SOFTWARE RELEASE DOCUMENT (SOFTDOC)

Product:	Shadowbase
Release:	Gravic Version 7.000 HP NonStop Shadowbase: T1122L70 (SB Repl/Guardian)
Release Date:	November 22, 2024
Copyright Notice:	Copyright Gravic, Inc. 1995 – 2024 (www.gravic.com)
File Name:	IPM7000.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 7.000. 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://shadowbasesoftware.com/support/shadowbase-software-product-release-and-support-policies/.

Table of Contents

Disclaimer

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

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://shadowbasesoftware.com/support/shadowbase-software-product-release-and-support-policies/. 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/support/shadowbase-software-product-release-and-support-policies/shadowbase-software-release-glossary/ 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 continued to be maintained for critical issues and tested, but will no longer be enhanced with new features and capabilities.	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

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	NSB 6.410 April, 2018	April, 2020

	consumer should immediately begin planning to move to the native version (AUDCONSN).		
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 exact queue file replication.	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
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 7.000 November, 2024	November, 2026

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.

Upgrade Considerations for Version 7.000 from 6.802

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) The FILTEREMPTYTX default value has been changed.

2) For FUP DCR environments, the collector will fail with an error if all three of the following are true:

   - CONS FUPCOORDINATION is enabled
   - TRACKTX is enabled
   - A QMGR is included

3) The SB_GEN command used when generating Shadowbase configs will now detect errors and be prevented from creating the config until the issue is fixed. SB_GEN ! can be run to skip this error checking.

4) The syntax for running LINKMXCL has changed.

5) If you are upgrading from certain older versions, there are also past upgrade considerations that still apply. Review the following subsections to see if they for the version you are upgrading from. These subjections are organized with the considerations applying for newer versions first. Please note that multiple sections may apply. For example, if you are upgrading from Version 6.310, then upgrade considerations for the current version, for upgrading from a version prior to 6.700, and for upgrading from a version prior to 6.320 would all apply.

Upgrading from a version prior to Version 6.700

1) As of Version 6.700, customers are required to update their license / SHADPASS files when upgrading from a version before 6.700 to version 6.700 or later. Shadowbase will not start if an old SHADPASS is used. Contact your HPE Shadowbase account representative for a renewed license.

2) The message format was modified in Version 6.700 in preparation for future enhancements and is incompatible with prior versions of Shadowbase for NonStop servers. Because of this change, ***you must upgrade all nodes at the same time***.

If you are replicating from NonStop to NonStop, and upgrading to Version 7.000 from a version prior to 6.700, the recommended upgrade procedure is as follows:

Prior to upgrading:

- Install Shadowbase 7.000 on each node involved in replication in a different location from your existing Shadowbase installation.
- Create a new configuration based upon your existing configuration, using the new location for the Shadowbase programs.
- **Do not** modify the locations of the Shadowbase files required for restarting. These files are compatible across versions and will be used to insure no data loss during the upgrade. These include:
    - The Collector's restart file location
    - The Queue Manager's restart file location
    - The Queue Manager's queue file locations
    - The Consumer's TIDFILE, TRANSLOG, TRACKTXFILE, and OVFLQFILE file locations.

Upgrading:

- Select a time window for performing the upgrade, preferably with little or no activity.
- Gracefully shutdown the existing replications threads using the AUDCOM SHUTDOWN command.
- Restart replication using the new configuration files.
- Shadowbase will resume replication from where it was shutdown.

QMGR Note: Version 6.700 and later of the Queue Manager support reading pre-V6.700 queue data and to convert it to the new format prior to sending to the Consumer. This facility allows Shadowbase to be upgraded with data in the queue files without data loss, as long as the queue file location is not changed during the upgrade.

If you would like assistance in planning your upgrade of your HPE NonStop Shadowbase environment, please contact HPE support.

Upgrading from a version prior to Version 6.320

1) As of version 6.320, the behavior of the Queue Manager ABNORMALSHUTDOWN parameter and the default value has been changed. As a result, when upgrading from a pre-6.320 version of Shadowbase in an environment with Queue Manager objects configured, the associated consumer processes will need to have a TIDFILE configured (if not already), or the ABNORMALSHUTDOWN parameters will have to be set to STOP for the Queue Manager objects.

Upgrading from a version prior to Version 6.100

1) If you are upgrading from a release prior to Version 6.100, note that the Shadowbase EMS subsystem owner and version (the EMS SSID token) changed in Version 6.100. The subsystem owner changed from ITI to GRAVIC, and the subsystem version is now composed of VNN, where NN are the two most significant digits of the Shadowbase release version id. As a result, the subsystem version has changed from D42. For any upgrade to Version 6.6nn, the version has thus been updated according to the VNN format to V66. Any EMS filters using these attributes for filtering must likewise be updated. Alternatively, the SBEMSOLDSSID TACL parameter can be enabled to have Shadowbase use the old values for the subsystem owner and version.

Special Notes for Version 7.000

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.

3) 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.

4) 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.

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

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

6) 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.

7) 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.

8) 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.

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

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

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

10) 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.

11) 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 this issue applies to you and you need to discuss options.

12) 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.

13) 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 been closed

and continue on as usual.

When the environment is started up again, 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.

14) 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.

15) This release of Shadowbase includes SBCREATP, a utility for converting SQL/MP schemas into various target SQL formats. 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 7.000 (since Version 6.802)

New Features

1) In the current release, Shadowbase introduces the concept of synchronous replication in a separately-orderable product called HPE Shadowbase Zero Data Loss (ZDL): Shadowbase ZDL is available as a separately licensable product released upon Controlled Availability. The ZDL product adds a new subsystem to the traditional Shadowbase product called the Sync Subsystem, which is comprised of new objects: the Shadowbase Sync Subsystem Monitor (SBSYNCMN), the Shadowbase Coordinator (SBCOOR), and the Shadowbase Metronome (SBMETRO). This subsystem is included in a separate PAK file (SBSYPAK) from the Replication Subsystem (SBPAK), but is installed as part of running the INSTALL script from SBPAK, as normal for HPE Shadowbase, if the associated files have been UNPAK'd from SBSYPAK. See the Installation Instructions included later in this document for further details on the installation process, and the HPE Shadowbase Synchronous Replication Manual included along with this softdoc for more information on the ZDL product and Sync Subsystem as a whole.

   As a result of the addition of the new ZDL product and Sync Subsystem, more informational and diagnostic output has been added to various AUDCOM outputs to accommodate new status and statistic information being tracked in this version of Shadowbase. See additions below:

   Additions to STATS COLL:

   For ZDL, we added a new section to STATS COLL to for coordinator communication info:

```
COORDINATOR MESSAGE INFO (RTC REPLIES):
       SENDS  MIN SIZE  MAX SIZE  AVG SIZE  BUSY
  ----------  --------  --------  --------  ----
         740        40        40        40     0
    BYTES SENT  NUM WRTRDS  WRTRD TIME    AVG RESP
  ------------  ----------  ------------  --------
         29600         740  00:00:00.450     0.608
    TOTAL OPS  MIN OPS  MAX OPS  AVG OPS  STATUS MSG
  -----------  -------  -------  -------  ----------
            8        0        1        1         732
```

   The top row, left to right represents the following:
   - The sum total number of message sends to all coordinators
   - The minimum IPC size of all messages sent
   - The maximum IPC size of all messages sent
   - The average IPC size of all messages sent
   - The number of messages to coordinators from this collector still outstanding and not completed

The second row, left to right represents the following:

- The sum total number of bytes of data sent to all coordinators
- The sum total number of individual WRITEREADs to all coordinators
- The total amount of time spent performing WRITEREADs to all coordinators
- The average coordinator message response time, in milliseconds

The third row, left to right represents the following:

- The total number of operations* in an IPC
- The minimum number of operations* in an IPC
- The maximum number of operations* in an IPC
- The average number of operations* in an IPC
- The number of status messages sent to the coordinators

* "operations" in the above cases implies notifications of transactions that are ready to complete (RTS (ready to sync) notifications)

We added a new section to measure different sync transaction timings:

```
SYNC TRANSACTION TIMINGS              AVG(MS)   MAX(MS)   MIN(MS)     COUNT
---------------------------------- --------- --------- --------- ---------
  ALL DATA READ-RTS RQST ACKED          8.574       580         0    485534
  RTS RQST ACKED-RTS REPLY SENT         0.166        77         0    485534
  LAST EVENT READ-ALL DATA READ      1733.491      2080         0    485534
```

For each timing, the average (AVG(MS)), maximum (MAX(MS)), minimum (MIN(MS)) values in milliseconds are displayed. The meaning of each statistic, top-down, follows:

- **ALL DATA READ – RTS RQST ACKED:**
  The time between it being known that all data has been read for the transaction, and receiving an acknowledgement from the QMGR process that it is ready for the commit of the transaction to be allowed to complete.

- **RTS RQST ACKED – RTS REPLY SENT:**
  The time between the acknowledge from the QMGR process that it is ready for the commit to be allowed to complete, and sending a notification to the coordinator process that it is ready for the transaction to complete.

- **LAST EVENT READ – ALL DATA READ:**
  The time between the last data event for the transaction being read, and all data having been read as being known.

When the DETAIL option is specified, one additional timing is displayed:

```
SYNC TRANSACTION TIMINGS                AVG(MS)   MAX(MS)   MIN(MS)     COUNT
----------------------------------- --------- --------- --------- ---------
  ALL DATA READ-RTS REPLY SENT          3.002       368         0      4929
    ALL DATA READ-RTS RQST ACKED        2.953       368         0      4939
  RTS RQST ACKED-RTS REPLY SENT         0.043        53         0      4929
  LAST EVENT READ-ALL DATA READ         0.024        11         0      4939
```

- **ALL DATA READ – RTS REPLY SENT:**
  The time between it being known that all data has been read for the transaction, and sending a message to the coordinator that it is ready for the commit of the transaction to be allowed to complete.

We also added a new section to measure the audittrail flush read delay, and information on how the collector treated the transactions:

```
AUDITTRAIL READ DELAY                   AVG(MS)   MAX(MS)   MIN(MS)     COUNT
----------------------------------- --------- --------- --------- ---------
  ADT FLUSH WRITTEN-ADT FLUSH READ     13.406       128         1    485534
ZDL STATS DISPLAYED FOR 485534 REPLICATED TXS:
  485534 (100.00%) TREATED SYNC BY COLLECTOR
  0 (0.00%) TREATED ASYNC BY COLLECTOR
```

- **ADT FLUSH WRITTEN-ADT FLUSH READ:**
  The time between the record indicating that all data has been flushed to the audit trail for the transaction was written and when the record was read by the collector. This is the portion of processing time imposed by audit trail reading lag.
- **ZDL STATS DISPLAYED FOR <number> REPLICATED TXS:**
  The total number of replicated transactions reported on in the ZDL statistics from this stats period.
- **TREATED SYNC BY COLLECTOR:**
  The count and percentage of replicated transactions processed synchronously by the collector in the stats period. This indicates that the coordinators(s) were connected during the transaction, the Collector was operating within the configured bounds of SYNCLATENCY, and a notification was sent to the coordinator when the data was made available on the target system and it is ready for the transaction to complete.
- **TREATED ASYNC BY COLLECTOR:**
  The count and percentage of replicated transactions processed asynchronously by the collector in the stats period. These transactions may have occurred when the coordinator was not connected, or while the collector had fallen out of band of the configured allowed SYNCLATENCY.

Information is also reported on certain Sync notifications statistics and related timings for each CPU where a Coordinator was connected and transactions were processed.

```
SYNC NOTIFICATION INFO:
CPU: 00 SYNC NOTIFICATION STATS
PITCHED NOTIFICATIONS
       PITCHED RTS     PITCHED FORGET   PITCHED ROLLBACK
  ---------------   ----------------   ----------------
                 0                  5                  0
SYNC NOTIFICATION TIMINGS             AVG(MS)   MAX(MS)   MIN(MS)     COUNT
----------------------------------- --------- --------- --------- ---------
  RTS REPLY SENT-END EVENT READ         4.131       954         0      7361
  END EVENT READ-RTS REPLY SENT         0.000         0         0         0
  FORGET RECV-COMMIT READ              53.542       938         0        85
  COMMIT READ-FORGET RECV               9.477      1786         0      7271
  ROLLBACK RECV-ABORT READ              0.000         0         0         0
  ABORT READ-ROLLBACK RECV              0.000         0         0         0
```

The top section of PITCHED NOTIFICATIONS statistics, left to right represents the following:

- The number of RTS requests that were ignored due to the transaction no longer being active in the collector's tracking.
- The number of FORGET notifications that were ignored due to the transaction no longer being active in the collector's tracking
- The number of ROLLBACK notifications that were ignored due to the transaction no longer being active in the collector's tracking

The bottom SYNC NOTIFICATION TIMINGS section has information on timings related to sync-related notifications. For each timing, the average (AVG(MS)), maximum (MAX(MS)), minimum (MIN(MS)) values in milliseconds are displayed. The meaning of each statistic, top-down, follows:

- **RTS REPLY SENT-END EVENT READ:**
  The time between it sending the RTS notification for the transaction to the coordinator, and reading an end (commit or abort) event for the transaction from the audit trail. This timing is counted in cases where the RTS notification is sent first, which is true for all transactions processed synchronously, which is generally the case for transactions processed in sync mode.

- **END EVENT READ-RTS REPLY SENT:**
  The time between the end (commit or abort) event being read and the RTS reply being sent. This timing is counted in cases where the end event is read first, which can occur in monitor mode, or if a transaction reverts to asynchronous processing (e.g. if the configured maximum prepare time is exceeded for the transaction).

- **FORGET RECV-COMMIT READ:**

The time between the forget notification being received from the coordinator and the commit event being read from the audit trail. This timing is counted in cases where the forget notification is received first.

- **COMMIT READ-FORGET RECV:**
  The time between the commit event being read from the audit trail and the forget notification being received from the coordinator. This timing is counted in cases where the commit event is read from the audit trail first.

- **ROLLBACK RECV-ABORT READ:**
  The time between the rollback notification being received from the coordinator and the abort event being read from the audit trail. This timing is counted in cases where the rolback notification is received first.

- **ABORT READ-ROLLBACK RECV:**
  The time between the abort event being read from the audit trail and the rollback notification being received from the coordinator. This timing is counted in cases where the abort event is read from the audit trail first.

Additions to DUMPTX/INFOTX COLL:

For ZDL, an additional section has been added to the output of DUMPTX/INFOTX COLL for information related to synchronous replication:

```
SYNC INFO:
  AUDIT FLUSH TS:      03/15/2024 14:55:34.267314
  SAFESTORE TS:        03/15/2024 14:55:34.269679
  COOR NOTIF SENT TS:  03/15/2024 14:55:34.269687
  FORGET RECEIVED FROM COOR
```

This section includes the following:

- The timestamp of the audit flush event, or a message that it has not been received yet.
- The timestamp of when the QMGR safestored the transaction, or a message that it has not been safestored yet.
- The timestamp of when the notification was sent to the coordinator, or a message that it has not been sent yet.
- If a forget or rollback message has been received from the coordinator.

2) The SBCREATP utility has a new parameter to add a qualifier that is prepended to the table name.

3) Consumer user exits are now able to retrieve values previously “put” using an SBPUT function or SBMAP using the new functions SBGETCURRCOLUMN, SBGETCURRRECORD, SBGETCURRRELATIVEADDR, and

SBGETCURRRELATIVEADDR64. See the New and Modified User Exit Functions for more information.

4) SBCOLL now supports automatically excluding Enscribe alternate key files from replication using the collector parameter SKIPALTFILEEVENTS.

5) The consumer parameter FILTEREMPTYTX is now set to ON by default. This will remove transactions that do no contain any change-data statements in them (for example, it will remove transactions that just consist of BEGIN TX and COMMIT TX).

6) The SB_LIST output for CONSUE objects is now clearer that it is for CONSUE objects instead of regular consumers.

7) SBMAP will now log the DBS name when a required SOURCEREC, SOURCEDICT, TARGETREC, or TARGETDICT parameter is missing.

8) Consumer user exits are now able to fan out a single Enscribe event into multiple events on the same target when replicating out to HPE Shadowbase for Other Servers using the new functions SBPROCESSAGAIN and SBPROCESSITER. This new functionality is useful, for example, when the Enscribe record contains an array of items that needs to be normalized into multiple target rows. See the New and Modified User Exit Functions for more information.

9) Added a new Enscribe Compare parameter, VVITERSOURCEREFETCH, to control whether or not the source records will be re-fetched on final iteration.

10) SOLVMGR now supports using a different config section for the first SOLV run. This can be used to specify different parameters or parameter values for the initial file only, such as to start from a specific position in the first file.

11) Added a new standalone program, SBBFCMP, that compares entry-sequenced Enscribe files. It now supports the comparing of partitioned entry-sequenced Enscribe files. Also, a new COMPAREBATCHSIZE parameter was added. The previous version of SBBFCMP had a 50 million combine total record limitation due to memory constraints. The COMPAREBATCHSIZE parameter will now allow SBBFCMP to compare the source and target files in batches of COMPAREBATCHSIZE. Thus, allowing for an infinite number of records in each file. Setting this parameter to something like one million will cause SBBFCMP to read one million records from each file at a time before doing the compare in batches. Running SBBFCMP without the COMPAREBATCHSIZE parameter enabled will produce a count of duplicated records in each file during the final result reporting. With COMPAREBATCHSIZE enable, the count of duplicate records will no longer be available in the final results. See the New Features Description section on SBBFCMP for Entry-Sequenced Enscribe Files for more details.

12) Added a new standalone program, SBQFUTL, that reads the QMGR queue files. It can output the contents to the terminal or to a file. There are several summary reports available or the individual headers or data can be output. The headers and event data can be filtered and sorted in a variety of ways. See the New Features Description section on Queue File Utility (SBQFUTL) for more details.

13) Added new standalone utilities for running wildcarded Compare jobs. SBCMPWLD can be used for Enscribe and SQL/MP files and tables, and SBCMXWLD can be used for SQL/MX tables. See the New Features Description section on Wildcard Compare Utilities for more details.

14) The ENSINST script has been removed as it is no longer needed (Shadowbase can internally use SQL and run on all systems regardless if they have an SQL license or not). The INSTALL script will now skip the SQLCOMP step if no catalog is specified when prompted.

15) The collector will now log a message and abend if the TACL parameter SBCOLLMULTIENDS (used with multi-ported DOC Writers) and the consumer parameter FILTEREMPTYTX are both enabled. These parameters are not logically compatible.

16) If the INSTALL script creates a catalog and the OWNER option was specified, the catalog will be given to OWNER.

17) Added a new SOLVMGR parameter, SOLVCONFIGSECTIONFIRST, that allows for a different section to be used for the first SOLV run from the rest of the SOLV runs.

18) Additional latency statistics have been added to the STATS output for the collector, queue manager, and consumer. The latency section for STATS COLL now looks like this:

```
LATENCY WARNING STATUS: DISABLED
  ETS: 08-23 00:49:44.925  DIFF (CUR): 14:41:44.032  (MAX): 15:31:20.804
  LAST EVENT TS (LTS): 08-23 00:49:44.925  READ AT TS: 08-23 15:31:28.956
                  READING LAG  (CUR): 14:41:44.031  (MAX): 15:31:20.804
  IDLE TIME (CUR): 00:00:00.000  (MAX): 00:00:00.000
  COLL EVENT PROCESSING TIME: 000:00:00.000009
```

The added information is as follows:

- READ AT TS: The timestamp at which the last read event was read
- IDLE TIME: If the collector is currently at EOF, how long since it last hit EOF
- COLL EVENT PROCESSING TIME: How long it took for the collector to process the most recent event it read

The latency section for STATS QMGR now looks like this:

```
LATENCY WARNING STATUS: DISABLED
  LAST RECV TS (LTS): 08-23 15:59:55.610  RECV AT TS: 08-23 16:00:02.559
      RECEIVING LAG (CUR): CAUGHT UP    (MAX): CAUGHT UP
          RECV IDLE (CUR): 00:00:07.388 (MAX): 00:00:09.614
  LAST SENT TS (LTS): 08-23 15:59:55.610  SENT AT TS: 08-23 16:00:02.559
        SENDING LAG (CUR): CAUGHT UP    (MAX): 00:00:06.950
          SENT IDLE (CUR): 00:00:17.162 (MAX): 00:00:17.162
  CONS ETS: 08-23 15:59:55.610  DIFF(CUR): CAUGHT UP    (MAX): 00:00:00.000
  QMGR IPM PROCESSING TIME: 000:00:00.000280
```

The added information is as follows:

- RECV AT TS: When the queue manager received the ipm containing the last received event.
- RECEIVING LAG: The difference between when the last received event occurred and when the queue manager received it
- RECV IDLE: If the queue manager is not receiving events from the collector, for how long has it been receiving pings from the collector.
- SENT AT TS: When the queue manager last sent an event to the consumer.
- SENDING LAG: The difference between when the last sent event occurred and when the queue manager sent it.
- SENT IDLE: If the queue manager is not sending events to the queue manager, for how long has it been sending pings to the consumer.
- QMGR IPM PROCESSING TIME: Time it took for the most recently sent data ipm to travel through the QMGR. This can include time spent processing other data ipms and time the ipm was in a queue file.

The latency section for STATS CONS now looks like this:

```
LATENCY WARNING STATUS: DISABLED
AUDITTRAIL PROCESSING SUMMARY:
       SEQ              RBA
  --------    -------------
  AA000118    0004911494996
  LAST EVENT TS (LTS): 08-23 15:59:57.256  RECV AT TS: 08-23 16:00:02.561
      RECEIVING LAG(CUR): CAUGHT UP     (MAX): 00:00:05.304
  IDLE TIME (CUR): 00:00:45.866  (MAX): 00:00:45.866
  CONS EVENT PROCESSING TIME: 000:00:00.005424
```

The added information is as follows:

- RECV AT TS: When the consumer received the last event received
- RECEIVING LAG: The difference between when the most recently received event occurred and when the consumer received it.
- IDLE TIME: If the consumer isn’t receiving events, how long it has been receiving pings from the collector or queue manager.
- CONS EVENT PROCESSING TIME: How long it took for the consumer to process the last event.

19) The LINKMXCL script has been enhanced to perform additional validation, prompt for configuration information, and log additional information on the steps and results. Running the LINKMXCL script will prompt for a name to use for the generated Consumer program file with the SQLMX client user exit linked in. Security settings and the SQL/MP catalog for the existing AUDCONSN program file will be used for the newly generated one. The new syntax for running LINKMXCL is as follows:

```
RUN LINKMXCL [<program name>]
```

If no <program name> is specified, the script will prompt for one.

20) The following enhancements have been made to the Shadowbase configuration macros (SBUNIDIR, SBBIDIR, SBUNDROS, and SBBIDROS).

a. The SB_GEN command was enhanced to perform additional error checking, and to halt configuration generation when an error is detected. It will also provide further information on existing files matching the prefix when cleaning up existing files prior to generating new versions.
b. In conjunction with the new SB_GEN command behavior, an '!' option was also added in conjunction with this to allow clearing of the error flag and to continue with attempting the SB_GEN operation. The syntax to specify this option is as follows:

```
SB_GEN !
```

c. Additional validation has been added, including validation of SBMON_PARAM options, and that a TGTNODE has been added in configurations where one is required.
d. Generated files now include information on their creation and creator, including the generating configuration filename and its owner, creation time, last modified time, and EOF value.
e. A new option SB_PASSWORD_PROMPTING for SBUNDROS and SBBIDROS will allow users to specify a hyphen (-) character instead of the SQL/MX target user's password in the configuration script. When the Shadowbase configuration is started, it will prompt the user for the real password which it will use when creating the target TRS/DW objects. This will ensure that the password is never stored nor displayed in clear text anywhere in the Shadowbase configuration. In order to enable this setting, use the following syntax:

```
SB_SET SB_PASSWORD_PROMPTING 1
```

21) An example file for supported SBCOOR and SBMETRO measure counters, MEASZDL, has been added.

22) A new feature called the ETL Toolkit has been added to the Shadowbase consumer. For more information about this feature, refer to the ETL Toolkit section in this softdoc.

23) The DUMPTX/INFOTX QMGR command now displays the current state of safestoring the transaction. The possible states are:

- **QUEUED** - The transaction is queued to be safestored
- **IN PROCESS** - The transaction is currently being safestored
- **STORED** - The transaction was just safestored

Problems Fixed

1) If the system time is changed use the SETTIME TACL command, AUDCOM could show incorrect times in STATS and STATUS output. The prior workaround for this to set the SBJTS-BYPASS TACL parameter to 1 is no longer needed.

2) If an SBMAP map block contains any assignments to a key column, the where clause for any updates or deletes would only contain the columns that were assigned to in the map block.

3) When using the MPLOG consumer parameter with MP DCR, it was possible for the consumer to trap when first trying to write to an MPLOG file that was empty.

4) Use of UPDATEDUPLICATE ON when replicating an Enscribe file via the alternate key through the use of the KEYSPECIFIER parameter without BASE24HANDLERID processing was not supported and would cause the Consumer to fail.

5) The ESTATS no longer requires authorization to run like the other stats commands.

6) The CHECK DBS will now properly warn about the ROOTPARTITION parameter not being set when it is required.

7) The user is now prevented from configuring FUPCOORDINATION in the same configuration as TRACKTX and a QMGR.

8) Altering FUPERRORACTION or MPERRORACTION should no longer occasionally fail.

9) The consumer will no longer ignore the EMSCOLLECTOR parameter in some scenarios.

10) Enscribe Compare will no longer log “UNDETERMINED TERMINATION” when mismatches are resolved during iterations while VVMAXSESSIONMISMATCHES is enabled.

11) Fixed an issue preventing the SHADTYPE file from being used in some cases.

12) Fixed a race condition between SUSPENDUPD and RESUMEUPD that could result in a consumer remaining in the SUSPENDUPD state until another RESUMEUPD command was given.

13) The SBDDLUTL utility now supports using an asterisk at the start of a line to denote a comment line.

14) The SBDDLUTL utility now correctly adds double quotes around identifiers in converted PostgreSQL output.

15) The SBDDLUTL utility no longer omits double quotes from VALUE IS text when the source schema has them.

16) Fixed an issue that could cause SBGETAFTERCOLUMN or SBGETBEFORECOLUMN to return the wrong value when using a CONSUE object and replicating from a source table containing VARCHAR2 or VARBINARY columns.

17) Fixed an issue that could cause SBGETAFTERCOLUMN or SBGETBEFORECOLUMN to return -4 for a field in a short Enscribe instead of -1.

18) An SBMAP map block can now use the SBHASH function for source tables that contain a varchar in their key.

19) The SBMAP function SUBSTRING’s length argument is now optional. If omitted, the substring will go until the end of the input string.

20) SBMXSERV now uses the correct consumer and DBS parameters for I/O and retry timing. The consumer parameter SQLLOCKWAIT is used to determine the timeout for SQL/MX locking errors. The parameter RETRYDELAY is now used to determine how long to wait before attempting another insert through SBMXSERV.

21) AUDCOM will no longer count having to use the SHADTYPE file as an error.

22) Enscribe Compare will no longer fail if the source and target file formats do not match.

23) The USRX function SBPUTCOLUMN will now always work when replicating a short Enscribe record to an open environment through an SQL/MP dummy table.

24) When calling the USRX function SBPUTCOLUMN on an Enscribe TYPE SQL VARCHAR field with length argument less than the maximum, the unused bytes will be filled with spaces instead of junk data.

25) Performing partition maintenance operations dynamically for DBSes using the ALLPARTITIONS ON parameter while replication is running could result in added, moved, or split partitions not being replicated. A race condition between events appearing for the new partition in the audit trail and when the new partition information would be returned from file system calls could cause the partition not to be in the partition list retrieved, and thus be excluded.

26) The SOLVMGR STARTFNAME parameter would not be used when specified. The behavior of the parameter is now fixed as intended.

27) Fixed an issue causing AUDMON to log the SHADTYPE warning message multiple times instead of once.

28) Fixed an issue causing the sub-second portion of parameters that take a date and time as a value to be parsed incorrectly.

29) Fixed an issue where the consumer could sometimes skip begins when FILTEREMPTYTX was ON even when the rest of the transaction was replicated.

30) SBCOLL is now able to replicate events containing SQL/MX DATE type2 columns.

31) The DBS parameter FIXUPDECIMAL can now be set to ABEND as expected.

32) Fixed some issues preventing OBEYFORM output from being used.

33) The SBDDLUTL utility no longer ignores the KEEP option of NONPRIMARYKEYFILLER and PRIMARYKEYFILLER when converting to SQL targets.

34) The SBDDLUTL utility will no longer abend when it handles specific syntax errors related to the /OUT <file>/ option.

35) The SBDDLUTL utility will now handle the '!', '+', and '^' options of the FILE and NETFORMATFILE commands correctly.

36) SBFUP's created FUP process will no longer start with the wrong default volume if the DEFAULTVOLUME parameter is shorter than SBFUP's volume.

37) The USRX functions SBGETSOURCE and SBGETTARGET will no longer cause the consumer to fail if called while processing a begin, commit, or abort event.

38) Enscribe Compare will no longer sometimes fail after multiple runs using the same marker file record.

39) AUDCOM will no longer upshift the KEYSPECIFIER parameter.

40) The fractional portion of the SOLVMGR parameters STARTIME and STOPTIME and the collector parameters ADTSTARTTIME and ADTSTOPTIME are now parsed correctly. The parameter documentation has been updated to show to proper way to specify fractional sections now (without a period between the millisecond and microsecond portions).

41) SOLVUTIL will no longer go into a debug prompt when it fails due to a SOLVMKR file pointing to a source file that no longer exits.

42) The USRX function SBPUTRELATIVEADDR will now correctly return an error -2 when called for an entry-sequenced format 1 file as documented.

43) The configuration macros, SBUNIDIR, SBBIDIR, SBUNDROS, and SBBIDROS, will no longer upshift the values of parameter overrides for types where case sensitive parameters exist.

44) Fixed an issue causing the QMGR to rarely restart from the wrong position under specific circumstances.

45) Fixed an issue causing AUDCOLLN to rarely skip events that have the same ADT position as already replicated events in specific circumstances on a restart when replicating with a QMGR.

New Features Description

SBBFCMP for Entry-Sequence Enscribe Files

SBBFCMP has released with the 7.000 version of HPE Shadowbase. Use it to compare and repair Enscribe Entry-Sequenced files. Either, both, or neither of the files to be compared may be audited.

SBBFCMP also has the ability to list file details about each file and all the existing processes that have the files opened. Running SBBFCMP does not require any of the other existing executables that are licensed and distributed with HPE Shadowbase Compare for Enscribe.

The primary functionality of SBBFCMP is to read the contents of two entry-sequenced files and generate a report listing any mismatches. Both entry-sequenced files must have the same record length for the compare to work. During a compare, SBBFCMP ignores the order of records in the file and any zero length records along with any duplicate records. A summary of the compare results is normally generated when SBBFCMP

completes. The record addresses and contents of the duplicate records can be added to the resulting report. Similarly, the record addresses and contents of any mismatches can also be included with the report output.

SBBFCMP uses hash functions to compress the contents of each record that it reads and compares. As a result, there is an extremely slim possibility of a hash collision. A hash collision is a random match in hash values that occurs when a hashing algorithm produces the same hash value for two distinct records. SBBFCMP can generate false duplicates and matches due to these collisions. A future version of SBBFCMP will try to further minimize the probability of false results that can occur from these hash collisions.

The repair program option of SBBFCMP allows a user to add missing records to the target file from the source file. Optionally, the missing records can be added to a third file that is also known as a repair file. This repair file will give the user a preview of the repair operation that would normally occur to the target file. The repair file must be pre-created with the same record length as the source file before running the repair option. Otherwise, a file not in directory (file system error 11) will occur on the repair file. The file must also be sized appropriately to hold all the records that will be written to the repair file or this will result in a file is full error (file system error 45).

SBBFCMP can be configured using TACL parameters or a SHADPARM file. The latter is the preferred method. Refer to the *HPE NonStop Shadowbase SOLV Manual* for further instructions on the method of how to set parameters. The particular parameters related to SBBFCMP are described in this manual. Note that TACL parameters override those in the SHADPARM file and specific section parameters override those in the general section. By default, SBBFCMP will look for parameters in the SBBFCMP section in the SHADPARM file located in the default subvolume.

To run SBBFCMP, a new 6.700 (or newer) version of the HPE Shadowbase license with the proper credentials is required. SBBFCMP will not run with previous versions of the HPE Shadowbase license file. The SHADPASS license file should be in the same location where the executable resides. If not, use the LICENSEFILENAME parameter to specify where to find the SHADPASS file.

This is the syntax to run SBBFCMP.

```
syntax: sbbfcmp [-f<shadparm-fname>]        (default SHADPARM in def subvol)
                [-s<shadparm-section-name>] (default SBBFCMP)
```

Only the SOURCEFILENAME and TARGETFILENAME parameters are required to run SBBFCMP to do a basic compare.

In order to run a repair, also include the PROGRAMOPTION variable and set it to REPAIRMISSINGTGTRECS.

Refer to the SBBFCMP Parameters Section in this manual for other program options and settings. Multiple sample configurations are also provided in this manual.

Running the program without any parameters will generate the following help text.

```
SBBFCMP - T1122L70 - V7000(L) - (04Jun24), COPYRIGHT GRAVIC, INC.
1995-2024. PORTIONS COPYRIGHTED BY AND LICENSED FROM THIRD PARTIES. SEE
README FILE. USAGE SUBJECT TO THE TERMS OF A WRITTEN LICENSE AGREEMENT.
PATENTS: WWW.GRAVIC.COM/GRAVICLABS/PATENTS/PRODUCTS

Started 2024-08-07:15:20:09.298110 on \VIV1(078466/254)

Shadparm file '\VIV1.$GPJ.SBBFCMP.SHADPARM ' does not exist

UNUSED PARAM(S): -NONE-

syntax: sbbfcmp [-f<shadparm-fname>]        (default SHADPARM in def subvol)
                [-s<shadparm-section-name>] (default SBBFCMP)
                [-h                         (help)

  Blistering Fast Compare Params:
    COMPAREBATCHSIZE   <num>
    LICENSEFILENAME    <fname>  (default SHADPASS in def subvol)
    LOGBYTESPERREC     <num>    (def  0=Hdr only)
                                     -1=full rec
                                     >0=record bytes
    LOGDUPS            <log>    (def NONE)
    LOGLINEWIDTH       <num>    (def  80 )
    LOGMISMATCH        <log>    (def NONE)
    LOGREADINTERVAL    <num>    (def   0 )   Report progress. Batch uses size
    NONPRINTCHAR       <char>   (def '.' )
    PROGRAMOPTION      <opt>    (def COMPARE)
    READSPERTHROTTLE   <num>    (def   1)
    READTHROTTLEDELAY  <num>    (def    0)   0.01 seconds
    REPAIRFILENAME     <fname>  (def none)   optional for REPAIRMISSINGTGTRECS
    REPAIRIGNOREREC0   <on/off> (def OFF)
   *SOURCEFILENAME     <fname>
    SOURCESTOPADDRESS  <addr>
   *TARGETFILENAME     <fname>
    TARGETSTOPADDRESS  <addr>
    VERBOSE            <on/off> (def OFF)
    WRITESPERTHROTTLE  <num>    (def   1)
    WRITETHROTTLEDELAY <num>    (def   0)     0.01 seconds
      where <opt>  = { COMPARE | GETINFO | LISTOPENERS | REPAIRMISSINGTGTRECS }
            <log>  = { NONE | SOURCE | TARGET | BOTH }
  TACL environment params override shadparm settings (* = required)

70>
```

Sample Configurations and Results

Compare

First, edit a shadparm file and add the following commands before running SBBFCMP to do a compare,

```
>fup copy shadparm

[compare_es]
sourcefilename=$GPJ.SBBFCMP.ES1
```

```
targetfilename=$GPJ.SBBFCMP.ES2
```

Now run SBBFCMP supplying the section name, compare_es, as entered above in the shadparm file. This is a sample of the results from running the compare.

```
42> v7000tst.sbbfcmp -scompare_es
SBBFCMP - T1122L70 - V7000(L) - (31Aug24), COPYRIGHT GRAVIC, INC.
1995-2024. PORTIONS COPYRIGHTED BY AND LICENSED FROM THIRD PARTIES. SEE
README FILE. USAGE SUBJECT TO THE TERMS OF A WRITTEN LICENSE AGREEMENT.
PATENTS: WWW.GRAVIC.COM/GRAVICLABS/PATENTS/PRODUCTS

Started 2024-09-18:13:05:04.016807 on \VIV1(078466/254)

Section 'compare_es' in '\VIV1.$GPJ.SBBFCMP.SHADPARM' does exist
UNUSED PARAM(S): -NONE-

SOURCEFILENAME \VIV1.$GPJ.SBBFCMP.ES1
TARGETFILENAME \VIV1.$GPJ.SBBFCMP.ES2

--File Info At Start-- ----------Source---------- ----------Target----------
File Type/Subtype               Ens file/EntrySeq          Ens file/EntrySeq
Format/BlockLen/RecLen                2/4096/4048                2/4096/4048
Audited/Auditcompress                      No/No                      No/No
Last Modified          2022-01-07:12:06:54.849846 2022-01-07:12:06:56.433032
Aggregate EOF                        386,891,776                444,379,136
Last Rec Address(FMT2)            405681135943680            465961001943040
                (hex2)            0x170f700000000            0x1a7ca00000000
                (FMT1)                  386887680                  444375040
                (hex1)                0x170f7000                 0x1a7ca000


----------Source----------                      ----------Target----------
                  188,951  records                                  217,024
              271,171,544  bytes                                311,465,404
       000-00:00:02.780750  time reading               000-00:00:03.129513
                   67,949  records/sec                               69,347
                97,517,412  bytes/sec                            99,525,198
           405681135943680  last       (fmt2)              465961001943040
           0x170f700000000   record    (hex2)              0x1a7ca00000000
                 386887680    compared (fmt1)                    444375040
                0x170f7000             (hex1)                   0x1a7ca000


                         0  zero len (ignored)                          0
                         0  dups (ignored)                         28,073
                   188,951  matches                                188,951
                         0  mismatches                                  0


Completed 2024-09-18:13:05:10.269496           Duration 000-00:00:06.252689

43>
```

Repair

Use the following lines in a SHADPARM file to configure SBBFCMP to generate repair records into a repair file. Note that because the REPAIRFILENAME variable is supplied, the repair file will need to be pre-created with the entry-sequenced file type and correct record length. The file must also be sized appropriately to hold all of the records that will be inserted during the repair operation.

```
>fup copy shadparm
[repair_es]
PROGRAMOPTION=REPAIRMISSINGTGTRECS
sourcefilename=$GPJ.SBBFCMP.ES1
targetfilename=$GPJ.SBBFCMP.ES2
repairfilename=repaires
```

Before running SBBFCMP a file named ***repaires*** will need to be created. The new file must be entry-sequenced with a record length the same as the source and target files. SBBFCMP can now be started with the section name, repair_es, as entered above in the shadparm file. This is a sample of the results from running the compare.

```
43> v7000tst.sbbfcmp -srepair_es
SBBFCMP - T1122L70 - V7000(L) - (31Aug24), COPYRIGHT GRAVIC, INC.
1995-2024. PORTIONS COPYRIGHTED BY AND LICENSED FROM THIRD PARTIES. SEE
README FILE. USAGE SUBJECT TO THE TERMS OF A WRITTEN LICENSE AGREEMENT.
PATENTS: WWW.GRAVIC.COM/GRAVICLABS/PATENTS/PRODUCTS

Started 2024-09-18:13:08:50.249547 on \VIV1(078466/254)

Section 'repair_es' in '\VIV1.$GPJ.SBBFCMP.SHADPARM' does exist
UNUSED PARAM(S): -NONE-

SOURCEFILENAME \VIV1.$GPJ.SBBFCMP.ES1
TARGETFILENAME \VIV1.$GPJ.SBBFCMP.ES2

--File Info At Start-- ----------Source---------- ----------Target----------
File Type/Subtype               Ens file/EntrySeq          Ens file/EntrySeq
Format/BlockLen/RecLen               2/4096/4048                2/4096/4048
Audited/Auditcompress                      No/No                      No/No
Last Modified          2022-01-07:12:06:54.849846 2022-01-07:12:06:56.433032
Aggregate EOF                     386,891,776                444,379,136
Last Rec Address(FMT2)          405681135943680            465961001943040
                (hex2)          0x170f700000000            0x1a7ca00000000
                (FMT1)                386887680                 444375040
                (hex1)               0x170f7000                0x1a7ca000

REPAIRFILENAME         \VIV1.$GPJ.SBBFCMP.REPAIRES
File Type/Subtype               Ens file/EntrySeq
Format/BlockLen/RecLen               2/4096/4048
Audited/Auditcompress                      No/No
Last Modified          2024-08-05:15:23:39.729877
Aggregate EOF                                 0
Last Rec Address(64)                          -
Last Rec Address(hex)                         -


----------Source----------                     ----------Target----------
```

```
                  188,951  records                                  217,024
              271,171,544  bytes                                311,465,404
      000-00:00:02.716009  time reading               000-00:00:03.077180
                   69,569  records/sec                           70,526
               99,841,916  bytes/sec                         101,217,804
          405681135943680  last       (fmt2)              465961001943040
          0x170f700000000   record    (hex2)              0x1a7ca00000000
                386887680    compared (fmt1)                   444375040
               0x170f7000              (hex1)                  0x1a7ca000


                       0  zero len (ignored)                          0
                       0  dups (ignored)                         28,073
                 188,951  matches                                188,951
                       0  mismatches                                  0

0 record(s) written(repaired) to \VIV1.$GPJ.SBBFCMP.REPAIRES


Completed 2024-09-18:13:08:56.487855           Duration 000-00:00:06.238308

44>
```

Getinfo

SBBFCMP can get the detailed information about the two files to be compared. Use this configuration to run the GETINFO option.

```
[getinfo_es]
programoption=getinfo
sourcefilename=$GPJ.SBBFCMP.ES1
targetfilename=$GPJ.SBBFCMP.ES2
```

These are the result from running with the GETINFO option.

```
44> v7000tst.sbbfcmp -sgetinfo_es
SBBFCMP - T1122L70 - V7000(L) - (31Aug24), COPYRIGHT GRAVIC, INC.
1995-2024. PORTIONS COPYRIGHTED BY AND LICENSED FROM THIRD PARTIES. SEE
README FILE. USAGE SUBJECT TO THE TERMS OF A WRITTEN LICENSE AGREEMENT.
PATENTS: WWW.GRAVIC.COM/GRAVICLABS/PATENTS/PRODUCTS

Started 2024-09-18:13:11:24.731149 on \VIV1(078466/254)

Section 'getinfo_es' in '\VIV1.$GPJ.SBBFCMP.SHADPARM' does exist
UNUSED PARAM(S): -NONE-

SOURCEFILENAME \VIV1.$GPJ.SBBFCMP.ES1
TARGETFILENAME \VIV1.$GPJ.SBBFCMP.ES2

--File Info At Start-- ----------Source---------- ----------Target----------
File Type/Subtype               Ens file/EntrySeq          Ens file/EntrySeq
Format/BlockLen/RecLen                2/4096/4048                2/4096/4048
Audited/Auditcompress                      No/No                      No/No
Last Modified          2022-01-07:12:06:54.849846 2022-01-07:12:06:56.433032
```

```
Aggregate EOF                          386,891,776               444,379,136
Last Rec Address(FMT2)            405681135943680            465961001943040
                (hex2)            0x170f700000000            0x1a7ca00000000
                (FMT1)                  386887680                  444375040
                (hex1)                 0x170f7000                 0x1a7ca000


Completed 2024-09-18:13:11:24.795977            Duration 000-00:00:00.064828

45>
```

Listopeners

Use SBBFCMP to get the information about which processes or applications have the two files opened. LISOPENERS will display the cpu and pin or the process name of the program along with the program name, open access, exclusion and the user id. Use this configuration to run the LISTOPENERS option.

```
[listopeners_es]
programoption=listopeners
sourcefilename=$GPJ.SBBFCMP.ES3
targetfilename=$GPJ.SBBFCMP.ES4
```

These are the result from running with the LISTOPENERS option.

```
8> v7000tst.sbbfcmp -slistopeners_es
SBBFCMP - T1122L70 - V7000(L) - (31Aug24), COPYRIGHT GRAVIC, INC.
1995-2024. PORTIONS COPYRIGHTED BY AND LICENSED FROM THIRD PARTIES. SEE
README FILE. USAGE SUBJECT TO THE TERMS OF A WRITTEN LICENSE AGREEMENT.
PATENTS: WWW.GRAVIC.COM/GRAVICLABS/PATENTS/PRODUCTS

Started 2024-09-18:13:52:11.201063 on \VIV1(078466/254)

Section 'listopeners_es' in '\VIV1.$GPJ.SBBFCMP.SHADPARM' does exist
UNUSED PARAM(S): -NONE-

SOURCEFILENAME \VIV1.$GPJ.SBBFCMP.ES3
TARGETFILENAME \VIV1.$GPJ.SBBFCMP.ES4

--File Info At Start-- ----------Source---------- ----------Target----------
File Type/Subtype               Ens file/EntrySeq          Ens file/EntrySeq
Format/BlockLen/RecLen                2/4096/4048                2/4096/4048
Audited/Auditcompress                      No/No                      No/No
Last Modified          2022-08-16:15:21:43.763044 2022-08-16:15:21:43.766926
Aggregate EOF                             106,496                    106,496
Last Rec Address(FMT2)               107374182400               107374182400
                (hex2)               0x1900000000
                (FMT1)                     102400                     102400
                (hex1)                    0x19000


\VIV1.$GPJ.SBBFCMP.ES3 Openers
\VIV1.1,1059 \VIV1.$GPJ.SBBFCMP.OPEN RW-S  255,008
```

```
\VIV1.$GPJ.SBBFCMP.ES4 Openers
none

Completed 2024-09-18:13:52:11.342829            Duration 000-00:00:00.141766

9>
```

In the results above, listopeners will display the process id or name, program name, the open access, exclusion and the user id of the openers.

Logdups

LOGDUPS will display all the detected duplicates in the same file regardless of how many times the same record occurs. The first occurrence of a duplicate record is not displayed. This option is only valid when COMPAREBATCHSIZE is also disabled. Duplicate records in the results are identified with a 'D'. By default, only the header of each record is displayed which include the filename, record address in decimal and hexadecimal. The record length is also included. Use the LOGBYTESPERREC parameter to have the display include the full or partial contents of each record.

Use this configuration to run with the LOGDUPS option.

```
[compare_es_1m]
sourcefilename=$GPJ.SBBFCMP.ES1m
targetfilename=$GPJ.SBBFCMP.ES1mfmt1
logdups=both
```

These are the result from running with the LOGDUPS option.

```
SBBFCMP - T1122L70 - V7000(L) - (31Aug24), COPYRIGHT GRAVIC, INC.
1995-2024. PORTIONS COPYRIGHTED BY AND LICENSED FROM THIRD PARTIES. SEE
README FILE. USAGE SUBJECT TO THE TERMS OF A WRITTEN LICENSE AGREEMENT.
PATENTS: WWW.GRAVIC.COM/GRAVICLABS/PATENTS/PRODUCTS

Started 2024-09-16:16:53:03.968730 on \VIV1(078466/254)

Section 'compare_es_1m' in '\VIV1.$GPJ.V7000TST.SHADPARM' does exist
UNUSED PARAM(S): -NONE-

SOURCEFILENAME \VIV1.$GPJ.SBBFCMP.ES1M
TARGETFILENAME \VIV1.$GPJ.SBBFCMP.ES1MFMT1

--File Info At Start-- ----------Source---------- ----------Target----------
File Type/Subtype               Ens file/EntrySeq          Ens file/EntrySeq
Format/BlockLen/RecLen                2/32768/10                  1/4096/10
Audited/Auditcompress                      No/No                       No/No
Last Modified          2023-11-13:19:02:52.037077 2023-11-15:09:18:20.733852
Aggregate EOF                         14,024,704                12,087,296
Last Rec Address(FMT2)              1833951037594             12670153523251
                (hex2)              0x1ab0000089a              0xb8600000033
                (FMT1)                         -                  12083251
                (hex1)                         -                  0xb86033
```

```
----------Source----------                        ----------Target----------
                1,000,102  records                                1,000,102
               10,001,020  bytes                                 10,001,020
      000-00:00:00.906502  time reading                 000-00:00:00.910650
                1,103,254  records/sec                           1,098,228
               11,032,540  bytes/sec                            10,982,287
            1833951037594  last       (fmt2)                12670153523251
            0x1ab0000089a   record    (hex2)                 0xb8600000033
                        -    compared (fmt1)                      12083251
                        -             (hex1)                       0xb86033


                        0  zero len (ignored)                           0
                        1  dups (ignored)                               1
                1,000,101  matches                              1,000,101
                        0  mismatches                                   0

D \VIV1.$GPJ.SBBFCMP.ES1M F2:1833951037594(0x1ab0000089a) Len:10
D \VIV1.$GPJ.SBBFCMP.ES1MFMT1 F1:12083251(0xb86033) Len:10

Completed 2024-09-16:16:53:08.568440            Duration 000-00:00:04.599710
```

Note that the F1 and F2 addresses above relate to format 1 versus format 2 files and are based on the corresponding file of the record. The addresses are also printed in decimal and hex representation.

Enabling LOGDUPS has no affect when COMPAREBATCHSIZE is also enable because duplicates cannot be identified when comparing in batches. SBBFCMP will log the following warning in such a configuration.

```
Warning: LOGDUPS is enabled with COMPAREBATCHSIZE enabled, ignoring LOGDUPS
```

SBBFCMP Parameters

COMPAREBATCHSIZE { <number> }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		0	0	$(2^{63})-1$	NO

If greater than zero, SBBFCMP will compare the source and target files in batches of COMPAREBATCHSIZE records. By default, the whole files will be compared at once. If comparing in batches, the count of duplicate records will no longer be available in the final results. In general, COMPAREBATCHSIZE can be any number. It should be set lower than 27 million due to the SBBFCMP limitations discusses in the limitations section of this manual. One million should be a practical number to use for this value.

LICENSEFILENAME { <license-file-name> }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		SHADPASS	-	-	NO

This is the filename of the license file. The SHADPASS file in the #default subvolume is the default.

LOGBYTESPERREC { <number> }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		0	-1	$(2^{31})-1$	NO

This is use when the LOGNONMATCH or LOGDUP parameter is enabled. This is the number of bytes to log to the report output from the record. The default is zero, meaning that only the record header information is logged and no bytes from the record values are logged. -1 will log the entire record in addition to the header. A value >0 will log that many bytes from the record to the report output.

LOGDUPS { NONE }
{ BOTH }
{ SOURCE }
{ TARGET }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		NONE	-	-	NO

LOGDUPS is used in compare mode. The default, NONE, is not to log duplicate records to the console. BOTH will log dup records from the source and target files. Alternatively, SOURCE or TARGET will only log dup records from the specified file.

LOGLINEWIDTH { <number> }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		80	0	$(2^{31})-1$	NO

Use LOGLINEWIDTH to specify the output line width. The default line width is 80 characters.

LOGNONMATCH { NONE }
{ BOTH }
{ SOURCE }
{ TARGET }

Initial	Changed

Version	Version	Values Default	Minimum	Maximum	Alter
6.710		NONE	-	-	NO

Use LOGNONMATCH to log records that do not match a record in the other file. The default, NONE, will not log missing/mismatched records to the console. BOTH will log mismatched records in the source and target. Alternatively, SOURCE or TARGET will only log the mismatched records from the specified file.

LOGREADINTERVAL { number }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		0	0	$(2^{31})-1$	NO

Use this parameter to report record reading progress. The default is 0 records, which disables progress reporting.

NONPRINTCHAR { <char> }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		'.'	-	-	NO

Use NONPRINTCHAR to specify the ascii character used to replace unprintable characters in the record output. The default is period ('.').

PROGRAMOPTION { COMPARE }
{ GETINFO }
{ REPAIRMISSINGTGTRECS }
{ LISTOPENERS }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		COMPARE	-	-	NO

By default, the SBBFCMP program will run in compare mode. The valid options are as follows:

- COMPARE - Will read both files and generate compare results.
- GETINFO - Report information about both the source and target files.
- REPAIRMISSINGTGTRECS - Will repair the target file with missing records from the source. When the REPAIRFILENAME variable is specified, the missing records will be written into this file instead of the target filename.
- LISTOPENERS - Will identify programs that have the source or target files opened.

READSPERTHROTTLE { <number> }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		1	1	$(2^{31})-1$	NO

READSPERTHROTTLE is used to control the throttling of the reading of records. The default is 1, causing the program to delay after every read. This parameter is ignored if READTHROTTLEDELAY is 0.

READTHROTTLEDELAY { <number> }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		0	0	$(2^{31})-1$	NO

READTHROTTLEDELAY is used to throttle the reading of records from both the source and target files. The default is 0.00 seconds causing the program to aggressively read both files as quickly as it can. This value is specified in 0.01 seconds.

REPAIRFILENAME { <repair-file-name> }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		-	-	-	NO

This is an optional setting for use with the REPAIRMISSINGTGTRECS variable. By default, the target file will be repaired. Meaning that missing target records will be written into the target file. When this value is specified, the repair records will be written to the repair file.

REPAIRIGNOREREC0 { ON }
{ OFF }

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

Some applications use record zero to store special information. Enabling this setting will prevent repair from inserting record 0 into the target or repair file. The default will insert record zero mismatches.

SOURCEFILENAME { source-file-name }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		-	-	-	NO

Source file name is a require parameter and is used in the compare.

SOURCESTOPADDRESS { number }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		-	0	$(2^{63})-1$	NO

This is an optional entry-sequence record address. If specified, reading from the source file will stop after reaching this address (inclusive). If not specified, the default behavior of SBBFCMP will continue reading until reaching eof.

This address is always specified as a 64-bit format 2 address where the 32 most significant bits pertain to the block number and the 32 least significant bits pertain to the record number within the block. See the section on Enscribe Entry-Sequenced Files in the *Enscribe Programmer's Guide* for more information on how block numbers and record numbers are stored in 32 bit addresses for format 1 files. The number of bits used for the block number and the record within the block for format 1 files are based on the file's block size.

TARGETFILENAME { target-file-name }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		-	-	-	NO

Target file name to use in the compare. Always supply this parameter, as it is required.

TARGETSTOPADDRESS { <number> }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.710		-	0	$(2^{63})-1$	NO

This is an optional entry-sequence record address. If specified, reading from the target file will stop after reaching this address (inclusive). If not specified, the default behavior is to read until reaching eof.

This address is always specified as a 64-bit format 2 address where the 32 most significant bits pertain to the block number and the 32 least significant bits pertain to the record number within the block. See the section on Enscribe Entry-Sequenced Files in the *Enscribe Programmer's Guide* for more information on how block numbers and record numbers are stored in 32 bit addresses for format 1 files. The number of bits used

for the block number and the record within the block for format 1 files are based on the file's block size.

VERBOSE { ON }
{ OFF }

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

The default is OFF. ON will display the runtime settings and additional informational messages.

WRITESSPERTHROTTLE { <number> }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000		1		1	$(2^{31})-1$	NO

WRITESPERTHROTTLE is used to control the throttling of the writing of records to the target or repair file when the repair option is enabled. The default is 1, causing the program to delay after every write. This parameter is ignored if WRITETHROTTLEDELAY is 0.

WRITETHROTTLEDELAY { number }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
6.710		0		0	$(2^{31})-1$	NO

Use WRITETHROTTLEDELAY to throttle the writing of records to the target or repair file when the repair option is enabled. The default is 0.00 seconds causing the program to aggressively write to the file as fast as it can. This value is specified in 0.01 seconds.

SOURCE/TARGETSTOPADDRESS Usage

Use SOURCESTOPADDRESS and TARGETSTOPADDRESS as the address to stop comparing records from either of the two files. This is particularly useful for files that are being replicated and there is a lag or a replication, system or network hiccup. The user wants to be absolutely certain that all records up to a particular address have been replicated to the target successfully without any duplicate entries.

In this scenario the user would first verify that replication is running normally. Then run SBBFCMP with the PROGRAMOPTION set to GETINFO to get the very last record address in the source file. The next step would be to wait until replication has processed the very last record using the last event timestamp retrieved from the corresponding

consumer and the wall clock time of the GETINFO command. Then after making sure the LTS is after the wall clock time, run the SBBMFCMP with the compare option using the SOURCESTOPADDRESS and verifying all records matched a target record without any duplicates.

Note that there are limitations with identifying duplicates when there are very large files involved in the compare. See the SBBFCMP limitations section.

SBBFCMP Limitations

The program has a limitation due to the amount of heap memory that is available to a Guardian process. As of the initial release of SBBFCMP, this memory limitation is 1.6 gigabytes. When running a standard sbbfcmp compare, the program can accommodate a total of approximately 54 million total records or 27 million records in each of the two files. These record counts are current approximations and are subject to change due to program changes and improvements.

SBBFCMP will display the following fatal error when running compare in the default mode (without COMPAREBATCHSIZE) and this limit is reached.

```
Fatal memory allocation error, consider using COMPAREBATCHSIZE, cannot continue
```

Enabling COMPAREBATCHSIZE should accommodate the largest of entry sequenced files as long as the total number of mismatches do not exceed approximately 54 million total records. Just remember that SBBFCMP cannot identify duplicate records when COMPAREBATCHSIZE is enabled.

Queue File Utility (SBQFUTL)

There is a program called SBQFUTL included in the HPE Shadowbase suite of software. This tool is used primarily for reading Queue Manager queue files and displaying (or storing) the information in human-readable format. For security purposes queue files are opened read-only and all new output files are written with a security of 'UUUU' to ensure that original queue files remain unchanged and only those with access to view the queue files can read the output of the tool.

In addition, the Queue File Utility provides various options to filter and sort the output as well as displays to summarize changes to data files and provide insight into the overall contents of the queue file. The Queue File Utility (SBQFUTL) can be run from the command prompt with an in file (SBQFUTL /IN <filename>/) or started with a semi-colon separated string of commands (SBQFUTL LOAD <filename>;START is one example). It can also be started with no initial arguments (SBQFUTL).

Licensing

The SBQFUTL software requires a valid Shadowbase license in order to run. By default, SBQFUTL software will look for a file called SHADPASS in the same subvolume as itself and attempt to validate that file as its license.

Queue File Contents

The Queue Manager Queue Files contain database DDL and DML operations (inserts, commits, etc). These database operations are sent to the receiving side in events. The events are grouped inside Inter-Process Commands (IPCs). The IPCs are grouped together and sent over in blocks. Each level of grouped data has its own header except the IPC which has two headers, one of which is added by the Queue Manager.

Commands

The SBQFUTL software runs interactively or obeys a file. In cases where errors arise the utility will alert the user via a message to the terminal even when not running interactively. Single commands may be entered with or without a semicolon, but multiple commands on the same line should be separated by a semicolon.

ADD

The ADD command is used to add files and transactions for filtering.

```
ADD <add-option>

<add-option>
     {  SRCFILE <source file name>  } -- May be wildcarded
     {  TRANSACTION <transid>       }

<transid>
     {  <num>                                    }
     {  \<node>[(<tm-flags>)].<cpu>.<sequence>   }
```

ADD SRCFILE is used to add source files by which the data is filtered. The source files are the files acted upon by database statements. Only block, Inter Process Communication (IPC), and event data relating to the listed source files will be displayed. Source files may be wildcarded with * and the user does not need to have access to the files in order to filter by them.
ADD TRANSACTION is used to add transactions by which the data is filtered. Only block, Inter Process Communication (IPC), and event data included in the

listed transactions will be displayed. Transactions may be input by either the internal or external format.

```
SBQFUTL 1> add transaction -72057391843639295
  Adding Transaction...
Transaction \VIV1.0.3085240 / -72057391843639295 added.
SBQFUTL 2> add transaction \VIV1.0.421342726
  Adding Transaction...
Transaction \VIV1.0.421342726 / -72029980921036800 added.
```

Transactions added using the external format may be wildcarded. The system name, cpu, sequence number may each be wildcarded with a single asterisk. For example the following are valid inputs:

```
SBQFUTL 1> add transaction \viv1.*.*
Adding Transaction...
Transaction \VIV1.*.* added.
SBQFUTL 2> add transaction \viv1.0.*
Adding Transaction...
Transaction \VIV1.0.* added.
SBQFUTL 3> add transaction \*.0.*
Adding Transaction...
Transaction \*.0.* added.
SBQFUTL 4> add transaction \*.*.3085240
Adding Transaction...
Transaction \*.*.3085240 added.
```

BEGIN

The BEGIN command sets the begin point for reading. The user can specify an audit trail (ADT) position, queue position, or time. Events before this point are ignored. By default, reading begins at the start of the first loaded queue file.

```
BEGIN <begin-option>

<begin-option>
     { ADT-POSITION <adt-pos>   } -- {<MAT sequence number>.[RBA]}
     { QUEUE-POSITION <q-pos>   } -- {<Queue sequence number>.[RBA]}
     { TIME <time>              } -- {[MM-DD-YYYY] [hh:mm[:ss[.mssssss]]]}
```

BEGIN ADT-POSITION sets the begin point to be the specified ADT position in the master audit trail.

BEGIN QUEUE-POSITION sets the begin point to be the specified queue file position.

BEGIN TIME sets the begin point to be the specified event time in the master audit trail.

DISPLAY

The DISPLAY command governs which portions of the queue files are displayed. There are several header-related options, summary reports, data-related options, and miscellaneous options that can be set. Requested items are displayed based on the events read and the events selected for display. Blocks and IPCs are selected for display based on the events they contain. For instance, if one particular event type is chosen, only blocks or IPCs containing events of that type will be displayed. By default the event headers, file counts report, and run summary report are displayed.

```
DISPLAY { <display-option> {OFF | ON}           }
        { EVENT-HEADER {OFF | SUMMARY | DETAIL} }

<display-option>
     { ALL                }
     { COMMA-DELIMITED    } -- OFF by default
     { EVENT-ABORT        } -- ON by default
     { EVENT-ALL          }
     { EVENT-ALTER        } -- ON      by default
     { EVENT-AUDIT-FLUSH  } -- ON      by default
     { EVENT-COMMIT       } -- ON      by default
     { EVENT-CREATE       } -- ON      by default
     { EVENT-DATA         } -- OFF     by default
     { EVENT-DELETE       } -- ON      by default
     { EVENT-HEADER       } -- SUMMARY by default
     { EVENT-INSERT       } -- ON      by default
     { EVENT-OTHER        } -- ON      by default
     { EVENT-PURGE        } -- ON      by default
     { EVENT-PURGEDATA    } -- ON      by default
     { EVENT-RENAME       } -- ON      by default
     { EVENT-UPDATE       } -- ON      by default
     { FILE-COUNTS        } -- ON      by default
     { IPC-HEADER         } -- OFF     by default
     { QBLOCK-HEADER      } -- OFF     by default
     { QFILE-DETAIL       } -- OFF     by default
     { QIPC-HEADER        } -- OFF     by default
     { SRCFILE-DETAIL     } -- OFF     by default
     { SUMMARY            } -- ON      by default
     { TRANSACTION-SUMMARY } -- OFF     by default
     { VERBOSE            } -- OFF     by default
```

ALL

Toggles the following display options:

- EVENT-ALL
- EVENT-DATA
- EVENT-HEADER
- FILE-COUNTS
- IPC-HEADER
- QBLOCK-HEADER
- QFILE-DETAIL
- QIPC-HEADER
- SRCFILE-DETAIL
- SUMMARY
- TRANSACTION-SUMMARY

When ALL is toggled ON, EVENT-HEADER is set to DETAIL and TRANSACTION-SUMMARY is set to OFF. Note that if all event types are off, no events, IPCs, or blocks will be selected for display.

COMMA-DELIMITED

Determines whether or not the output is displayed in comma-delimited format. Comma-delimited formats make it easier to view and manipulate data using other software products. When used for importing and analysis in other programs keep in mind that each option is on it's own line. For example, if the IPC-HEADER and EVENT-HEADER are both set to be displayed each will be on a separate line, so the document will have a non-homogenous mix of IPC and Event headers. If more than one sets of data are to be output it is recommended to output these in separate runs for ease of import and analysis. The DISPLAY ALL command does not toggle the COMMA-DELIMITED option off and on.

EVENT-ABORT

Determines if header, data, and file count information relating to abort events should be displayed. Note that this affects the abort events themselves, NOT the aborted events.

EVENT-ALL

Toggles all of the DISPLAY options related to displaying different types of audit events. Note that if all event types are off, no events, IPCs, or blocks will be selected. The following options are affected:

- EVENT-ABORT
- EVENT-ALTER
- EVENT-AUDIT-FLUSH
- EVENT-COMMIT
- EVENT-CREATE
- EVENT-DELETE
- EVENT-INSERT
- EVENT-OTHER
- EVENT-PURGE
- EVENT-PURGEDATA
- EVENT-RENAME
- EVENT-UPDATE

EVENT-ALTER

Determines if header, data, and file count information relating to alter events should be displayed.

EVENT-AUDIT-FLUSH

Determines if header and summary information relating to audit flush events should be displayed. Note that since these events are not associated with a source file they are not included in the file-counts report.

EVENT-COMMIT

Determines if header, data, and file count information relating to commit events should be displayed. Note that this affects the commit events themselves, NOT the commited events.

EVENT-CREATE

Determines if header, data, and file count information relating to create events should be displayed.

EVENT-DATA

Determines whether or not the data itself is displayed for each event. In order to display data the DISPLAY EVENT-HEADER may not be OFF. This is to provide context for the raw data. Before and after data are displayed only when applicable. For example an update will have both before and after data, but an insert will have only after data. Data will be displayed in hexadecimal as well as plain text. Where characters cannot be displayed in plain text they are represented by a period.

EVENT-DELETE

Determines if header, data, and file count information relating to delete events should be displayed.

EVENT-HEADER

Determines whether or not the event header is included in the output. Options are OFF, SUMMARY, and DETAIL. The OFF option does not display any event headers. This also precludes the display of event data. The SUMMARY option shows the most important fields. The DETAIL option includes all available information in the event header. The following fields are displayed if EVENT-HEADER is set to SUMMARY or DETAIL:
Event Type: Type of the event represented
MAT Address: Master Audit Trail address (prefix seqno:RBA)
AUX Address: Auxiliary Audit Trail address (prefix seqno:RBA)
Event Time: Timestamp of the event in the master audit trail
Trans ID: Transaction identifier (external format / internal format)
Source File: Source file/table name
Before Length: Length of the record/row before event execution (included when the event has before data)
After Length: Length of the record/row after event execution (included when the event has after data)

The following fields are only displayed if EVENT-HEADER is set to DETAIL:
Trans Status: For internal use
Expected Ends: The number of consumers involved in the transaction (only used with other servers targets where multi-ported objects may be in play)
Indicators: For internal use

TX Stop/Start Timestamp: The start time of Collector processing the transaction on the source, or for end transaction events (commit/abort), the time at which the end transaction was processed by the Collector.
Before Fill Length: Length of any additional filler present following the before image
Fill Length: Length of any filler present following the image data

EVENT-INSERT

Determines if header, data, and file count information relating to insert events should be displayed.

EVENT-OTHER

Determines if header, data, and file count information relating to miscellaneous events should be displayed.

EVENT-PURGE

Determines if header, data, and file count information relating to purge events should be displayed.

EVENT-PURGEDATA

Determines if header, data, and file count information relating to purgedata events should be displayed.

EVENT-RENAME

Determines if header, data, and file count information relating to rename events should be displayed.

EVENT-UPDATE

Determines if header, data, and file count information relating to update events should be displayed.

FILE-COUNTS

Determines whether or not to display the FILE-COUNTS report after processing the selected queue files. This report shows the number of inserts, updates, deletes, and transactions that were read for each file as well as total. It will also show if each event or transaction was aborted, committed, or neither. Events part of a transaction that is neither committed nor aborted are labeled ACTIVE. If there are no Events associated with ACTIVE, COMMITed, and ABORTed transactions for a particular source file those rows are not displayed. In the example below note that there are no ABORTed transactions, thus no row displaying them under the source file. DDL events and Shadowbase Generated transactions (SB GEN TX) are only displayed when present for the file. Certain DDL events are not always done as part of a transaction. In order to capture these transactionless events Shadowbase generates a synthetic transaction to aide in replication.

```
FILE-COUNTS:
SOURCEFILE                          INSERTS UPDATES DELETES IUD SUM TX
=================================== ======= ======= ======= ======= =============
\VIV1.$DATA1.DATA.FISH                 1950       0       0    1950    2217
  (COMMITS(DO))                        1950       0       0    1950    2128
  (SB GEN TX)                             0       0       0       0      89
  UNDO                                    0       0       0       0
  DDL EVENTS:
   CREATES:    0       ALTERS:     178
   RENAMES:    0       PURGES:     0
   PURGEDATAS: 89      OTHERS:     0
*** TOTAL SELECTED                    1950       0       0    1950    2217
(TOTAL COMMITS(DO))                   1950       0       0    1950    2128
(TOTAL ABORTS(DO+UNDO))                  0       0       0       0       0
(TOTAL ACTIVE)                           0       0       0       0       0
(TOTAL SB GEN TX)                        0       0       0       0      89
```

IPC-HEADER

Determines whether or not the IPC header is displayed in the output. The IPC-HEADER contains the following fields:
Version: The Shadowbase version number of the software that created the IPC.
Source: The source of the IPC
Destination: The destination of the IPC
Category: The type of IPC. While the QMGR typically records data IPCs containing data events, other types of IPCs are sent to the target from time to time.
Error: Error field
System: Node number of the node the IPC came from
Flag: for internal use
Message Number: The message sequence number
Collector Restart time: The time from which the collector will start its processing on a restart, at the time of the message
Collector Restart Position: The audit trail position from which the collector will start its processing on a restart, at the time of the message
Number of Events: The number of events in this IPC
Position: The queue position of this IPC

QBLOCK-HEADER

Determines whether or not the queue block header is displayed in the output. The QBLOCK-HEADER contains the following fields:
Eye-catcher: Special characters denoting the start of a block
Last Block: Indicator for whether or not it is the last block in a file,
Partial: Indicator for whether the block is full or only partially filled with data.
Block Number: Label for the block. Each block in a file has a unique block number
Length: Size of the used portion of the block
Sequence Number: Unique file identification number included in the QMGR queue file name
RBA: Relative Byte Address of the beginning of the block within the QMGR queue file
Write Timestamp: The date and time the block was initially written to the QMGR queue file

Update Timestamp: The date and time the block was last written to the QMGR queue file

QFILE-DETAIL

Determines whether or not to display additional information about each queue file that was processed. For each queue file, the following will be displayed both when the file is done being processed and when the run summary report (if enabled) is displayed:
First Queue Position: The sequence number and RBA of the first queue block read in this queue file
First Event Time: The event time of the first event in this queue file within the time/position range
First MAT Address: The ADT position of the first event in this queue file within the time/position range
Last Queue Position: The sequence number and RBA of the last queue block read in this queue file

Last Event Time: The event time of the last event in this queue file within the time/position range
Last MAT Address: The ADT position of the last event in this queue file within the time/position range
Last Valid Transaction in File: The transaction ID and abort or commit timestamp of the last transaction within the time/position range in this queue file.

QIPC-HEADER

Determines whether or not the header added on to each Inter-Process Communication (IPC) by the Queue Manager is displayed in the output. The QIPC-HEADER contains the following fields:
Eye-catcher: Special characters denoting the start of an IPC
Continuation: indicates whether or not the entire IPC is contained in the current block, and if not, which side of the IPC is included
Type: the general type of communication
Data Length: the size of the IPC that is in this block minus the size of the QIPC header
Received: the date and time the IPC was received by the QMGR
Counter: the IPC number in the queue

SRCFILE-DETAIL

Determines whether or not the FILE-COUNTS section includes additional information for each source file. For each source file, the number of inserts, updates, and deletes that were part of an UNDO transaction as well as any DDL events will be displayed. These items will not be displayed if the SRCFILE-DETAIL option is OFF. This option has no effect if FILE-COUNTS is OFF.

SUMMARY

Determines whether or not the report summarizing the run is displayed. This summary includes the following information:
RUN: When processing began
ELAPSED: How long SBQFUTL took to process all the queue files
QUEUE FILES: The queue files that were read from
SELECTION RANGE: Any filters from the ADD, BEGIN, and END commands that were used
BLOCKS: The number of blocks read during the run
IPCS: The number of IPCs read during the run
COMMITS: The number of commits read during the run
ABORTS: The number of aborts read during the run
NETWORK COMMITS: The number of network commits read during the run
NETWORK ABORTS: The number of network aborts read during the run
EVENTS: The total number of events read during the run
SELECTED (EVENTS): The total number of events selected by the filters.
FILES: The total number of source files read during the run
SELECTED (FILES): The total number of source files selected by the filters.
IPCS/BLOCK: Average number of IPCs per queue file block
IPC SIZE (BYTES): Average size of the IPCs read
EVENTS/IPC: Average number of events per IPC
EVENTS/TX: Average number of events per transaction
FILE PURGEDATAS: The number of purgedata events read
FILE PURGES: The number of purge events read
FILE ALTERS: The number of alter events read
FILE RENAMES: The number of rename events read
FILE CREATES: The number of create events read
AUDIT FLUSHES: The number of audit flush events read. Audit flush events are only seen when running ZDL.**UNDO EVENTS**: The number of events that were part of an UNDO transaction.
SB GEN TX: The number of Shadowbase generated transactions
LAST TRANSACTION: The transaction ID and abort or commit timestamp of the last transaction read
FIRST QUEUE POSITION: The sequence number and RBA of the first queue block read
LAST QUEUE POSITION: The sequence number and RBA of the last queue block read
FIRST EVENT TIME: The event time of the first event read
FIRST MAT ADDRESS: The ADT position of the first event read
LAST EVENT TIME: The event time of the last event read
LAST MAT ADDRESS: The ADT position of the last event read

TRANSACTION-SUMMARY

Determines whether or not the TRANSACTION-SUMMARY section is displayed. If this option is turned ON, the options SUMMARY, FILE-COUNTS, and SRCFILE-DETAILS will be turned OFF. This section displays a brief

summary of each transaction that was read. For each transaction, the following is displayed:

- The transaction ID
- Whether the transaction was committed, aborted, or was still active
- The number of inserts, updates, and deletes
- The number of undo events and undo inserts, updates, and deletes
- The source files that were affected by the transaction

VERBOSE

Determines whether additional informational messages are displayed during processing. DISPLAY ALL does not change this option.

END

The END command sets the end point for reading. The user can specify an audit trail (ADT) position, queue position, or time. Events after this point are ignored. By default, reading ends at the end of the last loaded queue file.

```
END <end-option>

<end-option>
     { ADT-POSITION <adt-pos>   } -- {<MAT sequence number>.[RBA]}
     { QUEUE-POSITION <q-pos>   } -- {<queue position.[RBA]}
     { TIME <time>              } -- {[MM-DD-YYYY] [hh:mm[:ss[.mssssss]]]}
```

END ADT-POSITION sets the end point to be the specified ADT position in the master audit trail.

END QUEUE-POSITION sets the end point to be the specified queue file position.

END TIME sets the end point to be the specified event time in the master audit trail.

EXIT

The EXIT command gracefully exits the utility.

```
EXIT
```

FC (Fix Command)

The FC command repeats a prior command with the option to edit it before executing.

```
FC [ [-] <num> | <string> ]
```

- If no argument is entered, the previous command is repeated.
- If a positive number is entered, the command on the <num> line number is repeated.
- If a negative number is entered, the command <num> commands back is repeated.
- If a string is entered, the most recent command that begins with <string> is repeated.

HELP

HELP displays the help menu.

```
HELP [<command name> | ALL]
```

HISTORY

The HISTORY command shows the previous commands.

```
HISTORY [ <num-commands> ]
```

Displays the last <num-commands> commands. If no argument is provided, the last 10 commands are displayed.

LIST

Displays the current options in use

```
LIST <list-option>

<list-option>
     { ALL          }
     { PARMS        }
     { QFILE        }
     { SRCFILE      }
     { TRANSACTION  }
```

LIST ALL displays all of the information displayed by the other LIST options.
LIST PARMS displays the options set by the OUT, BEGIN, END, and DISPLAY commands.
LIST QFILE displays the queue files set by the LOAD command.

LIST SRCFILE displays the names of the data (source) files to display data for.
LIST TRANSACTION displays the transaction IDs filtered against.

LOAD

Adds queue files created by the Queue Manager to the list of files to be read.

```
LOAD <queue file name>
```

Wildcarding is supported for the queue files. This list is maintained until the process is exited or stopped. The list of Queue Files can be cleared using the RESET command and viewed using the LIST and SHOW commands. Files added using this command must be viewable by the user running the SBQFUTL utility. If no Queue Files are added to the tool the utility will not produce meaningful output. The last 6 characters of the filename must be unique in the list of files added. For example, QF00001 and LL00001 cannot be included in the same file list and must be added to the SBQFUTL utility in separate runs.

OBEY

The OBEY command reads commands from a file instead of a terminal and executes them.

```
OBEY <filename>
```

OUT

Changes the output from writing to the terminal (default) to writing to a file.

```
OUT <output filename>
```

Error messages will still print to the terminal. If the file already exists the output will be appended to the file. If the file does not exist it will be created.

When displaying all of the event data of very large files the files must be pre-created outside the SBQFUTL software. Precreated files should be about 3 times the size of the total file size of all the qfiles to be read.

RESET

Returns options to their default values.

```
RESET <reset-option>

<reset-option>
     { ALL             }
     { OUT             }
     { PARMS           }
     { QFILE           }
     { SRCFILE         }
     { TRANSACTION     }
```

RESET OUT resets the output to the terminal.
RESET PARMS resets the parameters set by the DISPLAY, BEGIN, and END commands.
RESET QFILE clears the list of queue files to process.
RESET SRCFILE clears the list of source (data) files to filter on.
RESET TRANSACTION clears the list of transactions to filter on.
RESET ALL does all of the above.

SHOW

Displays all of the current options in use. Equivalent to the LIST ALL command.

```
SHOW
```

START

The START command starts processing queue files. No user input (except a break) is accepted while processing the files.

```
START
```

!

The ! command repeats a prior command.

```
!  [ [-] <num> | <string> ]
```

- If no argument is entered, the previous command is repeated.
- If a positive number is entered, the command on the <num> line number is repeated.
- If a negative number is entered, the command <num> commands back is repeated.

- If a string is entered, the most recent command that begins with <string> is repeated.

?

The ? command displays a prior command.

```
?  [ [-] <num> | <string> ]
```

- If no argument is entered, the previous command is displayed.
- If a positive number is entered, the command on the <num> line number is displayed.
- If a negative number is entered, the command <num> commands back is displayed.
- If a string is entered, the most recent command that begins with <string> is displayed.

Wildcard Compare Utilities

Description

The current COMPARE utilities (SOLV, SQLCMPE, and SQLCMPMX) are designed to compare only one file/table at a time which poses problems for customers having large numbers of files/tables needing to be compared. The following tools were created to allow users to specify wild-carded templates for multiple comparisons in one session and/or NetBatch job.

There are two tools, one for ENSCRIBE and SQL/MP comparisons (SBCMPWLD) and the other is for SQL/MX comparisons (SBCMXWLD). A separate ancillary tool (SBCMPWAK) is also provided to aid in the identification of Alternate Index files to build an exclusion file to skip them during comparison executions.

If there's a need to run BOTH tools from the same TACL session (or NetBatch job), please refer to the SB_CLEAR commands for the instructions needed to prevent errors that will be caused due to common/shared code and variables.

Usage

Both comparison tools were based on the same execution design as the SBUNIDIR/SBBIDIR tools that are used to generate the various forms of Shadowbase configurations. The user should find the commands familiar due to their similarity to the ones they've been using to configure their Shadowbase environments.

Both tools can also be invoked via a NetBatch job running TACL as the executor-program. Due to that capability, when not using NetBatch, the script output should ONLY be redirecected using the TACL built-in variable #OUT (see example in the "SBCMPWLD – Sample TACL Input File" section below).

SBCMPWLD – Command Reference

SB_CLEAR

Syntax:

SB_CLEAR [-NOBANNER]

Semantics:

Clear (reset) the entire TACL environment. The optional -NOBANNER parameter allows the suppression of the GRAVIC copyright information.

WARNING: This will RESET all parameters, including statistics, in the TACL execution to either NULL or a DEFAULT value.

NOTE: If there's a need to switch from the ENSCRIBE & SQL/MP comparison tool (SBCMPWLD) to the SQL/MX version (SBCMXWLD) this command MUST be executed before the TACL library is loaded.

Example:

```
SB_CLEAR -NOBANNER
```

SB_LIST

Syntax:

SB_LIST

Semantics:

This will display the current parameter values defined by the use of the SB_SET function.

Example:

```
SB_LIST
```

This will cause the wildcard tool to display its current settings.

SB_RUN

Syntax:

SB_RUN {<src-file-template>} {<tgt-fileset>} {<sqlcmp-obeyfile>|*|-} {<solv-shadparm>|*|-} [<shadparm-section>|-] [<src-IP-address>|-] [<tgt-IP-address>|-]

Semantics:

This command is a single-line submission for a comparison, it contains a "space separated" set of criteria to define the request parameters. The following is a list of the criteria and the possible values that can be supplied:

{<src-file-template>}	SOURCE file template using TACL wild card syntax.
{<tgt-fileset>}	TARGET fileset using GRAVIC syntax, either an EXACT name or a single asterisk (*) character.
{<sqlcmp-obeyfile>|*|-}	Either a fully-qualified file name, a single asterisk (*) character, or a dash (-) character. (*) indicates to use the SB_SQLCMP_CMDS value. This file is optional. (-) indicates to skip SQL/MP comparisons.
{<solv-shadparm>|*|-}	Either a fully-qualified file name, a single asterisk (*) character, or a dash (-) character. (*) indicates to use the SB_SHADPARM value. This file is optional. (-) indicates to skip ENSCRIBE comparisons.
{<shadparm-section>|-}	Either a valid section name or a single dash dash (-) character. A dash character (-) indicates to use the default section built-in value. This is to allow a SHADPARM with multiple sections to used and specified.
{<src-IP-address>|-}	An optional IP address:port if using the TCP/IP connection feature. A dash character (-) indicates to

	skip this feature. Use the standard raw IP number scheme (i.e., 127.0.0.1) that references the SOURCE system.
{<tgt-IP-address>|-}	An optional IP address:port if using the TCP/IP connection feature. A dash character (-) indicates to skip this feature. Use the standard raw IP number scheme (i.e., 127.0.0.1) that references the TARGET system.

Example:

```
SB_RUN $QA.CMPTEST1.* $*.CMPTEST2.* * - - - 10.1.50.220:1820
```

SB_RUN_BATCH

Syntax:

SB_RUN_BATCH [\<node>.][$<vol>.<svol>.]{<file>}

Semantics:

Defines an obey (EDIT-101) file that contains comparison templates (see SB_RUN for syntax values), one line per request. The wildcard tool will read this file serially and process the comparison as requested before moving to the next request.

Example:

```
SB_RUN_BATCH            $VOL.SVOL.FILE
```

This will cause the wildcard tool to read $VOL.SVOL.FILE and process its contents sequentially.

SB_SET

The SB_SET command is used to set specific SBWLDCRD settings.

SB_DATA_SVOL

Syntax:

SB_DATA_SVOL [\<node>.]{$<vol>.<svol>}

Semantics:

Defines a work subvolume to be used by the COMPARE utilities to store ancillary files needed to peform the comparisons. If set, this will also be the default subvolume for any files specified in the configuration if not fully qualified. If not specified, the tool will default to the current subvolume.

Example:

```
SB_SET SB_DATA_SVOL     $VOL.SVOL
```

This will set the work subvolume to $VOL.SVOL.

SB_EXCLUDE_2NDPARTS

Syntax:

SB_EXCLUDE_2NDPARTS{YES|NO}

Semantics:

Defines whether Secondary Partitions are excluded from the comparison, the DEFAULT is YES and the comparison will only be done using the Primary Partition name.

Example:

```
SB_SET SB_EXCLUDE_2NDPARTS    NO
```

This will cause Secondary Partitions to be compared as separated files/tables.

SB_EXCLUDE_FILE

Syntax:

SB_EXCLUDE_FILE [\<node>.][$<vol>.<svol>.]{<file>}

Semantics:

Defines an EXCLUSION file that can be used to bypass individual files from comparison. Each line will contain either one fully-qualified file name or a file name template using the TACL wildcard format characters (*, ?) to skip multiple

files. Both types of entries must have all four components (i.e., node, vol, svol, and file) of a file name to correctly match.

This feature is particularly useful in skipping the comparison of ALTERNATE INDEX files as they are not replicated due to RBA differences between systems. An ancillary tool (SBMKSKIP) has been provided to build an exclusion file of all ALT INDEX files found on a given system (see below for usages).

Example:

```
SB_SET SB_EXCLUDE_FILE  $VOL.SVOL.FILE
```

This will set a value for the EXCLUDE file to $VOL.SVOL.FILE.

SB_FTYPE_SKIP

Syntax:

SB_FTYPE_SKIP {(E|,K|,Q|,R|,U)}

Semantics:

Defines a list of file types that will be "skipped" for comparison:

(E)ntry-Sequenced
(K)ey-Sequenced
(Q)ueue Files
(R)elative Files
(U)nsctructured Files

Example:

```
SB_SET SB_FTYPE_SKIP    (E,Q,R,U)
```

This will cause entry-sequenced, queue, relative, and unstructured files to be skipped during the comparison run.

SB_MSG_FLAGS

Syntax:

SB_MSG_FLAGS {(A|,C|,E|,M|,N|,S|,T|,V)}

Semantics:

Defines a list of message levels to control the volume of output generated:

(A)LL
(C)RITICAL
(E)XCLUDED
(M)ATCHED
(N)O MATCH
(S)KIPPED
(T)ARGET MISSING
(V)ERBOSE

The (A)LL level will turn ON all of the other levels EXCEPT for (V)ERBOSE which has to be explicitly set. (V)ERBOSE will cause the output of the entire comparison detail and is not recommended unless diagnosing an issue. DEFAULT is (A)LL without (V)ERBOSE

Example:

```
SB_SET SB_MSG_FLAGS     (C,E,N,S,T)
```

This will cause messages to be generated for just exception issues (generally a good for a first attempt).

SB_NODE_SRC

Syntax:

SB_NODE_SRC {\<node>}

Semantics:

Defines the SOURCE system for the comparison requests.

Example:

```
SB_SET SB_NODE_SRC      \NODE
```

This will set the source system name to \NODE.

SB_NODE_TGT

Syntax:

SB_NODE_TGT {\<node>}

Semantics:

Defines the TARGET system for the comparison requests.

Example:

```
SB_SET SB_NODE_TGT      \NODE
```

This will set the target system name to \NODE.

SB_PROCESS_PREFIX

Syntax:

SB_PROCESS_PREFIX {<2-char prefix>}

Semantics:

Optional two character prefix for the purpose of uniquely setting the compare process names. Per standard process naming conventions, the first character is restricted to be alphabetic only. This value, when provided, will be prepended to create process names to the following values:

$<prefix>ESV - Enscribe compare process on the source system
$<prefix>EVV- Enscribe compare process on the target system
$<prefix>SCP - SQL compare process on the source system

If there is an existing process that matches a generated name, the script will skip that comparison.

Example:

```
SB_SET SB_PROCESS_PREFIX      A1
```

This will set the prefix to A1 and the process names will be: $A1ESV, $A1EVV, and $A1SCP respectively. If not specified, the script will default to system generated names for the *ESV and *SCP processes and $SOLVV for the *EVV process.

SB_SHADPARM

Syntax:

SB_SHADPARM [\<node>.][$<vol>.<svol>.]{<file>}

Semantics:

Defines a default SHADPARM file that can be used rather than uniquely specifying one on each compare request.

Example:

```
SB_SET SB_SHADPARM      $VOL.SVOL.FILE
```

This will set a default value for SHADPARM to $VOL.SVOL.FILE that will be used if an explicit specification is not provided on the command line.

SB_SOLV_SVOL

Syntax:

SB_SOLV_SVOL [\<node>.]{$<vol>.<svol>}

Semantics:

Defines the object subvolume for the Enscribe comparision tool (SOLV) on the source system. If not specified, the tool will default to the current subvolume.

Example:

```
SB_SET SB_SOLV_SVOL     $VOL.SVOL
```

This will set the SOLV object subvolume to $VOL.SVOL.

SB_SOLVNV_SVOL

Syntax:

SB_SOLVNV_SVOL[\<node>.]{$<vol>.<svol>}

Semantics:

Defines the object subvolume for the Enscribe comparision tool (SOLVNV) on the target system. If not specified, the tool will default to the current subvolume.

Example:

```
SB_SET SB_SOLVNV_SVOL   $VOL.SVOL
```

This will set the SOLVNV object subvolume to $VOL.SVOL.

SB_SQLCMP_CMDS

Syntax:

SB_SQLCMP_CMDS [\<node>.][$<vol>.<svol>.]{<file>}

Semantics:

Defines a default SQL obey file that can be used rather than uniquely specifying one on each compare request.

Example:

```
SB_SET SB_SQLCMP_CMDS   $VOL.SVOL.FILE
```

This will set a default value for SQL obey to $VOL.SVOL.FILE that will be used if an explicit specification is not provided on the command line.

SB_SQLCMP_SVOL

Syntax:

SB_SQLCMP_SVOL [\<node>.]{$<vol>.<svol>}

Semantics:

Defines the object subvolume for the SQL/MP comparision tool (SQLCMPE) on the source system. If not specified, the tool will default to the current subvolume.

Example:

```
SB_SET SB_SQLCMP_SVOL   $VOL.SVOL
```

This will set the SQLCMPE object subvolume to $VOL.SVOL.

SB_RUN_COMPARES

Syntax:

SB_RUN_COMPARES {YES|NO}

Semantics:

Defines whether the comparison executions are performed. This allows for a syntax check and file/table validation before actual comparisons are attempted.

Example:

```
SB_SET SB_RUN_COMPARES  NO
```

This will cause the suppression of the comparison executions and allow just syntax checking and file/table validataions.

SB_TGT_SQLCMP_SVOL

Syntax:

SB_TGT_SQLCMP_SVOL [\<node>.]{$<vol>.<svol>}

Semantics:

Defines the object subvolume for the SQL/MP comparision tool (DBSRVMP) on the target system. If not specified, the tool will skip this setting.

Example:

```
SB_SET SB_TGT_SQLCMP_SVOL     \NODE.$VOL.SVOL
```

This will set the DBSRVMP object subvolume to \NODE.$VOL.SVOL.

SB_STATS_PRINT

The SB_STATS_PRINT will output all comparison statistics collected to that point. A SB_STATS_RESET command will cause all collected statistics to be set to ZERO. The combination of these two commands will allow the grouping of various comparisons into logical sets as the user desires.

Example:

```
SB_STATS_PRINT
```

This will print the statistics gathered to this point. One or more of the following statistics may be printed:

TGT'S MISSING		- Number of target files/tables that could not be found to match the selection critieria entered.
SKIPPED	(Entry-Seq)	- Number of entry-sequenced source files/tables skipped due to the "E" specification in the SB_FTYPE_SKIP parameter.
SKIPPED	(Key-Seq)	- Number of key-sequenced source files/tables skipped due to the "K" specification in the SB_FTYPE_SKIP parameter.
SKIPPED	(Queue)	- Number of queue source files/tables skipped due to the "Q" specification in the SB_FTYPE_SKIP parameter.
SKIPPED	(Relative)	- Number of relative source files/tables skipped due to the "R" specification in the SB_FTYPE_SKIP parameter.
SKIPPED	(Unstruct)	- Number of unstructured source files/tables skipped due to the "U" specification in the SB_FTYPE_SKIP parameter.
EXCLUDED	(Exact)	- Number of files excluded due to an exact file specification in the file defined by the SB_EXCLUDE_FILE parameter.
EXCLUDED	(Tmplt)	- Number of files excluded due to a file template specification in the file defined by the SB_EXCLUDE_FILE parameter.
ENS_CALL		- Number of Enscribe COMPARE calls executed.
ENS_MATCH		- Number of Enscribe COMPARE calls that resulted in a MATCH (success) condition.
ENS_NOMATCH		- Number of Enscribe COMPARE calls that resulted in a NOMATCH (failure) condition.
ENS_CRITICAL		- Number of Enscribe COMPARE calls that resulted in a CRITICAL (extreme failure) condition.

ENS_PROC_EXISTS	- Number of Enscribe COMPARE calls that resulted in a DUPLICATE process name.
SQL_CALL	- Number of SQL/MP COMPARE calls executed.
SQL_MATCH	- Number of SQL/MP COMPARE calls that resulted in a MATCH (success) condition.
SQL_NOMATCH	- Number of SQL/MP COMPARE calls that resulted in a NOMATCH (failure) condition.
SQL_CRITICAL	- Number of SQL/MP COMPARE calls that resulted in a CRITICAL (extreme failure) condition.
SQL_PROC_EXISTS	- Number of SQL/MP COMPARE calls that resulted in a DUPLICATE process name.

SB_STATS_RESET

The SB_STATS_RESET will cause all collected statistics to be set to ZERO. The combination of this command with the SB_STATS_PRINT command will allow the grouping of various comparisons into logical sets as the user desires.

Example:

```
SB_STATS_RESET
```

This will reset all statistics to ZERO.

SBCMPWLD – Sample TACL Input File

```
== ?TACL MACRO
== Written for NetBatch execution, remove the double-equal above
== when running interactively


   #FRAME

   SINK [#LOAD /KEEP 1/ {$<vol>.<svol>}.SBCMPWLD

   ==
   == The following redirects the script output to the SPOOLER
   == REMOVE the comment characters to enable and set the spooler
   == destination value.
```

```
   ==
   == #PUSH #OUT
   == #SET  #OUT $S.#WLDCRD
   ==

   SB_CLEAR                    [-NOBANNER]

   SB_SET SB_DATA_SVOL         [\<node>.]{$<vol>.<svol>}
   SB_SET SB_EXCLUDE_2NDPARTS {YES|NO}
   SB_SET SB_EXCLUDE_FILE
      [\<node>.][$<vol>.<svol>.]{<file>}
   SB_SET SB_FTYPE_SKIP        {(E|,K|,Q|,R|,U)}
   SB_SET SB_MSG_FLAGS         {(A|,C|,E|,M|,N|,S|,T|,V)}
   SB_SET SB_NODE_SRC          {\<source-node>}
   SB_SET SB_NODE_TGT          {\<target-node>}
   SB_SET SB_PROCESS_PREFIX    {<2-char prefix>}
   SB_SET SB_RUN_COMPARES      {YES|NO}
   SB_SET SB_SHADPARM          [\<node>.][$<vol>.<svol>.]{<file>}
   SB_SET SB_SOLV_SVOL         [\<node>.]{$<vol>.<svol>}
   SB_SET SB_SOLVNV_SVOL             [\<node>.]{$<vol>.<svol>}
   SB_SET SB_SQLCMP_CMDS
      [\<node>.][$<vol>.<svol>.]{<file>}
   SB_SET SB_SQLCMP_SVOL             [\<node>.]{$<vol>.<svol>}
   SB_SET SB_TGT_SQLCMP_SVOL   [\<node>.]{$<vol>.<svol>}

   SB_RUN <src-fileset> <tgt-fileset> <sqlcmp-obeyfile> <solv-
shadparm> &
          <shadparm-section> <127.0.0.1> <127.0.0.2>
   SB_STATS_PRINT

   SB_STATS_RESET

   SB_RUN_BATCH [$<vol>.<svol>.]{<file>}
   SB_STATS_PRINT
#UNFRAME
```

SBCMXWLD – Command Reference

SB_CLEAR

Syntax:

SB_CLEAR [-NOBANNER]

Semantics:

Clear (reset) the entire TACL environment. The optional -NOBANNER parameter allows the suppression of the GRAVIC copyright information.

WARNING: This will RESET all parameters, including statistics, in the TACL execution to either NULL or a DEFAULT value.

NOTE: If there's a need to switch from the SQL/MX comparison tool (SBCMXWLD) to the ENSCRIBE & SQL/MP version (SBCMPWLD) this command MUST be executed before the TACL library is loaded.

Example:

```
SB_CLEAR -NOBANNER
```

SB_LIST

Syntax:

SB_LIST

Semantics:

This will display the current parameter values defined by the use of the SB_SET function.

Example:

```
SB_LIST
```

This will cause the wildcard tool to display its current settings.

SB_RUN

Syntax:

SB_RUN SOURCE: <src cat>.<src sch>.<src tab> TARGET: <tgt cat>.<tgt sch>.<tgt tab> [SRCIP: <src-ip-addr>] [TGTIP: <tgt-ip-addr>]

SOURCE: Keyword to identify source database

<src cat> SOURCE catalog name that conforms to the following conventions:

1) "*" – ALL possible values.
2) A fully-qualified exact-match catalog name (i.e., "AMRCAT").
3) SQL/MX template name using % and _ meta-characters (i.e., "AMR%").

<src sch> SOURCE schema name that conforms to the following conventions:
1) "*" – ALL possible values.
2) A fully-qualified exact-match schema name (i.e., "AMRSCH").
3) SQL/MX template name using % and _ meta-characters (i.e., "AMR%").

<src tab> SOURCE table name that conforms to the following conventions:
1) "*" – ALL possible values.
2) A fully-qualified exact-match table name (i.e., "AMRTBL").
3) SQL/MX template name using % and _ meta-characters (i.e., "AMR%").

TARGET: Keyword to identify target database

<tgt cat> TARGET catalog name that conforms to the following conventions:
1) "*" – Copy the **<src cat>** value specified
2) A fully-qualified exact-match catalog name (i.e., "AMRCAT2").

<tgt sch> TARGET schema name that conforms to the following conventions:
1) "*" – Copy the **<src sch>** value specified
2) A fully-qualified exact-match schema name (i.e., "AMRSCH2").

<tgt tab> TARGET table name that conforms to the following conventions:
1) "*" – Copy the **<src tab>** value specified
2) A fully-qualified exact-match table name (i.e., "AMRTBL2").

SRCIP: Optional keyword to identify the source system using a TCP/IP address:

<src-ip-addr> SOURCE system's IP address if using TCP/IP as the method of communication between systems. Use standard TCP/IP address notation (i.e., 127.0.0.1)

TGTIP: Optional keyword to identify the target system using a TCP/IP address:

<tgt-ip-addr> TARGET system's IP address if using TCP/IP as the method of communication between systems. Use standard TCP/IP address notation (i.e., 127.0.0.2)

Semantics:

This command is a single-line submission for a comparison, it contains a space-separated set of criteria to define the request parameters. The following is a list of the criteria and the possible values that can be supplied:

Example:

```
SB_RUN SOURCE:CJCCAT.CJCSCH.OINFTAB% TARGET:CJCCAT.CJCSCH.*
TGTIP:10.1.50.220
```

SB_RUN_BATCH

Syntax:

SB_RUN_BATCH [\<node>.][$<vol>.<svol>.]{<file>}

Semantics:

Defines an batch (EDIT-101) file that contains comparison templates (see SB_RUN for syntax values), one line per request. The wildcard tool will read this file serially and process the comparison as requested before moving to the next request.

Example:

```
SB_RUN_BATCH            $VOL.SVOL.FILE
```

This will cause the wildcard tool to read $VOL.SVOL.FILE and process its contents sequentially.

SB_SET

The SB_SET command is used to set specific SBCMXWLD settings.

SB_DATA_SVOL

Syntax:

SB_DATA_SVOL [\<node>.]{$<vol>.<svol>}

Semantics:

Defines a data subvolume to be used by the COMPARE utilities to store ancillary files needed to peform the comparisons. If set, this will also be the default subvolume for any files specified in the configuration if not fully qualified.

Example:

```
SB_SET SB_DATA_SVOL     $VOL.SVOL
```

This will set the data subvolume to $VOL.SVOL.

SB_MSG_FLAGS

Syntax:

SB_MSG_FLAGS {(A|,C|,E|,M|,N|,S|,T|,V)}

Semantics:

Defines a list of message levels to control the volume of output generated:

(A)LL
(C)RITICAL
(E)XCLUDED
(M)ATCHED
(N)O MATCH
(S)KIPPED
(T)ARGET MISSING
(V)ERBOSE

The (A)LL level will turn ON all of the other levels EXCEPT for (V)ERBOSE which has to be explicitly set. (V)ERBOSE will cause the output of the entire comparison detail and is not recommended unless diagnosing an issue. DEFAULT is (A)LL without (V)ERBOSE

Example:

```
SB_SET SB_MSG_FLAGS     (C,E,N,S,T)
```

This will cause messages to be generated for just exception issues (generally a good for a first attempt).

SB_NODE_SRC

Syntax:

SB_NODE_SRC {\<node>}

Semantics:

Defines the SOURCE system for the comparison requests.

Example:

```
SB_SET SB_NODE_SRC      \NODE
```

This will set the source system name to \NODE.

SB_NODE_TGT

Syntax:

SB_NODE_TGT {\<node>}

Semantics:

Defines the TARGET system for the comparison requests.

Example:

```
SB_SET SB_NODE_TGT      \NODE
```

This will set the target system name to \NODE.

SB_PROCESS_PREFIX

Syntax:

SB_PROCESS_PREFIX {<2-char prefix>}

Semantics:

Optional two character prefix for the purpose of uniquely setting the compare process names. Per standard process naming conventions, the first character is restricted to be alphabetic only. This value, when provided, will be prepended to create process names to the following values:

$<prefix>CP - SQL/MX compare process on the source system
$<prefix>MX - MXCI utility process on the source system

If there is an existing process that matches a generated name, the script will skip that comparison.

Example:

```
SB_SET SB_PROCESS_PREFIX      A1
```

This will set the prefix to A1 and the process names will be: $A1CP and $A1MX respectively. If not specified, the script will default to system generated names for the *CP and *MX processes.

SB_RUN_COMPARES

Syntax:

SB_RUN_COMPARES {YES|NO}

Semantics:

Defines whether the comparison executions are performed. This allows for a syntax check and file/table validation before actual comparisons are attempted.

Example:

```
SB_SET SB_RUN_COMPARES  NO
```

This will cause the suppression of the comparison executions and allow just syntax checking and file/table validataions.

SB_SQLCMP_CMDS

Syntax:

SB_SQLCMP_CMDS [\<node>.][$<vol>.<svol>.]{<file>}

Semantics:

Defines an obey file (EDIT-101) that will be read into the SQLCMPMX processing to allow user customization of the comparison.

Example:

```
SB_SET SB_SQLCMP_CMDS    $VOL.SVOL.FILE
```

This will set the obey filename to $VOL.SVOL.FILE.

SB_SQLCMP_SVOL

Syntax:

SB_SQLCMP_SVOL [\<node>.]{$<vol>.<svol>}

Semantics:

Defines the object subvolume for the SQL/MX comparision tool (SQLCMPMX) on the source system.

Example:

```
SB_SET SB_SQLCMP_SVOL    $VOL.SVOL
```

This will set the SQLCMPMX object subvolume to $VOL.SVOL.

SB_TGT_SQLCMP_SVOL

Syntax:

SB_TGT_SQLCMP_SVOL [\<node>.]{$<vol>.<svol>}

Semantics:

Defines the object subvolume for the SQL/MX comparision tool (DBSRV) on the target system.

Example:

```
SB_SET SB_TGT_SQLCMP_SVOL      $VOL.SVOL
```

This will set the DBSRV object subvolume to $VOL.SVOL.

SB_EXCLUDE_FILE

Syntax:

SB_EXCLUDE_FILE [\<node>.][$<vol>.<svol>.]{<file>}

Semantics:

Defines an EXCLUSION file that can be used to bypass individual files from comparison. Each line will contain either one fully-qualified file name or a file name template using the ASCII wildcard format characters ("%', "?") to skip multiple files. Both types of entries must have all three components (i.e., <catalog>.<schema>.<table>) separated by the period character to correctly match.

Example:

```
SB_SET SB_EXCLUDE_FILE  $VOL.SVOL.FILE
```

This will set a value for the EXCLUDE file to $VOL.SVOL.FILE.

SB_STATS_PRINT

The SB_STATS_PRINT will output all comparison statistics collected to that point. A SB_STATS_RESET command will cause all collected statistics to be set to ZERO. The combination of these two commands will allow the grouping of various comparisons into logical sets as the user desires.

Example:

```
SB_STATS_PRINT
```

This will print the statistics gathered to this point.

SB_STATS_RESET

The SB_STATS_RESET will cause all collected statistics to be set to ZERO. The combination of this command with the SB_STATS_PRINT command will allow the grouping of various comparisons into logical sets as the user desires.

Example:

```
SB_STATS_RESET
```

This will reset all statistics to ZERO.

SBCMXWLD – Sample TACL Input File

```
== ?TACL MACRO
== Written for NetBatch execution, remove the double-equal above
== when running interactively

   #FRAME

   SINK [#LOAD /KEEP 1/ {$<vol>.<svol>}.SBCMXWLD]

   SB_CLEAR                    [-NOBANNER]

   SB_SET SB_DATA_SVOL         [\<node>.]{$<vol>.<svol>}
   SB_SET SB_MSG_FLAGS         {(A|,C|,E|,M|,N|,V)}
   SB_SET SB_NODE_SRC          {\<source-node>}
   SB_SET SB_NODE_TGT          {\<target-node>}
   SB_PROCESS_PREFIX                 {2-char prefix}
   SB_SET SB_RUN_COMPARES      {YES|NO}
   SB_SET SB_SQLCMP_CMDS
      [\<node>.][$<vol>.<svol>.]{<file>}
   SB_SET SB_SQLCMP_SVOL             [\<node>.]{$<vol>.<svol>}
   SB_SET SB_TGT_SQLCMP_SVOL   [\<node>.]{$<vol>.<svol>}
   SB_SET SB_EXCLUDE_FILE
      [\<node>.][$<vol>.<svol>.]{<file>}

   SB_OBEY [$<vol>.<svol>.]{<file>}
   SB_STATS_PRINT

   SB_STATS_RESET

#UNFRAME
```

SBCMPWAK – Command Reference

Syntax:

[RUN] SBCMPWAK [\<node>.][$<vol>.][<svol>.][<file>]

Semantics:

This is an independent/standalone tool to scan an entire system for ALTERNATE INDEX files and create an EDIT (101) file suitable for use by the SBCMPWLD tool to exclude such files from comparison. If an explicit file name is not provided on the RUN line, the file will be created in the #DEFAULTS subvolume using the naming convention of ak<yy><mm><dd> using the current date. This file can be used as-is, renamed to another naming convention, and/or modified to

include other file names. Be aware, that the file names are fully qualified (\node.$vol.svol.file) to work in conjunction with the SBCMPWLD tool.

Example:

```
SBCMPWAK
```

ETL Toolkit

The HPE Shadowbase™ ETL Toolkit allows the user to perform two tasks:

1) Read/extract records from an HPE NonStop source file or table, or
2) Read/extract change data from the audit trail for an HPE NonStop source file or table,

into flat files or other target environments (such as Kafka or IBM MQ) with a configurable structure. This section describes basic functionality and configuration.

Introduction

The ETL Toolkit is a Consumer configuration that allows for the writing from an audit trail to a flat file or other targets (eg. Kafka). Valid sources of data are Enscribe files, SQL/MP tables, and SQL/MX tables. This feature is enabled by setting the Consumer parameter ETLPROCESSING to ON. When set to ON, the Consumer functions as a *consumptive* consumer that writes to flat files or other targets from the NonStop using a dummy target to format the data. The data format can be either a customizable delimited format (e.g. CSV) or JSON.

The ETL Toolkit has three options for where to put the processed events:

- A series of rolling Guardian flat files;
- A remote Kafka topic (via uLinga or directly when replicated out to a Windows or Linux Shadowbase target);
- An IBM MQ queue (local or remote, via TCP/IP)

For Guardian flat files, the names of these files will start with a prefix, followed by a series of sequential numbers. This prefix is 1 character long, followed by a 7-digit file number that is incremented sequentially when the file rolls. When the number reaches the maximum value (9999999), it rolls back to 0.

For replicating into a Kafka topic, uLinga must first be installed and configured on the NonStop system where the Shadowbase consumer will run. The consumer will connect into the remote Kafka topic using uLinga and write all of its data into it. Contact support

if replicating out to a Shadowbase target on Windows or Linux and feeding the data directly into Kafka from that environment (no uLinga required).

For replicating into an IBM MQ queue, the MQIC client DLL must be present on the NonStop system where the Shadowbase consumer will run. The consumer will establish a TCP/IP connection into the MQ queue using this DLL and write all of its data into it.

Running the ETL Toolkit

Configuring the Consumer

When running the ETL Toolkit, a dummy target can be used to add or remove columns. This dummy target can also add any of the well-known name columns. For more information about these columns, see the *HPE NonStop Shadowbase Operations Manual*. In addition, you can add the SHAD_SOURCE_NAME or SHAD_TARGET_NAME column to include the source or target table or file. If the dummy target is an Enscribe file, the field must be PIC X(35). For SQL/MP, the column should be a VARCHAR large enough to fit the table or file.

When writing to flat files, the SHAD_SRC_TIMESTAMP_GMT/LCT well-known names are not supported and the SHAD_TGT_TIMESTAMP_GMT/LCT well-known names are set as the current timestamp when the record is processed.

Parameters related to the ETL Toolkit are set using a file named ETLCONF (name can be changed with Consumer parameter ETLCONFIG). This file is set up in identical fashion to other *.ini* files. Each DBS or target table can have its own section with its own parameters and a "GENERAL" section used as default settings. This "GENERAL" section cannot contain a combination of parameters for different output methods.

Considerations

Flat File Rolling

When the Consumer is started, it will search for the file with prefix set by FILEPREFIX and the largest sequence number and roll to the next file. The three methods of rolling are:

- After N records are written to the file
- If the file about to be written to is older than N seconds
- If the file about to be written to would be larger than N megabytes.

The amount for each option is set by ROLLCOUNT, ROLLTIME, and ROLLSIZE, respectively. Each option has its own default value but if you do not want to roll on one or two of these options, set the value to 0. All three options cannot be set to 0 as there will be no way of rolling to the next file otherwise.

Outputting to multiple target flat files

In order to output to into separate files and different locations, some parameters must be set. For HPE NonStop, FILEPREFIX must be set to a unique prefix for each DBS or target table, and/or FILELOC should specify separate subvolumes.

Escaping characters in very large character fields

When using the ESCAPECHARACTERS parameter, if a character is encountered in the field, it is either surrounded by double quotes or escaped with a backslash. In addition, any existing double quotes in the field will be repeated. For example, EXAM”PLE would be escaped to be “EXAM””PLE”. Because potentially every character could be repeated, what would be written could be over double the actual size of the field. If escaping a character would cause the field to be larger than the internal buffer, the Consumer will abend.

Closing flat files with a timer

To limit the number of times a file is opened and closed by the ETL Toolkit, a timer is used to keep the file open until a certain amount of time has passed since the last event. Parameters FILECLOSEDELAY and FILECLOSEFREQ are used to configure the timer. Once the timer expiries, the flat files are checked to see if they have reached the allotted amount of time they should be open for and are closed if so. If not, the timer is started again the process is repeated.

ETL Toolkit Configuration Parameters

Parameters are used to configure format and contents of the replicated data. These parameters are set using an *.ini* style file configured using ETLCONFIG. These parameters are found during startup and cannot be altered while the Consumer is running.

AUDCONSN Parameters

These parameters are set for the Consumer and can be done through AUDCOM or a TACL start-up script. Refer to the New and Modified SET CONS Parameters section below for further details about these parameters:

ETLCONFIG
ETLPROCESSING

Formatting Parameters

SECTIONHEADER { DBS | TARGETFILE }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			DBS			NO

This parameter determines what the section headers should be in the ETLCONFIG file. When set to DBS, section headers should match DBS names. When set to TARGETFILE, section headers should match the full path of the target file. This parameter must be set within the GENERAL section for it to come into effect. For Wildcard DBS support, use the TARGETFILE option and specific sections with target file names. Attempting to use a Wildcarded DBS with the DBS option will result in schema mismatches if all the target files do not have the schema.

INCLUDERECORD { OFF | ON }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			OFF			NO

This parameter allows the entire event be placed into a column called “RECORD”. When set to ON, all columns in the source table or file are written to the column “RECORD” without any delimiters. The “RECORD” column should be enough space to hold the max size of all source columns. This option is ignored when OUTPUTFORMAT is set to JSON.

INCLUDERECDELIM { OFF | ON }

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

This parameter determines if records are delimited when writing to a flat file on NonStop.

RECDELIM <*delimiter*>

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			\n			

This parameter changes the record delimiter used. The delimiter can be up to 10 characters. If the delimiter starts with “0x”, the rest of the parameter is assumed to be a hexadecimal representation of the delimiter and will be converted.

INCLUDECOLDELIM { OFF | ON }

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

This parameter determines if columns are delimited when writing to a flat file on NonStop or between fields in the "RECORD" column.

COLDELIM <*delimiter*>

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		,			

This parameter changes the column delimiter used. The delimiter can be up to 10 characters. If the delimiter starts with "0x", the rest of the parameter is assumed to be a hexadecimal representation of the delimiter and will be converted.

INCLUDECOLLEN { OFF | CHARACTER | BINARY }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		OFF			NO

This parameter determines if the length of each field is included as a prefix. When set to CHARACTER , the prefix is added as a character field. When set to BINARY, the prefix is added as binary.

COLLENSIZE <col-len-size>

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		5	1	8	NO

This parameter determines the length of the character field prefix when INCLUDECOLLEN is on. When setting this parameter, make sure that the given length is large enough to contain any possible field length. For BINARY, the possible options are 2, 4, and 8.

COLLENENDIAN { HOST | LITTLE | BIG }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		HOST			NO

This parameter determines what Endian will be used when creating INCLUDECOLLEN when set to binary format. When set to HOST, the default host system Endian will be

used, when set to LITTLE, Little Endian will be used, and when set to BIG, Big Endian will be used.

INCLUDERECLEN { OFF | CHARACTER | BINARY }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			OFF			NO

This parameter determines if the entire record will be prefixed with the length of the record. When set to CHARACTER, the prefix is added as a character field. When set to BINARY, the prefix is added as binary. If enabled, this parameter will override INCLUDERECDELIM.

RECLENSIZE <rec-len-size>

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			4	1	8	NO

This parameter determines the length of the character field prefix when INCLUDERECLEN is on. When setting this parameter, make sure that the given length is large enough to contain any possible field length. For BINARY, the possible options are 2, 4, and 8.

RECLENENDIAN { HOST | LITTLE | BIG }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			HOST			NO

This parameter determines what Endian will be used when creating INCLUDERECLEN when set to binary format. When set to HOST, the default host system Endian will be used, when set to LITTLE, Little Endian will be used, and when set to BIG, Big Endian will be used.

PADFIXEDFIELD { OFF | ON }

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

This parameter determines if the fields will be padded with spaces to be the maximum possible size for the given field. The fields will be left-aligned. This parameter is only supported for CUSTOM format.

ESCAPECHARACTERS { OFF | DELIMITERS | UNPRINTABLES | BOTH | ALL}

Initial	Changed

Version	Version	Values Default	Minimum	Maximum	Alter
7.000		OFF			NO

This parameter determines if delimiters will be escaped. If set to DELIMITERS, the field will be surrounded with double quotes if any of the follow are true: The first character is a double quote, INCLUDECOLDELIM is enabled and COLDELIM is in the field, or if INCLUDERECDELIM is enabled and RECDELIM is in the field. Any double quotes in the field are escaped by repeating them. If set to UNPRINTABLES, any unprintable character in the field will be convert to hex with a backslash before it. An example is "\0D". Any backslash in the field will be escaped by repeating them. When set to BOTH, delimiters and unprintable characters are escaped. When set to ALL, every character is escaped into a two-character hex value.

OUTPUTFORMAT { CUSTOM | JSON }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		CUSTOM	CUSTOM	JSON	NO

This parameter determines the format of the output data. CUSTOM outputs custom delimited format, JSON outputs JSON formatting. When JSON format is set, RECDELIM will be set to ',' for proper JSON formatting.

JSONOUTPUTFORMAT { SINGLE | PRETTY }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		SINGLE	SINGLE	PRETTY	NO

This parameter determines the format of the JSON output on the flat file. SINGLE outputs the JSON formatting into single lines. PRETTY outputs the JSON formatting into pretty version for improved readability.

JSONSTRINGFORMAT { ASCII | HEX }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		ASCII	ASCII	HEX	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.

JSONCONVERTTOSTRING { STRING | ALL }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		STRING	STRING	ALL	NO

This parameter determines how the JSON formatting will handle converting data types to strings. The default STRING will only treat non-numeric data types as strings. If set to ALL, all data types will be converted to strings. This will ensure that decimal values do not lose precision. If JSONSTRINGFORMAT is set to HEX and JSONCONVERTTOSTRING is set to ALL, every value will be converted to hex.

OUTPUTMETHOD { FLATFILE | KAFKA | MQ }

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		FLATFILE			NO

This parameter determines what method is used for outputting data. FLATFILE outputs the data to flat files. KAFKA outputs the data to a Kafka Topic. MQ outputs data to a MQ Queue. A DBS can only be configured with one output method and should only contain parameters for that output method.

PARAMLOGGING { OFF | ON }

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

This parameter logs all parameters for the GENERAL section and each specified DBS or target table section within the ETLCONF file. The parameters are logged when the first event for the DBS or target table appears. Parameters COLDELIM and RECDELIM are displayed in hex format (ex. 0x0A).

Flat File Parameters

FILEPREFIX <*character*>

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		A			NO

This parameter determines a one-character prefix when writing to a flat file on the NonStop. WARNING: When using multiple DBSes or target tables, a different FILEPREFIX and/or FILELOC should be specified for each DBS or target table to avoid running into file open errors.

FILELOC <*\node.$volume.subvolume*>

Initial	Changed

Version	Version	Values Default	Minimum	Maximum	Alter
7.000					NO

This parameter determines where flat files are written to. The default is the location of Consumer program.

ROLLCOUNT <*count*>

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		100000	0	2147483647	NO

This parameter determines how many records will be written to a flat file before rolling. Set this value to 0 to never roll on the number of records.

ROLLTIME <*time*>

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		68400	0	15372286728	NO

This parameter determines how old in seconds the flat file can be until rolling to a new one. The maximum time is equal to the maximum size of a signed long long. Set this value to 0 to never roll on time.

ROLLSIZE <mega*byte-size*>

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		2048	0	4096	NO

This parameter determines how large in megabytes the flat file can be before rolling to a new one. The default is 2048MB. Set this value to 0 to never roll on size.

FILECLOSEFREQ <*time*>

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		60	1	922337203685	NO

This parameter determines how much time in seconds must pass until the ETL Toolkit attempts to close flat files. If FILECLOSEFREQ expires and FILECLOSEDELAY had not yet been reached, then a timer is set again until all files are closed.

FILECLOSEDELAY <*time*>

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
7.000		60	1	922337203685	NO

This parameter determines how long in seconds a flat file will stay open since the last event. This value is checked every time the timer expires and depending on if FILECLOSEDELAY is reached or not, then the file is closed or the timer is started again.

FILETYPE { UNSTRUCTURED | ENTRY }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000		UNSTRUCTURED	UNSTRUCTURED	ENTRY		NO

This parameters determines what file type is created when writing to flat files. When set to UNSTRUCTURED the file type is unstructured. When set to ENTRY, the file type is entry-sequenced. Unstructured files will not have the ability to omit replayed data and abort data, unlike entry-sequenced files.

FILECODE { *code* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			0	0	32767	NO

This parameters determines what file code will be used with the created flat files. If not specified, the default value of 0 will be used.

INCLUDEHEADER { OFF | ON }

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

This parameter determines whether or not the first record written to the file is a record of the column names. OFF does not include the header record. ON includes the header record. This option will only work if OUTPUTFORMAT is set to CUSTOM.

Kafka Parameters

KAFKAPROCESSNAME { *process-name* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			$ULKFK			NO

This parameter determines the name of the uLinga for Kafka process. The default value is “ULKFK”. The ‘$’ must be included in this name.

KAFKAOPENNAME { *open-name* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter determines what the open name of this specific DBS or target table will be. This is used by the uLinga for Kafka process to specify what Kafka Topic to send the data to. The '#' should NOT be included in this name.

KAFKARESPONSE { OFF | ON }

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

This parameter determines if the ETL Toolkit is expecting a response from the target Kafka application. If this parameter is turned on, each event will be sent to the Kafka Cluster one at a time and will not process the next event until a response is received. WARNING: This can lead to slow processing if the target Kafka application takes time to process the data and send a response.

KAFKARUNFILE { *node.$volume.subvolume.file* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter determines what the name of the TACL macro file used to start uLinga for Kafka is called. The processes that are started in this file must have **IN** and **OUT** specified to a valid value (e.g. $ZHOME, $NULL, etc) or the uLinga for Kafka processes will not properly start.

MQ Parameters

MQDLL { *node.$volume.subvolume.file* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			MQIC			NO

This parameter is used to specify the full pathname of the *MQIC* client DLL needed from the MQ installation subvolume. The ETL Toolkit utilizes the MQ Client DLL for sending and receiving data from MQ. These functions and defines are dynamically loaded into the Toolkit and therefore the location is required of this DLL for the Toolkit to support MQ target. This parameter should be placed in the "GENERAL" section of the ETLCONF file.

MQQUEUE { *queue* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter is used to specify the MQ Queue name to send the data to.

MQQUEUEMANAGER { *queuemanager* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter is used to specify the MQ Queue Manager name that holds the MQ Queue you are sending the data to.

MQCONNECTION { *host(port)* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter is used to specify the connection name used to connect to the MQ Server. The format of this parameter should be *host (port)* where *host* is the IP or connection name of the system that the MQ Server is on and *port* is the port that the MQ Channel is listening to.

MQCHANNEL { *channel* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter is used to specify the MQ Channel name that will have the data from the ETL Toolkit sent to. This should match the name of a SVRCONN channel that is already defined as the Toolkit will connect to MQ via a CLNTCONN.

MQID { *id }*

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter is used to specify the MQ User ID for the connection into MQ. This parameter should be used in combination with MQPASS.

MQPASS { *password }*

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter is used to specify the MQ User ID password for the connection into MQ. This parameter should be used in combination with MQID.

MQCIPHERSPEC { *cipher-spec* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter is used to specify the CipherSpec that will be used for SSL/TLS connection to the MQ Queue. This should match the value set for SSLCIPH on the SVRCONN channel you are connecting to. For more details, refer to *IBM MQ Documentation* and the topic *Working with SSL or TLS on HP Integrity NonStop Server*.

MQKEYREPO { *key-repository* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter is used to specify the directory of the key repository. This directory is the one that stores digital certificates and certificate revocation information. For more details, refer to *IBM MQ Documentation* and the topic *Working with SSL or TLS on HP Integrity NonStop Server*.

MQCERTLABEL { *certificate-label* }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000						NO

This parameter is used to specify the certificate label that is being used by the MQ Queue Manager that is being connected to. This should match the value set for CERTLABL on the SVRCONN channel you are connecting to. For more details, refer to *IBM MQ for HPE NonStop Documentation* and the topic *TLS Channels*.

Example Configuration

This is an example configuration for the ETL Toolkit on HPE NonStop. This configuration is outputting to entry-sequenced flat files, in a csv format. Below is an example of the ETLCONF ini file.

```
[DBS-NAME]
```

```
OUTPUTMETHOD=FLATFILE
FILEPREFIX=X
FILETYPE=ENTRY
```

Messages generated by ETL Toolkit

All messages begin with an “ETL: “ prefix that is not shown below.

Error loading file <FILENAME>: <ERROR>

Cause: An attempt was made to load in a file.

Effect: The file load fails and the consumer abends.

Recovery: Ensure that the file being loaded is properly setup.

Invalid value for parameter <PARAMETER> = <PARAM-VALUE> Error setting <PARAMETER> parameter Invalid parameter set for <PARAMETER>: <PARAM-VALUE> Nonnumeric input for parameter <PARAMETER>

Cause: An attempt was made to set the parameter.

Effect: Parsing parameters fails and the consumer abends.

Recovery: Ensure that the parameter is properly being set with valid values.

<PARAMETER> set to <PARAM-VALUE> for <SECTION-NAME>

Cause: A parameter was set to this value for this section.

Effect: The ETL Toolkit continues to parse parameters until finished.

Recovery: N/A

Rounding up <PARAMETER> to <PARAM-VALUE> for binary format.

Cause: A parameter value was rounded up to properly support binary format.

Effect: The parameter value is rounded up and parsing is continued.

Recovery: N/A (This message will appear everytime Shadowbase is started unless the parameter value is one of 2, 4, or 8).

WARNING: <PARAMETER> is set to either CHARACTER or BINARY and OUTPUTFORMAT is set to JSON. This configuration will result in invalid JSON format. Turn <PARAMETER> OFF to ensure JSON format is valid

Cause: The parameter and OUTPUTFORMAT are in a configuration that would result in invalid JSON data.

Effect: Parameter parsing and processing will continue.

Recovery: Shutdown the consumer and make a change to the parameter or OUTPUTFORMAT.

PROCESS_GETINFO_ Error: <ERROR> Additional Info: <ERROR-DETAIL>

Cause: An attempt was made to get the full name of the consumer.

Effect: The read fails and a file-system error code is returned.

Recovery: Refer to the *Guardian Procedure Errors and Messages Manual* for the recovery procedure for the returned file-system error.

Unable to start uLinga for Kafka Process

Cause: ETL Toolkit was unable to start uLinga for Kafka process.

Effect: The consumer will abend.

Recovery: Ensure that the KAFKARUNFILE parameter is a valid TACL Macro that can be run by the consumer.

uLinga for Kafka Process started

uLinga for Kafka Process is assumed to have already been started

Cause: Started or assumed that the uLinga for Kafka process was already started.

Effect: ETL Toolkit processing continues.

Recovery: N/A

Error: <FUNCTION> returned <ERROR>

Cause: A function returned an error code that specified a failure.

Effect: The function fails.

Recovery: Refer to the *Guardian Procedure Errors and Messages Manual*, *HPE Shadowbase Programming Manual*, or C documentation for more information for the returned error.

Error: Interval with size <SIZE>

Cause: An SQL interval was encountered with an internal size that was not 2, 4, or 8 bytes.

Effect: The interval is skipped.

Recovery: Modify the data type so that it has a valid size.

Interval end <END> not valid

Cause: An SQL interval was encountered with an end unit of time that was invalid.

Effect: The interval is skipped.

Recovery: Modify the data type so that it has a valid end unit.

Interval start <START> not valid

Cause: An SQL interval was encountered with a start unit of time that was invalid.

Effect: The interval is skipped.

Recovery: Modify the data type so that it has a valid start unit.

Error: Illegal datetime bounds

Cause: Either the start or end value for the date datatype was invalid.

Effect: The field may be missing data.

Recovery: Modify the data type so that it has a valid range.

Error: Not all of <COL-NAME> read

Cause: A SQL datetime column was not completely read.

Effect: Not all of the column is converted to a string.

Recovery: Retry.

<COL-NAME>: Interval length 0 after being converted to string

Cause: An SQL interval attempted to be converted to a string, but the resulting string had a length of 0.

Effect: The interval does not appear in the flat file.

Recovery: Retry.

<COL-NAME>: Datatype not supported.

Cause: A date type was encountered that is not supported by the consumer.

Effect: The column is skipped.

Recovery: Modify the column so that it has a valid data type.

<COL-NAME>: Unknown datatype <DATATYPE>.

Cause: An unknown datatype was encountered.

Effect: The column is skipped.

Recovery: Modify the column so that it has a valid data type.

Escaping a <CHARACTER> caused a field to become larger than the buffer, abending.

Cause: The field would be larger than the buffer if the characeter was escaped.

Effect: The consumer abends.

Recovery: Disable escaping of the character which could be DELIMITERS, UNPRINTABLES, BOTH, or ALL

Unable to add data for column <COL-NAME>, added NULL instead

Cause: ETL Toolkit was unable to add the value as a number and then string so it will be added as a NULL.

Effect: ETL Toolkit continues processing.

Recovery: Ensure that no binary data is being used with JSON formatting. If this is the case, JSONHANDLEBINARY should be set to HEX.

Converted a value for column <COL-NAME> to a string

Cause: ETL Toolkit was unable to add the value as a number and then did so as a string.

Effect: ETL Toolkit continues processing.

Recovery: Ensure that data being used for specific column is valid for its data type.

SBGETBEFORECOLUMN Error <ERROR> when getting <COL-NAME> SBGETAFTERCOLUMN Error <ERROR> when getting <COL-NAME>

Cause: An error was returned by SBGETBEFORECOLUMN or SBGETAFTERCOLUMN.

Effect: The column is skipped.

Recovery: Refer to the *HPE Shadowbase Programming Manual* for more information about this error.

Error: Total source record length <LENGTH>

Cause: The total length of the “record” field is 0

Effect: The “record” field will be empty.

Recovery: Modify the parameters or source table so something is included even if every column is empty.

Unidentified event type: <EVENT-ID>

Cause: An unknown event id was encountered

Effect: ETL Toolkit returns control to the consumer.

Recovery: N/A

Unknown column beginning with <SBDATAPREFIX> found

Cause: A column starting with SHAD-, or SBDATAPREFIX, was found and was not a well known name column or SOURCE-NAME.

Effect: The column is ignored.

Recovery: Change the value of SBDATAPREFIX or the name of the column.

Error: Total source record length <LENGTH>

Cause: The total length of the “record” field is 0

Effect: The “record” field will be empty.

Recovery: Modify the parameters or source table so something is included even if every column is empty.

Error: SBGETMXSOURCE unable to find MXSOURCETABLE

Cause: SBGETMXSOURCE returned 0, indicating that the length of the source table is 0.

Effect: The SHAD-SOURCE-NAME column, if used, will be empty.

Recovery: Refer to the *HPE Shadowbase Programming Manual* for more information about this error.

Error: SBGETSOURCE unable to find real source file name

Cause: SBGETSOURCE returned 0, indicating that the length of the source file is 0.

Effect: The SHAD-SOURCE-NAME column, if used, will be empty.

Recovery: Refer to the *HPE Shadowbase Programming Manual* for more information about this error.

Failed to open <SECTION-NAME> flat file: <ERROR>

Cause: There was an error trying to open the file pointer.

Effect: The file is not written to.

Recovery: Make sure the given path is valid and the user has the ability to write to it.

Create Error <ERROR> when creating <FILE-NAME>

Cause: There was an error trying to create the given file using FILE_CREATE_.

Effect: The file is not created.

Recovery: Make sure the file is able to be created.

Failed to write to <DBS> flat file Error: <ERROR> Failed to write to <DBS> entry-sequence file. FILE_WRITE64_ returned <ERROR> Failed to remove close bracket to <DBS> flat file Error: <ERROR>

Cause: There was an error trying to write to the given file.

Effect: The file is not written to and the consumer abends.

Recovery: Make sure the user has the ability to write to the file.

Failed to close <SECTION-NAME> flat file: <ERROR>

Cause: There was an error trying to close the flat file.

Effect: The file is not closed properly.

Recovery: Make sure the given path is valid and the user has the ability to write to it.

<SECTION-NAME> file with prefix <FILE-PREFIX> closed

Cause: The timer has stopped for this specific section resulting in the file being closed.

Effect: The consumer waits for remaining opened files to be closed.

Recovery: N/A

All files closed

Cause: All files that were open with a timer have been closed.

Effect: ETL Toolkit processing continues.

Recovery: N/A

Attempt <NUM> of 3 trying to connect to IPCSERVER. Will try again after 18 seconds

Cause: The ETL Toolkit attempted to connect to the uLinga for Kafka IPCSERVER.

Effect: ETL Toolkit will retry connection after 18 seconds.

Recovery: Ensure that the uLinga for Kafka process is up and running.

Unable to connect to IPCSERVER. Abending

Cause: The ETL Toolkit was unable to connect to the uLinga for Kafka IPCSERVER.

Effect: The consumer will abend.

Recovery: Ensure that the uLinga for Kafka process is up and running.

Attempting to reconnect to IPCSERVER. Will try again aftere 18 seconds. If this message
continues, either bring down environement or check to see that IPCSERVER is running.

Cause: The ETL Toolkit lost connection with the uLinga for Kafka IPCSERVER.

Effect: The ETL Toolkit will attempt reconnection every 18 seconds.

Recovery: Ensure that the uLinga for Kafka process is up and running.

dlopen on <FILE> error: <ERROR>

Cause: The ETL Toolkit was unable to dynamically load in the file.

Effect: The consumer will abend.

Recovery: Ensure that the file exists and is in correct format, and/or permission is allowed. Refer to error message.

dlsym on <FUNCTION> in <FILE> error: <ERROR>

Cause: The ETL Toolkit was unable to dynamically load the function from the file.

Effect: The consumer will abend.

Recovery: Ensure that the the correct file is being read from. Refer to error message.

MQI <MQ-FUNCTION> [<COMP-CODE>] failure; Reason Code [<REASON-CODE>]; (Queue/QueueManager) [<(QUEUE/QUEUEMANAGER>]

Cause: The MQI function failed to perform its operation

Effect: The consumer will abend.

Recovery: Refer to reason code and IBM MQ Documentation to understand error.

Kafka Quickstart on NonStop

Outputting into a Kafka topic via the ETL Toolkit is done through the help of uLinga for Kafka. To achieve this, the ETL Toolkit must be configured as well as a few uLinga for Kafka objects. This quick start will go over what must be done to get the ETL Toolkit configured properly for Kafka output. It does not matter which configuration you do first as long as they are both configured before attempting to start the ETL Toolkit (if KAFKARUNFILE is not set, then the uLinga for Kafka objects should be running before starting the ETL Toolkit).

uLinga for Kafka Configuration

The configuration for uLinga for Kafka with the ETL Toolkit must have an IPCSERVER, and KAFKAPRODUCER configured (a KAFKACONSUMER should be configured if KAFKARESPONSE is set to ON). Below shows the minimum configuration requirements for an IPCSERVER and KAFKAPRODUCER. These configurations are being set through the *ConsoleCon* but they can also be set via *WebMan*. For more details, refer to the *uLinga for Kafka Reference Manual.*

```
ADD KAFKAPRODUCER                    PROD1,
    BOOTSTRAP_PRIMARY_HOST_REMOTE    10.1.10.100,
    BOOTSTRAP_PRIMARY_SERVICE_REMOTE 9092,
    TOPIC                            producer1


ADD IPCSERVER     IPCSER1,
    KAFKAPRODUCER PROD1,
    OPENNAME      KAFKA
```

KAFKAPRODUCER is the name of the KAFKAPRODUCER object.
BOOTSTRAP_PRIMARY_HOST_REMOTE specifies the DNS name or numeric address of the Kafka Broker you are attempting to connect to.
BOOTSTRAP_PRIMARY_SERVICE_REMOTE specifies the network service name, or decimal port number of the Kafka Broker you are attempting to connect to.
TOPIC is the name of the Kafka Topic you wish to send data to.

IPCSERVER is the name of the IPCSERVER object.
KAFKAPRODUCER is the name of the KAFKAPRODUCER you are linking with the IPCSERVER.
OPENNAME is a valid IPC open name that will be used to when sending datat to the IPCSERVER.

ETL Toolkit Configuration

The necessary parameters for Kafka output for the ETL Toolkit within the ETLCONF in file are OUTPUTMETHOD, KAFKAPROCESSNAME and KAFKAOPENNAME. Below is an example of the ETLCONF file.

```
[DBS-NAME]
OUTPUTMETHOD=KAFKA
KAFKAPROCESSNAME=$ULKFK
KAFKAOPENNAME=KAFKA
```

KAFKAPROCESSNAME must match the process name you give uLinga for Kafka when you start it.
KAFKAOPENNAME must match the OPENNAME you gave the IPCSERVER without the # symbol.

Starting uLinga for Kafka and ETL Toolkit

After these configurations are set, you are able to run the uLinga for Kafka process. This can be done through a TACL MACRO like the one below. uLinga for Kafka ships the below TACL MACRO, however if using this file in unision with the KAFKARUNFILE parameter then the sections in bold will have to be added. When specifying the name for the process, make sure it matches the KAFKAPROCESSNAME. Before doing this ensure that the Kafka Broker is up and running so that the uLinga for Kafka process is able to connect to it.

```
?TACL MACRO


== Stop any running uLinga for Kafka process

STOP $ULKFK


== Remove any existing XML control file - this file is
defined == in the uLinga for Kafka params file

PURGE ULKFKCFG


== Start uLinga for Kafka

== NOTE: uLinga for Kafka is shipped with a params file
named
== "PARAMS". It is recommended that you make a copy of it
named == "ULKFKPM" to match the value of the PARAMS_FILE
parameter in == the line below.

RUN ULKAFKA/NAME $ULKFK, NOWAIT, CPU 0, IN $NULL, OUT
$NULL/ --PARAMS_FILE ULKFKPM


== Configure uLinga for Kafka

RUN CCON /IN $NULL, OUT $NULL/; OBEY ULKFKOB
```

The process name ($ULKFK) should match KAFKAPROCESSNAME with the $ symbol.
The obey file (ULKFKOB) would be the file that was used to configure the different uLinga for Kafka objects.
Once the uLinga for Kafka process is running you are able to start the ETL Toolkit and send data to the Kafka Topic.

New and Modified Commands

New and Modified SET AUD Parameters

New and Modified SET COLL Parameters

SKIPALTFILEEVENTS { ON }
{ OFF }

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

Determines whether to replicate or skip all Enscribe alternate key file (alt file) events at the collector level. This applies only to SBCOLL. Defaults to OFF - meaning the collector will not automatically skip Enscribe alt file events and they will be passed to any consumers that have an appropriate SOURCEFILE setting.

When this feature is enabled, the collector may need to do a certain amount of audit trail pre-reading in order to ensure that any potential alt file events flushed out of order in the audit trail are detected and skipped properly. If the events being flushed to the audit trail are extremely out of order, it's possible that the collector will end up replicating some alt file events that should otherwise be skipped (however, a warning message will be logged).

A new section of the STATS COLL output will be included like so:

```
ALTFILE FILTERING INFO:
  NUM PREREADS:          2
  TOTAL ALTFILES SKIPPED:          2  ALTFILES REPLICATED:          0
  TOTAL PREREAD EVENTS:      50303  TOTAL TIME PREREADING: 00:01:30.319
```

`NUM PREREADS`: The number of times the collector has started pre-reading the audit trail to determine whether an event is for an alt file or not.
`TOTAL ALTFILES SKIPPED`: The total number of alt files that the collector has seen in the audit trail and is skipping.
`ALTFILES REPLICATED`: The total number of alt files that were flushed out of order to the extent that the collector's pre-read could not initially detect it, causing at least one event to be replicated before events started being skipped for the alt file.
`TOTAL PREREAD EVENTS`: The total number of events that the collector has processed during all of its pre-reads.
`TOTAL TIME PREREADING`: The total amount of time that the collector has spent doing pre-reads.

New and Modified SET CONS Parameters

ADTSTARTTIME { (start_date start_time) }
{ (-1) }

Initial Version	Changed Version	Values	Default	Value Specifications	Alter
2.000	7.000		None	-1 reset Date Time	No

The starting event time for the Collector to start processing from the audit trail files. Start_date is optional. If not entered, the current date is assumed. If start_date is entered it must be in the format MMM DD YYYY, where MMM are the first three characters of the month name (e.g., JUN), DD are the day digits and YYYY is the four-digit year; YYYY is optional and if omitted the current year is assumed (e.g., JUN 01 2006). If entered, start_time must be in the format of HH:MM:SS.ffffff. If it is omitted 00:00:00.0 is assumed. Entering -1 will clear the value previously set for the parameter. This is an example of setting this parameter: SET COLL ADTSTARTTIME (SEP 26 2005,09:13:00.0)

If a specific starting audit trail has been identified with an ADTSTARTNAME Collector parameter, the Collector will open and read this audit trail first. If the ADTSTARTTIME indicated is lower than the earliest time in the first audit trail read, the Collector will search the TM/MP catalog for the audit trails that contain the desired time. This is an optional parameter. This parameter cannot be used to position the collector for starting in a bi-directional replication environment.

Notes: This parameter will be ignored upon HPE Shadowbase startup if the file indicated by the RESTARTFILE parameter exists.

Note: This parameter will be ignored if the parameter ADTSTARTEOF is set to ON.

ADTSTARTTIME uses the TMF master (MAT) audit trail time stamps for positioning. On systems configured to use master (MAT) and auxiliary (AUX) audit trail files, the specified date and time values are significant only for the master audit trail. The TMF subsystem stores event data in the AUX file(s) and stores a pointer record in the MAT file following the successful AUX write(s). For example, there can be events in the AUX file that are timestamped between 07:30:01 and 07:30:59 , but the related AUX Pointer record in the MAT file may not have been written until 07:33:01. When this situation is the case and you set your starting ADTSTARTTIME to 07:32:00, these events will be processed (assuming other HPE Shadowbase parameters, such as the ADTAUXMASK, do not filter off these events). Similarly, if you set the ADTSTARTTIME to 07:29:00 and the ADTSTOPTIME to 07:32:00, these AUX events will not be processed even though they occurred during this interval (because their AUX pointer record does not occur within this interval).

ADTSTOPTIME { (stop_date, stop_time) }
{ (-1) }

Initial	Changed

Initial Version	Changed Version	Values	Default	Value Specifications		Alter
2.000	7.000		None	-1 reset	Date Time	Yes

The audit trail event time at which the Collector will stop reading from the audit trail files. Reading will stop when this time is reached. Stop_date is optional. If not entered, the current date is assumed. If stop_date is entered it must be in the format MMM DD YYYY, where MMM are the first three characters of the month name (e.g., JUN), DD are the day digits and YYYY is the four-digit year; YYYY is optional and if omitted the current year is assumed. If entered, stop_time must be in the format of HH:MM:SS.ffffff. If it is omitted 00:00:00.0 is assumed. Entering -1 will clear any value previously entered in the field. This is an optional parameter. If a stop file has been explicitly defined, this parameter will have no effect if the time of the last audit record read in the last file is less than ADTSTOPTIME.

> **Note**: Refer to the ADTSTARTTIME parameter for a discussion regarding the timestamps of the events in the MAT and AUX to determine the time range you selected will pick up the events from the audit trails.

ETLCONFIG { [\system.] [$volume.] [subvolume.] file }

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

This parameter specifies the name of the *.ini* file used to specifiy ETL Toolkit related parameters. Aside from ETLPROCESSING, all other parameters should be placed in this file with an .ini style format, using a GENERAL sections for default parameters and section names for DBSs or target files (depending on SECTIONHEADER parameter).

ETLPROCESSING { ON }
{ OFF }

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

This parameter determines if the Consumer will be operating under ETL Toolkit processing. Doing this will stop the Consumer from outputting changes to the target table and rather output into flat files, Kafka, or MQ as configured.

FILTEREMPTYTX { ON }
{ OFF }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
6.800	7.000		ON	n/a	n/a	NO

If set to ON, the consumer will not replicate begin, commit, or abort events over TCPIP for transactions where all events were ignored by calling SBSETIGNORE or failing the SBWHERE clause in SBMAP.

New and Modified SET DBS Parameters

New and Modified SET QMGR Parameters

LATENCYCLOCKADJ hundredths

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			0	-2147483647	2147483647	Yes

Defines a + or - differential to be used for calculating latency when the source and target system clocks are not synchronized. The value is subtracted from or added to the system clock before LATENCYTHRESHOLD is evaluated. The valid values are 2147483647 through 2147483647. Input a minus sign (-) for a negative adjustment; enter no sign for a positive adjustment. The default is 0.

LATENCYTHRESHOLD hundredths

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			0 Off	0	2147483647	Yes

This sets the tolerance for latency monitoring. The value is added following the processing of an audittrail event (i.e., replicated to target), the Queue Manager will add this to the associated event timestamp and compare the result to current time to see if the threshold has been exceeded. The default is 0 and means that Queue Manager latency monitoring is off. Valid values are 0 to 2147483647.

LATENCYWARNRATE seconds

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000			0	0	2147483647	Yes

Sets the rate at which EMS messages are generated when the Queue Manager has exceeded or recovers from exceeding the LATENCYTHRESHOLD. Valid values are 0 to 2147483647. Note that you should choose a reasonable value for this parameter for your operating environment. If the value is set too low, it is quite possible that the EMS log will become flooded with messages and impede HPE Shadowbase performance. While the default is 0; a value of 60 is recommended, when the parameter LATENCYTHRESHOLD is set to a value greater than 0, to avoid flooding EMS when a "behind" condition exists.

New and Modified SET SOLVMGR Parameters

SOLVCONFIGSECTIONFIRST section_name

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000		None		n/a	n/a	No

If set, the SOLVMGR will use this section for the first SOLV load instead of the one specified with SOLVCONFIGSECTION. If not set, SOLVCONFIGSECTION will be used on the first load.

STARTTIME { (start_date start_time) }
{ (-1) }

Initial Version	Changed Version	Values	Default	Value Specifications		Alter
5.000			None	-1 reset	Date Time	No

Is used by SOLVMGR to start processing on the file from the file set whose file's modified time occurs after this date and time. The file list is generated from the DBSs configured for the SOLVMGR. Start_date is optional. If not entered, the current date is assumed. If start_date is entered it must be in the format MMM DD YYYY, where MMM are the first three characters of the month name (e.g., JUN), DD are the day digits and YYYY is the four-digit year; YYYY is optional and if omitted the current year is assumed (e.g., JUN 01 2011). If entered, start_time must be in the format of HH:MM:SS.ffffff. If it is omitted 00:00:00.0.0 is assumed. Entering -1 will clear the value previously set for the parameter. This is an example of setting this parameter: SET SOLVMGR STARTTIME (SEP 26 2011,09:13:00.0)

Only one of STARTTIME or STARTFNAME can be specified. STARTTIME is ignored if restart (marker) records exist for the SOLVMGR.

If neither STARTTIME nor STARTFNAME are specified, SOLV loading will start with the file having the earliest modified timestamp. Essentially this means that all files in the file set will be loaded.

STOPTIME { (stop_date stop_time) }
{ (-1) }

Initial Version	Changed Version	Values	Default	Value Specifications		Alter
5.000			None	-1 reset	Date Time	No

Is used by SOLVMGR to stop processing on the file from the file set whose file's modified time occurs after this date and time. The file list is generated from the DBSs configured for the SOLVMGR. If SOLV is loading a particular file and

the modified time exceeds the STOPTIME, SOLVMGR will instruct SOLV to stop loading when EOF is reached.

Stop_date is optional. If not entered, the current date is assumed. If stop_date is entered it must be in the format MMM DD YYYY, where MMM are the first three characters of the month name (e.g., JUN), DD are the day digits and YYYY is the four-digit year; YYYY is optional and if omitted the current year is assumed (e.g., JUN 01 2011). If entered, stop_time must be in the format of HH:MM:SS.ffffff. If it is omitted 00:00:00.0 is assumed. Entering -1 will clear the value previously set for the parameter. This is an example of setting this parameter: SET SOLVMGR STOPTIME (SEP 26 2011,09:13:00.0)

Only one of STOPTIME or STOPFNAME can be specified.

If neither STARTTIME nor STARTFNAME are specified, SOLV loading will continue indefinitely.

New and Modified SOLV AND SOLVNV Parameters

VVITERSOURCEREFETCH { ON }
{ OFF }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000		ON				NO

This is an optional parameter. VVITERSOURCEREFETCH is used to determine if, during the final iteration of a run, SOLV will reread the source file for any records that are still different. This prevents changes to the source file that occur during the compare job from appearing as mismatches.

New or Modified EMS Messages

4556	STARTED WRITING TO QMGR QUEUE FILE <file name>, BLOCK FIRST EVENT POSITION <pos>, BLOCK FIRST EVENT TIME <time>
4556	CONTINUED WRITING TO QMGR QUEUE FILE <file name>, BLOCK FIRST EVENT POSITION <pos>, BLOCK FIRST EVENT TIME <time>

Cause: The Queue Manager started writing (or continued in the case of a restart) to the specified queue file with an event at the audit trail position and timestamp listed.

Effect: Informational only.

Recovery: None needed.

2017	FILEMNT: FAILED TO PURGE FILE <file-name> ERROR <error-num>, STOPPING. TURN SKIPFATALDDLERR ON TO CONTINUE INSTEAD.

Cause: Informational. The purge event did not complete successfully. SKIPFATALDDLERR was OFF, so the Consumer stops.

Effect: The file was not completely purged. Previous EMS messages detail the errors. The Consumer stops.

Recovery: Manually purge the target file and restart the Consumer.

1432	TACL PARAMETER SBCOLLMULTIENDS NOT SUPPORTED WITH CONSUMER PARAMETER FILTEREMPTYTX FOR <consumer-pname>

Cause: The Collector attempted to start add a thread with the consumer parameter FILTEREMPTYTX enabled, while the TACL parameter SBCOLLMULTIENDS is also enabled.

Effect: The Collector stops.

Recovery: Disable SBCOLLMULTIENDS or FILTEREMPTYTX in all connected Consumers and restart the Collector.

2017	FILEMNT: FAILED TO CREATE FILE <file-name> ERROR <error-num>, STOPPING. TURN SKIPFATALDDLERR ON TO CONTINUE INSTEAD.

Cause: Informational. The create event did not complete successfully. SKIPFATALDDLERR was OFF, so the Consumer stops.

Effect: The file was not completely created. Previous EMS messages detail the errors. The Consumer stops.

Recovery: Manually correct the problem to make the target file match the source file.

2017	FILEMNT: FAILED TO PURGEDATA FILE <filename> ERROR <error number>, STOPPING. TURN SKIPFATALDDLERR ON TO CONTINUE INSTEAD.

Cause: The Consumer failed to purgedata the specified file. This EMS message is critical, and is reported for EMS event number 2017 (PROCESS_ABEND_EVENT).

Effect: The Consumer abends.

Recovery: See the *Guardian Procedure Errors and Messages Manual* for the recovery procedure for the returned file-system error.

2017	FILEMNT: FAILED TO ALTER FILE <filename> ERROR <error number>, STOPPING. TURN SKIPFATALDDLERR ON TO CONTINUE INSTEAD.

Cause: The Consumer failed to alter the specified file. This EMS message is critical, and is reported for EMS event number 2017 (PROCESS_ABEND_EVENT).

Effect: The Consumer abends.

Recovery: See the *Guardian Procedure Errors and Messages Manual* for the recovery procedure for the returned file-system error.

2095	FILEMNT: FAILED TO PURGE FILE <file-name> ERROR <error-num>, CONTINUING. TURN SKIPFATALDDLERR OFF TO STOP INSTEAD.

Cause: Informational. The purge event did not complete successfully. SKIPFATALDDLERR was ON, so the Consumer continues to process events.

Effect: The file was not completely purged. Previous EMS messages detail the errors. The Consumer continues.

Recovery: Manually purge the target file.

2095	FILEMNT: FAILED TO CREATE FILE <file-name> ERROR <error-num>, CONTINUING. TURN SKIPFATALDDLERR OFF TO STOP INSTEAD.

Cause: Informational. The create event did not complete successfully. SKIPFATALDDLERR was ON, so the Consumer continues to process events.

Effect: The file was not completely created. Previous EMS messages detail the errors. The Consumer continues.

Recovery: Manually correct the problem to make the target file match the source file.

2095	FILEMNT: FAILED TO ALTER FILE <file-name> ERROR <error-num>, CONTINUING. TURN SKIPFATALDDLERR OFF TO STOP INSTEAD.

Cause: Informational. The alter event did not complete successfully. SKIPFATALDDLERR was ON, so the Consumer continues to process events.

Effect: One or more attributes of the target file were not altered to match the source. Previous EMS messages detail the errors. The Consumer continues.

Recovery: Manually alter the target file to match the source.

2095	FILEMNT: FAILED TO PURGEDATA FILE <filename> ERROR <error number>, CONTINUING. TURN SKIPFATALDDLERR OFF TO STOP INSTEAD.

Cause: The Consumer failed to purgedata the specified file. This EMS message is informative and is reported for EMS event number 2095(FILEMNT_INFO_EVENT).

Effect: The event is ignored and the Consumer continues processing.

Recovery: See the *Guardian Procedure Errors and Messages Manual* for the recovery procedure for the returned file-system error.

2096	SBBASE24: SBGETBEFORERECORD - UNEXPECTED ERROR (<error-num>) RETURNED FOR <file-name>

Cause: A call to SBGETBEFORERECORD returned an unexpected error.
Effect: The handler will ignore the event and continue.
Recovery: Depends on the error returned from the call to SBGETBEFORERECORD. Check the HPE Shadowbase NonStop Programmers Manual for return codes.

2096	SBBASE24: SBGETAFTERRECORD - UNEXPECTED ERROR (<error-num>) RETURNED FOR <file-name>

Cause: A call to SBGETAFTERRECORD returned an unexpected error.
Effect: The handler will ignore the event and continue.
Recovery: Depends on the error returned from the call to SBGETAFTERRECORD. Check the HPE Shadowbase NonStop Programmers Manual for return codes.

2096	SBBASE24: ERROR (<error-num>) RETURNED FROM FILE_SETKEY_ ON <filename>, SOURCE RECORD ADDRESS <source-record-address>

Cause: A relative file position failed in the relative file BASE24 handler.

Effect:	The IO event that caused the problem is ignored. The Consumer continues processing.
Recovery:	Take appropriate action depending on the file system error returned from the call to file_setkey_.

2096	SBBASE24: AWAITIOX ERROR <error-num> on <target-filename> FOR KEY <alt-key-value>, SOURCE RECORD ADDRESS <source-record-address>

Cause:	After attempting to read the relative file, an await failed.
Effect:	The IO event that caused the problem is ignored. The Consumer continues processing.
Recovery:	Take appropriate action depending on the file system error returned from the call to awaitiox

2096	SBBASE24: <file-name> (SOURCE) AND <file-name> (TARGET) FILES MUST BE ENTRY SEQUENCED ENSCRIBE FILES FOR DBS <dbs-name>

Cause:	File types are not entry sequenced as expected.
Effect:	The event is ignored and the Consumer continues.
Recovery:	None

2096	SBBASE24: <file-name> (SOURCE) AND <file-name> (TARGET) FILES MUST BE RELATIVE ENSCRIBE FILES FOR DBS <dbs-name>

Cause:	File types of the source and target files are not relative as expected.
Effect:	The event is ignored and the Consumer continues.
Recovery:	None.

2017	SBBASE24: FILE_GETINFOLISTBYNAME_ FAILED ON <file-name>, ERROR <error-number>

Cause:	An attribute of the source file could not be retrieved due to the specified file system error.

Effect:	The Consumer abends.
Recovery:	Use the file system error number in the message to determine and correct the problem

2017	SBBASE24: FILE_GETINFOLISTBYNAME_ FAILED ON <file-name>, ERROR <error-number>, ITEM CODE <item-code> ITEM INDEX <item-index>

Cause:	An attribute of the source file could not be retrieved due to the specified file system error.
Effect:	The Consumer abends.
Recovery:	Use the file system error number in the message to determine and correct the problem.

2017	SBBASE24: ERROR, COULD NOT FIND <key-specifier> KEY SPECIFIER FOR FILE <file-name> IN DBS <dbs-name>

Cause:	Could not find a required alternate key for the listed file.
Effect:	The Consumer abends.
Recovery:	The handler requires the specified key specifier to exist. Verify that the DBS KEYSPECIFIER parameter is set to the correct key specifier.

2017	SBBASE24: ERROR, <key-specifier> KEY SPECIFIER FOR FILE <file-name> DBS <dbs-name> IS NOT UNIQUE

Cause:	The specified alternate key was not unique. The specified key must be unique for PTDF type files.
Effect:	Consumer abends.
Recovery:	Specify a unique alternate key in the DBS for the file.

2017	SBBASE24: INTERNAL ERROR, COULD NOT FIND FILELIST INFO FOR DBS <dbs-name>

Cause:	Unexpected internal BASE24 handling error.

Effect:	Consumer fails.
Recovery:	Contact HPE Shadowbase Support. If available, a saveabend file will be requested.

2017	SBBASE24: FILE_OPEN_ ERROR <error-num> ON <file-name>

Cause:	A BASE24 handler could not successfully open a file.
Effect:	Consumer fails
Recovery:	Determine the cause of the FILE_OPEN error. Error-num is a system file error.

2017	SBBASE24: DURING THE PROCESSING OF AN UPDATE FOR <source-filename> WITH ADDRESS <source-relative-address> AN UNEXPECTED INSERT SET-A-SIDE EVENT WAS FOUND

Cause:	An unexpected condition occurred while process an update on the set-a-side list.
Effect:	Fatal error.
Recovery:	Contact HPE Shadowbase support for assistance.

2017	SBBASE24: INVALID HANDLER ID (<handler-id-number>) ENCOUNTERED FOR DBS <dbs-name>

Cause:	The specified handler id was not one of the allowed values.
Effect:	Consumer abends.
Recovery:	Change the handler id to one of the allowed values, and restart.

2017	SBBASE24: ERROR, DBS KEYSPECIFIER PARAMETER REQUIRED FOR BASE24HANDLERID <handler-id-number>, ENCOUNTERED FOR DBS <dbs-name>

Cause:	The DBS parameter KEYSPECIFIER is not set.
Effect:	Consumer abends.
Recovery:	Set the KEYSPECIFIER parameter and restart.

2096	SBBASE24: SBGETBEFORERECORD UNEXPECTED ERROR (<error-num>) RETURNED FOR <file-name>, SOURCE RECORD ADDRESS <source-record-address>

Cause:	A call to SBGETBEFORERECORD returned an unexpected error.
Effect:	The handler will ignore the event and continue.
Recovery:	Depends on the error returned from the call to SBGETBEFORERECORD. Check the HPE Shadowbase NonStop Programmers Manual for return codes.

2096	SBBASE24: SBGETAFTERRECORD UNEXPECTED ERROR (<error-num>) RETURNED FOR <file-name>, SOURCE RECORD ADDRESS <source-record-address>

Cause:	A call to SBGETAFTERRECORD returned an unexpected error.
Effect:	The handler will ignore the event and continue.
Recovery:	Depends on the error returned from the call to SBGETAFTERRECORD. Check the HPE Shadowbase NonStop Programmers Manual for return codes.

2096	SBBASE24: WRITEX TO TARGET <file-name> SUCCESSFUL, ORIGINAL EVENT AS AN UPDATE

Cause:	Informational.
Effect:	None.
Recovery:	None.

2096	SBBASE24: SBGETBEFORERECORD UNEXPECTED ERROR (<err-num>) RETURNED FOR <file-name>, COULD NOT DUMP VALUE OF ALT KEY FOR PREVIOUS MESSAGE

Cause:	A call to SBGETBEFORERECORD failed due to the displayed error. Check the HPE Shadowbase Programming Manual for a detailed description of the error returned for the specific procedure call. The Consumer was trying to display the altkey value for the source record address that can be found in the in the previous EMS message.
Effect:	The Consumer continues and ignores the IO associated with the Consumer's previous message.
Recovery:	Use the source record address in the previous message to verify that the source record is identical to the target. Any corrections will need to be applied manually. Then send the message and error number to HPE Shadowbase support.

2096	SBBASE24: SBGETAFTERRECORD UNEXPECTED ERROR (<err-num>) RETURNED FOR <file-name>, COULD NOT DUMP VALUE OF ALT KEY FOR PREVIOUS MESSAGE

Cause:	A call to SBGETAFTERRECORD failed due to the displayed error. Check the HPE Shadowbase Programming Manual for a detailed description of the error returned for the specific procedure call. The Consumer was trying to display the altkey value for the source record address that can be found in the in the previous EMS message.
Effect:	The Consumer continues and ignores the IO associated with the Consumer's previous message.
Recovery:	Use the source record address in the previous message to verify that the source record is identical to the target. Any corrections will need to be applied manually. Then send the message and error number to HPE Shadowbase support.

2096	SBBASE24: USED SBGETBEFORERECORD at ADDRESS <source-record-address> OF <source-file-name>(SOURCE) TO GET TARGET POSITION FOR <target-file-name>(TARGET) AT ADDRESS <target-record-address>

Cause:	Informational message only when BASE24 message level is set to 2.
Effect:	None.
Recovery:	None.

2096	SBBASE24: USED SBGETAFTERRECORD at ADDRESS <source-record-address> OF <source-file-name>(SOURCE) TO GET TARGET POSITION FOR <target-file-name>(TARGET) AT ADDRESS <target-record-address>

Cause:	Informational message only when BASE24 message level is set to 2.
Effect:	None.
Recovery:	None.

2096	SBBASE24: UNEXPECTED UPDATE EVENT ENCOUNTERED FOR A NON-HEADER RECORD AT ADDRESS <address> - <source-file> (SOURCE) - <target-file>(TARGET) - DBS <dbs-name> - IGNORING EVENT AND CONTINUING

Cause:	An updated of a non-header record (a record whose address is not equal to zero) was encountered.
Effect:	The event is ignored and the Consumer continues.
Recovery:	None.

2096	SBBASE24: COULD NOT OVERRIDE THE HEADER EVENT (RECORD ADDRESS ZERO) FROM AN UPDATE TO AN INSERT <file-name> (SOURCE) - <file-name> (TARGET) - DBS <dbs-name> - SBSETSTATEMENTTYPE ERROR <sb-return-code> - IGNORING EVENT AND CONTINUING

Cause:	A call to SBSETSTATEMENTTYPE failed with the given HPE Shadowbase error number for an entry sequenced file.
Effect:	The event is ignored and the Consumer continues.
Recovery:	None.

2096	SBBASE24: REMOVING EVENT FROM THE RELATIVE FILE SET-A-SIDE LIST DUE TO ABORT EVENT, SOURCE FILE <file-name> AND RECORD ADDRESS <source-record-address>

Cause: Informational.
Effect: None.
Recovery: None.

4003	ERROR <error-code> CALLING PROCEDURE OLDFILENAME_TO_FILENAME_ WHILE CONVERTING INFILE <value> LOADED FROM STARTUP MESSAGE FROM OLDFILENAME FORMAT. VALID IN AND OUT SETTINGS ARE REQUIRED. NOTE THAT IF CONFIGURING IN SCF, THESE VALUES MUST BE SUPPLIED.
4003	ERROR <error-code> CALLING PROCEDURE OLDFILENAME_TO_FILENAME_ WHILE CONVERTING OUTFILE <value> LOADED FROM STARTUP MESSAGE FROM OLDFILENAME FORMAT. VALID IN AND OUT SETTINGS ARE REQUIRED. NOTE THAT IF CONFIGURING IN SCF, THESE VALUES MUST BE SUPPLIED.

Cause: The program was unable to convert a valid filename for the IN or OUT filename (as specified in the EMS message) loaded from the startup message. Valid filenames are required for these, and can be set explicitly by specifying the IN and OUT options when starting the process, or configuring it in SCF.
Effect: The process is unable to load the information needed from the startup message, and the related operation may fail.
Recovery: Correct or supply the IN or OUT file name when starting the process or configuring it in SCF, then restart the process and retry the operation.

2017	ENTERING DRAIN MODE DUE TO PERIODIC AUDMON CHECK (AUDMON MAY BE IN AN UNKNOWN STATE): <consumer name>(<consumer process>)

Cause: The consumer detects that AUDMON is either stopped or inaccessible. The consumer runs this check every 5 minutes. The consumer will enter drain mode if still attached to a QMGR and not paused.

Effect: The consumer goes into drain mode to finish processing any events out of the QMGR and shut down when complete.

Recovery: Restart the replication environment.

2017	ENTERING DRAIN MODE DUE TO PROCESS_CLOSE MSG RECEIVED FROM AUDMON (AUDMON MAY BE IN AN UNKNOWN STATE): <consumer name>(<consumer process>)

Cause: The consumer received a PROCESS CLOSE message from AUDMON after AUDMON has stopped running. The consumer will enter drain mode if still attached to a QMGR and not paused.

Effect: The consumer goes into drain mode to finish processing any events out of the QMGR and shut down when complete.

Recovery: Restart the replication environment.

2017	ENTERING ABNORMALSHUTDOWN DRAIN MODE DUE TO LOCAL CPU DOWN (AUDMON MAY BE IN AN UNKNOWN STATE): <consumer name>(<consumer process>)

Cause: The consumer detects that AUDMON's CPU has gone down. The consumer will enter drain mode if still attached to a QMGR and not paused.

Effect: The consumer goes into drain mode to finish processing any events out of the QMGR and shut down when complete.

Recovery: Restart the replication environment.

2017	ENTERING ABNORMALSHUTDOWN DRAIN MODE DUE TO REMOTE CPU DOWN (AUDMON MAY BE IN AN UNKNOWN STATE): <consumer name>(<consumer process>)

Cause: The consumer detects that AUDMON's CPU has gone down. The consumer will enter drain mode if still attached to a QMGR and not paused.

Effect: The consumer goes into drain mode to finish processing any events out of the QMGR and shut down when complete.

Recovery: Restart the replication environment.

2017	ENTERING ABNORMALSHUTDOWN DRAIN MODE DUE TO REMOTE COMM DOWN (AUDMON MAY BE IN AN UNKNOWN STATE): <consumer name>(<consumer process>)

Cause: The consumer has lost connection with AUDMON's node. The consumer will enter drain mode if still attached to a QMGR and not paused.

Effect: The consumer goes into drain mode to finish processing any events out of the QMGR and shut down when complete.

Recovery: Restart the replication environment.

2017	CONSUMER PROCESS STOPPED DUE TO PROCESS CLOSE (AUDMON MAY BE IN AN UNKNOWN STATE) <consumer name>(<consumer process>)

Cause: The consumer received a PROCESS CLOSE message from AUDMON after AUDMON has stopped running, but did not enter drain mode.

Effect: The consumer shuts down.

Recovery: Restart the replication environment.

2017	CONSUMER PROCESS STOPPED DUE TO LOCAL CPU FAILURE (AUDMON MAY BE IN AN UNKNOWN STATE) <consumer name>(<consumer process>)

Cause: The consumer detects that AUDMON's CPU has gone down, but did not enter drain mode.

Effect: The consumer shuts down.

Recovery: Restart the replication environment.

2017	CONSUMER PROCESS STOPPED DUE TO REMOTE CPU FAILURE (AUDMON MAY BE IN AN UNKNOWN STATE) <consumer name>(<consumer process>)

Cause: The consumer detects that AUDMON's CPU has gone down, but did not enter drain mode.

Effect: The consumer shuts down.

Recovery: Restart the replication environment.

2017	CONSUMER PROCESS STOPPED DUE TO REMOTE COMMUNICATION FAILURE (AUDMON MAY BE IN AN UNKNOWN STATE) <consumer name>(<consumer process>)

Cause: The consumer has lost connection with AUDMON's node, but did not enter drain mode.

Effect: The consumer shuts down.

Recovery: Restart the replication environment.

2017	CONSUMER PROCESS STOPPED DUE TO PERIODIC AUDMON CHECK (AUDMON MAY BE IN AN UNKNOWN STATE) <consumer name>(<consumer process>)

Cause: The consumer detects that AUDMON is either stopped or inaccessible, but did not enter drain mode. The consumer runs this check every 5 minutes.

Effect: The consumer shuts down.

Recovery: Restart the replication environment.

2137	NSK DATA MAPPING IS DEPRECATED, CONSIDER USING SBMAP INSTEAD

Cause: Informational.
Effect: None.
Recovery: None.

2019	FILE_GETINFOLISTBYNAME_ RETURNED ERROR <error> WHILE ATTEMPTING TO CHECK DCOMPRESS ATTRIBUTE OF FILE <filename> FOR MAYBE COMPRESSED AUDIT TRAIL IMAGE

Cause: A file system error occurred when attempting to check the DCOMPRESS attribute of the file or table listed.
Effect: The check fails. A following EMS message will indicate the action taken by the Collector program as a result.
Recovery: Check the follow-up EMS from the Collector program to determine the impact of the error, and whether recovery action is needed.

2019	UNABLE TO DETERMINE DCOMPRESS ATTRIBUTE OF FILE <file>, ASSUMING IT HAD DCOMPRESS ENABLED AND CONTINUING WITH COMPRESSION LENGTH BYTE REPLACEMENT PROCESSING

Cause:	Attempting to fetch the after image of a file or table returned an ARW_MAYBE_COMPRESSED warning, causing the Collector program to need to check the datafile to see if the DCOMPRESS attribute is set. The attempt to check the DCOMPRESS attribute failed, due to a file system error which will be logged in the prior message from the Collector process.
Effect:	Informational. The Collector continues by assuming that the event contained compression, and performing the requisite swap of the first byte of the image to handle that.
Recovery:	None needed. However, it is recommended to review the file system error in question, to see if it is expected or if there may be error-dependent action desired to resolve it.

2017	UNABLE TO DETERMINE DCOMPRESS ATTRIBUTE OF FILE <file>, ABENDING DUE TO CONFGIURED SETTING OF SBCOLLDCOMPCHECKLEVEL

Cause:	Attempting to fetch the after image of a file or table returned an ARW_MAYBE_COMPRESSED warning, causing the Collector program to need to check the datafile to see if the DCOMPRESS attribute is set. The attempt to check the DCOMPRESS attribute failed, due to a file system error which will be logged in the prior message from the Collector process.
Effect:	The Collector process abends.
Recovery:	Take appropriate action based on the specific file system error reported in the prior EMS message from the Collector process. If resolution of the file system error is not feasible and the file or table was known to have the DCOMPRESS attribute set, the SBCOLLDCOMPCHECKLEVEL param value may be changed to 2 to tell the Collector to proceed with handling by assuming the file or table has the DCOMPRESS attribute set.

4952	QMGR QFILE <file_name> FIRST EVENT TIME: <event_time>; FIRST EVENT MAT ADDRESS: <event_pos>

Cause:	Informational.
Effect:	None.
Recovery:	None.

4953	QMGR QFILE <file_name> LAST EVENT TIME: <event_time>; LAST EVENT MAT ADDRESS: <event_pos>

Cause: Informational.
Effect: None.
Recovery: None.

2002	COLL <name>(<process>): COLLECTOR STARTING ADT OR ARC <sequence num> AT EOF (APPROX EOF=<eof>)

Cause: The Collector was started and has begun reading audit at end of file, at approximately the specified location. Note that there could be some variance due to additional data being written between when the Collector checked the audit trail EOF, and when the Collector started reading.

Effect: None.

Recovery: Informational only.

New and Modified SOLV User/EMS Messages

New and Modified TACL Parameters

SBCOLLMAXFNAMES

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
3.964A	7.000		1000000	1	2147483647	No

As part of the performance enhancements made in release 3.964A, by default, the AUDCOLLN Collector supported up to 131,070 unique file names on the Filename Prescreen search list. This default value was increased to 1,000,000 in version 7.000. You can increase or decrease this limit by entering the following TACL command prior to starting AUDMON.

```
PARAM SBCOLLMAXFNAMES <number>
```

Note that this parameter has no effect on the SBCOLL Collector program, which uses a different approach to managing its file information search list.

SBCOLLMSFASTSAMPLE

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
7.000		1 if ZDLENABLED ON 0 Otherwise		0	1	No

This parameter allows for explicitly changing the behavior of the SBCOLL collector program when FASTSAMPLE is ON. Normally, FASTSAMPLE will causes ADTSAMPLEDELAY and TURBOWAITTIME to be in either centi-seconds (if ZDLENABLED is OFF), or milli-seconds (if ZDLENABLED is ON). To use centi-seconds in all cases for FASTSAMPLE ON affected parameters, specify the following:

```
PARAM SBCOLLMSFASTSAMPLE 0
```

Alternatively, to use milli-seconds in all cases for FASTSAMPLE ON affected parameters, specify the following:

```
PARAM SBCOLLMSFASTSAMPLE 1
```

Note that SBCOLLMSFASTSAMPLE has no effect for the AUDCOLLN collector program.

New and Modified User Exit Functions

SBGETCURRCOLUMN

The SBGETCURRCOLUMN function is used to return the value that the consumer will use for a field or column. If the field or column has previously been modified by the consumer as part of this event, the most recent value will be returned. Otherwise, the source value will be retrieved.

Product	Availability	Calling Module Usage
HPE NonStop Server	Yes	USRXPROCESS USRXEXCEPTION
Other Servers	No	N/A

Syntax for COBOL Programmers

```
ENTER C "SBGETCURRCOLUMN" IN USRXLIB.
    USING COL, IN-WHERE, STORE-TYPE, LEN, VAL
    GIVING RETURN-CODE.
```

Syntax for C Programmers

```
short SBGETCURRCOLUMN(char *col,
                      short *in_where,
                      short *store_type,
                      short *len,
                      char *val)
```

Parameters

RETURN-CODE returned value

- Indicates whether call completed successfully or encountered an error:
  -5 = partial source field found
  -4 = column name not found in schema (without overriding target)
  -3 = bad column name format
  -2 = Schema not available
  -1 = Not available
  0 = found null value (for SQL sources)
  1 = value returned
- If a column for a SQL target table has a null value, the length returned is set to -1 and 0 is returned.

COL input

- Null terminated
- Value in DDL or SQL format
- Is the name of the field in the DDL record definition defined with the DBS TARGETREC parameter or the SQL column name.

IN_WHERE output

- Indicates whether the field/column value returned is part of the where clause: 0 = column not part of where clause, 1 = column is part of where clause

STORE_TYPE output

- Storage type (internal or external) of the value being returned.
  0 = column is in internal format & is present in target schema
  1 = value is an external string (quotes will be added to value)
  2 = value is an external non-string value (no quotes will be added),
- See SBPUTCOLUMNOPEN for more information.

LEN output

- Length of the field/column value returned

VAL output

- Data type must match that of field/column sought
- Size must be large enough to hold maximum returned value

Considerations

- Will return values put by SBPUTCOLUMN, SBPUTCOLUMNOPEN, SBPUTWHERECOLUMN, and SBPUTWHERECOLUMNOPEN.
- For Enscribe target files, a DBS TARGETREC DDL definition must be configured.
- Field or column name can be in DDL (e.g., ABC-DEF) or SQL (e.g., ABC_DEF) format.
- Make sure that the data area for the value returned is large enough to hold the entire contents of the returned column and that it is the correct data type.
- For insert and update events, the "after image" of the field/column will be returned if no put was previously done for it. See considerations for SBGETAFTERCOLUMN.
- For delete events, the "before image" of the field/column will be returned if no put was previously done for it. See considerations for SBGETBEFORECOLUMN.
- IN_WHERE will always return 0 for insert events.
- STORE_TYPE will always return 0 when returning values from the source image or from SBPUTCOLUMN/SBPUTWHERECOLUMN.

SBGETCURRRECORD

The SBGETCURRRECORD function is used to return the value that the consumer will use for the Enscribe record. If the record has previously been modified by the consumer as part of this event, the most recent value will be returned. Otherwise, the source value will be retrieved.

Product	Availability	Calling Module Usage
HPE NonStop Server	Yes	USRXPROCESS USRXEXCEPTION
Other Servers	No	N/A

Syntax for COBOL Programmers

```
ENTER C "SBGETCURRRECORD" IN USRXLIB.
    USING LEN, VAL
    GIVING RETURN-CODE.
```

Syntax for C Programmers

```
short SBGETCURRRECORD(short *len, char *val)
```

Parameters

RETURN-CODE returned value

- Indicates whether call completed successfully or encountered an error:
  -1 = Not available
  1 = record returned

LEN output

- Length of the record returned

VAL output

- Record return by HPE Shadowbase

Considerations

- Will return values put by SBPUTRECORD.
- Make sure that the data area for the returned record is large enough.
- For insert and update events, the after image of the Enscribe record will be returned if not put was previously done for it.
- For delete events, the before image of the Enscribe record will be returned if not put was previously done for it.
- SBGETCURRRECORD can only be used for a direct environment.
- Enscribe Only

SBGETCURRRELATIVEADDR

The SBGETCURRRELATIVEADDR function is used to retrieve the relative record number of the current record from an Enscribe file. If the record number has previously been modified by the consumer as part of this event, the most recent value will be returned.

Product	Availability	Calling Module Usage
HPE NonStop Server	Yes	USRXPROCESS USRXEXCEPTION
Other Servers	No	N/A

Syntax for COBOL Programmers

```
ENTER C "SBGETCURRRELATIVEADDR" IN USRXLIB.
    GIVING RETURN-CODE.
```

Syntax for C Programmers

```
long SBGETCURRRELATIVEADDR()
```

Parameters

RETURN-CODE returned value

- Indicates whether call completed successfully or encountered an error:
  -3 = Wrong API used for format 2 file.
  -2 = Wrong API used for format 1 Enscribe entry-sequenced file.
  -1 = Not available
  >-1 = relative record number

Considerations

- Will return values put by SBPUTRELATIVEADDR and SBPUTRELATIVEADDR64.
- Relative record addresses returned by the "SBGET..." API's are zero-relative. Note that if one is using this address to do COBOL run-time library I/O, 1 needs to be added to the Shadowbase returned address since COBOL record addresses are relative to 1.
- Enscribe Only
- Can only be used for Format 1 Enscribe files. Use SBGETRELATIVEADDR64 for Format 2 Enscribe files.
- SBGETRELATIVEADDR64 must be used for format 1 Enscribe entry-sequenced files.

SBGETCURRRELATIVEADDR64

The SBGETCURRRELATIVEADDR64 function is used to retrieve the relative record number or SYSKEY of the current record from a Format 2 (“big”) Enscribe file. If the record number has previously been modified by the consumer as part of this event, the most recent value will be returned.

Product	Availability	Calling Module Usage
HPE NonStop Server	Yes	USRXPROCESS USRXEXCEPTION
Other Servers	No	N/A

Syntax for COBOL Programmers

```
```

```
ENTER C "SBGETCURRRELATIVEADDR64" IN USRXLIB.
    GIVING RETURN-CODE.
```

Syntax for C Programmers

```
Long long SBGETCURRRELATIVEADDR64()
```

Parameters

RETURN-CODE returned value

- Indicates whether call completed successfully or encountered an error:
  -1 = Not available
  >-1 = relative record number

Considerations

- Will return values put by SBPUTRELATIVEADDR and SBPUTRELATIVEADDR64.
- Relative record addresses returned by the "SBGET..." API's are zero-relative. Note that if one is using this address to do COBOL run-time library I/O, 1 needs to be added to the Shadowbase returned address since COBOL record addresses are relative to 1.
- Enscribe Only

SBPROCESSAGAIN

The SBPROCESSAGAIN function is used to indicate to the consumer that it should process this event for this target an additional time.

Product	Availability	Calling Module Usage
HPE NonStop Server	Yes	USRXPROCESS USRXEXCEPTION
Other Servers	No	N/A

Syntax for COBOL Programmers

```
ENTER C "SBPROCESSAGAIN" IN USRXLIB.
    GIVING RETURN-CODE.
```

Syntax for C Programmers

```
long SBPROCESSAGAIN()
```

Parameters

RETURN-CODE returned value

- Indicates whether call completed successfully or encountered an error:
  -3 = Event processed max number of times
  -2 = Not supported with current DBS configuration
  -1 = Event type not supported
  0 = This event will be processed an additional time

Considerations

- As of 7.000, only supported with an Enscribe source and when replicating to an Other Servers environment using an Enscribe dummy target.

SBPROCESSITER

The SBPROCESSITER function is used to get the number of times that this event has been processed again by calling SBPROCESSAGAIN

Product	Availability	Calling Module Usage
HPE NonStop Server	Yes	USRXPROCESS USRXEXCEPTION
Other Servers	No	N/A

Syntax for COBOL Programmers

```
ENTER C "SBPROCESSITER" IN USRXLIB.
    GIVING RETURN-CODE.
```

Syntax for C Programmers

```
long SBPROCESSITER()
```

Parameters

RETURN-CODE returned value

- The number of times that this event has been processed again by calling SBPROCESSAGAIN

Considerations

- As of 7.000, only supported with an Enscribe source and when replicating to an Open environment using an Enscribe dummy target.

New and Modified Measure Counters

Process	Counter Name	Type	Description
SBCOLL	ADTRECSUSED	Accum	# of ADT records used
	ADTBEGINSUSED	Accum	# of begins used
	ADTNETCOMMITUSED	Accum	# of network commits used
	ADTNETABORTUSED	Accum	# of network aborts used
	ADTEOFIDLE	Busy	Time spent idling at EOF
	ADTEOFHITS	Faccum	# of times EOF was hit while reading from the audit trail
	LATESTREADADTTS	FSnapshot	Snapshot of the event timestamp of most recent event read from audit
	LATESTREADATTS	FSnapshot	Snapshot of when the most recent event read from audit
	LATESTREADLAG	FSnapshot	Snapshot of the read lag for the most recent event read from audit
	ALTPREREADBUSY	Busy	Time spent prereading in order to filter altfiles
	ADTFLUSHREAD	Accum	# of adt flushes read
	ADTFLUSHUSED	Accum	# of adt flushes used
(modified)	SYNCTXCOUNT	Faccum	# of transactions that were treated synchronously
	ASYNCTXCOUNT	Faccum	# of transactions that were treated asynchronously
	FLUSH2COMMITTIME	Faccum	Time in ms from when the audit flushed event is read to when the end event is read
	FLUSH2COMMITSAMP	Faccum	# of samples in FLUSH2COMMITTIME
	FLUSH2RTCTIME	Faccum	Time in ms from when the audit flushed event is read to when the ready to commit notification is sent to the coordinator
	FLUSH2RTCSAMP	Faccum	# of samples in FLUSH2RTCTIME
AUDQMGR	ADTOPSRECV	Faccum	Total # of operations received from COLL
	ADTOPSSENT	Faccum	Total # of operations sent to CONS
	ADTRECVMSGS	Faccum	# of data IPMs received from COLL
	ADTSENTMSGS	Faccum	# of data IPMs sent to CONS
	ADTSENTQTIME	Queue	Queue of data IPMs being sent to CONS
	LATESTRECVADTTS	FSnapshot	Snapshot of the event timestamp of the most recently received event
	LATESTRECVATTS	FSnapshot	Snapshot of when the most recent event was received

	LATESTRECVLAG	FSnapshot	Snapshot of the receiving lag for the most recent event received.
	LATESTSENTADTTS	FSnapshot	Snapshot of the event timestamp of the most recently sent event
	LATESTSENTATTS	FSnapshot	Snapshot of when the most recent event was sent
	LATESTSENTLAG	FSnapshot	Snapshot of the sending lag for the most recent event sent
	TOTALQFILECNT	FSnapshot	Snapshot of the # of queue files at a time
	INUSEQFILECNT	FSnapshot	Snapshot of the # of queue files in use at a time
	ADTFLUSHRECV	Faccum	# of adt flushes received
	ADTFLUSHSENT	Faccum	# of adt flushes sent
AUDCONSN	ADTOPSRECV	Accum	Total # of operations received
	ADTOPSUSED	Accum	Total # of operations used
	ADTBEGINRECV	Accum	# of begins received
	ADTBEGINUSED	Accum	# of begins used
	ADTCOMMITRECV	Accum	# of commits received
	ADTCOMMITUSED	Accum	# of commits used
	ADTABORTRECV	Accum	# of aborts received
	ADTABORTUSED	Accum	# of aborts used
	ADTNETCOMMITRECV	Accum	# of network commits received
	ADTNETCOMMITUSED	Accum	# of network commits used
	ADTNETABORTRECV	Accum	# of network aborts received
	ADTNETABORTUSED	Accum	# of network aborts used
	ADTALTRECV	Accum	# of alters received
	ADTALTUSED	Accum	# of alters used
	ADTCRTRECV	Accum	# of creates received
	ADTCRTUSED	Accum	# of creates used
	ADTPRGRECV	Accum	# of purges received
	ADTPRGUSED	Accum	# of purges used
	ADTPRGDATARECV	Accum	# of purgedatas received
	ADTPRGDATAUSED	Accum	# of purgedatas used
	ADTINSRECV	Accum	# of inserts received
	ADTINSUSED	Accum	# of inserts used
	ADTUPDRECV	Accum	# of updates received
	ADTUPDUSED	Accum	# of updates used
	ADTDELRECV	Accum	# of deletes received
	ADTDELUSED	Accum	# of deletes used
	LATESTRECVADTTS	FSnapshot	Snapshot of the event timestamp of the most recently received event
	LATESTRECVATTS	FSnapshot	Snapshot of when the most recent event was received
	LATESTRECVLAG	FSnapshot	Snapshot of the receiving lag for the most recent event received

	USRXTIMEOUTBUSY	Busy	Time spent in the USRXTIMEOUT function
	FUPDCRCOUNT	Accum	# of FUP DCR commands used
	FUPDCRBUSY	Busy	Time spent executing FUP DCR commands
	MPDCRCOUNT	Accum	# of MP DCR commands used
	MPDCRBUSY	Busy	Time spent executing MP DCR commands
	ENSLOCKWAIT	Faccum	Total amount of time waiting on Enscribe locks
	ENSRETRYDELAY	Faccum	Total amount of time delaying between Enscribe retries
	SQLLOCKWAIT	Faccum	Total amount of time waiting on SQL locks
	SQLRETRYDELAY	Faccum	Total amount of time delaying between SQL retries
	WAITCOMMITBUSY	Busy	Time spent waiting for commits to complete
	SCHEMASTORED	Snapshot	# of schemas stored in memory
SBCOOR	APPSYNCTXCOUNT	Faccum	# of transactions that matched a sync user app
	APPASYNCTXCOUNT	Faccum	# of transactions that didn't match a sync user app
	SYNCTXNODBS	Faccum	# of transactions that matched a sync user app, but wasn't processed synchronously because it wasn't replicated by the collector.
	SYNCTXCOLLBEHIND	Faccum	# of transactions that matched a sync user app, but wasn't processed synchronously because the collector fell back to async mode due to high latency
	SYNCTXASYNCMODE	Faccum	# of transactions that matched a sync user app, but wasn't processed synchronously because the coordinator is set to ASYNC mode
	SYNCTXNOTALLCOLL	Faccum	# of transactions that matched a sync user app, but wasn't processed synchronously because ALL_COLLECTORS_REQUIRED is on, and not all collectors are connected
	SYNCTXNOCOLLOPEN	Faccum	# of transactions that matched a sync user app, but wasn't processed synchronously because no collectors are connected

	SYNCTXNOPREPRECV	Faccum	# of transactions that matched a sync user app, but wasn't processed synchronously because the coordinator didn't receive a prepare from TMF
(modified)	SYNCTXCOUNT	Faccum	# of transactions that were processed fully synchronously
	ASYNCTXCOUNT	Faccum	*(removed, previous counter is now APPASYNCTXCOUNT)*

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 3.40282347e+38. However, the HPE Shadowbase for Other Servers ODBC Server gets a range error starting with 3.40282337e+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 use of DBS parameters FETCHSOURCE and INSERTNOTFOUND for DBSs containing key-sequenced Enscribe files with a key-offset of 1 causes issues with auditcompressed updates. Under such circumstances, Shadowbase would remove the offset in the target file, replacing the first non-key byte with the leftmost byte of the key, modifying the primary key value. This causes INSERTNOTFOUND to trigger, causing Shadowbase to effectively double the target file. This behavior has only been observed with Enscribe files.

    The work-around for this is to set the FETCHSOURCE DBS parameter to (ON, SOURCE).

50) 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.

51) 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%

52) 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.

53) 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.

54) 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.

55) 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.

56) 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.

57) 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.

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

59) 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.

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

61) 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.

62) SBMAP may parse minus signs in the map file incorrectly and return a syntax error when it shouldn’t.
    - If a minus sign used for subtraction is followed immediately by a numeric literal (e.g. 5-1), a syntax error will occur. A workaround for this is to add a space between the minus and the number.
    - A minus sign can only be used to negate a numeric literal and not any other numeric expression. A workaround for this is to subtract the expression from zero.

63) 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.

64) 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.

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

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

67) When obeying a file with SBQFUTL, lines containing the word "obey" may fail if it isn't the OBEY command.

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

69) 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.

70) When using ETL Toolkit processing, the following SQL/MX datatypes will log "unknown datatype" and insert a NULL value: VARCHAR2, NCHAR VARYING, BINARY, and VARBINARY.

71) When using ETL Toolkit processing, the following SQL/MX datatypes have the possibility to be corrupted: FLOAT and DOUBLE PRECISION.

72) When replicating IBM MQ environments, there is a possibility when partitioning an MQ queue that a secondary partition being replicated to a virtual drive will be created corrupted.

73) 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.

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