SOFTWARE RELEASE DOCUMENT (SOFTDOC)

Product:	Shadowbase
Release:	Gravic Version 7.010 HPE Nonstop Shadowbase: T1122L70^ABL (SB Repl/Guardian)
Release Date:	May 31, 2026
Copyright Notice:	Copyright Gravic, Inc. 1995 – 2026 (www.gravic.com)
File Name:	IPM7010.pdf

VERY IMPORTANT: Due to licensing changes in Version 6.700, existing Shadowbase installations running versions prior to (earlier than) 6.700 will require a new license file in order to install and run Version 7.010. This is true for any Shadowbase upgrade when the prior release is before Version 6.700 and the new release you plan to install is version 6.700 or after (more recent)…

Contact the HPE License Manager to request a new license file <license.manager@hpe.com>. DO NOT INSTALL Shadowbase Version 6.700 (or later) software when upgrading from a version prior to Version 6.700 until a new license file has been received.

NOTE: If this is a TCD delivery, please see NOTE FOR TCDS for TCD delivery information.

NOTE: This softdoc covers new features and corrected problems for Shadowbase for 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 T1122L70, 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 T1122L70.

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. In addition, visit the http://www.ShadowbaseSoftware.com web site for the list of supported platforms and databases, and the Shadowbase Software Support Policy for Software Versions by clicking here https://www.shadowbasesoftware.com/glossary-policy-and-versions/.

Table of Contents

SOFTWARE RELEASE DOCUMENT (SOFTDOC) 1
Table of Contents 1
Disclaimer 3
Note for TCDs 4
Mature and Deprecated Features 4
Previous Version End of Support Dates 6
Supported Database and Operating System Versions 7
Upgrade Considerations for Version 7.010 from 7.000 8
Special Notes for Version 7.010 9
Changes in Release 7.010 (since Version 7.000) 13
New Features 13
Problems Fixed 15
Major New Feature Descriptions 17
SBMXSERV 17
New Features 17
Sample SBMXSERV Shadowbase Configuration 19
List of Ignore, Retry, and Stop SQL/MX SQLCODEs 19
User Exit DLL Support 20
USRXDLL.h 20
LINKDLL 20
USEREXITDLL Parameter 21
USEREXITDLLLOADERROR Parameter 21
HPE Nonstop Shadowbase IBM MQ Disaster Recovery 21
Overview 21
Considerations 21
General Shadowbase Information 23
Procedures 23
Example Configurations 26
Configuring a Drain Session (SBUNIDIR) 31
SB_ADD TGTNODE 31
SB_GEN_DRAIN 32
Configuring a Drain Session (SBBIDIR) 32
SB_GEN_DRAIN 33
New and Modified Commands 33
QSETSEND Command 33
QSENDRESUME Command 35
SBBIDROS SB_ADD CONSUE 35
SBBIDROS SB_ADD CONSUE_PARAM 36
SBBIDROS SB_ADD LINK 36
New and Modified SET AUD Parameters 36
New and Modified SET COLL Parameters 37
New and Modified SET CONS Parameters 37
New and Modified SET DBS Parameters 42
New and Modified SET QMGR Parameters 47

New and Modified SET SOLVMGR Parameters 47
New and Modified SOLV AND SOLVNV Parameters 47
New and Modified Coordinator (SBCOOR) Parameters 47
RM_TIMEOUT 47
RM_TIMEOUT_USAGE_OVERRIDE 48
New or Modified EMS Messages 49
New and Modified SOLV User/EMS Messages 70
New and Modified TACL Parameters 70
New and Modified User Information Messages 71
New and Modified User Exit Functions 72
New and Modified Measure Counters 73
New and Modified ETL Toolkit Parameters 73
Validating Downloaded Files 75
Installation Instructions (HPE Nonstop Shadowbase) 76
HPE Release File Structure 78
Known Problems Remaining 79

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.

We expect customers of the Shadowbase product suite to "stay current" on Shadowbase releases. This means that you, the customer, should periodically upgrade your Shadowbase software to a newer release that is under support before support ends on your current release. For most customers, this means that you will want to upgrade while your release is in 'ACTIVE' support. Otherwise, you run the risk of not being able to get full (or even any if the release has gone 'OBSOLETE' end-of-service-life) support for the version you are running.

The Shadowbase Software Policy for Software Versions is described here: https://www.shadowbasesoftware.com/glossary-policy-and-versions/. We encourage all customers to periodically review this material and plan for periodic upgrades to their Shadowbase software. Contact Support if you need additional information.

NOTICE: Each user's experiences will vary depending on its system configuration, hardware and other software compatibility, operator capability, data integrity, user procedures, backups and verification, network integrity, third party products and services, modifications and updates to this product and others, as well as other factors. As a result, the ZDL product does not guarantee that you will not lose any data; all user warranties are provided solely in accordance with the terms of the product License Agreement. Please consult with your supplier and review our License Agreement for more information.

Note for TCDs

TCD (Temporary Code Delivery) – A software update delivered via an SPR downloadable from an FTP dropbox. A TCD is an early version, intended for customer testing only (not production usage). A TCD by definition is restricted to certain customers. Note that a "Gravic TCD" is delivered directly from Gravic, not via HPE, but otherwise has the same attributes.

A TCD is provided only to the specified customer for the purposes agreed between the customer and Gravic as to how it will be used. A TCD is provided subject to the following terms and conditions in addition to the existing written license governing the use of Shadowbase:

- A TCD is provided for evaluation and test purposes only for no more than ninety (90) days use, and is not to be used in production systems
- A TCD may not have been fully tested by Gravic, no warranties are implied as to its behavior
- A TCD is delivered directly from Gravic to the customer, it is not available from HPE/SCOUT
- As testing proceeds, iterative TCD deliveries may be necessary as issues are identified/resolved
- A TCD is temporary, after evaluation it is to be withdrawn from use by the customer
- After testing completes, a TCD may or may not subsequently be released as a Shadowbase TCF or otherwise be included in the Shadowbase product line

Please see https://www.shadowbasesoftware.com/glossary-policy-and-versions/ for additional information.

Mature and Deprecated Features

This section provides a list of mature features ("***mature features***") and deprecated features ("***deprecated features***") for HPE Nonstop Shadowbase. Gravic will continue to support ***mature features*** and will continue to include them in future releases; however, Gravic will not enhance those features further or qualify them on newer versions of operating systems and databases as they become available. If you are using a ***mature feature***, Gravic highly recommends moving off the ***mature feature*** to the suggested replacement approach as soon as possible. New installations should avoid using ***mature features***; please discuss this issue with Gravic if you feel you must use a ***mature feature***.

Gravic will also continue to support ***deprecated features*** in the current release. However, Gravic plans to eliminate the deprecated features in a future release and *strongly* recommends that customers begin the planning and implementation to move off the feature immediately. Contact Gravic if you need assistance in planning your migration.

Table 1, below, lists the features that are currently considered 'mature' by Gravic, along with the release/date where they were declared mature. Again, please note that the *mature* designation means the feature will not be enhanced and will eventually be deprecated (retired).

Table 2, below, lists the deprecated features. The deprecated features are no longer being enhanced, and are on a schedule to be retired. The table also contains a column indicating the date of the last release that will contain the deprecated functionality (the last release before this end date will be the last release to contain this functionality). Please note that release plans are subject to change without notice and delivery dates are not guaranteed. However, Gravic currently expects that all general availability releases up to and including the date of the last release will contain the deprecated feature(s).

Please see Shadowbase Software Product Release and Support Policies on www.shadowbasesoftware.com for additional information regarding the Shadowbase Software Support Policy.

Table 1 - Mature Features in HPE Nonstop Shadowbase

Feature	Description	Mature Release
AUDCOLLN	The original native mode collector, AUDCOLLN is now mature and is being replaced by the new SBCOLL collector. AUDCOLLN will continue to be maintained for critical issues and tested, but will no longer be enhanced with new features and capabilities. **We suggest customers plan to upgrade to SBCOLL as soon as they can. Contact Support if you would like assistance with this process.**	NSB 6.410
CONSTXBQ	A user exit-bound consumer that will process queue file events by use of a mapping file and batches transactions. Consider replicating queue files using the ALLOWQUEUEFILE REPLICATEKEY option along with the CONSTXB user exit-bound consumer instead if transaction batching is needed.	NSB 6.600
ENS2SQL	Tool used to convert Enscribe DDL records to SQL/MP tables, effectively replaced by SBDDLUTL	NSB 6.600
Loader	The original Shadowbase Loader (LOADER) is considered mature for usage with KREP (the key replay feature). Outside of KREP, LOADER should no longer be used and the Shadowbase Online Loader (SOLV) should be used for any loading of files or tables.	NSB 6.300 May, 2017
RDTAPE	Object program called by TACL macro used to restore audit trail file from tape to disk, made obsolete by better audit trail recovery methods.	NSB 7.010 July, 2025

Table 2 - Deprecated Features in HPE Nonstop Shadowbase

Feature	Description	Deprecated Release	Last Release Date
Non-native Consumer	The non-native version of the consumer (AUDCONS) is deprecated. Customers still using the non-native version of the consumer should immediately begin planning to move to the native version (AUDCONSN).	NSB 6.410 April, 2018	April, 2020
SBLINK	The original program used to connect SEM to a Nonstop environment, replaced by listener AUDCOM process.	NSB 6.220 May, 2016	NSB 6.320 May, 2017
SBQFILE	Program used in conjunction with replicating Enscribe queue files to target non-queue key-sequenced files, replaced with the EXACT queue file replication feature (i.e., it is no longer needed).	NSB 6.220 May, 2016	NSB 6.320 May, 2017
SQL92	Protocol used by Shadowbase components between the Nonstop and Other Servers, replaced by SQLSBV1.	NSB 6.320 May, 2017	NSB 6.410 April, 2018
Shadowbase Enterprise Manager (SEM)	Windows tool used to monitor Shadowbase environments in a GUI setting.	NSB 7.000 November, 2024	November, 2026
NSK Data Mapping	Consumer feature for implementing common data transformation operations without a user exit, replaced by SBMAP.	NSB 7.000 November, 2024	November, 2026
Format 1 TMF Audit Trails	While still supported (but limited to HPE's support), Gravic recommends that any customers still using Format 1 Audit Trails migrate to Format 2 TMF Audit Trails as soon as possible.	NSB 7.000 November, 2024	November, 2026
SBPERF	Tool for testing TCP/IP comm connectivity and speeds. Similar features such as BYPASS mode, have been added directly into Shadowbase.	NSB7.010 December, 2024	December, 2026

Previous Version End of Support Dates

With the release of this version of Shadowbase OSB version 7.010, the end of support dates for the previous version (7.000) are now published as so:

Shadowbase NSB version 7.000

Release date: December 2024
End of ***Active*** State: **May 2028**
End of ***Mature*** State: **May 2029**
End of ***Limited*** State/Obsolete: **May 2030**

For more details, refer to the Shadowbase glossary at https://www.shadowbasesoftware.com/glossary-policy-and-versions/.

Supported Database and Operating System Versions

The database and operating systems versions supported by HPE Shadowbase can be found on the Gravic Shadowbase website. Please see Supported Databases and Platforms on www.shadowbasesoftware.com for additional information regarding Shadowbase support for databases and operating systems. Please note that, as the tables show, there may be many more versions and combinations supported by Shadowbase beyond the official list. Contact Support if you need further clarification for specific versions and combinations.

Upgrade Considerations for Version 7.010 from 7.000

This section is intended to give users an overview of changes that might impact their existing Shadowbase environments when they upgrade to this version. For this list of topics, refer to the details elsewhere in this document to understand whether this might have an impact on your environment.

1) SBMXSERV no longer requires a special user exit in order to be used with the consumer.
    a. The files LINKMXCL and SBMXCLIN have been removed.
    b. Instead of using USEREXITID to determine which DBSes should be replicated through SBMXSERV, the DBS parameter SBMXSERVENABLED should be used.
    c. Instead of using USEREXITPARAM to configure the SBMXSERV process, the CONS parameters prefixed with SBMXSERV should be used.

2) SBUNDROS and SBBIDROS now set the “Transaction Processing” flag to N by default whenever configuring DOC Writer objects.

3) The new NODBSEVENTACTION parameter defaults to STOP which may cause existing environments to fail if they are misconfigured and data loss is detected (e.g., if events sent by the Collector to the Consumer are automatically discarded by the Consumer, most likely due to a configuration error).

4) The new HASHBUCKETSKIPACTION parameter defaults to STOP which may cause existing environments to fail if they have HASHBUCKETCOUNT or HASHBUCKETID misconfigured and data loss is detected.

5) Some file validation behavior has been fixed (problem fixed #56) which may cause existing environments to detect configuration issues that were not detected in previous versions.

Special Notes for Version 7.010

1) 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.

2) 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.

1) For those that have user exits, you must use this version's USRXLIBN or USRXLIBX API library included in the distribution subvolume. If you have copies of the 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 USRXLIBN or USRXLIBX with this version of Shadowbase.

2) 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.

3) 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 (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.

4) 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.

5) 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.

6) 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.

7) 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 observed that TMF may save updates of large rows (e.g., those greater than 4k in size) as a delete event of the old row followed by an insert of the revised row. Shadowbase will replay this sequence on the target as a delete followed by an insert as well. If cascading deletes are configured on the target, 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.

8) The EMS subsystem version has been updated with this version from V68 to V70 to match the two most significant digits of the current Shadowbase release version id.

9) 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. Contact Support if you are using multi-ported DOC Writers in any configuration, especially important if this issue applies to you and you need to discuss options.

10) SBHADW configuration scripts now sets the TACL parameter SBCOLLAUTOADVMODE in the AMON script to a default value of 0. This parameter used to default to 1.

11) If a Shadowbase process on another node does not respond to a STOP request from the AUDMON, the AUDMON will attempt to kill the process using the PROCESS_STOP_ system call. This attempt will often fail if the remote process is

PROGID'd, as all of the Shadowbase processes are. Once the AUDMON has passed its STOPPROCESSMAXRETRIES threshold, it will assume the process has stopped and continue on as usual.

When the environment is started up again, the if the remote process remains running, the restart of the remote process will fail to start because it still exists from before. This problem can be prevented by creating a SAFEGUARD Access Control List (ACL) for the process to allow access to the user that AUDMON is PROGID'd under. Once the user has been whitelisted with the ACL, AUDMON is able to properly kill the remote process.

12) The default value for SBCOLLMAXFNAMES has increased from 131,070 to 1,000,000. See the SBCOLLMAXFNAMES entry in the New and Modified TACL Parameters section for more info.

13) This release of Shadowbase includes SBCREATP, a utility for converting SQL/MP schemas into various target SQL formats, such as Oracle and SQL Server. The SBCREATP utility is being provided as a ***Shadowbase TCD (Temporary Code Delivery).*** **NOTE**: This utility is made available solely subject to the terms enclosed in this SOFTDOC. Do not use the utility if you do not agree with the terms.

Please note the following:

- This TCD is an early software version, intended for specific customer/partner testing purposes only (not production usage).
- This TCD is provided only to the specified customer/partner for the purposes agreed between the customer/partner and Gravic/HPE as to how it will be used.
- This TCD is Confidential, and is not to be discussed, shared or installed outside of the specified customer/partner facilities/personnel.
- This TCD may or may not subsequently be released as a Shadowbase TCF or otherwise be included in the Shadowbase product line.
- This TCD is provided subject to the above and the following terms and conditions in addition to the existing written license governing the use of Shadowbase software:
  - TCD is provided for evaluation and test purposes, and is not to be used in production systems.
  - TCD may not have been fully tested by Gravic, no warranties are implied as to its behavior or fitness for a particular purpose.
  - TCD is delivered as part of the normal Shadowbase release from HPE/SCOUT.
  - TCD is temporary, after evaluation it is to be withdrawn from use by the customer/partner.

For additional information, please see https://www.shadowbasesoftware.com/glossary-policy-and-versions/.

NOTE: Usage of this SBCREATP documentation, software and/or validating password(s) and any other documentation, software, and password(s) for Product is only authorized under the terms of a valid end-user or trial license agreement. If you or your

company have not paid license and maintenance fees to Gravic, Inc. or an authorized Shadowbase reseller for use of the software product for which this and any other documentation, software or passwords are designed, you may only use the product for evaluation purposes for a limited period of time (up to 60 days). Unless you have executed a written Product usage agreement with Gravic, Inc. or an authorized Shadowbase reseller, usage of this license file or any other documentation, software or password in this case connotes acceptance by your company of the Shadowbase Confidential Testing and Evaluation Agreement. A review copy of this agreement can be found on the Shadowbase website at https://www.shadowbasesoftware.com/wp-content/uploads/2015/06/ShadowbaseEvaluAgmt.pdf. In the event that your company is licensing the Product for production use, has paid license and maintenance fees to Gravic, Inc. or an authorized Shadowbase reseller, and does not yet have a written end-user license agreement with Gravic, Inc. or an authorized Shadowbase reseller, usage of this license file and any other documentation, software and/or password(s) connotes acceptance by you and your company of the terms of the shrink wrap license which can be found at https://www.shadowbasesoftware.com/wp-content/uploads/2015/06/ShadowbaseTCAgreement.pdf.

Please contact Shadowbase Support if you have any questions.

Changes in Release 7.010 (since Version 7.000)

New Features

1) Added a new DBS parameter PRIPARTONLY to filter secondary partitions. See the *New and Modified SET DBS Parameters* section for more information.
2) Wildcarding has been made more flexible for TARGETFILE and MAPTOFILE. See the *New and Modified SET DBS Parameters* section for more information.
3) Added new DBS parameters, CATCHALL and CATCHALLMODE, that can be used to configure DBSes that only match files that don't match other DBSes. See the *New and Modified SET DBS Parameters* section for more information.
4) The consumer will now log a message and abend if it hits an error while trying to resolve the target filename.
5) New CONS parameter NODBSEVENTACTION which handles situations where an event appears on the target where the source file does not belong to any of the DBSs. This can occur when the MAPTOFILE is incorrectly configured.
6) Added the ability to run the AUDCONXN SQL/MX Consumer on the target system during SQL/MX replication, via the use of MAPTOFILE and a table exactly matching the source table on the target system.
7) Added the new COLL parameter HASHBUCKETSKIPACTION to control behavior when an event is skipped due to it not matching any HASHBUCKETID value. See the *New and Modified SET COLL Parameters* section for more information.
8) When using SBCOLL, ALLPARTITIONS can now be used with wildcarded DBSes.
9) SBCOLL now supports replicating enhanced key sequenced (EKS) files.
10) SBCOLL now supports wildcard characters in the MAPTOFILE parameter.
11) SBCOLL now supports the MAPTOPRIPTN option of the PURGEDATAOPTIONS DBS parameter.
12) The PULSE command and PULSERATE parameter can now be used with SBCOLL.
13) ZDL functionality is enhanced:
    a. ZDL can now be used with bidirectional replication.
    b. A new timeout parameter for the Coordinator, adding support for configuring and controlling the Coordinator's resource manager being timed out by TMF in the event it is held up too long.
14) QMGRs are now required for all replication threads for ZDL enabled collectors. For proper ZDL operation, the QMGRs should be configured to run on the target system.
15) The Queue Manager is now capable of stopping and resuming processing at a particular time, TMF address, or EOF. See the QSETSEND and QSENDRESUME in the *New and Modified Commands* section for further information.
16) Added new QMGR parameters, QFILEROLLTIMEMODE and QFILEROLLTIME, that can be used to roll queue files based on time. See the *New and Modified SET QMGR Parameters* section for more information.
17) Once the QMGR reaches queue file sequence number 999999, it will now wrap back around to sequence number 1.
18) The QMGR will now only open queue files that it is writing to or reading from.
19) QMGRs can now be configured without a collector. QMGRs without a collector can be used to replicate data stored in a queue file without having to start a collector. This

is particularly useful when a catastrophic failure takes out the source system and is no longer accessible.

20) Autorestart logic related to QMGRs has been improved. AUDMON will no longer restart all QMGRs when one fails. During an autorestart, if a consumer that is attached to a QMGR fails to start, AUDMON will still try to restart the QMGR and collector.
21) ETL Toolkit support for MX data types VARCHAR2, NCHAR VARYING, BINARY, and VARBINARY.
22) ETL Toolkit recognizes MX table name as a section header name in ETLCONF.
23) New option for ETL Toolkit parameter JSONSTRINGFORMAT, UTF8, which allows support for standard extended ASCII encoding when in JSON format.
24) Added ETL Toolkit parameter KAFKANOWAITTIME which can be used to specify the nowait time when opening and writing to the uLinga for Kafka process. Defaults to 10 seconds.
25) AUDMON will no longer send expedited PULSE commands to the Collector that are not replicating events.
26) The ability to enter the commands NETUSERNAME and NETUSERPASSWORD into AUDCOM without having value echoed back to the screen. More information can be found in the *New and Modified SET CONS Parameter*.
27) AUDMON will no longer prevent RESOLVELOCKS, RESOLVETRANS, and ABORTTRANS from being enabled for entry-sequenced files if ENTRYSEQEXACT is ON.
28) Removed the EXCLUSIONMSG parameter from SOLV, SOLVMX, and LOADER.
29) C++ user exits are now supported and can be compiled using RUNCCOMP or LINKDLL. Refer to the comments in the files for directions on how to compile for C++.
30) In SBMAP, the SBHASH function can be used in any expression, instead of just as part of the SBWHERE clause.
31) A message is now written to the IOTRACE file right before SBMAP processing occurs.
32) Column names that are reserved words in SBMAP can now be referenced by enclosing the reserved word in curly braces ({}).
33) Added the ability to configure consumptive consumers for a SQL/MX source with SBBIDROS. See the *New and Modified Commands* section for more information.
34) Added the ability to generate additional configuration files with SBUNIDIR and SBBIDIR to drain target QMGRs when the source node is not accessible. See the *Configuring a Drain Session (SBUNIDIR)* and *Configuring a Drain Session (SBBIDIR)* sections for more information.
35) Options were added when configuring Shadowbase to not store the STP network username and password, and instead prompt at runtime, and prevent display of configured values from AUDCOM.
36) SBUNDROS and SBBIDROS now set the Transaction Processing setting of DOC Writers to ‘N’ by default.
37) The SB_PASSWORD_PROMPTING option for SBUNDROS and SBBIDROS has been updated to additionally prompt for the NETUSERPASSWORD CONS param and corresponding DOC Writer/Direct Writer password.

38) The SQL/MX Wildcard Compare script, SBCMXWLD, will now show whether an exclusion occurs based on an exact or templated exclusion match.

Problems Fixed

1) Fixed an issue in SBMAP preventing minus signs from being parsed correctly in all cases.
2) In SBMAP, quotation marks in string literals are now escaped correctly. Quotation marks matching the kind surrounding the string literal can now be specified by repeating the quotation mark. Quotation marks not matching the kind surrounding the string literal are no longer escaped.
3) The consumer parameter PULSERATE now functions correctly for consumers that are connected to a QMGR.
4) STATUS QMGR and STATUS CONS will now correctly display the EMS collector's process name.
5) SBCOOR will now properly close the collector if it rejects the open.
6) ETL Toolkit now supports MX FLOAT and DOUBLE PRECISION data types.
7) AUDMON will no longer fail when using a SHADTYPE file and BASE24, ETL Toolkit, or SBMAP processing is enabled,
8) Fixed an issue in ETL Toolkit where COLDELIM would not be added if the first column was NULL.
9) Fixed an issue in ETL Toolkit where RECDELIM and COLDELIM would not log the value NULL correctly.
10) Fixed an issue in ETL Toolkit to properly abend on timer errors.
11) Fixed an issue in ETL Toolkit to properly display large numeric values when using JSON format.
12) Fixed an issue displaying the ALTERS parameter with SHOW DBS.
13) Fixed an issue with FUPSERVER that could cause the consumer to rarely trap on startup.
14) Fixed an issue where the consumer would fail to log the correct error number when cleaning up old SOLV marker records.
15) AUDCOLLN no longer traps when an expedited PULSE is entered.
16) Fixed an issue where AUDMON would not log an error when a PULSE command did not generate any pulses.
17) The consumer will no longer close EMS after logging that it has successfully connected to STP.
18) When SBCOLL abends while looking for audit that has rolled off system, it will now log an additional message and not generate a saveabend file.
19) The TARG I/O fields in the STATS CONS output now accurately displays the number of I/O operations against the target tables.
20) Fixed an issue that sometimes caused SBCOLL to not replicate internal sequence generator events down some threads.
21) Fixed some AUDQMGR messages that labelled ADT positions as "AUX", even when they were on the MAT audit trail.
22) SBCOLL will now cleanly stop on startup if the ADTSTARTSEQLO parameter is not an existing audit trail file.

23) The consumer no longer upshifts commands from SBFUP.
24) SBCOLL will now fall back to asynchronous processing when the REPLICATION parameter is set to SUSPEND or SKIP.
25) The HELP text for the INFO command has been fixed to show that an asterisk can be used to display the values for all objects of a type.
26) AUDMON will no longer use the incorrect valid range of values of LABELCACHEDEPTH for SBCOLL.
27) KREP will now properly use $0 as the default EMS collector.
28) SQL/MX internal sequence generator events will now correctly be replicated, even if MXTARGETTABLE is not present.
29) Fixed the error message when a direct STP server consumer is started with QMGRNAME configured.
30) SBBIDROS will now correctly output a warning when the SB_ADD CONSOS argument <shadtrs-tab-vol> is a virtual drive.
31) The consumer will no longer abend if it processes an event on a file from a node that has gone down.
32) ETL Toolkit parameter KAFKAPROCESSNAME now allowed to specify a node name in process name.
33) SBQFUTL no longer displays a higher EVENTS count from SELECTED when set to DISPLAY ALL.
34) SBQFUTL RESET command properly resets VERBOSE setting.
35) SBQFUTL properly handles comments "==" when reading from an obey file.
36) SBQFUTL displays a meaningful message when entering an invalid date and/or time value.
37) SBQFUTL now uses '-' instead of '/' for displaying event header time values.
38) SBQFUTL now uses '.' Instead of ':' for separating seqno and rba in event header.
39) Consumer will display a message and stop when configured for Enscribe to SQL (E2S) replication with ETL Toolkit enabled.
40) The banner that gets output when starting a program has been made consistent across all programs.
41) SBSYNCMN will now immediately close the processes it opens when they are stopped.
42) SBUNDROS now correctly uses the <trans_proc> argument of SB_ADD CONSOS.
43) The consumer will no longer abend while trying to log a varchar to the IOTRACE file after a SQL error.
44) The DBS parameter ALLUPDATES now works as expected when using SBCOLL.
45) When determining whether to replicate a file to a consumer, SBCOLL will no longer just look at the parameters of the first DBS that matches the file.
46) SOLVMX can now handle reserved word column names in primary keys and partitions.
47) The message flags ((N)O MATCH, and (C)RITICAL) in the Wildcarded Compare scripts were adjusted to properly reflect the expected states based on the completion code returned from the compare programs.
48) Fixed an issue where SOLV would sometimes fail to parse a partition's first key value if the last field was a DATETIME or INTERVAL.
49) SBQFUTL now handles number arguments for FC/!/? commands correctly.

50) SBQFUTL now only stores up to 100 commands in its history.
51) Fixed an issue where SBQFUTL could trap when trying to create the OUT file if the loaded queue files were very large.
52) SBCOLL will no longer fail with the error "INCONSISTENT HASHBUCKETCOUNT DEFINITIONS" if a SOLV marker file or control file is included by a DBS.
53) FETCHSOURCE (ON,AUDIT) will no longer incorrectly change bytes before the key in key-sequence files where the key offset is 1 or greater.
54) Fixed an issue where SBQFUTL could run out of memory while writing to extremely large output files.
55) Fixed an issue where the consumer could incorrectly log that the source data length was greater than the target column size when processing a delete after a varchar column had been added to the table.
56) Fixed AUDMON's file attribute and parameter validation during ADD DBS processing when FASTSTART is ON to only check for sourcefiles on the same node as AUDMON. Previously, it was incorrectly comparing the node of the sourcefile to that of the Consumer process instead when determining whether to perform validation. This validation will also now occur for expand configurations when the sourcefile is on the same node as AUDMON.

Major New Feature Descriptions

SBMXSERV

An option for replicating SYSKEY columns *EXACTLY* for key-sequenced SQL/MP tables is via SBMXSERV. The consumer creates a separate process that executes inserts, preserving the source SYSKEY value in the target row's SYSKEY. Previously, the consumer required using a special user exit to interface with SBMXSERV, but that is no longer the case.

New Features

1) SBMXSERV is now configured with consumer and DBS parameters. See the *New and Modified SET DBS Parameters* and *New and Modified SET CONS Parameters* sections for more information.
2) The consumer now handles errors returned by SBMXSERV similarly to how it handles errors from the consumer itself, either ignoring, skipping or abending for the error.
3) The consumer will now consider all SQLCODEs returned by SBMXSERV, instead of just the first.
4) SBMXSERV can now be configured to run additional CONTROL TABLE and CONTROL QUERY DEFAULT commands using the SBMXSERVCONTROLTABLE DBS parameter and SBMXSERVCQD CONS parameter.
5) If IOTRACE is enabled, the consumer will now write the SQL/MX insert statement and SQL/MX errors to the IOTRACE file.
6) Added SBMXSERV process info to STATUS AUD and STATUS CONS:

STATUS AUD

```
…
      CONS-DIR-KTCN     \VIV1.$KTCN       CONS  STARTED    2024-10-29:15:07:45
                                        LTS (LAST EVENT):  NONE RECEIVED
        MXSERV-KTMXS    \VIV1.$KTMXS      MXSV  RUNNING
…
```

STATUS CONS

```
…
  CONSNAME          PROCESS           STATE      TRANS BUSY             MSGSEQ
  ----------------  ----------------  ---------  ----------  -----------------
  CONS-DIR-KTCN     \VIV1.$KTCN       RUNNING             0                   0
    MXSERV-KTMXS    \VIV1.$KTMXS      RUNNING
…
```

7) Added SBMXSERV statistics to STATS CONS:

```
…
    SBMXSERV INFO:
       MSGS SENT  AVG SENT BYTES  AVG RECV BYTES
       ----------  --------------  --------------
                1              22              17
       STMTS BUILT    BUILD TIME   %BUILD  AVG BUILD
       -----------  ------------  -------  ---------
                 0  00:00:00.000     0.00      0.000
        STMTS SENT      I/O TIME     %I/O    AVG I/O
       -----------  ------------  -------  ---------
                 0  00:00:00.000     0.00      0.000
           LAST STMT SENT TIME
       -----------------------
       0000-00-00:00:00:00.000
…
```

- MSGS SENT - # of messages sent to SBMXSERV. This counts all messages including pings
- AVG SENT BYTES - Average size of a message sent to SBMXSERV
- AVG RECV BYTES - Average size of a reply from SBMXSERV
- STMTS BUILT - # of statements built to be sent to SBMXSERV
- BUILD TIME - Total time spent building statements to be sent to SBMXSERV
- %BUILD - Percent of stats duration spent building statements to be sent to SBMXSERV
- AVG BUILD - Average time spent to build a statement to be sent to SBMXSERV
- STMTS SENT - # of statements sent to SBMXSERV. This will differ from STMTS BUILT if an insert is retried.
- I/O TIME - Total time spent sending statements to SBMXSERV
- %I/O - Percent of stats duration spent sending statements to SBMXSERV
- AVG I/O - Average time spent sending a statement to SBMXSERV
- LAST STMT SENT TIME - The last time a statement was sent to SBMXSERV

8) The consumer will now use RESOLVELOCKS to handle SBMXSERV locking errors if configured.
9) SBMXSERV can now be used for non-like-to-like replication.

10) SBMXSERV can now be used with user exits and SBMAP.
    a. USRXEXCEPTION can be used, but the SQL/MX SQLCODE must be specified for SBSETEXCEPTIONCODES. USRXEXCEPTION will be entered if any of the returned SQL/MX SQLCODEs are set as exception codes.
    b. If SBSETTARGET is used to override the target, SBMXSERV will not be used.

Sample SBMXSERV Shadowbase Configuration

Here is a sample config showing how to configure a single SQL/MP SYSKEY table using SBMXSERV:

```
LOAD/KEEP 1/SBUNIDIR

SB_CLEAR ALL
SB_SET SB_CONFIGURATION_TYPE DIRECT
SB_SET SB_TRACKTX 1

SB_ADD SRCNODE \SRC \SRC.$DATA1.SB7010 \SRC.$DATA1.SBDATA ~[#MYTERM~]
\SRC.$DATA1.SB7010.SHADPASS $SBCOL 0 1 100 $SBMON 0 1 100
SB_ADD TGTNODE \TGT \TGT.$DATA1.SB7010 \TGT.$DATA1.SBDATA

SB_ADD CONS $SBCON 1 0 100 \TGT.$DATA1.SBDATA.TIDFIL1
SB_ADD CONS_PARAM * * SBMXSERVPROGRAM \TGT.$DATA1.SB7010.SBMXSERV
SB_ADD CONS_PARAM * * SBMXSERVPROCESS \TGT.$SBMXS
SB_ADD CONS_PARAM * * SBMXSERVCPU 1
SB_ADD CONS_PARAM * * SBMXSERVPRI 99

SB_ADD DBS DBS1 \SRC.$DATA1.DATSVOL.TESTSYSK $SBCON
\TGT.$DATA1.DATSVOL.TESTSYSK - - -
SB_ADD DBS_PARAM * * SBMXSERVENABLED ON

SB_GEN
```

This will replicate an SQL/MP syskey table called TESTSYSK from \SRC to \TGT, preserving the SYSKEY value.

List of Ignore, Retry, and Stop SQL/MX SQLCODEs

The following SQL/MX SQLCODEs are either ignored, retried or stopped for when they are encountered by SBMXSERV. This can be modified via the SBMXSERVIGNORECODES, SBMXSERVRETRYCODES, and SBMXSERVSTOPCODES consumer parameters.

SQL/MX SQLCODEs	Default Action
1123	STOP
2108	STOP
3001-3402	STOP

4025	STOP
4030	STOP
4039	STOP
4069	STOP
4122	STOP
7001	STOP
8402-8422	STOP
8574	RETRY
8575	IGNORE
8576	RETRY
8808-8809	STOP
15001	STOP

Translated SQL/MX SQLCODEs

The following SQL/MX SQLCODEs are translated into equivalent HPE Nonstop errors. The behavior for these errors can be modified with the IGNOREERROR, RETRYERROR, and STOPERROR parameters.

SQL/MX SQLCODEs	HPE Nonstop Error
100	11
8102	10
8605	78

User Exit DLL Support

A new method of including User Exits is now available via Nonstop DLLs. This option can replace the current method of compiling and binding the User Exit into the consumer. For User Exit DLL support to work, the User Exit must be compiled using the LINKDLL TACL script. User Exit DLL support is not available at the moment for COBOL user exits.

There are a few new files and functions to support DLL functionality:

USRXDLL.h

This file is a new header file that should be used in place of USRXLIB.h when creating a User Exit. This new header file allows the User Exit API functions to be accessible by the new DLL.

LINKDLL

This TACL script is used to both compile a User Exit and convert the compiled object into a DLL. This script by default is set up to compile and build for TANDEM_FLOAT, but can easily be changed to work for IEEE_FLOAT by following comments located in the script. The output of this script will be a DLL, USRXDLL, which should be used in the USEREXITDLL Consumer parameter.

USEREXITDLL Parameter

This parameter specifies the name of the User Exit DLL that is compiled using the LINKDLL TACL script. The LINKDLL will compile a User Exit along with converting it to a DLL. The USEREXITDLL file will be dynamically loaded into the Consumer on startup.

USEREXITDLLLOADERROR Parameter

This parameter specifies how the Consumer should handle a symbol not found error when loading the User Exit DLL. This can occur when one or multiple User Exit entry point functions are missing from the DLL. The options are to display a message and stop, or display a message and continue processing.

HPE Nonstop Shadowbase IBM MQ Disaster Recovery

Overview

This manual serves as a guide on how to configure HPE Shadowbase to replicate an IBM MQ for HPE Nonstop environment. It includes procedures to follow for setting up an Active/Passive replication for MQ, failing over to a backup MQ installation, and reversing the flow of replication. Along with these procedures, setup instructions and example configurations can be found in this manual. This manual focuses on the Shadowbase configuration needs and procedures for MQ Disaster Recovery. For more MQ related information on MQ Disaster Recovery, refer to the IBM document *Disaster Recovery – IBM MQ for HPE Nonstop*.

Considerations

File System Names

IBM MQ for the HPE Nonstop requires all objects it uses to be the same in both the source and target environments. This includes data volume names, subvolume names, file names, TCP/IP process names, etc. If any of these are different or need to be changed when failing over from the Active to the Passive MQ environment, please see the IBM document *Disaster Recovery – IBM MQ for HPE Nonstop* for how to do this.

Safeguard Permissions

Due to file permissions on IBM MQ created files, proper security settings must be configured to allow HPE Shadowbase to replicate MQ files. The Shadowbase user must be allowed to create, read, alter, and purge MQ files. Here is an example of using HPE Nonstop Safeguard ACL's to achieve the proper security settings. To do this, log in as

SUPER.SUPER (or a user that has full access to Safecom) and open up Safecom. From there this command can be entered:

```
> safecom
SAFEGUARD COMMAND INTERPRETER - T9750L01^AIP - (31MAR2023)
SYSTEM  \NS2SQL
= ADD SUBVOLUME $DATA.<mq-subvol>, ACCESS \*.255,<sb-user-
id> (R,W,E,P,C,O) ; \*.50,<mqm-user-id> (R,W,E,P,C,O)
```

It is also important to check that Safeguard is configured to allow ACL's for subvolumes. This can be checked and enabled by issuing the following commands:

```
= info safeguard
…
…
CHECK-SUBVOLUME        = OFF
= ALTER SAFEGUARD CHECK-SUBVOLUME ON
```

If permissions are not properly set for every subvolume that HPE Shadowbase will be replicating, security errors could occur when creating or altering an IBM MQ file on the target. To avoid hitting these errors, ensure that every subvolume of the HPE Shadowbase configuration is accounted for.

IBM MQ Queue Migration and Partitioning

IBM MQ for HPE Nonstop allows for IBM MQ Queues to be migrated to different locations and/or partitioned across several volumes. For HPE Shadowbase to properly replicate these migrations and partitions, the subvolumes that these new files will be created on must be included in the HPE Shadowbase configuration. Since IBM MQ randomly creates the names of these files, the subvolume must be wildcarded. Refer to the *Example Configuration* section for an example on how to configure Shadowbase replication to pick up these files.

Starting IBM MQ on the Backup Node

Warning: IBM clearly states in their manuals that the IBM MQ product can only be configured in a uni-directional Active/Passive architecture for disaster recovery on the HPE Nonstop. Attempting to configure the MQ product in a bi-directional Active/Active architecture should not be attempted, as doing so may lead to a corrupted MQ environment that cannot be started nor accessed, and lead to queue data loss.

In an Active/Passive Replication environment, starting the backup IBM MQ Queue Managers to verify that changes are on the target is possible but must be done very carefully. Here are few steps that should be followed when starting the backup:

1. When attempting to start a backup IBM MQ Queue Manager, ensure that the source Queue Manager has been **stopped**. Having both Queue Mangers running at the same time can corrupt that Queue Manager resulting in data loss.
2. When verifying backup IBM MQ Queue Managers, Queues, and other objects, it is important to **not change any of the data** in these Queues or attributes of any of these objects.
3. Please refer to the IBM MQ documentation for more information.

EXCLUSIONMODE DBS Parameter

In order to ensure that the MQ application does not make changes to any MQ Queue Managers or Queues while Shadowbase has the files opens, the Shadowbase DBS parameter EXCLUSIONMODE should be set to EXCLUSIVE. This will prevent any application from opening the file while Shadowbase has it open as the target for replication. Using the consumer parameter FILECLOSEDELAY and setting it to 0 will keep the files open at all times once an event occurs against them. How to set these parameters is shown in the Example Configuration section.

General Shadowbase Information

Shadowbase is controlled using Shadowbase operation scripts that are created when users obey the Shadowbase configuration file. See the Example Configuration section for an example Shadowbase configuration file with an MQ environment.

The following operation scripts are created:

```
SRconf   - Shadowbase configuration for source node
SRstrtsb - Start source Shadowbase
SRamon   - Start AUDMON
SRcu     - Clean up
SRstrtdr - Start direct CONS and collector
SRstopdr - Stop collector and direct consumers
SRrun    - Run Shadowbase
SRstopsb - Stop Shadowbase
SRSHUTDN - Shadowbase shutdown
SRload   - loads data from the source node
SRsolvld - loads data with solv from the source node
```

These scripts are referenced in the procedures below.

Procedures

Active/Passive Replication

The first step in setting up a Disaster Recovery environment for IBM MQ is to set up a working Active/Passive replication environment. HPE Shadowbase will replicate the creation of IBM MQ Queue Managers, Queues and other objects to a target backup system. For this to happen correctly there is a list of steps that should be followed when creating an MQ environment, and when making changes to an existing MQ environment.

When first creating an IBM MQ environment, here is a list of steps that should be followed:

1. Ensure that proper permissions have been created for all subvolumes that will contain IBM MQ files. These permissions should allow the user of HPE Shadowbase to create, read, alter, and purge IBM MQ files. Refer to the section *Safeguard Permissions* for more information on this.
2. Configure HPE Shadowbase to include all locations where IBM MQ files may be created. Since MQ creates most of these files with random names, these locations will need to be wildcarded. It is important to have every possible location included in the configuration to avoid missing any files created by MQ.
3. After HPE Shadowbase has been properly configured for IBM MQ, start Shadowbase to begin replication.
4. Once replication has begun, the IBM MQ objects can now be created and configured accordingly. Do not create any MQ objects before HPE Shadowbase has begun replicating. Failing to do so will result in Shadowbase missing creates for these files and not having them created on the backup system.

Note: if the IBM MQ environment already exists before adding Shadowbase to the environment, make sure the Shadowbase configuration and security matches the requirements listed above. In this case, the user will be responsible to PAK and UNPAK the MQ environment to the target before enabling Shadowbase replication. This type of procedure is detailed in the *Procedures: Failover Scenario* section.

If changes need to be made to an IBM MQ environment that require changes to the HPE Shadowbase configuration, then the following steps should be taken:

1. Stop any active IBM MQ Queue Managers or processes.
2. Stop HPE Shadowbase replication.
    a. RUN SRSTOPSB
3. Make changes to HPE Shadowbase configuration. Ensure that any new IBM MQ changes will be picked up by the existing or adjusted/new Shadowbase configuration.
4. Ensure that proper permissions have been created for all subvolumes that will contain new IBM MQ files. These permissions should allow the user of HPE Shadowbase to create, read, alter, and purge IBM MQ files. Refer to the section *Safeguard Permissions* for more information on this.
5. Start HPE Shadowbase replication.
    a. RUN SRSTRTSB

6. Start IBM MQ objects and make desired MQ configuration changes or creation of new objects.

Failover Scenario

Once an Active/Passive environment is configured, a Disaster Recovery environment can also be configured. This process will involve changing the Passive node (backup) to the Active node as part of the failover process. Once the original Active node is up again, replication can begin flowing in a reverse order. Here is a list of steps to properly failover in a disaster recovery scenario:

1. Stop all IBM MQ active Queue Managers and processes on the Active node (assuming the Active node is still accessible/exists).
2. Issue the SUSPENDUPD COLL * command from Audcom
    a. From TACL:
        i. Run $sb.sb7000.audcom $MQD1
    b. From Audcom:
        i. SUSPENDUPD COLL *
3. Make sure all Consumers are in SUSUPENDUPD state
    a. From Audcom:
        i. STATUS AUD
4. Issue the RESUMEUPD COLL * command from Audcom
    a. From TACL:
        i. Run $sb.sb7000.audcom $MQD1
    b. From Audcom:
        i. RESUMEUPD COLL *
5. Stop HPE Shadowbase replication.
    a. From TACL:
        i. RUN SRSTOPSB
6. Configure HPE Shadowbase replication to flow in the reverse order. This replication should begin from the moment the previous Active node was stopped. This can be done by setting the Collector parameter ADTSTARTTIME to the time when replication was stop. Refer to the section *Failover Configuration* for how to configure this.
    a. Make sure the Collector Restartfile does not exist before starting Shadowbase using ADTSTARTTIME. By default, the Collector Restartfile will be created in the Shadowbase Data directory. Use the SRCONF to find the Collector's Restartfile.
7. Start HPE Shadowbase replication.
8. Start IBM MQ Queue Managers and processes on what was previously the backup node, but now Active node.

If replicating to a clean system and not the original Active system then the following steps should be taken:

1. Stop all IBM MQ active Queue Managers and processes on the Active node.
2. Issue the SUSPENDUPD COLL * command from Audcom
    a. From TACL:
        i. Run $sb.sb7000.audcom $MQD1
    b. From Audcom:
        i. SUSPENDUPD COLL *
3. Make sure all Consumers are in SUSUPENDUPD state
    a. From Audcom:
        i. STATUS AUD
4. Issue the RESUMEUPD COLL * command from Audcom
    a. From TACL:
        i. Run $sb.sb7000.audcom $MQD1
    b. From Audcom:
        i. RESUMEUPD COLL *
5. Stop HPE Shadowbase replication.
6. Configure HPE Shadowbase replication to point to the new system. This replication should begin from the moment the previous Active node was stopped. This can be done by setting the Collector parameter ADTSTARTTIME to the time when replication was stop. Refer to the section *Failover Configuration* for how to configure this.
    a. Make sure the Collector Restartfile does not exist before starting Shadowbase using ADTSTARTTIME. By default the Collector Restartfile will be create in the Shadowbase Data directory. Use the SRCONF to find the Collector's Restartfile.
7. Ensure that proper permission have been created for all subvolumes that will contain new IBM MQ files. These permissions should allow the user of HPE Shadowbase to create, read, alter, and purge IBM MQ files. Refer to the section *Safeguard Permissions* for more information on this.
8. PAK all IBM MQ files that are being replicated by HPE Shadowbase and transfer them to the new system. From there UNPAK all the files.
9. Start HPE Shadowbase replication.
10. Start IBM MQ Queue Managers and processes on what was previously the backup node, but now Active node.

Example Configurations

Active/Passive Configuration

Below is a sample Active/Passive HPE Shadowbase configuration. This a fairly simple base configuration to showcase the necessary configuration parameters to allow for Shadowbase to properly replicate an IBM MQ environment. The below TACL script uses

the Shadowbase SBUNIDIR script to configure Shadowbase. For more information on SBUNIDIR refer to the *HPE Shadowbase Tools and Utilities Manual* and for information on Shadowbase configuration parameters, refer to *HPE Shadowbase Command Reference Manual*.

```
?tacl macro

#frame
#push sb_source_node sb_target_node
#push sb_source_system_install sb_target_system_install
#push sb_source_system_data sb_target_system_data
#push mq_source_install mq_target_install mq_source_qmgr mq_target_qmgr

== Source and target nodes.
#set sb_source_node \PRIM
#set sb_target_node \BACK

== Locations of source and target Shadowbase installs.
#set sb_source_system_install [sb_source_node].$SB.SB7000
#set sb_target_system_install [sb_target_node].$SB.SB7000

== Locations on source and target where Shadowbase will be run from.
#set sb_source_system_data [sb_source_node].$sb.sbdata
#set sb_target_system_data [sb_target_node].$sb.sbdata

== Locations of source and target MQ installs.
#set mq_source_install [sb_source_node].$SSD1.MQDR
#set mq_target_install [sb_target_node].$SSD1.MQDR

== Locations of source and target MQ Queue Managers.
== Repeat for each Queue Manager that should be replicated.
#set mq_source_qmgr [sb_source_node].$SSD1.MQDRQ0
#set mq_target_qmgr [sb_target_node].$SSD1.MQDRQ0

load /keep 1/ [sb_source_system_install].SBUNIDIR

SB_CLEAR ALL
SB_SET SB_CONFIGURATION_TYPE DIRECT
SB_SET SB_CONFIG_FILE %0%
SB_SET SB_TRACKTX 1

SB_ADD SRCNODE [sb_source_node] [sb_source_system_install] [sb_source_system_data]
~[#MYTERM~] [sb_source_system_install].SHADPASS $MQLL1 1 0 140 $MQD1 0 1 90

SB_ADD TGTNODE [sb_target_node] [sb_target_system_install] [sb_target_system_data]

SB_ADD CONS $MQCS1 0 1 130 [sb_target_system_data].TIDFIL1

== This collector parameter prevents replication from stopping when deleting MQ
Queue
== Managers
SB_ADD COLL_PARAM GETRECADDRERROK ON

== This consumer parameter will keep the target files open at all times
== once there is an event for that file.
SB_ADD CONS PARAM * * FILECLOSEDELAY 0

== Adds the three files that needed to be replicated from the MQ install to
replication.
SB_ADD DBS DBS1 [mq_source_install].AMQMQINS $MQCS1 [mq_target_install].AMQMQINS -
-
SB_ADD DBS DBS2 [mq_source_install].AMQPARMS $MQCS1 [mq_target_install].AMQPARMS -
-
SB_ADD DBS DBS3 [mq_source_install].AMQSINI $MQCS1 [mq_target_install].AMQSINI - -

== Adds the subvolume where the MQ Queue Manager will be created to replication.
== Excludes the altkey file AMQPDBA to avoid replicating it twice.
SB_ADD DBS DBS4 [mq_source_qmgr].* $MQCS1 [mq_target_qmgr].* - -
SB_ADD EXCLUDE_DBS DBS5 [mq_source_qmgr].AMQPDBA $MQCS1

== These lines show how to include a queue migration or partition. The subvolume
is
== wildcarded due to MQ creating random names for these files.
SB_ADD DBS DBS6 [sb_source_node].$SSD2.MQDRQ0.* $MQCN
[sb_target_node].$SSD2.MQDRQ0.* - -
```

```
SB_ADD DBS DBS7 [sb_source_node].$DATA2.MQDRQ0.* $MQCN
[sb_target_node].$DATA2.MQDRQ0.* -

== The below DBS parameters are need to properly replicate MQ. The DDL related
parameters
== should only be set for DBSs that are MQ Queue Manager subvolumes.
SB_ADD CONS_PARAM $MQCS1 * CONTROLFILE [sb_source_system_data].CONTC1
SB_ADD DBS_PARAM * * TRACKTX ON
SB_ADD DBS_PARAM * * INSERTNOTFOUND OFF
SB_ADD DBS_PARAM * * UPDATEDUPLICATE OFF
SB_ADD DBS_PARAM DBS4 * EXCLUSIONMODE EXCLUSIVE
SB_ADD DBS_PARAM DBS4 * CREATES ON
SB_ADD DBS_PARAM DBS4 * CREATEOPTIONS (RECREATEALTFILES, PARTITIONS,
ALTFILEMAPPING, NOCOORDINATION)
SB_ADD DBS_PARAM DBS4 * ALTERS ON
SB_ADD DBS_PARAM DBS4 * ALTEROPTIONS (NOCOORDINATION)
SB_ADD DBS_PARAM DBS4 * PURGEDATAS ON
SB_ADD DBS_PARAM DBS4 * PURGEDATAOPTIONS (PURGEDATAALTFILES, ALLPARTS,
NOCOORDINATION)
SB_ADD DBS_PARAM DBS4 * PURGES ON
SB_ADD DBS_PARAM DBS4 * PURGEOPTIONS (PURGEALTFILES, ALLPARTS, NOCOORDINATION)

SB_GEN
```

Failover Configuration

The example below will be almost identical to the Active/Passive configuration from above, but includes a few key changes. One of these changes is to flip the source and target nodes which will result in the reversing of the Active/Passive environment. The next change is to add a Collector start time by using the COLL Parameter ADTSTARTTIME and setting ADTSTARTEOF to OFF. The time ADTSTARTTIME should be set to is the moment when replicated was stopped and no more data was applied to the target.

```
?tacl macro

#frame
#push sb_source_node sb_target_node
#push sb_source_system_install sb_target_system_install
#push sb_source_system_data sb_target_system_data
#push mq_source_install mq_target_install mq_source_qmgr mq_target_qmgr

== Source and target nodes.
#set sb_source_node \BACK
#set sb_target_node \PRIM

== Locations of source and target Shadowbase installs.
#set sb_source_system_install [sb_source_node].$SB.SB7000
#set sb_target_system_install [sb_target_node].$SB.SB7000

== Locations on source and target where Shadowbase will be run from.
#set sb_source_system_data [sb_source_node].[vol].[subvol]
#set sb_target_system_data [sb_target_node].[vol].[subvol]

== Locations of source and target MQ installs.
#set mq_source_install [sb_source_node].$SSD1.MQDR
#set mq_target_install [sb_target_node].$SSD1.MQDR

== Locations of source and target MQ Queue Managers.
== Repeat for each Queue Manager that should be replicated.
#set mq_source_qmgr [sb_source_node].$SSD1.MQDRQ0
#set mq_target_qmgr [sb_target_node].$SSD1.MQDRQ0

load /keep 1/ [sb_source_system_install].SBUNIDIR

SB_CLEAR ALL
SB_SET SB_CONFIGURATION_TYPE DIRECT
SB_SET SB_CONFIG_FILE %0%
SB_SET SB_TRACKTX 1

SB_ADD SRCNODE [sb_source_node] [sb_source_system_install] [sb_source_system_data]
~[#MYTERM~] [sb_source_system_install].SHADPASS $MQLL1 1 0 140 $MQD1 0 1 90

SB_ADD TGTNODE [sb_target_node] [sb_target_system_install] [sb_target_system_data]

SB_ADD CONS $MQCS1 0 1 130 [sb_target_system_data].TIDFIL1

== This collector parameter prevents replication from stopping when deleting MQ
Queue
== Managers
SB_ADD COLL_PARAM GETRECADDRERROK ON

== This consumer parameter will keep the target files open at all times
== once there is an event for that file.
SB_ADD CONS PARAM * * FILECLOSEDELAY 0

== Sets start time for the Collector to start processing and replicating files.
SB_ADD COLL_PARAM ADTSTARTEOF OFF
SB_ADD COLL_PARAM ADTSTARTTIME (JAN 29 2025,12:59:59.0.0)

== Adds the three files that needed to be replicated from the MQ install to
replication.
SB_ADD DBS DBS1 [mq_source_install].AMQMQINS $MQCS1 [mq_target_install].AMQMQINS -
-
SB_ADD DBS DBS2 [mq_source_install].AMQPARMS $MQCS1 [mq_target_install].AMQPARMS -
-
SB_ADD DBS DBS3 [mq_source_install].AMQSINI $MQCS1 [mq_target_install].AMQSINI - -

== Adds the subvolume where the MQ Queue Manager will be created to replication.
== Excludes the altkey file AMQPDBA to avoid replicating it twice.
SB_ADD DBS DBS4 [mq_source_qmgr].* $MQCS1 [mq_target_qmgr].* - -
SB_ADD EXCLUDE_DBS DBS5 [mq_source_qmgr].AMQPDBA $MQCS1

== These lines show how to include a queue migration or partition. The subvolume
is
```

```
== wildcarded due to MQ creating random names for these files.
SB_ADD DBS DBS6 [sb_source_node].$SSD2.MQDRQ0.* $MQCN
[sb_target_node].$SSD2.MQDRQ0.* - -
SB_ADD DBS DBS7 [sb_source_node].$DATA2.MQDRQ0.* $MQCN
[sb_target_node].$DATA2.MQDRQ0.* -

== The below DBS parameters are need to properly replicate MQ. The DDL related
parameters
== should only be set for DBSs that are MQ Queue Manager subvolumes.
SB_ADD CONS_PARAM $MQCS1 * CONTROLFILE [sb_source_system_data].CONTC1

SB_ADD DBS_PARAM * * TRACKTX ON
SB_ADD DBS_PARAM * * INSERTNOTFOUND OFF
SB_ADD DBS_PARAM * * UPDATEDUPLICATE OFF
SB_ADD DBS_PARAM DBS4 * EXCLUSIONMODE EXCLUSIVE
SB_ADD DBS_PARAM DBS4 * CREATES ON
SB_ADD DBS_PARAM DBS4 * CREATEOPTIONS (RECREATEALTFILES, PARTITIONS,
ALTFILEMAPPING, NOCOORDINATION)
SB_ADD DBS_PARAM DBS4 * ALTERS ON
SB_ADD DBS_PARAM DBS4 * ALTEROPTIONS (NOCOORDINATION)
SB_ADD DBS_PARAM DBS4 * PURGEDATAS ON
SB_ADD DBS_PARAM DBS4 * PURGEDATAOPTIONS (PURGEDATAALTFILES, ALLPARTS,
NOCOORDINATION)
SB_ADD DBS_PARAM DBS4 * PURGES ON
SB_ADD DBS_PARAM DBS4 * PURGEOPTIONS (PURGEALTFILES, ALLPARTS, NOCOORDINATION)

SB_GEN
```

Configuring a Drain Session (SBUNIDIR)

SBUNIDIR has been enhanced to support generating additional configuration files to start the target node when the source node has gone down. This is done with the new SB_GEN_DRAIN command, which should be called after generating the normal configuration files with SB_GEN.

The following considerations apply when generating drain configuration files.

- SB_GEN_DRAIN is only supported when the configuration type is set to DIRECT.
- The files generated by SB_GEN_DRAIN will use the prefix from SB_TARGET_CONFIG_PREFIX.
- If the source and target files are not identical, a MAPTOFILE must be configured in order to drain the consumer while the source node is not accessible. If the source and target are different and a MAPTOFILE is not used, data corruption may occur.
- Override parameters can be set between SB_GEN and SB_GEN_DRAIN in order to have different settings between the normal environment and drain environment.
- The drain configuration will only include replication threads that have a QMGR on the target node.

SB_ADD TGTNODE

When generating a drain configuration, optional arguments can be added to the SB_ADD TGTNODE command in order to configure the AUDMON process that will run on the target node. Arguments that are not specified will default to corresponding argument from SB_ADD SRCNODE. Optional arguments can be skipped with a dash.

```
  SB_ADD TGTNODE node-name appl-subvol data-subvol [home-term] [license-fname]
                 [audmon-name] [audmon-cpu-p] [audmon-cpu-b] [audmon-pri]


    where

      <node-name>          is the target node name.
      <appl-subvol>        is the HPE Shadowbase's installation subvolume on the target.
      <data-subvol>        is the subvolume on the target where the HPE Shadowbase runtime
                           files are stored.
      <home-term>          the location where the process 'in', 'out' and 'term' are set
                           to on the target node.  ( i.e. $VHS or ~[#myterm~] )
      <license-fname>      is the filename of the HPE Shadowbase password file on the
                           target node.
      <audmon-pname>       a valid name for the audmon process name on the target node.
      <audmon-cpu-p>       a valid primary cpu number.
      <audmon-cpu-b>       a valid backup cpu number (use -1 for no backup).
      <audmon-pri>         a valid audmon priority number.

Example:
sb_set sb_configuration_type tcpip
sb_add srcnode \s1 \s1.$shad.srcappl \s1.$shad.srcdata ~[#myterm~] &
              $shad.pass.word $coll 0 1 120 $audm 1 0 120
sb_add tgtnode \h1 \h1.$shad.app1 \h1.$shad.data - - $audmt
```

SB_GEN_DRAIN

SB_GEN_DRAIN generates the additional configuration files for a drain configuration.

```
   SB_GEN_DRAIN
```

This is a sample of the output and files generated by SB_GEN_DRAIN:

```
Create the drain configuration files for \NS2SQL ([Y]/N)?
Creating TGconf   - Shadowbase configuration for target node (\NS2SQL)
Creating TGstrtsb - Start target Shadowbase
Creating TGamon   - Start AUDMON
Creating TGcu     - Clean up
Creating TGstrtdr - Start direct CONS
Creating TGstopdr - Stop direct consumers
Creating TGrun    - Run Shadowbase
Creating TGstopsb - Stop Shadowbase
Creating TGSHUTDN - Shadowbase shutdown
```

Configuring a Drain Session (SBBIDIR)

SBBIDIR has been enhanced to support generating additional configuration files to start the target node when a source node has gone down. This is done with the new SB_GEN_DRAIN command, which should be called after generating the normal configuration files with SB_GEN.

The following considerations apply when generating drain configuration files.

- SB_GEN_DRAIN is only supported when the connection type is set to DIRECT.

- SB_GEN_DRAIN will generate a set of drain configuration files for each combination of nodes, in both directions. The files generated by SB_GEN_DRAIN will include both node ids as a prefix.
- If the source and target files are not identical, a MAPTOFILE must be configured in order to drain the consumer while the source node is not accessible. If the source and target are different and a MAPTOFILE is not used, data corruption may occur.
- Override parameters can be set between SB_GEN and SB_GEN_DRAIN in order to have different settings between the normal environment and drain environment.
- The drain configuration will only include replication threads that have a QMGR on the target node.

SB_GEN_DRAIN

SB_GEN_DRAIN generates the additional configuration files for a drain configuration.

```
SB_GEN_DRAIN
```

This is a sample of the output and files generated by SB_GEN_DRAIN:

```
Create the drain K1 -> K2 configuration files for \NS2SQL? ([Y]/N)?
Creating K1K2conf - Shadowbase configuration (code 101) for K1 -> K2 (\NS2SQL)
Creating K1K2strs - Start Shadowbase (cold/cool)
Creating K1K2amon - Start AUDMON
Creating K1K2cu   - Clean up
Creating K1K2strd - Start direct CONS and QMGR
Creating K1K2run  - Run Shadowbase
Creating K1K2stps - Stop Shadowbase
Creating K1K2stpd - Stop collector and direct consumers
Creating K1K2shut - Shadowbase shutdown
…
Create the drain K2 -> K1 configuration files for \VIV1? ([Y]/N)?
Creating K2K1conf - Shadowbase configuration (code 101) for K2 -> K1 (\VIV1)
Creating K2K1strs - Start Shadowbase (cold/cool)
Creating K2K1amon - Start AUDMON
Creating K2K1cu   - Clean up
Creating K2K1strd - Start direct CONS and QMGR
Creating K2K1run  - Run Shadowbase
Creating K2K1stps - Stop Shadowbase
Creating K2K1stpd - Stop collector and direct consumers
Creating K2K1shut - Shadowbase shutdown
```

New and Modified Commands

QSETSEND Command

The QSETSEND command allows the operator to start and stop data flowing from the QMGR to the Consumer at a position in audit, at a time (local), or at EOF. If no event starts exactly at the requested time/position the consumer's new start point is set to the first event after the requested time/position. Before issuing the QSETSEND command, the associated consumer must be stopped. The collector does not need to be stopped and

the QMGR will continue receiving and storing data from the collector in QMGR Queue Files during replay operations.

Note that for environments where TRACKTX is enabled, the contents of the TRACKTX file must be deleted (e.g. via a PURGEDATA) before changing the START position. All start times/positions must have been sent to the QMGR before issuing the QSETSEND START command. QSETSEND STOP commands may use times/positions in the past or the future and can be used with TRACKTX without issuing a PURGEDATA on the TRACKTX file for events that have not yet been applied to the target database.

The QSETSEND command supports starting and stopping QMGR flow to the consumer. The command syntax for the QSETSEND command is as follows:

```
QSETSEND [QMGR] [<audmon-name>.]<qmgrname>
                        {START}
                        {STOP}
                             {<date> [<time>]}
                             {<time>}
                             {<MAT position> [<ADT position>]}
                             {EOF}
                                 <date> is in the format:
                                     MM-DD-YYYY
                                 <time> is in the format:
                                     HH:MM:SS.FFFFFF
                                 <position> is in the format:
                                     <prefix><sequence-number>.<rba>
START
    Replays data from the qfiles from the start criteria (if present).
STOP
    Processing continues until the stop criteria (if present).
    Use the QSENDRESUME command to resume processing after a stop.
```

To issue a QSETSEND command first the associated consumer must be stopped. Next the QSETSEND command is issued. Finally sending from the queue manager begins when the consumer is re-started and continues indefinitely, or until the stop criteria are met (if a STOP is specified). The example below includes both a START and STOP command. In this example, the data starting at and including MAT AA000123.45678 and AUX CC000091.23456 will be sent to the consumer. Data at and after 10-31-2025 12:01.23.456789 will not be sent to the consumer and the QMGR will enter a SENDSTOP state in which it does not send data on to the consumer. To exit this state the QSENDRESUME command must be used. Please note that if the QMGR is stopped via a QSETSEND STOP command, transactions that have not been committed or aborted at that point of processing will be automatically aborted in the Consumer after your system-defined time-out. The Consumer will also maintain a lock on these in progress records/rows until the QMGR is resumed with the QSENDRESUME command or the lock times out.

```
+ STOP CONS <consname>
+ QSETSEND QMGR <qmgrname> START AA000123.45678 CC000091.23456
+ QSETSEND QMGR <qmgrname> STOP 10-31-2025 12:01.23.456789
+ START CONS <consname>
```

The STATS output for the QMGR has been modified to show any stop point set (in time or in TMF audit) that has not yet been reached. EMS messages are also issued for each START and STOP position change. After setting a START time or position, statistics for queued events will be temporarily unavailable. A message in the STATS output for the QMGR indicates the time or position when that data will be available again.

QSENDRESUME Command

The QSENDRESUME command allows the operator to re-enable QMGR processing after a QSETSEND STOP command. This is similar to the RESUME command, but is specific to the SENDSTOP state. After issuing QSENDRESUME, consumer processing will proceed as normal. The command syntax for the QSENDRESUME command is as follows:

```
QSENDRESUME  [QMGR] { [<audmon-name>.]<qmgrname>}
                    { [<audmon-name>.]* }
Puts queue managers stopped due to a QSETSEND STOP command (SENDSTOP state)
    in a run state.
```

SBBIDROS SB_ADD CONSUE

The SB_ADD CONSUE command allows for consumptive user exits to be configured for a SQL/MX source without the need for a DOC writer. DBSes are added to consumers configured with this command with SB_ADD BIDIRDBS or SB_ADD UNIDIRDBS, just as for consumers configured with SB_ADD CONSOS.

When using a consumer added with CONSUE, all DBSes must have the USEREXITID DBS parameter set greater than 0. Additionally, after returning from USRXPROCESS, the consumer will ignore the event, as if SBSETIGNORE was called.

The syntax for SB_ADD CONSUE is as follows:

```
SB_ADD CONSUE <cons-id> <cpu-p> <cpu-b>

    where

        <cons-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)
```

SBBIDROS SB_ADD CONSUE_PARAM

The SB_ADD CONSUE_PARAM command overrides the default consumer parameters for consumers configured with the SB_ADD CONSUE command.

The syntax for the SB_ADD CONSUE_PARAM is as follows:

```
SB_ADD CONSUE_PARAM <src-node-id|*> <tgt-node-id|*> <cons-id|*> <param-name-and-value>

    where

        <src-node-id>          is an existing source node id. This id can be
                               wildcarded with an asterisk.

        <tgt-node-id>          is an existing target node id. This id can be
                               wildcarded with an asterisk.

        <cons-id>              is an existing consumer id. This id can be wildcarded
                               with an asterisk.

        <param-name-and-value> is the parameter name and value which to override.
```

SBBIDROS SB_ADD LINK

An optional argument called <cons-type> has been added to this command.

```
  SB_ADD LINK link_id cons-id node1 port1p port1b node2 port2p port2b [cons-type]

    where

       <link-id>       is a link id that is generated by sbbidros.
       <cons-id>       is an existing consumer id
       <node1>         is an existing node-id
       <port1p>        is the port to use for node 1, primary.
       <port1b>        is the port to use for node 1, backup.
       <node2>         is an existing node-id
       <port2p>        is the port to use for node 2, primary.
       <port2b>        is the port to use for node 2, backup.
       <cons-type>     (opt) is the type of consumer that is part of the link. Defaults
                       to NRML.
                              NRML: Consumer configured with SB_ADD CONSOS
                              UEXT: Consumer configured with SB_ADD CONSUE
```

New and Modified SET AUD Parameters

None in this version.

New and Modified SET COLL Parameters

HASHBUCKETSKIPACTION { STOP }
{ WARN }
{ CONTINUE }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.010		STOP				YES

This parameter controls the behavior of the collector when an event is skipped due to it not matching any of the HASHBUCKETID values. When set to STOP, an EMS message is logged and the collector abends. When set to WARN, an EMS message is logged once per file and the event is skipped. When set to CONTINUE, the event is skipped without logging a message. This parameter is only support for SBCOLL.

New and Modified SET CONS Parameters

SBMXSERVPROGRAM [\system.$volume.subvolume] filename

Initial Version	Changed Version	Values	Default	Value Specifications	Alter
7.010		SBMXSERV		Valid program file name	No

This parameter specifies the name of the SBMXSERV object file. By default, the volume and subvolume used will be the same as the object file for the consumer.

SBMXSERVPROCESS [\systemname.] $sbmxserv_name

Initial Version	Changed Version	Values	Default	Value Specifications	Alter
7.010		None		Valid process name	No

This parameter specifies the process name of the SBMXSERV process.

SBMXSERVPRI <number>

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.010		None		1	199	No

This parameter specifies the priority of the SBMXSERV process. If omitted, the consumer's priority will be used.

SBMXSERVCPU <cpu-number>

Initial Version	Changed Version	Values Default	Value Specifications	Alter
7.010		None	Valid CPU Number	No

This parameter specifies the CPU of the SBMXSERV process. If omitted, the consumer's CPU will be used.

SBMXSERVCQD <control clause>
(<control clause> [,(<control clause>)]…)

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		None			NO

This parameter can be used to issue custom SQL/MX CONTROL QUERY DEFAULT statements for SBMXSERV. See the CONTROL QUERY DEFAULT statement in the SQL/MX User Guide manual for all possible options. The parameter should be entered as a comma separated list of <option> '<value>' pairs.

Before executing the CQDs from SBMXSERVCQD, or if SBMXSERVCQD is not set, SBMXSERV will run the following commands:

```
>> CONTROL QUERY DEFAULT TIMEOUT '<SQLLOCKWAIT>'
>> CONTROL TABLE * TABLELOCK 'OFF'
```

SBMXSERVIGNORECODES {sqlcode[, sqlcode, ...]}
{-1}

Initial Version	Changed Version	Values Default	Value Specifications	Alter
7.010		-1	Valid SQL/MX SQLCODE	No

This parameter is related to SQL I/O done through SBMXSERV. The SBMXSERVIGNORECODES parameter can be used to modify the list of SQLCODEs that the consumer will ignore. The SQLCODEs entered must be positive and ignore sign when comparing against the SQLCODEs returned by SBMXSERV. See the *HPE Nonstop Shadowbase Operations Manual* for the list of default SQLCODEs that can be ignored. In some cases, SQLCODEs either translate to or come with HPE Nonstop errors. In these cases, the IGNOREERROR parameter should be used instead.

When determining which action to take after an error is returned from SBMXSERV, the consumer will look at all SQLCODEs returned, using the following order of precedence:

a. First, if any of the SQLCODEs are in the stop list, the consumer will abend.
b. Next, if any of the SQLCODEs are in the skip list, the consumer will skip the event.

c. Next, if any of the SQLCODEs are in the ignore list, the consumer will ignore the event.
d. Finally, if none of the SQLCODEs are in any list, the consumer will abend.

SBMXSERVRETRYCODES {sqlcode[, sqlcode, ...]}
{-1}

Initial Version	Changed Version	Values	Default	Value Specifications	Alter
7.010		-1		Valid SQL/MX SQLCODE	No

This parameter is related to SQL I/O done through SBMXSERV. The SBMXSERVRETRYCODES parameter can be used to modify the list of SQLCODEs that the consumer will retry. The SQLCODEs entered must be positive and ignore sign when comparing against the SQLCODEs returned by SBMXSERV. See the *HPE Nonstop Shadowbase Operations Manual* for the list of default SQLCODEs that can be retried. In some cases, SQLCODEs either translate to or come with HPE Nonstop errors. In these cases, the RETRYERROR parameter should be used instead.

When determining which action to take after an error is returned from SBMXSERV, the consumer will look at all SQLCODEs returned, using the following order of precedence:
a. First, if any of the SQLCODEs are in the stop list, the consumer will abend.
b. Next, if any of the SQLCODEs are in the skip list, the consumer will skip the event.
c. Next, if any of the SQLCODEs are in the ignore list, the consumer will ignore the event.
d. Finally, if none of the SQLCODEs are in any list, the consumer will abend.

SBMXSERVSTOPCODES {sqlcode[, sqlcode, ...]}
{-1}

Initial Version	Changed Version	Values	Default	Value Specifications	Alter
7.010		-1		Valid SQL/MX SQLCODE	No

This parameter is related to SQL I/O done through SBMXSERV. The SBMXSERVSTOPCODES parameter can be used to modify the list of SQLCODEs that the consumer will stop for. The SQLCODEs entered must be positive and ignore sign when comparing against the SQLCODEs returned by SBMXSERV. See the *HPE Nonstop Shadowbase Operations Manual* for the list of default SQLCODEs that will be stopped for. In some cases, SQLCODEs either translate to or come with HPE Nonstop errors. In these cases, the STOPERROR parameter should be used instead.

When determining which action to take after an error is returned from SBMXSERV, the consumer will look at all SQLCODEs returned, using the following order of precedence:

a. First, if any of the SQLCODEs are in the stop list, the consumer will abend.
b. Next, if any of the SQLCODEs are in the skip list, the consumer will skip the event.
c. Next, if any of the SQLCODEs are in the ignore list, the consumer will ignore the event.
d. Finally, if none of the SQLCODEs are in any list, the consumer will abend.

SBMXSERVTRACE { ON }
{ OFF }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		OFF	OFF	ON	NO

When this parameter is ON, SBMXSERV will log additional EMS messages containing statements that it is executing and errors that it receives. Note that tracing should be disabled at all times unless necessary to troubleshoot SBMXSERV replication issues.

USEREXITDLL [\system.$volume.subvolume] filename

Initial Version	Changed Version	Values Default	Value Specifications	Alter
7.010		None	Valid DLL Filename	NO

When this parameter is set to a DLL filename, AUDCONS will dynamically load User Exit functions into the consumer from the specified DLL. This parameter should point to a compiled DLL User Exit. User Exit functionality works the same, just avoids having to bind the User Exit into the Consumer.

USEREXITDLLLOADERROR { STOP }
{ WARN }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		STOP	STOP	WARN	No

This parameter specifies how the Consumer should handle a symbol not found error when loading the User Exit DLL. This can occur when one or multiple User Exit entry point functions are missing from the DLL. The default, STOP, will display an error message and stop. The option WARN will display a warning message and continuing processing.

NODBSEVENTACTION { STOP }
{ WARN }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.010		STOP	STOP	WARN	NO	

This parameter determines how events that appear in the Consumer with no DBS specification are handled. This can occur when a MAPTOFILE is incorrectly configured and does not match any source files from the listed DBSs. The default STOP logs a message and stops. WARN will log a message and continue processing by skipping the event.

NETUSERNAME username

Initial Version	Changed Version	Values	Default	Value Specifications	Alter
3.900	7.010	None		Alphanumeric up to 31 bytes	No

This is used for authorizing an STP connection used for a TCP/IP replication environment. NETUSERNAME is also used during the connection process between two Consumers when operating in a HPE Nonstop-to-Nonstop over TCP/IP mode as well as for a "standalone" Consumer used for an Open-to-Nonstop replication environment. NETUSERNAME can be up to 31 bytes long and contain any alphanumeric string character except semicolon (;). When setting the parameter through AUDCOM, the value can be entered securely by entering 'SET CONS NETUSERNAME' without a value. AUDCOM will then prompt the user to enter a value that will not be echoed back to the terminal.

NETUSERPASSWORD password

Initial Version	Changed Version	Values	Default	Value Specifications	Alter
3.900	7.010	None		Alphanumeric up to 31 bytes	No

This is used for authorizing an STP connection used for a TCP/IP replication environment. NETUSERPASSWORD is also used during the connection process between two Consumers when operating in a HPE Nonstop-to-Nonstop over TCP/IP mode as well as for a "standalone" Consumer used for an Open-to-Nonstop replication environment. NETUSERPASSWORD can be up to 31 bytes long and contain any alphanumeric string character except semicolon (;).

Note: The INFO command will only display the NETUSERPASSWORD value if the user is SUPER.SUPER or the user that started AUDMON. Additionally, a SET CONS LIKE... will not set the NETUSERPASSWORD.

When setting the parameter through AUDCOM, the value can be entered securely by entering 'SET CONS NETUSERPASSWORD' without a value. AUDCOM

will then prompt the user to enter a value that will not be echoed back to the terminal.

New and Modified SET DBS Parameters

PRIPARTONLY { ON }
{ OFF }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.010		OFF	OFF	ON	NO	

When this parameter is ON, secondary partitions that match SOURCEFILE will not be replicated, unless ALLPARTITIONS is ON and the primary partition also matches SOURCEFILE. This parameter is only supported when using SBCOLL.

ALLPARTITIONS { ON }
{ OFF }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
4.091	7.010		OFF	OFF	ON	No

This parameter is used to indicate that all local partitions (i.e., those partitions on the COLL's node), associated with the primary partition specified in the SOURCEFILE parameter, are to be selected. The default is OFF. When set to ON and not using SBCOLL, the DBS SOURCEFILE must be set to a fully-qualified file name and must be the primary partition.

The file name passed from the COLL to the CONS is for the primary partition (the COLL replaces the secondary partition volume with the primary partition volume name). Thus, the extended file statistics (STATS AUD output) for the CONS will only show the primary partition because the CONS only has the primary file in its selection/tracking list.

Note: Only local partitions are loaded into a COLL's selection list. This is because the COLL uses ARLIB to access the audit trail events, and ARLIB can only access data for local files/tables, not remote files or tables (or partitions thereof). If the COLL finds distributed partitions for a file/table (i.e., partitions on other nodes), the condition is printed to EMS. It is possible that HPE Shadowbase replication may need to be configured on the other node to pick up and replicate the distributed partitions.

Note: If you plan to utilize the SOLV loader for a related DBS object that has ALLPARTITIONS set ON, you should set the SOLV TARGETFNAME parameter to the subvol of the primary partition of the DBS SOURCEFILE value. Additionally, if your DBS object has MAPTOFILE set, then you should set the SOLV TARGETFNAME parameter to the subvol of the DBS MAPTOFILE value.

CATCHALL { ON }
{ OFF }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		OFF	OFF	ON	NO

When this parameter is ON, files will only be included by this DBS if it matches SOURCEFILE and no DBS with CATCHALL OFF matches the file. This parameter is only supported when using SBCOLL.

CATCHALLMODE { CONS }
{ COLL }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		COLL			NO

When the parameter CATCHALL is ON, this parameter determines which DBSes are checked to determine if another DBS has already matched it. When CATCHALLMODE is COLL, the CATCHALL DBS will only match if no DBS with CATCHALL OFF in the same collector matches the file. When CATCHALLMODE is CONS, the CATCHALL DBS will only match if no DBS with CATCHALL OFF in the same consumer matches the file. This parameter is only supported when using SBCOLL.

SBMXSERVENABLED { ON }
{ OFF }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		OFF	OFF	ON	NO

When this parameter is ON, inserts into key-sequenced SQL/MP tables will be done through the SBMXSERV process instead of by the consumer.

SBMXSERVCONTROLTABLE <control clause>
(<control clause> [,(<control clause>)]…)

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		None			NO

This parameter can be used to issue custom SQL/MX CONTROL TABLE statements against a target table for SBMXSERV. See the CONTROL TABLE statement in the SQL/MX User Guide manual for all possible options. The parameter should be entered as a comma separated list of <option> '<value>' pairs. By default, SBMXSERV sets the TABLELOCK option to 'OFF' for all tables.

TARGETFILE filename

Initial Version	Changed Version	Values	Default	Value Specifications	Alter
3.900	7.010		None	Valid target file name Include NODE	No

For a DBS configured under a consumer applying events directly to a Nonstop target (i.e. CONNECTIONTYPE DIRECT), this parameter identifies the name of the destination file. This parameter is required for this scenario, unless a USEREXITID is specified in which case it is assumed that the user exit will supply the name of the target file to apply events to. If volume and subvolume are not entered, the default specified with the TARGETVOL command is used. If the TARGETVOL command is omitted, the default volume and subvolume are used based on the environment where the AUDMON process is running.

The filename can be a specific filename or a fileset selected by using wildcard characters. 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. For example,

  $DATA.*.TABLE1

  Indicates all files called TABLE1 found in all subvolumes on the volume $DATA.

  If a part of the filename is a single asterisk, it will match the entire corresponding part of the source file, regardless of how it is wildcarded.

? (question mark) matches one character in the position where it appears. For example,

  $D?.*.TABLE1

  Indicates all files called TABLE1 in all subvolumes on all disks beginning with the letter D whose name is followed by one character (i.e., it would select $D1, but not $DATA1).

When a part of the filename is not static or a single asterisk, additional restrictions apply:

- The part must contain the same wildcard characters, in the same order, as the corresponding part of SOURCEFILE and MAPTOFILE if specified.
- The part cannot contain multiple asterisks.
- MAPFROMSOURCE must be off.

For a DBS configured under a consumer acting as a source sending events to a target Other Servers system, this parameter identifies the name of the dummy file used to modify the events before sending them to the target system. This parameter can be set to the same file as SOURCEFILE if necessary. This parameter is only required in this scenario when a dummy file is needed. When TARGETFILE is present in a DBS and PATHNAME is omitted, the filename portion of the TARGETFILE file will be used as the table name in the replicated events. If PATHNAME is set, that value will be used as the table name instead. If neither TARGETFILE nor PATHNAME are set, the filename portion of the SOURCEFILE file will be used.

For a DBS configured under a consumer acting as a source sending events to a target consumer over TCP/IP, this parameter should not be set.

Notes:

- In an HPE Shadowbase TCP/IP Nonstop to Nonstop environment using a source and target consumer, the source consumer's CONNECTIONTYPE is set to TCPIP and the target consumer's CONNECTIONTYPE is set to DIRECT.
- For Enscribe-to-Enscribe replication without DBS SOURCEREC and TARGETREC parameters (DDL), the Consumer will use the minimum of the TARGETFILE record length or the audit trail event length for the target I/O. Note that if ABENDONMISMATCH is set to ON, this minimum setting is not performed and the audit trail event length is used.
- A check by the Consumer will be done to make sure that a TARGETFILE is set to the primary partition of an Enscribe file during startup or if a new DBS object is added. A warning will be logged to the user and the CONS will search for, and open, the primary partition of a partitioned Enscribe file when necessary.

MAPTOFILE file_name

Initial Version	Changed Version	Values	Default	Value Specifications	Alter
3.600	7.010		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 target file name. 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.

  If a part of the filename is a single asterisk, it will match the entire corresponding part of the source file, regardless of how it is wildcarded.

? (question mark) matches one character in the position where it appears. For example,

  $D?.*.TABLE1

  Indicates all files called TABLE1 in all subvolumes on all disks beginning with the letter D whose name is followed by one character (i.e., it would select $D1, but not $DATA1).

When a part of the filename is not static or a single asterisk, additional restrictions apply:

- The part must contain the same wildcard characters, in the same order, as the corresponding part of SOURCEFILE.
- The part cannot contain multiple asterisks.
- MAPFROMSOURCE must be off.

MAPTOFILE may also be used to allow the SQL/MX consumer, AUDCONXN, to run on the target system. In this use case, MAPTOFILE must be used as follows:

- A table with the same exact schema, and the same CATALOG.SCHEMA.TABLE ANSINAME as exists on the source must exist on the target.
- Wildcarding is not supported for this usage of MAPTOFILE.
- The MAPTOFILE must be set to the underlying guardian file name associated with the matching SQL/MX table on the target system. The underlying guardian file name can be determined from MXCI by using the SHOWDLL command against the table on the target system exactly matching the source. The LOCATION attribute in the output of the SHOWDDL command shows the underlying Guardian filename.

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

NOTE: If your DBS object has MAPTOFILE set, then you should set the SOLV TARGETFNAME parameter to the subvol of the DBS MAPTOFILE value.

New and Modified SET QMGR Parameters

QFILEROLLTIMEMODE { OFF }
{ CLOCK }
{ AUDIT }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		OFF			YES

This parameter enables rolling queue files based on time, as well as when they become full. When set to CLOCK, the queue manager will roll to the next queue file QFILEROLLTIME seconds after first writing an event to the queue file. When set to AUDIT, the queue manager will roll to the next queue file if the current IPM's last event occurred more than QFILEROLLTIME seconds after the first event of the current queue file or if it has been QFILEROLLTIME seconds since the last event was received.

QFILEROLLTIME <seconds>

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		60	0	32767	YES

When QFILEROLLTIMEMODE is set to CLOCK or AUDIT, this parameter is the number of seconds before the queue manager rolls the queue file early.

New and Modified SET SOLVMGR Parameters

None in this release.

New and Modified SOLV AND SOLVNV Parameters

None in this release.

New and Modified Coordinator (SBCOOR) Parameters

RM_TIMEOUT

Initial Version	Changed Version	Default Values	Minimum	Maximum	Occurs
7.010		60	-1 (excluding 0)	21474836	1

RM_TIMEOUT specifies the maximum time limit for TMF to receive any response from the coordinator. If the timeout is exceeded, TMF will timeout the resource manager of the coordinator, allowing all active transactions to complete

without waiting for the coordinator to vote. These transactions will complete with replication handled asynchronously, as well as any interim transactions that occur until the coordinator reopens the resource manager and re-establishes the capability for synchronous replication. This is a safety feature to prevent a scenario such as a deadlock from being able to prevent all transaction activity in the CPU. The coordinator will perform version detection to determine availability of this feature, and disable it if it is determined not to be available. This detection can be overridden by the usage of the RM_TIMEOUT_USAGE_OVERRIDE parameter. The current requirements for RM_TIMEOUT are RVU L22.09 to L25.02 with T8607L01^ANZ and its requisite SPRs.

Usage with L25.09 requires a different set of SPRs, not yet available at the time of publication; inquire with Shadowbase support for current information.

Please note that Gravic strongly encourages customers to only run on versions where the RM_TIMEOUT is available. Otherwise, a deadlock scenario as described above can occur causing significant disruption to the application and the TMF environment.

The parameter is specified in seconds, with a default value of 60 (1 minute). Valid values are:
-1 – No timeout
1 through 21474836 – The value in seconds of the timeout.

RM_TIMEOUT_USAGE_OVERRIDE

Initial Version	Changed Version	Default Values	Minimum	Maximum	Occurs
7.010		-1	-2	1	1

The RM_TIMEOUT_USAGE_OVERRIDE parameter allows for overruling the coordinator logic determining the usage (or lack thereof) of the RM_TIMEOUT feature. The default value (-1), will use the coordinators determination. Allowed values are:

-2 – Allow the coordinator to determine the availability of the feature, but always log the TMF versions detected.
-1 – Allow the coordinator to determine the availability of the feature, only logging a message in the event that the RM_TIMEOUT feature is disabled.
0 – Override the coordinator logic, disabling usage of the RM_TIMEOUT feature.
1 – Override the coordinator logic, enabling usage of the RM_TIMEOUT feature. Note that if the underlying functionality used by this feature is not available on the system, this will cause the coordinator to trap during startup processing.

New or Modified EMS Messages

2083	COLL FAILED TO PROCESS EXPEDITED PULSE FOR FILE <file>

Cause: The collector was either unable to get the file info for the pulse file, or unable to match the pulse to a consumer thread.

Effect: The expedited pulse is not sent to the consumer.

Recovery: Ensure that the pulse file still exists and the consumer exists, and then retry the command.

2116	ERR 2193: EXPEDITED PULSES CANNOT BE SENT TO COLLECTORS THAT ARE NOT REPLICATING EVENTS

Cause: An expedited pulse was sent to a collector whose REPLICATION parameter is set to SKIP or SUSPEND.

Effect: The expedited pulse is not sent to the consumer.

Recovery: Alter the REPLICATION parameter to ON and retry the command.

2080	COLLECTOR DETECTED INCONSISTENT MAPTOFILE DEFINITIONS FOR DBS OBJECT <dbs name> (VALUE < MAPTOFILE value>, FILE <file>); ALL MATCHING DBS FOR THE SAME CONSUMER MUST HAVE IDENTICAL MAPTOFILE OPTIONS

Cause: The collector outputs this message to indicate that there are ambiguous definitions for the MAPTOFILE parameter in the DBS objects. The message identifies the offending DBS object, the MAPTOFILE value that is being used, and the file matched the offending DBS objects.

Effect: The collector continues to run.

Recovery: Review all DBS objects that use the same SOURCEFILE as the offending DBS specification and verify the MAPTOFILE settings are the same.

2080	COLLECTOR DETECTED INCONSISTENT ROOTPARTITION DEFINITIONS FOR DBS OBJECT <dbs name> (VALUE < ROOTPARTITION value>, FILE <file>); ALL MATCHING DBS FOR THE SAME CONSUMER MUST HAVE IDENTICAL ROOTPARTITION OPTIONS

Cause: The collector outputs this message to indicate that there are ambiguous definitions for the ROOTPARTITION parameter in the DBS objects. The message identifies the offending DBS object, the ROOTPARTITION value that is being used, and the file matched the offending DBS objects.

Effect: The collector continues to run.

Recovery: Review all DBS objects that use the same SOURCEFILE as the offending DBS specification and verify the ROOTPARTITION settings are the same.

2080	COLLECTOR DETECTED INCONSISTENT ALLPARTITIONS DEFINITIONS FOR DBS OBJECT <dbs name> (VALUE < ALLPARTITIONS value>, FILE <file>); ALL MATCHING DBS FOR THE SAME CONSUMER MUST HAVE IDENTICAL ALLPARTITIONS OPTIONS

Cause: The collector outputs this message to indicate that there are ambiguous definitions for the ALLPARTITIONS parameter in the DBS objects. The message identifies the offending DBS object, the ALLPARTITIONS value that is being used, and the file matched the offending DBS objects.

Effect: The collector continues to run.

Recovery: Review all DBS objects that use the same SOURCEFILE as the offending DBS specification and verify the ALLPARTITIONS settings are the same.

2080	COLLECTOR DETECTED INCONSISTENT FETCHSOURCE DEFINITIONS FOR DBS OBJECT <dbs name> (VALUE < FETCHSOURCE value>, FILE <file>); ALL MATCHING DBS FOR THE SAME CONSUMER MUST HAVE IDENTICAL FETCHSOURCE OPTIONS

Cause: The collector outputs this message to indicate that there are ambiguous definitions for the FETCHSOURCE parameter in the DBS objects. The message identifies the offending DBS object, the FETCHSOURCE value that is being used, and the file matched the offending DBS objects.

Effect: The collector abends.

Recovery: Review all DBS objects that use the same SOURCEFILE as the offending DBS specification and verify the FETCHSOURCE settings are the same.

2080	COLLECTOR DETECTED INCONSISTENT HASHBUCKETCOUNT DEFINITIONS FOR DBS OBJECT <dbs name> (VALUE < HASHBUCKETCOUNT value>, FILE <file>); ALL MATCHING DBS FOR THE SAME CONSUMER MUST HAVE IDENTICAL HASHBUCKETCOUNT OPTIONS

Cause: The collector outputs this message to indicate that there are ambiguous definitions for the HASHBUCKETCOUNT parameter in the DBS objects. The message identifies the offending DBS object, the HASHBUCKETCOUNT value that is being used, and the file matched the offending DBS objects.

Effect: The collector abends.

Recovery: Review all DBS objects that use the same SOURCEFILE as the offending DBS specification and verify the HASHBUCKETCOUNT settings are the same.

2080	COLLECTOR DETECTED INCONSISTENT HASHBUCKETID DEFINITIONS FOR DBS OBJECT <dbs name> (VALUE < HASHBUCKETID value>, FILE <file>); ALL MATCHING DBS FOR THE SAME CONSUMER MUST HAVE IDENTICAL HASHBUCKETID OPTIONS

Cause: The collector outputs this message to indicate that there are ambiguous definitions for the HASHBUCKETID parameter in the DBS objects. The message identifies the offending DBS object, the HASHBUCKETID value that is being used, and the file matched the offending DBS objects.

Effect: The collector abends.

Recovery: Review all DBS objects that use the same SOURCEFILE as the offending DBS specification and verify the HASHBUCKETID settings are the same.

2080	COLLECTOR DETECTED INCONSISTENT MAPTOPRIPTN DEFINITIONS FOR DBS OBJECT <dbs name> (VALUE <MAPTOPRIPTN value>, FILE <file>); ALL MATCHING DBS FOR THE SAME CONSUMER MUST HAVE IDENTICAL MAPTOPRIPTN OPTIONS

Cause: The collector outputs this message to indicate that there are ambiguous definitions for the MAPTOPRIPTN option of the PURGEDATAOPTIONS parameter in the DBS objects. The message identifies the offending DBS object, the MAPTOPRIPTN value that is being used, and the file matched the offending DBS objects.

Effect: The collector abends.

Recovery: Review all DBS objects that use the same SOURCEFILE as the offending DBS specification and verify the MAPTOPRIPTN settings are the same.

2123	SOURCEFILE (<file pattern>) MUST BE FULLY-QUALIFIED IF ALLPARTITIONS IS ON for DBS <dbs name>

Cause: The DBS ALLPARTITIONS parameter is set ON but the SOURCEFILE does not contain a fully-qualified filename.

Effect: The DBS object is not added.

Recovery: Set the SOURCEFILE to a fully-qualified primary partition filename or change the collector to SBCOLL and try again.

2026	SBCOLL STOPPING: UNABLE TO READ AUDIT. AUDIT FILE FOR POSITION <prefix> <seqno> DOES NOT EXIST and MUST BE RESTORED TO ALLOW READING

Cause: The collector tried to look for an audit file that has rolled off the system.

Effect: The collector abends.

Recovery: Restore the missing audit file and restart the collector.

ERR 2190	TARGETFILE (<file>) PARTS MUST BE EITHER STATIC, A SINGLE ASTERISK, OR CONTAIN THE SAME WILDCARD CHARACTERS AS SOURCEFILE
ERR 2190	MAPTOFILE (<file>) PARTS MUST BE EITHER STATIC, A SINGLE ASTERISK, OR CONTAIN THE SAME WILDCARD CHARACTERS AS SOURCEFILE
ERR 2190	TARGETFILE (<file>) PARTS MUST BE EITHER STATIC, A SINGLE ASTERISK, OR CONTAIN THE SAME WILDCARD CHARACTERS AS MAPTOFILE

Cause: Each part of TARGETFILE/MAPTOFILE must either contain no wildcard characters, be a single asterisk, or contain the same wildcard characters as the respective part in SOURCEFILE/MAPTOFILE.

Effect: The DBS is not added.

Recovery: Fix the TARGETFILE/MAPTOFILE parameter and try again.

ERR 2191	TARGETFILE (<file>) PARTS MUST BE EITHER STATIC OR A SINGLE ASTERISK IF MAPFROMSOURCE IS ON
ERR 2191	MAPTOFILE (<file>) PARTS MUST BE EITHER STATIC OR A SINGLE ASTERISK IF MAPFROMSOURCE IS ON

Cause: If MAPFROMSOURCE is ON, each part of TARGETFILE/MAPTOFILE must either contain no wildcard characters or be a single asterisk.

Effect: The DBS is not added.

Recovery: Fix the TARGETFILE/MAPTOFILE parameter or disable MAPFROMSOURCE and try again.

ERR 2192	TARGETFILE (<file>) PARTS CANNOT CONTAIN MULTIPLE ASTERISKS
ERR 2192	MAPTOFILE (<file>) PARTS CANNOT CONTAIN MULTIPLE ASTERISKS

Cause: Each part of TARGETFILE/MAPTOFILE cannot contain multiple asterisks.

Effect: The DBS is not added.

Recovery: Fix the TARGETFILE/MAPTOFILE parameter and try again.

2080	UNABLE TO RESOLVE WILDCARDED MAPTOFILE (<mapto>) IN DBS <dbs name> FOR FILE/TABLE <file>, ABENDING

Cause: The collector failed to resolve the wildcarded MAPTOFILE. Either the pattern in MAPTOFILE is invalid or the resolved file is invalid.

Effect: The collector abends.

Recovery: Fix the MAPTOFILE parameter for the DBS and restart the collector.

ERR 1604:	UNABLE TO RESOLVE TARGET FILE FROM <file> DUE TO ISSUE WITH TARGETFILE (<targetfile>)

Cause: The Consumer was unable to generate a valid target file name from the specified file due to an issue with the TARGETFILE parameter.

Effect: The consumer abends.

Recovery: Check the TARGETFILE specification. (Use CHECKTARGETS to determine which files/tables are included.)

ERR 1605:	UNABLE TO RESOLVE TARGET FILE FROM <file> DUE TO ISSUE WITH SOURCEFILE (<sourcefile/maptofile>)

Cause: The Consumer was unable to generate a valid target file name from the specified file due to an issue with the SOURCEFILE or MAPTOFILE parameter.

Effect: The consumer abends.

Recovery: Check the SOURCEFILE/MAPTOFILE specification. (Use CHECKTARGETS to determine what files/tables are included.)

ERR 2188	CONS <cons name> MISSING REQUIRED SBMXSERV PARAMETER SBMXSERVPROCESS

Cause: A DBS has SBMXSERVENABLED ON, but its consumer does not have SBMXSERVPROCESS set.

Effect: The DBS is not added.

Recovery: Set the SBMXSERVPROCESS parameter and try again.

2017	CONSUMER CAN'T PURGE SQLCI OUTFILE <file name> FOR SQL DDL EVENT, FILE CODE NOT 101

Cause: The file name associated with the SQLCI OUTFILE for processing SQL DDL events is not an edit file.

Effect: The Consumer abends.

Recovery: Check the file name reported in the message. The SQLCI OUTFILE should be located in the HPE Shadowbase program subvolume. It contains the Consumer process name along with the suffix "SCI". Correct the condition that caused the file error. You can purge the file if HPE Shadowbase is down. Then, restart HPE Shadowbase.

4556	QUEUE FILE <filename> FILLED WITH BLOCK AT RBA <rba>, LAST EVENT POSITION <pos>, LAST EVENT TIME <time>

Cause: The Queue Manager wrote the final block to the specified queue file with an event at the audit trail position and timestamp listed.

Effect: Informational only.

Recovery: None needed.

4556	QUEUE FILE <filename> READY TO ROLL; LAST EVENT TIME <time>

Cause: The consumer restart position has been updated to past the last event in the specified queue file, so the Queue Manager has begun to roll the file.

Effect: Informational only.

Recovery: None needed.

2116	EVENT SKIPPED DUE TO MISSING HASHBUCKETID DBS FOR FILE <file>, HASHBUCKETID VALUE(S): (<values>)

Cause: An event on the specified file did not match the HASHBUCKETID of any of the matching DBSes, so the event was skipped. If DBSes in different consumers have different HASHBUCKETCOUNT parameters, the same hash can result in different HASHBUCKETID values. If this happens, <values> will be a list of HASHBUCKETID values.

Effect: If HASHBUCKETSKIPACTION is set to STOP, the collector abends. If HASHBUCKETSKIPACTION is set to WARN, the collector skips the event and continues.

Recovery: Fix your configuration so that all HASHBUCKETID values are specified by a DBS. If intentional, change HASHBUCKETSKIPACTION to WARN or CONTINUE.

ERR 2201:	ALL CONSUMERS MUST HAVE A QMGR FOR A ZDL ENABLED COLLECTOR

Cause: AUDMON attempted to start a ZDL enabled collector, but not all of its consumers had a QMGR configured.

Effect: The collector is not started.

Recovery: Configure a QMGR for all consumers attached to the collector.

4542	COLLECTOR ZDL PROCESSING DISABLED, REPLICATION ACTION SET TO '<action>'

Cause: The collector was started in synchronous mode, but the REPLICATION parameter is currently set to SUSPEND or SKIP.

Effect: All transactions will be processed asynchronously until the collector's REPLICATION parameter is set to ON.

Recovery: None needed. The collector will resume replicating synchronously once the REPLICATION parameter is set to ON.

4542	COLLECTOR ZDL PROCESSING ENABLED, REPLICATION ACTION SET TO 'ON'

Cause: The collector was started in asynchronous mode, but the REPLICATION parameter was set to ON.

Effect: The collector will resume synchronous replication.

Recovery: None needed.

4542	OPEN REQUEST RECEIVED FROM COORDINATOR <process name> WHILE COLLECTOR IS CONFIGURED FOR SYNC OPERATION, BUT COLLECTOR IS CONFIGURED TO NOT REPLICATE EVENTS

Cause: The collector was started in synchronous mode, but the REPLICATION parameter is currently set to SUSPEND or SKIP.

Effect: All transactions will be processed asynchronously until the collector's REPLICATION parameter is set to ON.

Recovery: None needed. The collector will begin replicating synchronously once the REPLICATION parameter is set to ON.

ERR 1387:	INCORRECT LABELCACHEDEPTH PARAMETER, VALID RANGE FOR COLLECTOR IS <low valid number> TO <high valid number> (MEGABYTES)

Cause: The COLL LABLECACHEDEPTH parameter is invalid.

Effect: The LABELCACHEDEPTH is set to the default value.

Recovery: Set LABELCACHEDEPTH within the range in the message.

ERR 2202:	QMGRNAME MUST NOT BE SPECIFIED FOR A DIRECT STP SERVER

Cause: The QMGRNAME parameter was specified incorrectly for a direct STP server. Note that there is no Queue Manager associated with a direct STP server.

Effect: The Consumer is not added.

Recovery: Remove the QMGRNAME parameter from the configuration or use the RESET command.

ERR 2197:	CAN'T SUSPEND CONSUMER, QMGR NOT STARTED

Cause: The SUSPEND CONS command was entered but failed because the QMGR was not started.

Effect: The SUSPEND command did not occur.

Recovery: Start the QMGR and reenter the SUSPEND command.

ERR 2198:	CAN'T RESUME CONSUMER, QMGR NOT STARTED

Cause: The RESUME CONS command was entered but failed because the QMGR was not started.

Effect: The RESUME command did not occur.

Recovery: Start the QMGR and reenter the RESUME command.

ERR 2200:	SUSPEND QMGR NOT SUPPORTED FOR RECOVERY QMGR
ERR 2200:	PULSE NOT SUPPORTED FOR RECOVERY QMGR
ERR 2200:	SUSPENDUPD NOT SUPPORTED FOR RECOVERY QMGR
ERR 2200:	RESUMEUPD NOT SUPPORTED FOR RECOVERY QMGR
ERR 2200:	UNDO NOT SUPPORTED FOR RECOVERY QMGR

Cause: A command was entered that is not supported for QMGRs that are configured with a collector.

Effect: The command is not executed.

Recovery: None required.

4546	MISSING EXPECTED FILE, QUEUE SEQNO <seqno> FOR QMGR RECOVERY

Cause: The queue manager was building a restart file based on existing queue files, but there was a gap in the queue sequence numbers.

Effect: The queue manager abends.

Recovery: Restore the missing queue file and restart the queue manager.

2121	ERR 2121: CONSUMER MUST BE BROUGHT DOWN BEFORE QSETSEND COMMANDS EXECUTE

Cause: A QSETSEND command has been issued, but the associated consumer was running.

Effect: The QSETSEND command causes the error message to be issued, but makes no other changes.

Recovery:	Stop the Consumer associated with this Queue Manager before re-sending the QSETSEND command. The consumer can be started after the QSETSEND command is accepted.
4565	CONS REPLICATION FLOW STOPPED DUE TO QSETSEND STOP COMMAND FOR CONSUMER <name>(<process name>) WITH ACTIVE TRANSACTION(S)

Cause: The Queue Manager has reached the stop point set by the QSETSEND command, but some transactions have not been committed or aborted.

Effect: If left in this state all transactions that have not been completed will be automatically aborted when the system-defined timeout is reached.

Recovery: Issue a QSENDRESUME command before the timeout window to resume the flow of data between the Queue Manager and the Consumer.

4565	CONS REPLICATION FLOW STOPPED DUE TO QSETSEND STOP COMMAND FOR CONSUMER <name>(<process name>)

Cause: The Queue Manager has reached the stop point set by the QSETSEND command.

Effect: Data from the collector will be written to the QMGR Queue Files, but will not be applied to the target database.

Recovery: When desired issue a QSENDRESUME command to resume the flow of data between the Queue Manager and the Consumer.

4567	QSETSEND STOP CRITERIA SET FOR CONSUMER <name>

Cause: A stop point has been set for the Queue Manager using the QSETSEND command.

Effect: When it reaches the stop point the Queue Manager will stop sending data to the Consumer. The stop point can be viewed using the STATS QMGR <name> command.

Recovery: None. This is informational.

4545	CONSUMER RESTART POINT SET TO <time> ADT POS <filename><seqno>.<rba>. START AND RUN CONSUMER TO REPLAY DATA

Cause: A QSETSEND START command was issued. The time/ADT position is the first event at or after the requested time/ADT position.

Effect: The Queue Manager will send data to the Consumer from the start time/ADT position listed once the Consumer has been started and a run has been issued.

Recovery: None. This is informational.

6015	ERR 6015: REQUESTED START POINT IS NOT IN QUEUE FILES. QSETSEND COMMAND ABORTED CURRENTLY AVAILABLE RANGE OF EVENTS IN QUEUE FOUND AS FOLLOWS: FIRST EVENT AT <begin position> LAST EVENT AT <end position>

Cause: A QSETSEND START command was issued, but the requested start is outside the bounds of the current QMGR Queue Files.

Effect: The Queue Manager will not replay data from the QMGR Queue Files from the requested time/position.

Recovery: Either re-issue the QSETSEND command using a time or ADT position within the bounds of the current files or start and run the consumer to continue processing from where replication for this thread was paused.

6016	ERR 6016: QMGR QUEUE FILES ARE EMPTY. QSETSEND COMMAND ABORTED.

Cause: A QSETSEND START command was issued, but the QMGR Queue Files do not contain any events.

Effect: The Queue Manager will not replay data.

Recovery: Either stop the Queue Manager, restore the desired QMGR Queue Files, restart the Queue Manager, and re-issue the QSETSEND command, or generate data during normal operations and issue the QSETSEND command once DML or DDL events have started flowing from the Collector to the Queue Manager.

6018	ERR 6018: EVENT DATA NOT FOUND IN QMGR QUEUE FILES. QSETSEND COMMAND ABORTED.

Cause: A QSETSEND START command was issued, but the QMGR Queue Files do not contain any events matching the requested audit trail position.

Effect: The Queue Manager will not replay data.

Recovery: Using the QSETSEND help in AUDCOM, ensure that the position is correctly entered. Evaluate the QMGR Queue Files to ensure the requested start position is in the current set of files, then re-send the command.

4545	CONSUMER RESTART POINT SET TO EOF. START AND RUN CONSUMER TO REPLAY DATA

Cause: A QSETSEND START command was issued. The requested start position was EOF.

Effect: The Queue Manager will send data to the Consumer from the EOF position that was current when the QSETSEND command was issued once the Consumer has been started and run.

Recovery: None. This is informational.

4565	CONS REPLICATION FLOW STOPPED DUE TO QSETSEND STOP COMMAND FOR CONSUMER <name>(<process name>) WITH ACTIVE TRANSACTION(S)

Cause: A QSETSEND STOP command was issued and the consumer is now at that stop point and there are transactions applied to the target, but not committed or aborted at the time of this message.

Effect: No data will flow through the consumer until a QSENDRESUME command is sent. There may be holds on the target table until the transaction is applied (committed) or rolled back (aborted). The transaction will be automatically aborted after a system-defined period of time.

Recovery: Send a QSENDRESUME command before the transaction timeout time to prevent losing the data in that transaction.

4566	CONS REPLICATION FLOW STOPPED DUE TO QSETSEND STOP COMMAND FOR CONSUMER <name>(<process name>)

Cause: A QSETSEND STOP command was issued and the consumer is now at that stop point.

Effect: No data will flow through the consumer until a QSENDRESUME command is sent.

Recovery: Send a QSENDRESUME command when ready to start sending data to the consumer again.

4566	QSETSEND STOP CRITERIA SET FOR CONSUMER <name> at <stop criteria>

Cause: A QSETSEND STOP command was issued for the specified consumer.

Effect: When the consumer reaches the stop point it will stop replicating to the target. The collector will continue to collect data from TMF and store it in the QMGR's Queue Files to potentially apply later.

Recovery: None. This is informational.

4532	EVENTS BETWEEN <first adt address> AND <second adt address> WILL NOT BE APPLIED TO THE TARGET DATABASE DUE TO A QSETSEND COMMAND. IF THIS IS NOT DESIRED BEHAVIOR USE THE QSETSEND COMMAND TO SET THE START POSITION TO THE DESIRED POSITION BEFORE STARTING THE CONSUMER.

Cause: A QSETSEND START command was issued for the specified QMGR. The new start position is beyond the associated Consumer's current restart position.

Effect: When the consumer is started events between the consumer's current position in audit and the position requested in the QSETSEND START command will not be applied to the target.

Recovery: This is informational, but if skipping these events is unintended the consumer's start point can be set to the current position using a QSETSEND START command before starting the consumer. (<first_adt_address> listed in the EMS message is the consumer's current position)

4568	QSENDRESUME SENT, BUT QSETSEND STOP CRITERIA NOT MET: <stop criteria>

Cause: A QSENDRESUME command was issued, but previously a stop point had been set using the QSETSEND STOP command. The QSENDRESUME command was sent before the consumer got to that stopping point.

Effect: When the QSENDRESUME command is sent it re-sets the stop point and the Queue Manager will not longer stop at that point.

Recovery: None. This is informational. If the stop is still desired a new QSETSEND STOP command should be sent.

4545	CONSUMER RESTART POINT SET TO <time> MAT POS <filename>.<rba>. START AND RUN CONSUMER TO REPLAY DATA

Cause: A QSETSEND START command was issued, moving the consumer's current position to the position indicated in this EMS message.

Effect: When the QSETSEND START command is sent it sets the consumer's start point to the event at the specified position. The consumer will replicate from this point forward.

Recovery: None. This is informational.

4545	CONSUMER RESTART POINT SET TO <time> ADT POS <filename>.<rba>, MAT POS <mfilename>.<mrba>. START AND RUN CONSUMER TO REPLAY DATA

Cause: A QSETSEND START command was issued, moving the consumer's current position to the position indicated in this EMS message.

Effect: When the QSETSEND START command is sent it sets the consumer's start point to the event at the specified position. The consumer will replicate from this point forward.

Recovery: None. This is informational.

4545	CONSUMER RESTART POINT SET TO EOF. START AND RUN CONSUMER TO REPLAY DATA

Cause: A QSETSEND START command was issued, moving the consumer's current position to the current QMGR Queue File EOF.

Effect: When the QSETSEND START command is sent it sets the consumer's start point to the current QMGR Queue File EOF. The consumer will replicate from this point forward and skip any events between the consumer's previous position and the current QMGR Queue File EOF.

Recovery: None. This is informational.

4545	QMGR QUEUE FILES ARE EMPTY. QSETSEND COMMAND ABORTED.

Cause: A QSETSEND START command was issued, but there are no events in the current QMGR Queue Files.

Effect: No change will be made as a result of this command.

Recovery: None. This is informational.

4545	NO DATA FOUND IN QFILES BEFORE <begin_time> ADT POS <begin_adt> OR AFTER <end_time> ADT POS <end_position>

Cause:	A QSETSEND START command was issued, but there are no events in the current QMGR Queue Files that match or exceed the requested time/position.
Effect:	No change will be made as a result of this command.
Recovery:	None. If the consumer's start position should be modified the operator should choose a start point between the begin criteria and the end criteria or restore previous QMGR Queue Files.

7600	RM TIMEOUT DETECTED. ACTIVE AND INTERIM TRANSACTIONS PROCEEDED ASYNCHRONOUSLY. SBCOOR WILL ATTEMPT TO REOPEN RM TO REESTABLISH CAPABILITY FOR SYNCHRONOUS PROCESSING.

Cause:	The coordinator process was unable to respond to TMF within the configured RM_TIMEOUT period, causing TMF to remove the resource manager from voting.
Effect:	The active and interim transactions were allowed to commit without waiting for synchronous replication. The coordinator will automatically reopen the resource manager to rejoin transaction voting and allow for synchronous transaction replication again.
Recovery:	No action needed to restore synchronous replication, as the coordinator will automatically attempt to reopen the resource manager and resume synchronous processing.

7600	DETECTED INSUFFICIENT TMF VERSION FOR USE OF COORDINATOR RM_TIMEOUT FEATURE. IDENTIFIED TMF VERSION: <version> SUB-VERSION: <sub-version> PLATFORM: <platform-id> SPR ID: <spr>. MINIMUM SPR VERSION FOR USE IS: ANZ. SBCOOR WILL CONTINUE WITHOUT USE OF THE RM_TIMEOUT FEATURE

Cause:	The coordinator process detected that the version of TMF on the system is insufficient to use the RM_TIMEOUT feature.
Effect:	The coordinator will proceed without configuring a timeout value for the resource manager.

Recovery: None needed, though use of the RM_TIMEOUT feature is preferred. TMF SPR T8607L01^ANZ or a superceding version of TMF is required for usage of the RM_TIMEOUT feature.

7600	DETECTED SUFFICIENT TMF VERSION FOR USE OF COORDINATOR RM_TIMEOUT FEATURE. IDENTIFIED TMF VERSION: <version> SUB-VERSION: <sub-version> PLATFORM: <platform-id> SPR ID: <spr>. MINIMUM SPR VERSION FOR USE IS: ANZ. SBCOOR WILL CONTINUE WITHOUT USE OF THE RM_TIMEOUT FEATURE

Cause: The coordinator process detected that the version of TMF on the system is sufficient to use the RM_TIMEOUT feature, and logging of this detection was enabled by the setting of the RM_TIMEOUT_USAGE_OVERRIDE parameter.

Effect: Informational. The coordinator will enable usage of the configured RM_TIMEOUT value.

Recovery: None needed.

7600	ENABLING USE OF RM_TIMEOUT FEATURE DUE TO THE SETTING OF THE RM_TIMEOUT_USAGE_OVERRIDE PARAMETER (<value>)
7600	DISABLING USE OF RM_TIMEOUT FEATURE DUE TO THE SETTING OF THE RM_TIMEOUT_USAGE_OVERRIDE PARAMETER (<value>)

Cause: The RM_TIMEOUT_USAGE_OVERRIDE parameter was set to override the coordinator's default determination on the availability and usage of the RM_TIMEOUT feature.

Effect: Informational only. The stated action will take place, enabling or disabling usage of the RM_TIMEOUT feature.

Recovery: None needed.

7600	ATTEMPT TO CHECK TMF VERSION FAILED WITH ERROR <error>. SBCOOR WILL ATTEMPT TO USE CONFIGURED " "RM_TIMEOUT OF <rm-timeout>

Cause: The specified error occurred when attempting to call the guardian procedure TMF_VERSION_EXT_ to check the version of TMF running on the system.

Effect: Informational. The coordinator will attempt to proceed with configuring the RM_TIMEOUT feature with the output configured value.

Recovery: None needed, unless the RM_TIMEOUT feature is not available on the system. Otherwise, take corrective action based on the file system error code returned.

New and Modified SOLV User/EMS Messages

None in this release.

New and Modified TACL Parameters

SBMXSERVIPCTIMEOUT

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		2 * SQLLOCKWAIT	1	32767	No

By default, when the consumer is waiting for a response from SBMXSERV, it will wait 2 * SQLLOCKWAIT hundredths of seconds before logging an error and abending. You can increase or decrease this timeout value by entering the following TACL command prior to starting AUDMON.

```
PARAM SBMXSERVIPCTIMEOUT <number>
```

SBCHECKMATCONTROLFILE

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.010		1		0	1	No

By default, the consumer will not allow the parameter CONTROLFILE to be created on non MAT data volumes. Setting this TACL parameter value to 0 will allow the consumer parameter CONTROLFILE to be created on MAT or AUX data volumes. The default value of 1 enforces the CONTROLFILE to be on MAT data volumes.

```
PARAM SBCHECKMATCONTROLFILE < 0 | 1 >
```

SBSKIPADTCOMPRESSINF

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.010		0		0	1	No

In situations where the consumer sees an audit-compressed updated for a row that does not exist in the target table and FETCHSOURE is not enabled it will fail and log a message. This TACL parameter allows the consumer to skip these events while logging a message. The default 0 will log message and fail, option 1 will log message and skip the event.

```
PARAM SBSKIPADTCOMPRESSINF < 0 | 1 >
```

New and Modified User Information Messages

The new user information messages are:

6017	ERR 6017: CONSUMERS MUST BE BROUGHT DOWN BEFORE REPLAYING QMGR DATA

Cause: A QSETSEND START command was issued, but the consumer associated with this replication thread is running.

Effect: No change will be made as a result of this command.

Recovery: Stop the consumer and re-issue the QSETSEND START command.

6015	ERR 6015: REQUESTED START POINT IS NOT IN QUEUE FILES. QSETSEND COMMAND ABORTED CURRENTLY AVAILABLE RANGE OF EVENTS IN QUEUE FOUND AS FOLLOWS: FIRST EVENT AT <position in audit> LAST EVENT AT <position in audit>

Cause: A QSETSEND START command was issued, but there are no events in the current QMGR Queue Files that match or exceed the requested time/position.

Effect: No change will be made as a result of this command.

Recovery: Choose a start position between the begin criteria and the end criteria and re-send the command.

6016	ERR 6016: QMGR QUEUE FILES ARE EMPTY. QSETSEND COMMAND ABORTED.

Cause: A QSETSEND START command was issued, but there are no events in the current QMGR Queue Files.

Effect: No change will be made as a result of this command.

Recovery: None. This is informational.

6018	ERR 6018: EVENT DATA NOT FOUND IN QMGR QUEUE FILES. QSETSEND COMMAND ABORTED.

Cause: A QSETSEND START command was issued, but there are no events in the current QMGR Queue Files.

Effect: No change will be made as a result of this command.

Recovery: None. This is informational.

New and Modified User Exit Functions

None in this release.

New and Modified Measure Counters

None in this release.

New and Modified ETL Toolkit Parameters

ADTCOMPRESSMETHOD { OMIT }
{ NULL }
{ USER }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		OMIT	n/a	n/a	NO

This parameter will determine what value is placed in a column when that value is missing. This will most likely occur during an audit compressed update. The default value OMIT, will leave the value blank and not add anything in its place. The value NULL will add "NULL" for CUSTOM format and null for JSON format. The value USER will add the value used in ADTCOMPRESSVALUE.

ADTCOMPRESSVALUE < value >

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		"NULL"	n/a	n/a	NO

This parameter is used to set the value of what will replace missing column values when ADTCOMPRESSMETHOD is set to USER. This value should be string of max size 50. The default value is the string "NULL".

JSONSTRINGFORMAT { ASCII }
{ HEX }
{ UTF8 }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000	7.010	ASCII	n/a	n/a	NO

This parameter determines how JSON formatting will handle displaying strings. When the default, ASCII, is set, data will be output in ASCII format and any value that contains an unprintable character will be converted to the format '\uXXXX' with the X's being the 4-digit hex value (some values will be converted to the control character they represent. Ex. TAB will be displayed as '\t' rather than '\u0009'). If set to HEX, then all character datatypes will be converted into two-character hex values. If set to UTF8, all values will be converted into UTF-8 with the ISO 8859-1 standard.

KAFKANOWAITTIME < *time* >

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.010		10	1	21474836	NO

This parameter is used to set the timeout value for opening and writing to the uLinga for Kafka process. These operations are perform nowaited and this value will be the timeout for the following AWAITIO call. The values are in seconds with a default of 10.

Validating Downloaded Files

Gravic provides an SHA1 checksum and the size of the release package (file) in bytes. The purpose of this is so that the user can confirm that the release package they downloaded is valid and free from corruption/tampering.

- For HPE Shadowbase releases, this information is published in the HPE SOFTDOC that corresponds with the specific SPR version of the software. The HPE SOFTDOC for a specific SPR version is available on the HPE SCOUT portal. This information can also be found on the Gravic Shadowbase website here (https://www.shadowbasesoftware.com/releases).
- For non-HPE Shadowbase releases, for example Shadowbase releases obtained directly from Gravic (regardless if an HPE TCD release or a direct Gravic licensee release), this information is included in the body of the Gravic email that provides the link to download the software. This information can also be found on the Gravic Shadowbase website here (https://www.shadowbasesoftware.com/releases/gravic).

All Shadowbase releases are provided as Windows format download files. The user should validate the downloaded release file size by comparing it with the published release file size.

In order to validate the SHA1 checksum of the downloaded file, users can run the following command on a Windows system where the file was downloaded in order to generate the checksum in their own environment:

```
certutil -hashfile <downloaded filename> SHA1
```

Users can compare the certutil-generated checksum value against the published checksum value to make sure they match. Note that any other SHA1 generation tool can also be used to create the checksum value on the downloaded file. The certutil method is just one available option for Windows environments, and is included as part of Certificate Services.

Installation Instructions (HPE Nonstop Shadowbase)

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

1a) 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 H-based and J-based systems:

```
FUP ALTER SBPAK, CODE 800
```

For L-based systems:

```
FUP ALTER SBPAK, CODE 500
```

1b) Optionally (if installing the ZDL product), also binary FTP the SBSYPAK file 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 L-based systems:

```
FUP ALTER SBSYPAK, CODE 500
```

2a) 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

Note: Using UNPAK to extract the files instead can result in errors and fail to unpack the files with older versions of UNPAK. Using the RUN command as listed above does not have this issue.

2b) If the SBSYPAK was transferred in step 1b, to "unpak" the files from the self-extracting SBSYPAK, enter the following command:
RUN SBSYPAK, *.*.*, VOL <restore volume and subvolume>, LISTALL, MYID

Note: SBSYPAK should be UNPAKed using the same volume and subvolume used to UNPAK SBPAK in step 2a.

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 the INSTALL 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 USRXLIBN (the native library) IS ADDED TO THE RESULTING OBJECT FILE THAT WILL SUBSEQUENTLY BE INCLUDED IN THE FINAL LINK.

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

HPE Release File Structure

If this release is obtained through HPE Scout, the downloaded file is a single zip file that is intended to be extracted on a Windows PC. When the zip file has been extracted, it will create a folder structure on the PC with all of the various pieces of Shadowbase in appropriate folders. This folder structure is as follows:

CONTENTS
Each component's installation files are in a separate directory in the ZIP file or on the product DVD under the directory \T1122L70. This file set contains the following files:

File or Directory	Description
\readme.txt	Readme file for the release.
\SBEnterpriseManager	Directory containing the files required to install SEM on a PC.
\TNS-E	Directory containing the installation files for the HPE Integrity Nonstop i versions of NS Repl, SAR, and Compare.
SBAuditReader	Directory containing the installation files for SAR.
SBCompare	Directory containing the installation files for Compare.
SBReplication	Directory containing the installation files for Nonstop Guardian replication.
\TNS-X	Directory containing the installation files for the HPE Integrity Nonstop X versions of NS Repl, SAR, and Compare.
SBAuditReader	Directory containing the installation files for SAR.
SBCompare	Directory containing the installation files for Compare.
SBReplication	Directory containing the installation files for Nonstop Guardian replication.

Known Problems Remaining

1) Replication of SQL/MP tables with DCOMPRESS 2 is not supported at this time due to an issue with the HPE TMF ARLIB2 audit reading library in handling audit data from these tables, which will result in the Collector failing. The issue is described in the HPE knowledgebase article titled "TMFARLB2 - ARFETCHAFTERIMAGE Returns Garbage When DCOMPRESS 2 Is Used", found here: https://support.hpe.com/hpesc/docDisplay?docLocale=en_US&docId=ns0106884en_us. SQL/MP tables with DCOMPRESS 2 must be removed from the replication configuration. The attribute is not alterable, so these tables would need to be recreated with a different DCOMPRESS option (i.e. DCOMPRESS 1 or NO DCOMPRESS) in order to be replicated, and only audit trail data from the period with a different option for DCOMPRESS used could be replicated.

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

3) For SQL/MX, the documented maximum value for the 32-bit float is 3.4028234**7e**+38. However, the HPE Shadowbase for Other Servers ODBC Server gets a range error starting with 3.4028233**7e**+38 (difference highlighted and underlined). The cause of this issue remains under investigation.

4) 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.

5) 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 HPE for further investigation. At this time, it appears that this is caused by a problem in SQL/MP. A fix by HPE has yet to be released. HPE 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
```

HPE 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.

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 an 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) 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.

15) When replicating from an Enscribe source file using a DDL definition out to an 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.

16) 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. This can result in a consumer process replicating MX data to log errors similar to the following:

```
\NODE1.$DATA1.ZSDQAG7S.G2CS4Q00 / MXCAT_V.MXSCH_T.TABL_1
(DBS=DBS-SB-01) -> TARGET IS MISSING. CONSUMER ABENDING.
```

In order to get around this issue, the MX catalog metadata must be upgraded so that there is only one version of the schemas define. The following command can be run to upgrade the catalog metadata:

```
UPGRADE ALL METADATA IN CATALOG catalog
    [ RESTRICT | CASCADE ]
    [ output-spec ]

output-spec is:
  [ log-to-spec ] [ REPORTONLY ]

log-to-spec is:
  { [ LOG TO ] OUTFILE oss-file [ CLEAR ] | LOG TO HOMETERM }
```

17) 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.

18) SQL/MX bidirectional replication sometimes sends events to the target even if those events are part of a transaction that will ultimately be aborted. These events can trigger the Collector to change from SUSPNEXT to REPLSUSP mode, which can seem misleading since the events will not be applied to the target.

19) The DUMPTX COLL command does not display information for any tracked peer consumer transactions when using bidirectional Enscribe or SQL/MP replication with SBCOLL. The peer consumer transactions are counted in the number of active transactions displayed by the command, but they are not output.

20) The AUD level parameters for global translog support, PEERTRANSLOG and TRANSLOG, do not work for bidirectional replication with SBCOLL.

21) Replicating a created table using the new CREATEOPTIONS AUDITED parameter setting can lead to inconsistency between the Collector stats for the source and target tables.

22) The checking for the DOIFNOTRGATRUN DBS is only performed when an event is encountered for a target file for which the consumer does not have the schema information in memory. For SQL/MP tables, if the target table is dropped after one or more events have been replicated, and the schema is already loaded, the consumer will ultimately receive an error 11 for the event, ignore it, and continue, even if DOIFNOTRGATRUN is set to STOP.

23) When replicating ALTER events between multiple tables in succession (e.g. Table A to Table B, then Table B to Table C), extra ALTER and COMMIT events can be logged in the STATS.

24) The Consumer's I/O Tracing output can contain garbage characters when the Consumer encounters an error 73 (table or record is locked).

25) Regarding SOLV/SOLVnV shadparm input processing, Shadowbase currently only supports KEYVALUESTART and KEYVALUESTOP values up to 45 bytes long. With other methods SOLV can be pushed to use a KEYVALUESTART value above 239 or 240 bytes, but goes into an infinite loop.

26) SBGETREALTARGET will cause the Consumer to trap with an illegal address reference if called for an "end trans" (commit or abort) event, and tracing is enabled. SBGETREALTARGET should never be called for an end trans event.

27) The CONS SOLVMAXSESSIONS parameter cannot be set to 0 if a QMGR is configured. Attempting to do so will result in error 1235: REQUIRED PARAMETER NOT SET -> COLLNAME when attempting to add the Consumer object.

28) No validation occurs when the DBS VIEWNAME parameter is set to confirm that the DBS SOURCEFILE is a SQL/MP table. This parameter is only valid for SQL/MP source tables.

29) Internal Gravic testing revealed that setting the VVMKRHSTEXTENT overwrites the SOLV VVMKREXTENT value. The workaround is to create the SOLVMKRH file with the desired extents separately.

30) When attempting to SOLV load a SQL/MX table with more than 3 primary key columns with a mix of ascending and descending columns in the key, the SQL optimizer for SQL/MX will choose a poor access path to the data which results in SOLVMX eventually encountering an error 35 or slowing to a crawl. A different method must be used to load this table other than SOLV.

31) Attempting to use UNPAK to unpack the Shadowbase self-extracting PAK files will fail with older versions of UNPAK up to T1255H01^AAI. The workaround is to run the self-extracting PAK files instead of using UNPAK, using the method outlined in the installation instructions.

32) Modifications to a DDL record definition during replication will not be picked up by the consumer when either performing a RELEASE DBS followed by a CAPTURE DBS, or removing the existing DBS and adding a new DBS. The consumer process must be restarted in order for the change to be picked up.

33) The Queue Manager will abend with the following error if MAXCONSWRITES is set to 1 and the Consumer process is on a remote system:
QMGR QMGR01(\GRAVIC1.$GSQM1): ERROR 28 CALLING PROCEDURE SETMODENOWAIT ON FILE \VIV1.$GSCS1 , OPERATION 71, PARAM1 0, PARAM2 115
QMGR QMGR01(\GRAVIC1.$GSQM1): ABENDING: UNABLE TO SET EXPAND MESSAGE TRANSMISSION PRIORITY

    The workaround for this issue is the set MAXCONSWRITES to a value of 2 or higher. The default value for the MAXCONSWRITE parameter is 15.

34) The Shadowbase Consumer sporadically encounters SQL error -8204 / FS error 1061 when doing inserts into partitioned entry-sequenced SQL/MP tables. Based on internal testing, a work-around for this issue is to do a SET DBS CONTROLTABLEEXT1 (SEQUENTIAL INSERT OFF) for the related DBS object. Note that the CONTROLTABLEEXT parameter has superseded CONTROLTABLEEXT<n>, and if present, CONTROLTABLEEXT<n> will be ignored (CONTROLTABLEEXT1, in this case).

35) Shadowbase BASE24 handling has a potential efficiency issue when processing events associated with a DBS where the source or target file definitions could be associated with multiple files. This is especially apparent when using "true dynamic" selection mapping (i.e. no MAPFROMSOURCE and SOURCEFILE/TARGET is used only for mapping) and when using MAPFROMSOURCE ON and the SOURCEFILE has a wild-card (*) in the volume or sub-volume. For now, a workaround would be to limit the use of these types of DBS specifications for BASE24.

36) Enscribe DDL definitions with data types containing negative scale causes SBDDLUTL to generate invalid SQL/MP output. For instance, an Enscribe DDL data type of TYPE BINARY 64,-16 will generate a corresponding SQL/MP type of NUMERIC(18, -16).

37) SBDDLUTL traps if supplied an input file it cannot open (e.g. if the input file does not exist, or if SBDDLUTL does not have permission to open it)

38) When VARCHAR fields are included in the primary key of a SQL/MP table with AUDITCOMPRESS on, SQL/MP will pad VARCHAR columns in the WHERE clause of UPDATE statements with spaces, up to the configured size of the VARCHAR. This can result in data corruption.

39) MP DCR name mapping does not currently have awareness of the relationship between physical and virtual drives, i.e. when performing name mapping, SBSQLCI does not know ahead of time if the mapping performed for the PHYSVOL clause will generate a command with a physical volume that is in the same storage pool as the virtual volume used in the rest of the command. For example:

Given the following name mapping definitions:

```
MAP NAME $VDRV1.*.* TO $VDRV1B.*.*
MAP NAME $QA TO $QAX
```

The following CREATE TABLE statement will cause the target MPSERVER consumer to fail:

```
CREATE TABLE $VDRV1.TEST.NEWTAB (
    ...
)
CATALOG $VOL.CAT
PHYSVOL $QA;
```

The failure will occur if the physical drive $QAX is not within the same storage pool as the virtual drive $VDRV1B.

The work-around here is to just be conscientious about what name mappings are configured for the MP DCR environment if you're going to be running commands with the PHYSVOL clause. Check the storage pools on the target node ahead of time to make sure you properly configure name mapping. To do so:

```
$VOL SUBVOL 10> SCF
SCF - T9082H01 - (01MAY18) (02MAR18) - 08/02/2019 17:56:25 System \GRAVIC1
(c) Copyright 1986-2016 Hewlett Packard Enterprise Development LP
(Invoking \GRAVIC1.$SYSTEM.SYSTEM.SCFCSTM)
1-  INFO POOL $*, DETAIL
```

40) When generating an explain plan using SOLVMX, certain conditions can cause the explain output to contain this error: '*** ERROR[2105] This query could not be compiled because of incompatible Control Query Shape (CQS) specifications. Inspect the CQS in effect.' This is the result of the SQL compiler selecting an invalid SPLIT_TOP plan from the supplied select statements and CQS generated by SOLVMX. This problem does not seem to affect the SOLVMX load.

41) The MPSERVER consumer will not back out target-side transactions following a PAUSE that happens within the transaction/session.

For now, it is recommended to either avoid running multiple commands under a single explicit transaction on the source, or avoid using MPERRORACTION PAUSE until this issue is resolved.

42) Running an environment with some form of coordination (DCR coordination / DDL coordination), a Queue Manager on any thread, and TRACKTXFILE(S) has the potential to lock up the environment in some scenarios, and is not supported as a result. The following are options for workarounds:
    - Not using a QMGR in the Shadowbase environment if MP DCR or DDL Replication with coordination will occur under the same AUDMON
    - Turning MPCOORDINATION OFF for MP DCR or turning DDL coordination off (SET DBS CREATES NOCOORDINATION) for DDL replication
    - Using TID files with TIDFILECONSTANT ON instead of TRACKTXFILE, or otherwise disable TRACKTX processing.

43) SBSQLCI currently has a problem parsing SQL DDL commands that contain SQL/MP reserved/key words as part of column/file names. For instance, any place in a command that could take any user-specified data value cannot be a reserved/key word. Example:

```
SBSQLCI 1> CREATE TABLE DCRSRC.PARTTB (
mul(1, 2)> ID INT NOT NULL,
mul(1, 3)> NAME CHAR(30),
mul(1, 4)> PRIMARY KEY(ID))
mul(1, 5)> CATALOG $DATA9.VPCAT
mul(1, 6)> PARTITION (
mul(1, 7)> $VDV001.DCRSRC.PARTTB FIRST KEY(1001) CATALOG $DATA9.VPCAT,
mul(1, 8)> $VDV002.DCRSRC.PARTTB FIRST KEY(2001) CATALOG $DATA9.VPCAT,
mul(1, 9)> $VDV003.DCRSRC.PARTTB FIRST KEY(3001) CATALOG $DATA9.VPCAT
mul(1,10)> );
-- SQL operation complete.

SBSQLCI 2> CREATE INDEX DCRSRC.PARTIND1 ON DCRSRC.PARTTB (ID, NAME) PARTITION (
mul(2, 2)> $VDV001.DCRSRC.PARTIND1 FIRST KEY(1001) CATALOG $DATA9.VPCAT,
mul(2, 3)> $VDV002.DCRSRC.PARTINT1 FIRST KEY(2001) CATALOG $DATA9.VPCAT,
mul(2, 4)> $VDV003.DCRSRC.PARTIND1 FIRST KEY(3001) CATALOG $DATA9.VPCAT);
...DCRSRC.PARTTB (ID, NAME) PARTITION ( $VDV...
                            ^
SBSQLCI ERROR: syntax error (unexpected "NAME")
```

In the preceding example, the column name in an index column list cannot be NAME because NAME is a SQL/MP reserved/key word (in this case in a WITH SHARED ACCESS clause).

44) AUDMON will not allow a DIRECT environment to be configured without any DBSs and Shadowbase will fail to start if this is done. This may impact DCR-only environments where the only desired replication thread is the DCR thread, and no DBSs are needed. To work around this issue, add a DBS to the *dir configuration script. The source and target table specifications can be anything as this DBS is not effectively in use.

45) When SBSQLCI evaluates a TACL define contained within a SQL DDL statement, it does not remove the node name from the statement. This will cause a problem if replicating DCR events to systems that have a different node name than the source system. A work around is to use table/catalog/physvol names directly instead of TACL defines.

46) SBSQLCI's config file settings CTRL_FILENAME and CONFIG_FILENAME override the =SBCONTROLFILE and =SBSQLCICONFIG TACL defines. A work-around to this is to delete the CTRL_FILENAME and CONFIG_FILENAME parameters from the config file prior to defining the =SBCONTROLFILE and =SBSQLCICONFIG TACL defines.

47) The MPSERVER consumer does not currently have the ability to process certain SQL DDL commands such as ALTER or DROP when replicating MP DCR commands on un-audited target tables. The user will see a SQL error 1353 when attempting to UPDATE or DROP a SQL table that is not audited and perform the actions dictated by the MPERRORACTION parameter. To work around this issue:
    - MPERRORACTION is set to SKIP, either
        - Alter the given table to turn audit ON in a separate SQLCI session, then re-execute the DDL commands via SBSQLCI to have them re-replicate, or
        - Manually apply the DDL commands to the target table in a separate SQLCI session
    - MPERRORACTION is set to PAUSE: ALTER the table to turn AUDIT ON in a separate SQLCI session, then do RESUMEDDL RETRY in AUDCOM to retry the command
    - MPERRORACTION is set to STOP: either manually enter the DDL commands in a separate SQLCI session, or ALTER the table to turn audit ON, then restart Shadowbase

48) The use of the NEXTDOCTIME and NEXTDOCTRIGGER collector parameters causes several issues in the DOC rolling logic:
    a. The DOC rolls repeat once per second, even if NEXTDOCTRIGGER is set to TIMEOFDAY
    b. The NEXTDOCTIME parameter is ignored, and the DOC rolls begin as soon as the Shadowbase environment is started
    c. The EMS message pertaining to the initiated DOC roll shows a timestamp that is off by several hours; between 2 and 6 hours seen in internal testing.

49) The REPLICATION collector parameter cannot be altered from ON to SUSPEND in a bidirectional replication environment, specifically for SBCOLL, while the environment is running. The work-around for this is to shut down Shadowbase, alter the parameter, and re-start Shadowbase.

50) In replication environments between tables where primary keys differ and no user-exit is supplied, Shadowbase creates a faulty WHERE clause for non-

auditcompressed UPDATE events. The WHERE clause includes only the primary key columns that exist in both the source and target tables, which can cause unintended effects if the target table has columns in the primary key the source is missing (e.g. the update might affect multiple rows).

DBS mapping can be used as a work-around by using the FLD++ syntax to move the addition source table primary keys not present on the target into the WHERE clause. For example, in a source table containing keys C1, C2, and C3, where only key C1 is shared between the source and target table, the following DBS mapping will force the update statement to contain all three source table key columns in the WHERE clause:

[DBS+]U:DBS-01
[FLD++]C2=%C2%
[FLD++]C3=%C3%

51) SOLV does not support the use of the PARTONLYFNAME parameter in conjunction with partition key breaks whose data type does not match the type of the primary key column(s). For instance, if a table is defined with a primary key column of TYPE DATETIME YEAR TO FRACTION(6) and is partitioned with FIRST KEY values of type DATETIME YEAR TO DAY, then the SOLV load would fail with SQL error -7011:

```
The two date-time (DATE, TIME, TIMESTAMP, or DATETIME) operands in this
arithmetic or relational expression do not have the same range of DATETIME
fields.
```

The work-around for this issue is to remove the use of PARTONLYFNAME and instead prefer to use the SQLWHERE parameter instead to manually specify the key ranges for the SOLV load.

52) There is an existing issue in Shadowbase with dynamically adding replication threads to a running NSK -> Other Servers configuration. Shadowbase will allow you to run the necessary SET, ADD, START, and RUN commands within AUDCOM to make the changes to the environment, and the newly added objects can be seen within the STATUS AUD output and INFO DBS output, however, the collector will not recognize the new replication thread and will not select events for the newly added DBS. The work-around for this is to stop and re-start the collector. Note that this is not an issue for NSK -> NSK environments.

53) SBMAP can potentially encounter SQL error -7600:

```
Internal error: Expressions Generator.
```

This is the result of the map block being too large or complex. The work-around for this is to lower the complexity of the map block.

54) The SBMAP function NUMTOSTRING currently does not support NUMERIC or DECIMAL fields that have a scale greater than 0. In order to prevent data corruption, calling NUMTOSTRING on one of these unsupported fields will cause SBMAP to abend. As a workaround, these datatypes can still be converted using the CAST function.

55) SBMAP cannot handle SQL/MP compressed updates properly. For compressed updates, unchanged columns will not be present in the audit trail event. There is currently no way for SBMAP to ignore these columns only when they are not present. A workaround for this is to disable audit compression on your source table or set DBS FETCHSOURCE ON.

56) SBMAP currently does not support expressing floating-point literals in scientific notation. In order to include a floating-point literal in the mapping input file, decimal notation must be used instead.

57) The SBDDLUTL commands LOAD RENAMES and LOAD REDEFINES are not fully implemented and should not be used at this time.

58) When replicating PURGEDATA events using the MAPTOPRIPTN option, only the initial PURGEDATA event is mapped to the primary partition. All subsequent PURGEDATAs within a short time are skipped. If there are multiple separate PURGEDATA events issued on the same file within several minutes, the latter ones could be lost when using MAPTOPRIPTN.

59) Comparisons between floating-point data and floating-point literals in SBMAP may result in unexpected results due to internal rounding and casting.

60) Using HASHBUCKETCOUNT and HASHBUCKETID to have multiple consumers replicate into the same unstructured Enscribe file may cause insert events to be dropped or inserted at the wrong RBA.

61) The consumer may abend while processing well-known names for an update or delete for an Enscribe file. If a field in the key is missing, the consumer will log a message and abend. A workaround for this is to use SBMAP to add the well-known name columns into the target.

62) When using ETL Toolkit processing, a TARGETFILE is required when configuring a DBS, otherwise the DBS will fail to be added to the environment. A workaround is to set TARGETFILE to the same value as SOURCEFILE.

63) Transaction support for ETL Toolkit processing when outputting data to IBM MQ is not implemented as of this release.

64) When using SBQFUTL, the START command should either be entered on its own line, or as the last command on the line.

65) When obeying a file with SBQFUTL, lines containing the word “obey” may fail if it isn’t the OBEY command.

66) When obeying a file with SBQFUTL, nested OBEY commands may fail to be parsed correctly.

67) There are certain cases where invalid or incorrect input to the SBUNIDIR, SBBIDIR, SBUNDROS, and SBBIDROS configuration macros may not prevent execution of SB_GEN (such as some TACL errors resulting from incorrect input). Do not rely solely on this error detection to prevent incorrect configurations being generated from bad input; the output of the configuration macros and the generated configuration files should still be reviewed for correctness prior to execution.

68) When using Enscribe Compare, SOLVNV will trap if the report file fills up completely. A workaround is to purge the report file and configure it to be larger using the VVREPORTEXTENT parameter.

69) AUDCOLLN can potentially run into an issue when replicating partitioned files with ALLPARTITIONS ON where it can send events to unexpected consumers. The issue will happen if an ALLPARTITIONS DBS is pointing to a secondary partition of a table while another ALLPARITIONS DBS is pointing to the primary partition of the same table. A workaround is to disable ALLPARTITIONS for the DBS for the secondary partition. This issue does not occur with SBCOLL.

70) SOLVMX is unable to load tables containing BINARY, VARBINARY, or VARCHAR2 columns.

71) The SOLV parameter PARTONLYFNAME cannot be used for Enscribe Compare.

72) AUDMON can potentially hang attempting to autorestart multiple source side consumers in a TCPIP environment if the target environment has been lost. When this issue occurs, AUDCOM will not be able to communicate with AUDMON. To resolve this issue, either recover the target environment so that the consumers can successfully start or manually stop the AUDMON process.

73) When using SBDDLUTL, the ^ character only works as expected with out files for the SAVE, FILE, and NETFORMATFILE commands.

74) SOLV is unable to do a part-only load of a SQL/MP table where the partition first key is a different datatype as the primary key. A workaround for this is to use the SQLWHEREPARAM or SQLWHEREFNAME parameters to select the desired rows.

75) The collector may potentially skip SQL/MP compressed updates where no columns were changed. This can occur if a single consumer has multiple DBSes with the same

source table and different values for the ALLUPDATES parameter. The event should only be skipped for the DBS with ALLUPDATES OFF. When using AUDCOLLN, the event will be skipped for all DBSes in the consumer. When using SBCOLL, the event will be sent to the consumer but may not be used. A workaround for this is to use separate consumer objects for the DBSes with ALLUPDATES ON and OFF.