SOFTWARE RELEASE DOCUMENT (SOFTDOC)

Product:	Shadowbase
Release:	Gravic Version 6.800 HP NonStop Shadowbase: T1122-ABE (SB Repl/Guardian)
Release Date:	September 30, 2023
Copyright Notice:	Copyright Gravic, Inc. 1995 – 2023(www.gravic.com)
File Name:	IPM6800.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 6.800. 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 T1122H06, including:

- HPE Shadowbase Audit Reader,
- HPE Shadowbase Enterprise Manager, and
- HPE Shadowbase Compare for SQL.

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

NOTE: **Supported Release Version Updates (RVUS)**: This softdoc applies to all currently supported database and operating system versions as described by the Shadowbase Supported Versions Cross-Reference. 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.

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 - Mature Features in HPE NonStop Shadowbase, 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 - Deprecated Features in HPE NonStop Shadowbase, 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
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.	NSB 6.410
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
Shadowbase Enterprise Manager (SEM)	Windows tool used to monitor Shadowbase environments in a GUI setting.	NSB 6.710

Table 2 - Deprecated Features in HPE NonStop Shadowbase

Feature	Description	Deprecated Release	Last Release Date
Non-native Consumer	The non-native version of the consumer (AUDCONS) is deprecated. Customers still using the non-native version of the consumer should immediately begin planning to move to the native version (AUDCONSN).	NSB 6.410 April, 2018	April, 2020
SBLINK	The original program used to connect SEM to a NonStop environment, replaced by listener AUDCOM process.	NSB 6.220 May, 2016	NSB 6.320 May, 2017
SBQFILE	Program used in conjunction with replicating Enscribe queue files to target non-queue key-sequenced files, replaced with 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

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.

Special Notes for Version 6.800

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 6.800 from a version prior to 6.700, the recommended upgrade procedure is as follows:

Prior to upgrading:

- Install Shadowbase 6.800 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 NonStop Shadowbase environment, please contact HPE support.

3) 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.

4) 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.

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

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

7) 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.

8) 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.

9) 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.

10) 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.

11) 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.

12) 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.

13) 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.

14) The EMS subsystem version has been updated with this version from V67 to V68 to match the two most significant digits of the current Shadowbase release version id.

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

16) 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.

17) As of Version 6.400, the SBLINK and SBQFILE programs have been deprecated will no longer be shipped alongside the rest of Shadowbase.

18) 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.

19) 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.

Changes in Release 6.800 (since Version 6.700)

New Features

1) SQL/MP exact syskey replication is now supported without the need for using ODBC or OSS (however, the existing method of using OSS and ODBC/MX remains supported). The new approach that does not use OSS/ODBC is added via a new HPE Shadowbase MX Server (SBMXSERV) process, with a special user exit linked consumer program. For more information, see the HPE Shadowbase SBMXSERV portion of the New Features Description section.

2) Entry-sequenced SQL/MP tables can now be replicated via a **unique** index by using the KEYSPECIFIER DBS parameter. This allows exact and correct UPDATE and DELETE WHERE clause processing as the source to target “primary key” may not be the same.

3) Audit DDL replication of CREATE events supports new enhancements, allowing for more sophisticated mapping of secondary partitions and alternate key files for the file created on the target system. These enhanced mapping is configured via the new DBS parameters, CREATELIKE and CREATEFILEPREFIX. See the related new DBS parameter documentation for further details.

4) The STATS command will now additionally display the maximum value for certain lag statistics. This applies to STATS COLL, QMGR, and CONS. Bolded sections are new.

   STATS COLL (SBCOLL only):

```
LATENCY WARNING STATUS: DISABLED
      ETS: 06-21 16:39:33.736  DIFF (CUR): 00:00:00.000  (MAX): 00:00:00.000
      LTS: 06-21 16:39:33.736  LAG  (CUR): 00:00:00.000  (MAX): 00:00:00.000
```

   STATS QMGR:

```
LATENCY WARNING STATUS: DISABLED
      COLL:     06-21 16:37:40.029  LAG (CUR): 00:08:59.093  (MAX): 00:08:59.093
      CONS ETS: 06-21 16:37:40.029  LAG (CUR): 00:08:59.093  (MAX): 00:08:59.093
      CONS LTS: 06-21 16:37:40.029  LAG (CUR): 00:08:59.093  (MAX): 00:08:59.093
```

   STATS CONS:

```
        LAST EVENT TS (LTS): 07-10 14:09:39.287
        PROCESSED AT TS:     07-10 14:09:44.295    LAG(CUR): 00:00:05.008
                                                      (MAX): 00:00:05.008
```

5) SBUNDROS configurations support the SB_ADD SBMON_PARAM command. This command can override the settings of the SBMON ADD command written to the Guardian file *SBMON and can override additional settings in the Guardian file *SPINI and the OSS file shadparm.ini where appropriate. Overrides can be specified for a DOC Writer, TRS, Direct Writer, or DOC Cleaner. The override options are as follows:

   DOC Writer overrides and values

   a. **NAME** – A unique 5-character name for the DOC Writer. Defaults to the process name of the object.

      Modifies the following *SPINI/shadparm.ini file parameters:

      i. SHAD_RUN_PARAMS_<doc-name>_1
      ii. SHAD_<doc-name>_IP
      iii. SHAD_<doc-name>_PORT
      iv. SHAD_<doc-name>_PROCESS

   b. **PROCESS_NAME –** An existing HPE NonStop TCP/IP process name. Defaults to the TCP/IP process name of the target node.

      Modifies the following *SPINI/shadparm.ini file parameter:

      i. SHAD_<doc-name>_PROCESS

   c. **SERVER_TYPE** – A number ranging from 1 to 10, which represents an OSB object type. For example, 1 is DOC Writer, 2 is Direct Writer, 3 is TRS, etc. Defaults to 1 for DOC Writer objects.

      Note: This override should not be used unless directed to do so because it can break the environment and requires all of the object's other parameters to be overridden!

   d. **EXECUTABLE_NAME** – A valid executable type for the corresponding object. Defaults to SBOCONSC for cached DOC Writers.

   e. **RECORD_FORMAT** – A number ranging from 1 to 3. Defaults to 3 for cached SQL statements.

   f. **IP_ADDRESS** – A valid TCP/IP address available on the system. Defaults to the TCP/IP address associated with target node.

   g. **PORT** – A valid port number on the system. Defaults to the port number associated with target node.

   h. **CPU** – A valid CPU on the system. Defaults to the primary CPU specified for the DOC Writer.

i. **PRIORITY** – A number ranging from 0 to 199. Defaults to 130.

j. **NETWORK_PACKET_SIZE** – Defaults to 65534 bytes, which is the recommended size.

k. **TRANSACTION_PROCESSING** – An option of Y or N. Defaults to Y.

l. **TRANSACTION_BOUNDRIES** – An option of Y or N. Defaults to Y.

m. **STARTING_SEQUENCE_NUMBER** – A number ranging from 1 to 999. Defaults to 1.

n. **SQL_FORMAT** – A valid SQL format type. Defaults to SQLSBV1.

o. **STP_USER –** A username for the Shadowbase connection between the Consumer and DOC Writer. Defaults to OPEN.

p. **STP_PASSWORD** – A password for the Shadowbase connection between the Consumer and DOC Writer. Defaults to OPEN.

TRS overrides and values

a. **NAME** – A unique 5-character name for the TRS. Defaults to the process name of the object.

   Modifies the following *SPINI/shadparm.ini file parameters:

   i. SHAD_RUN_PARAMS_<trs-name>_1

b. **SERVER_TYPE** – A number ranging from 1 to 10, which represents an OSB object type. For example, 1 is DOC Writer, 2 is Direct Writer, 3 is TRS, etc. Defaults to 3 for TRS objects.

   Note: This override should not be used unless directed to do so because it can break the environment and requires all of the object's other parameters to be overridden!

c. **EXECUTABLE_NAME** – A valid executable type for the corresponding object. Defaults to SBODBCTRS for TRS objects.

d. **READ_OPEN_DOCS** – An option of Y or N. Defaults to Y.

e. **CLEANUP_DOCS** – An option of Y or N. Defaults to N.

f. **CONVERT_INS_TO_UPD** – An option of Y or N. Defaults to N.

g. **STARTING_SEQUENCE_NUMBER** – A number ranging from 1 to 999. Defaults to 1.

h. **SQL_FORMAT** – A valid SQL format type. Defaults to SQLSBV1.

i. **CATALOG** – The name of the target SQL/MX catalog. Defaults to the catalog specified in the SB_ADD CONSOS command.

j. **SCHEMA** – The name of the target SQL/MX schema. Defaults to the schema specified in the SB_ADD CONSOS command.

k. **DB_USER** – The target database user name. Defaults to the database user name specified in the SB_ADD CONSOS command.

l. **DB_PASSWORD** – The target database password. Defaults to the database password specified in the SB_ADD CONSOS command.

m. **DSN** – The target DSN. Defaults to the DSN specified in the SB_ADD CONSOS command.

n. **CPU** – A valid CPU number. Defaults to the CPU specified in the SB_ADD CONSOS command.

o. **PRIORITY** – A valid number ranging from 0 to 199. Defaults to 130.

Direct Writer overrides and values

a. **NAME** – A unique 5-character name for the Direct Writer. Defaults to the process name of the object.

   Modifies the following *SPINI/shadparm.ini file parameters:

   i. SHAD_RUN_PARAMS_<dw-name>_1
   ii. SHAD_<dw-name>_IP
   iii. SHAD_<dw-name>_PORT
   iv. SHAD_<dw-name>_PROCESS

b. **SERVER_TYPE** – A number ranging from 1 to 10, which represents an OSB object type. For example, 1 is DOC Writer, 2 is Direct Writer, 3 is TRS, etc. Defaults to 3 for TRS objects.

   Note: This override should not be used unless directed to do so because it can break the environment and requires all of the object's other parameters to be overridden!

c. **EXECUTABLE_NAME** – A valid executable type for the corresponding object. Defaults to SBODBCDW for Direct Writer objects.

d. **IP_ADDRESS** – A valid TCP/IP address available on the system. Defaults to the TCP/IP address associated with target node.

e. **PORT** – A valid port number on the system. Defaults to the port number associated with target node.

f. **NETWORK_PACKET_SIZE** – Defaults to 65534 bytes, which is the recommended size.

g. **CONVERT_INS_TO_UPD** – An option of Y or N. Defaults to N.

h. **SQL_FORMAT** – A valid SQL format type. Defaults to SQLSBV1.

i. **CATALOG** – The name of the target SQL/MX catalog. Defaults to the catalog specified in the SB_ADD CONSOS command.

j. **SCHEMA** – The name of the target SQL/MX schema. Defaults to the schema specified in the SB_ADD CONSOS command.

k. **STP_USER –** A username for the Shadowbase connection between the Consumer and Direct Writer. Defaults to OPEN.

l. **STP_PASS** – A password for the Shadowbase connection between the Consumer and Direct Writer. Defaults to OPEN.

m. **DSN** – The target DSN. Defaults to the DSN specified in the SB_ADD CONSOS command.

n. **CPU** – A valid CPU number. Defaults to the CPU specified in the SB_ADD CONSOS command.

o. **PRIORITY** – A valid number ranging from 0 to 199. Defaults to 130.

DOC Cleaner overrides and values

a. **NAME** – A unique 5-character name for the DOC Cleaner. Defaults to the process name of the object.

   Modifies the following *SPINI/shadparm.ini file parameters:

   i. SHAD_RUN_PARAMS_<dcl-name>_1

b. **SERVER_TYPE** – A number ranging from 1 to 10, which represents an OSB object type. For example, 1 is DOC Writer, 2 is Direct Writer, 3 is TRS, etc. Defaults to 9 for DOC Cleaner objects.

   Note: This override should not be used unless directed to do so because it can break the environment and requires all of the object's other parameters to be *overridden!*

c. **STARTING_SEQUENCE_NUMBER** – A number ranging from 1 to 999. Defaults to 1.

d. **CPU** – A valid CPU number. Defaults to the CPU specified in the SB_ADD CONSOS command.

e. **PRIORITY** – A valid number ranging from 0 to 199. Defaults to 130

6) The ESTATS command has been updated to include new options for formatting output and limiting what gets printed. These options can be used with ESTATS AUD and ESTATS DBS. The new options are as follows:

BRIEF
Limits the output to one line per DBS/file. With this option, the following information will be shown:

DBS stats:

```
             DBS          DO        UNDO TRANSFORMED     IGNORED
---------------- ----------- ----------- ----------- -----------
```

File stats:

```
             DBS                               SOURCE
  TARGET       TOTAL
---------------- ------------------------------------ ----------------------------
-------- -----------
```

For SQL/MX tables, the SOURCE and TARGET columns will contain the ANSI name. If longer than 36 characters, the last 36 characters of the ANSI name will be used.

LISTCOUNT <num>
Limits the output to the first <num> DBSes/files per consumer.

SORT {NAME | ACTIVITY}
Determines the order of output per consumer. NAME is the default behavior. ACTIVITY sorts by total number of events, showing the most active first.

7) Replication of Enscribe Entry-sequenced files can now be configured to use a non-unique alternate key, instead of just unique alternate keys. Please refer to the manual section describing this feature to understand exactly how positioning works when there are multiple records in the target file with the same non-unique alternate key values.

8) When replicating from an SQL/MX source, the user exit function SBGETREALSOURCE will now output the physical location of the source table.

9) While a queue manager’s queue files are all full, the QMGR’s status during a STATUS command will appear as SUSPQFULL, indicating that the QMGR cannot accept any additional data from the Collector. Additionally, while in this state, a warning message will be repeated every 5 minutes.

10) Queue Managers can now be started without a collector in order to process (send to the Consumer) data already present in the queue files.

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

12) Consumers can now handle timers in user exit code. The functions SBTIMERSTART and SBTIMERSTOP should be used to start and stop user exit timers. When the timer started by SBTIMERSTART elapses, the entry point USRXTIMEOUT will be called, supplying the same tag that was used when the timer was created.

13) SBMAP will now abend if a target column or field does not exist or if a NULL literal is being applied to a column that is not nullable.

14) When replicating events using SBMAP, if an SBWHERE clause resolves to false, the event will no longer immediately be ignored. Instead, it will continue attempting to match to a map block where the SBWHERE clause resolves to true.

15) DDL Command Replication (DCR) now supports replicating events for Enscribe files using the SBFUP utility. See the HPE Shadowbase FUP DCR section for more details, and the HPE NonStop Shadowbase DDL Command Replication manual for more in-depth information on how to configure and use this new feature. FUP DCR related parameters are listed in the “New and Modified SET CONS Parameters” section of the SOFTDOC.

16) SBSQLCI and SBFUP now support the SYSCALL command in order support replication of TACL commands. See the HPE Shadowbase FUP DCR section for the syntax and how to restrict usage.

17) SBCOLL now supports splitting the replication of events from the same source table between multiple replication threads with the DBS parameters HASHBUCKETCOUNT and HASHBUCKETID. This feature aids Shadowbase in splitting replication traffic for “heavy hitter” files and tables across many replication threads to avoid “hot spot” areas.

18) The consumer now supports replication for the SQL/MX datatypes VARCHAR2, BINARY, and VARBINARY.

19) When the FETCHSOURCE DBS parameter is enabled, the consumer will now only fetch the source record/row if required, instead of for every compressed update. There is a new FETCHSOURCE option to force the consumer to fetch the source for every compressed update.

20) SBSQLCI and SBFUP's name mapping has been updated. Volume and subvolume mappings will now be applied to mapped files using a "more specific to less specific" algorithm, meaning the most specific mapping for a filename is used first.

    For example, if there is a volume mapping command present mapping $VOLA to $VOLB, with no other mappings defined, a subsequent command to CREATE FILE $VOLA.SUBVOL.FILE will now map to $VOLB.SUBVOL.FILE. If instead there was also an existing mapping for $VOLA.SUBVOL to $VOLC.SUBVOL, the source $VOLA.SUBVOL.FILE command would instead be mapped to $VOLC.SUBVOL.FILE.

21) In the FUPSERVER consumer's FUPLOG output, the session started and session ended messages will now include the process name of the SBFUP process.

22) In SBSQLCI and SBFUP, the config file parameters have been updated to be consistent with the SB_SET parameter names. The old parameter names with underscores are still supported, but the new name will be used over it if both are included (note that eventually, the old parameter names will be deprecated). See the HPE NonStop Shadowbase DDL Command Replication Manual for more information.

23) Dictionaries generated with DDL2 are now supported.

24) SBDDLUTL is now able to generate flattened records and tables from DDL definitions (in addition to RECORD constructs). Definitions are loaded using the DEFINITIONS command. When converting a DDL definition into a table definition, the table name will be based on the definition name. When converting to Hana, SQL/MP, or PostgreSQL, dashes in the definition name will be removed. When converting to DB2, SQL/MX, MySQL, Oracle, SQL Server, or Sybase, dashes will be converted to underscores.

25) Added the consumer parameter FILTEREMPTYTX. If enabled, consumers replicating over TCPIP will not replicate begin, commit, and abort events for transactions where all events have been ignored through SBMAP or a user exit. Note: in most cases, FILTEREMPTYTX should be enabled as replaying empty transactions at the target can decrease replication performance.

26) The Shadowbase configuration macros (SBBIDIR, SBBIDROS, SBUNIDIR, and SBUNDROS) were enhanced in a number of ways.
    a. Further validation logic was added for command inputs.

    i. All object-based parameter overrides (e.g. CONS_PARAM, QMGR_PARAM, etc.) are now validated to ensure that the specified object exists.
    ii. All SB_ADD commands to add an object now check for attempts to add a duplicate object.
    iii. SBUNIDIR now prevents attempting to configure a TIDFILE name prefix argument that is greater than 7 characters.

   If any of these validation checks fail on the input, an error message will be output, and the macro will abort generation of the configuration files.

b. The configuration macros now support adding identifying information about the input file used to generate the output configuration files. To enable this, set the new SB_CONFIG_FILE global variable option using the SB_SET command. The following command can be used to set this from a ?TACL MACRO configuration input file:

```
SB_SET SB_CONFIG_FILE %0%
```

c. SBUNDROS and SBBIDROS will now set the TCPIP^PROCESS^NAME TACL define based on the target primary TCP/IP process name specified for the last CONSOS added if no TCPIP^PROCESS^NAME define is either present in the TACL session, or explicitly set with a SB_ADD TACL_DEFINE override.

Problems Fixed

1) Version 6700 license files were limited to 8 HPE NonStop systems in a single license file. More systems could be handled by splitting the nodes into multiple license files; only nodes being replicated over Expand need be collocated in the same license file.

2) The validation logic for preventing use of the AUDITED CREATEOPTIONS DBS parameter option was incorrectly checking the last Consumer added, instead of the Consumer associated with the DBS. This could cause the AUDMON backup process to fail during takeover in the event of a failure of the primary AUDMON process, due to mistakenly determining it to be an invalid configuration. This could occur when all of the following conditions are met:
   1) The environment is bidirectional (i.e. TRANSLOG files are in use).
   2) There are a mix of Consumers with and without a TRANSLOG configured (e.g. a mix of unidirectional and bidirectional Consumer threads in the environment).
   3) There are one or more DBS objects configured with the CREATEOPTIONS AUDITED option specified.

3) When a consumer has multiple DBSes that use the same TARGETFILE, but different DDL records, each DBS will use the correct DDL record.

4) The consumer will no longer abend when attempting to write a long user exit log message to the USERTRACE file.

5) When using the REPLICATION parameter SKIP mode option with the AUDCOLLN collector program, transactions containing multiple events with different replication thread usage profiles could cause the end events to be buffered and not fully cleaned up. This would cause the replication restart point to stick, and repeated occurrences could cause the collector to run out of buffers and stop collecting.

6) RESOLVETRANS was not able to resolve the transaction usage limit error 83 for the consumer process under certain specific circumstances. It would fail on a subsequent error 83 after RESOLVETRANS processing in the event that all of the transactions in use were associated with aborted transactions where it had seen one or more of the undo backout events, but had not yet received the final abort record.

7) The collector will no longer abend if a SQL/MP view is created while the MAPTOPRIPTN or ALLPARTITIONS parameters are enabled.

8) AUDMON will no longer abend when processing the STATS COLL, LAG command if the collector is very far behind.

9) The consumer will no longer abend during a SOLV load if the DBS TARGETFILE parameter is set incorrectly.

10) The consumer will no longer open a file multiple times if it retries an error that does not prevent the file from being opened, such as an error 4.

11) The consumer will now log a message and abend cleanly if configured to replicate a file by an alternate key that doesn’t exist.

12) The SBDDLUTL command LIST uses “CONCATS” and “REPLACES” as options instead of “CONCATENTATIONS” and “REPLACEMENTS”.

13) The following syntax errors will no longer occur in an SBMAPCONF file:
    - Using -1 as a literal
    - Having no digits to the left of the decimal point for a float literal

14) SBMAP will now abend if it parses an unterminated comment.

15) The SBMAP function NUMTOSTRING now properly handles all of the different signed PIC 9 variants supported by Enscribe.

16) SBMAP will no longer abend when handling float literals with high precision.

17) When running SBSQLCI or SBFUP with an OUT file specified, but not an IN file, prompts are properly written to the terminal instead of the OUT file.

18) Fixed an issue with license validation at startup for Expand configurations between certain system types. A workaround for the issue was available via a SHADTYPE file.

19) When replicating to OSB, the "Latest TX Source Commit" timestamp will now properly be the source commit timestamp value (when the commit occurred in the source database) instead of when the consumer processed the commit event.

20) SBSQLCI and SBFUP will now log any map file errors before abending due to hitting the ALLOWERRORS limit.

21) In SBSQLCI and SBFUP, the HELP SB_SET <param> text will now include the default value for all parameters.

22) SBSQLCI and SBFUP will now print an error when a filename with a syntax error is entered for an SB_SET command.

23) SBSQLCI and SBFUP will now print an error if SB_SAVE fails to write to the config file.

24) SBSQLCI and SBFUP will no longer trap after an invalid history command.

25) SBCOLL will now log a warning message if it is coordinating for a DCR event while a downstream consumer or qmgr is suspended. In both AUDCOLLN and SBCOLL, the collector parameter COORDWAITINGWARNRATE will now apply to this parameter.

26) The consumer will no longer trap while replicating to OSB if the source table was dropped and recreated with a different schema before the event was processed.

27) License validation will no longer fail if CPU 0 is down.

28) SBMAP will no longer trap if a statement or assignment is included before any map block headers.

29) SBMAP will no longer trap when encountering invalid characters in certain scenarios.

30) The INFO AUD command will now display ALLOWUSERID is alphabetical order instead of at the bottom of the settings.

31) A FUPSERVER or MPSERVER consumer will now properly abend if an AUDMON failure is detected while the consumer is in the PAUSE or PAUSEABT state.

32) SBCOLL now correctly handles ROOTPARTITION while ALLPARTITIONS is ON.

33) When SBSQLCI or SBFUP prompts for a control file on startup, the entered filename will now be expanded to include the volume and subvolume.

34) The consumer will no longer sometimes miss the definition of one or more partitions when replicating SQL/MP create events for tables with a large number of partitions.

35) EMS messages related to the MPERRORACTION and FUPERRORACTION options STOP and PAUSE are now critical.

36) When running the CHECK DBS command, errors related to Base24 handling of alt keys and replicating entry-sequenced files by alt key or index properly increment the "NUMBER OF SRC'S/TRG'S NEEDING DETAIL REVIEW" count.

37) Added the LOGCTRLEVENTS parameter to SBSQLCI and SBFUP. When turned on, a message will be logged for each command written to the control file.

38) SBSQLCI and SBFUP will no longer add commands executed from an OBEY file to the command history.

39) AUDMON will no longer prevent stopped FUPSERVER or MPSERVER consumers from being altered.

40) SBMAP will now correctly replicate Enscribe records whose key does not match the key from SOURCEREC or TARGETREC.

41) Enabling the DBS ALLPARTITIONS parameter did not work for DDL replication when using the SBCOLL Collector program, and required a workaround to use either a DBS SOURCEFILE parameter value with a wildcarded volume, or to use multiple DBSes for the individual partitions. This problem has been corrected, and the workaround is no longer required.

42) Issuing a TACL SETTIME on a system would cause an incorrect time to be reported by Shadowbase (offset by the amount the time was adjusted). Changing the time by a large amount could cause AUDMON to abend during a STATS QMGR command. Both problems have been corrected.

43) SOLV would not properly handle loading of unstructured files when using a logical record size which would cause a logical record to span across partitions. It would abend if NUMTRANSRECS set to 1, and otherwise could load incorrect data for the logical record that spanned partitions.

44) SBUNDROS and SBBIDROS param overrides were previously upshifted (to create capital letters in the param), which would cause issues for parameters which are case sensitive.

New Features Description

HPE Shadowbase SBMXSERV

Overview

The HPE Shadowbase MX Server (SBMXSERV) allows for exact SQL/MP SYSKEY replication for SQL/MP clustering key and other key-sequenced SQL/MP SYSKEY tables without using OSS or ODBC.

There are two components in use here. The first part is a Consumer user exit. On start-up, this user exit will create a server process (SBMXSERV). Whenever the consumer receives an insert that it is configured to replicate using SBMXSERV, it will send the insert operation to the server process. SBMXSERV will execute the operation, preserving the source-to-target SYSKEY value, and then return the result of the operation, including any error messages, to the Consumer user exit for handling.

Implementing SBMXSERV Replication

Setting up the SBMXSERV replication environment involves configuration of the standard HPE NonStop Shadowbase replication environment with a few extra parameters.

Prerequisites

The SBMXSERV program must be MX compiled. This will be done as part of the INSTALL script when the MXCOMP option is included.

Linking the SBMXSERV User Exit

Before running HPE Shadowbase with SBMXSERV, the SBMXSERV client user exit must be linked into the Consumer. HPE Shadowbase comes with a TACL macro, LINKMXCL that will link in the user exit and SQL compile the Consumer. To use this macro, enter:

```
LINKMXCL [ CATALOG <catalog> ] [ OWNER <user> ]
```

Where:

<catalog>
Is the $volume.subvolume location of the SQL catalog into which the Consumer is to be SQL compiled. If you do not enter a catalog, the catalog identified in the =_defaults define is used.

<user>
Is the username of the owner for the consumer program. Set this to the user who the consumer program should run as. After the consumer is built, it will be 'given' to this user and secured properly.

HPE NonStop Shadowbase Configuration

HPE Shadowbase replication environments configured with SBMXSERV can only support like-to-like SQL/MP replication (this means that the target schema must match the source schema).

Enabling SBMXSERV is done via DBS and Consumer parameters. This section lists the available SBMXSERV-related parameters.

CONS Parameters

PROGRAM { [\system.$volume.subvolume] filename }

Initial	Changed			
Version	**Version**	**Values Default**	**Value Specifications**	**Alter**
1.000	5.002	AUDCONSN	**Valid program file name**	No

Is the name of the Consumer object file. It can be fully qualified with the volume and subvolume names. Unless otherwise specified, the volume and subvolume names are the same as the location of the object file for AUDMON. Must be set to the name of the Consumer object file with the SBMXSERV user exit linked in. If the LINKMXCL TACL macro was used, the new consumer name will be AUDCONSN.

DBS Parameters

MAXRETRIES number

Initial	Changed				
Version	**Version**	**Values Default**	**Minimum**	**Maximum**	**Alter**
4.070		-1	-2	32767	No

This parameter is used by SBMXSERV in addition to its normal usage. It specifies the number of retries following SBMXSERV responding with a lock timeout error (file-system error 73). The valid range is -2 to 32767 and takes the following values:

MAXRETRIES = -2: Retry indefinitely.
MAXRETRIES = -1: Use the value set for the CONS.
MAXRETRIES = 0: No retries.
MAXRETRIES > 0: The number of times to retry the operation after the initial failure.

The default value is -1: use the value set for the CONS MAXRETRIES parameter.

RETRYDELAY seconds

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
4.070			-1	-1	32767	No

This parameter is used by SBMXSERV in addition to its normal usage. It specifies the number of seconds that SBMXSERV will direct SQL/MX to wait for a lock to become available before returning with an error. If set to -1, the value specified for the associated CONS object is used. If the value used the consumer is also -1, SQL/MX will not timeout.

USEREXITID { num }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
3.900			0	0	32767	No

Must be set to a value greater than 0 for all DBSes that need to be replicated through SBMXSERV.

USEREXITPARAM { value-with-no-spaces }
{ (value with spaces) }

Initial Version	Changed Version	Values	Default	Value Specification	Alter
6.220		None		See below	No

Used to configure the SBMXSERV process. Options are specified as a colon separated list. Options must be specified in order, but optional options can be skipped by leaving the value blank. The list of available options is shown below.

NOTE: The consumer will only launch one instance of SBMXSERV that is used by all DBSes. The first DBS configured to use SBMXSERV that gets added must have this parameter set and any other DBSes will ignore this parameter.

USEREXITPARAM Options

Server Program Name

Initial Version	Changed Version	Values Default	Value Specification	Alter
6.800		None	Valid program filename	No

This required parameter is the name of the SBMXSERV program file.

Server Process Name

Initial Version	Changed Version	Values Default	Value Specification	Alter
6.800		None	Valid process name	No

This required parameter is the process name that the consumer will use to create the SBMXSERV process.

Server CPU

Initial Version	Changed Version	Values Default	Value Specification	Alter
6.800		See below	Valid CPU number	No

Specifies the processor where the SBMXSERV runs. If omitted, the processor of the consumer will be used.

Server Priority

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.800		See below	1	199	No

Specifies the priority at which the SBMXSERV process will run at. If omitted, the priority of the consumer will be used.

EMS Collector

Initial Version	Changed Version	Values Default	Value Specification	Alter
6.800		See below	Valid process name	No

Specifies an alternative EMS collector that SBMXSERV will send EMS messages to. If omitted, SBMXSERV will send EMS messages to the same EMS collector as the consumer.

IPC Timeout

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.800		See below	-1	32767	No

This optional parameter specifies the number of seconds that the consumer will wait for a response from SBMXSERV. If set to -1, the consumer will wait indefinitely. If omitted, the RETRYDELAY DBS parameter times 2 will be used.

Behavior on error

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.800		1 (ABEND)	0 (SKIP)	1 (ABEND)	No

Specifies how the consumer should handle receiving a SQL/MX error from SBMXSERV. If set to 0, the consumer will log a message and skip the event. If set to 1, the consumer will abend. The default is 1.

Tracing

Initial Version	Changed Version	Values Default	Minimum	Maximum	Alter
6.800		0 (OFF)	0 (OFF)	1 (ON)	No

Enables or disables EMS tracing. If set to 0, tracing is disabled. If set to 1, both the consumer and SBMXSERV will log additional details for every event. The default is 0.

Note: Tracing generates a lot of additional overhead, and will significantly affect performance. Tracing should never be enabled except under direction from HPE Shadowbase Support.

Limitations

Due to differences between SQL/MP and SQL/MX (limitations with inserting data into SQL/MP tables through SQL/MX), the following restrictions apply to tables being replicated through SBMXSERV.

SBMXSERV does not support replicating SQL/MP tables with the following data types:

- Fraction-only DATETIMEs
- Fraction-only INTERVALs
- Character data types that are not using the ISO88591, KANJI, or KSC5601 character sets.
- Character columns using a collation other than the default collation
- If running a SQL/MX version prior to 3.6, character columns using the KANJI or KSC5601 character sets.

SBMXSERV does not support tables defined in the following ways:

- Column defaults cannot contain:
    - DATETIME string portions with nonstandard formatting
        - i.e. DATE “2023-1-1” instead of DATE “2023-01-01”
- Constraints cannot contain:

    - The UNITS function
    - DATETIME string portions with nonstandard formatting
    - FRACTION-only DATETIME or INTERVAL literals
    - Interval literals with negative signs inside quotation mark delimiters
        - i.e. INTERVAL '-5' DAY instead of INTERVAL -'5' DAY
    - Identifiers named after reserved words
- First keys cannot contain:
    - DATETIME string portions with nonstandard formatting
    - FRACTION-only DATETIME or INTERVAL literals
    - Interval literals with negative signs inside quotation mark delimiters

Lastly, the DELUIDXDUP DBS parameter is not supported with SBMXSERV at this time.

HPE Shadowbase FUP DCR

HPE Shadowbase DDL Command Replication (DCR) offers the ability to execute, process, and replicate FUP commands through DCR for FUP, also known as FUP DCR. FUP DCR features a front-end command-line utility called SBFUP, which functions and acts like the FUP program, but with added enhancements for DCR.

Should I Use DCR or DDL Audit Replication?

Feature	SB DDL Audit Replication	SB DDL Command Replication (FUP DCR)
Supports operations on un-audited files	No	Yes
Target-side human intervention on errors	No	Yes[2]
File name mapping	Yes	Yes
Selective application of commands	Yes	No
ALLOCATE command	No	Yes
ALTER command	Yes	Yes
COPY command	No	Yes
CREATE command	Yes	Yes
DEALLOCATE command	No	Yes
DUPLICTAE command	No	Yes
LOAD command	No	Yes
LOADALTFILE command	No	Yes
PURGE command	Yes	Yes
PURGEDATA command	Yes	Yes
RENAME command	No	Yes
RESTART command	No	Yes
CONFIGURE command	No	Yes
SET command	No	Yes
RESET command	No	Yes

Figure 1 - Comparison of FUP DCR and DDL Audit Replication

Architecture and Workflow

The following section outlines the high-level architecture and workflow surrounding DDL Command Replication for FUP. It describes the general logic that the FUP DCR environment uses to dictate what happens with commands entered into SBFUP.

With HPE Shadowbase configured and running, the user runs SBFUP:

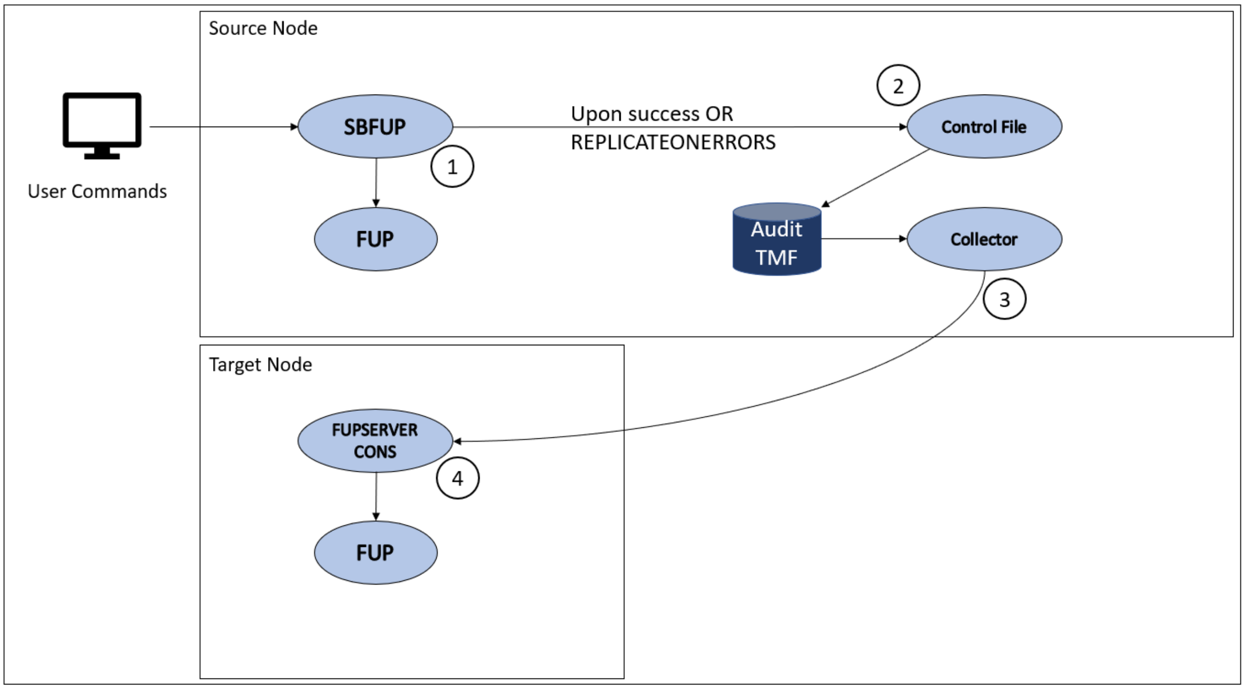
- SBFUP performs set up and initialization; spawning a child FUP process.
- At the same time, the target consumer will also spawn a child FUP process that will match SBFUP's child process' state.
- A portion of the set-up involves reading in a configuration from a file.

The user enters commands into SBFUP one of several ways:

- Interactively via the terminal / SBFUP command prompt
- Commands entered via the TACL prompt (including INLINE commands)
- Input/OBEY file passed in with the /IN <file>/ parameter

SBFUP processes entered commands differently depending on type:

- **SBFUP command:** processed on the source; usually has the effect of configuring an aspect of how SBFUP works, displaying stats, etc. These commands are not replicated. These are SBFUP-specific, and do not exist in FUP.
- **FUP informational command:** processed on the source; comprised of commands that just display information like FILENAMES, INFO, SHOW, etc. There commands do not get replicated.
- **FUP control command:** the commands that actually affect the file system, such as ALTERS, CREATES, and PURGES.
- **FUP session command:** the commands modify the state of the FUP session, so they are replicated, but do not affect the file system, so these commands do not cause coordination, even if FUPCOORDINATION is ON. In the event the consumer must restart while FUP DCR is running, session commands are replayed, but completed control commands are not.



1. **SBFUP passes entered commands to FUP, executing them on the source**
   a. Commands are parsed and processed, file names are expanded, and TACL DEFINES are evaluated.
   b. Commands are passed to FUP and executed on the source
   c. Results and errors are logged to the terminal or configured log file
2. **If the command needs to be replicated, SBFUP performs file name mapping, then writes it into the control file; only if:**
   a. The command was executed successfully on the source

   b. The REPLICATEONERRORS program setting is turned ON.

The CONTROLFILE is audited by TMF, so all interactions with it show up in the audit trail, which is read by the Collector.

3. **The Collector (AUDCOLLN or SBCOLL) picks up the CONTROLFILE events in the audit trail, and decides what to do next:**
   - **a.** Typically, the Collector will coordinate with all configured Consumers and Queue Managers. See the DCR Coordination section for more information.
   - **b.** The Collector sends the CONTROLFILE events to the downstream FUPSERVER consumer.

4. **The FUPSERVER consumer replays the FUP command on the target node by:**
   - **a.** Passing the command to the FUP process to be executed.

   Any issues are reported via. EMS and/or the FUPSERVER consumer's log file (FUPLOG).

Note that while an FUPSERVER consumer can perform audit data replication, it is recommended to keep the FUPSERVER consumer dedicated solely to performing FUPSERVER work.

In the event that the FUPSERVER consumer shuts down abnormally and is restarted after a FUP DCR command is executed on the target, but before the control file event's commit is processed. In this event, the DCR command may be replayed and cause an error. If this happens, the duplicate command can be skipped with the RESUMEDDL command if FUPERRORACTION is set to PAUSE, or by altering FUPERRORACTION to SKIP or SKIPNOLOG.

SBFUP

SBFUP is the front-end command-line utility the user will primarily interact with when engaging in DDL Command Replication for FUP. SBFUP is designed to work similarly to FUP and fit in with existing user workflows as much as possible. It supports various configuration commands, statistics, and FUP commands.

SBFUP Commands

Overview

This section covers the syntax, behavior, and considerations for each of the SBFUP-specific commands, and any FUP command whose behavior/support may differ from what might be expected of FUP. For in-depth help and documentation on FUP commands, refer to HPE’s File Utility Program (FUP) Reference Manual.

To see a listing of all supported commands, run SBFUP and enter the HELP SB command; a portion of the output is printed below:

```
SBFUP COMMANDS:
-----------------
CLS         EXIT        FC          HELP                HISTORY
OBEY        SB_LOAD     SB_RESET    SB_RESET_ERRORS     SB_SAVE
SB_SHOW     SB_SET      SYSCALL     VALIDATE            VERSION
!           ?

SUPPORTED FUP COMMANDS (NOT REPLICATED):
------------------------------------------
FILENAMES    FILES     INFO    SHOW
SUBVOLS      SYSTEM    VOLS    VOLUME

SUPPORTED FUP CONTROL COMMANDS:
---------------------------------
ALLOCATE       ALTER         BUILDKEYRECORDS    COPY
CREATE         DEALLOCATE    DUP[LICATE]        LOAD
LOADALTFILE    PURGE         PURGEDATA          RELOCATE
RENAME         RESTART

SUPPORTED FUP SESSION COMMANDS:
---------------------------------
CONFIG[URE]    DISPLAYBITS    SET    RESET
REPORTWIDTH

UNSUPPORTED FUP COMMANDS:
---------------------------
ALLOW     CHECKSUM    LISTLOCKS    LISTOPENS
RELOAD    STATUS      SUSPEND      TRUST
```

Commands

FC

Performs the standard, Nonstop Guardian fix command on the specified line. Note that unlike FUP, FC can operate on multi-line commands.

```
FC [ <number>    ]
   [ < -number > ]
   [ < text >    ]
   [ "< text >"  ]
```

HELP

Displays help text for various commands and topics. Note that SBFUP’s help command also wraps the FUP help command, so help text can be obtained for any SBFUP command as well as any of the original help text for FUP topics.

```
HELP [ / OUT listfile / ] [ SB | FUP ] [ ALL [ , SYNTAX ] | <topic> ]
```

SB_RESET

Resets the program settings back to their default values.

```
SB_RESET
```

See the following list of program settings that get reset and what their default values are:

PROGRAM SETTING	DEFAULT VALUE
WORKING SUBVOL	The vol and subvol SBFUP was started from
CONFIG FILE	<user's default $VOL.SUBVOL>.SBFPCSTM
CONTROL FILE	REMAINS UNCHANGED (No default)
MAPPING FILE	<current $VOL.SUBVOL>.MAP
TRACE FILE	<current $VOL.SUBVOL>.TRACE
LOG FILE	STD_OUT
CHECK MODE	OFF
ERROR MESSAGES	ON
INFO MESSAGES	ON
NAME MAPPING	ON
ALLOW COLLISIONS	OFF
REPLICATE ON ERRORS	OFF
TRACE MESSAGES	OFF
ALLOWERRORS	ON

SB_SHOW

Displays the current SBFUP session settings and program statistics, including:

- **Working Subvol:** the current working directory; changed via. the SYSTEM and VOLUME commands. Initialized on start-up to the directory from which SBFUP was started.
- **Config File:** the configuration file used to load program settings from on start-up. Defaults to <user's default sub-volume>.SBFPCSTM, whether or not it exists.
- **Control File:** The current control file SBFUP will write to after processing and executing a command. Does not have a default, will be prompted for if not specified prior to start-up.
- **Name Mapping File:** the current name mapping file used to map file and catalog names when processing FUP commands. Does not have a default, if not specified names will not be mapped.
- **AUDMON Control File:** the AUDMON control file. Used to determine if the user is permitted to execute SYSCALL commands.
- **Trace File:** the current tracing file. Defaults to <current sub-volume>.TRACE.
- **FUP Out File:** the current output destination for FUP results and errors. Defaults to STD_OUT.

- **Log File:** the current log file SBFUP will use to log messages to; defaults to STD_OUT.
- **FUP Process Name:** the process name used for the underlying FUP process spawned by SBFUP. In addition to the process name, this field will display whether or not the process name was system-generated or user-selected
- **Check Mode:** whether or not CHECKMODE is enabled
- **DEFMODE:** whether or not DEFMODE is enabled
- **Error Messages:** whether or not to log error messages to the log file/std out
- **Info Messages:** whether or not to log informational messages to the log file/std out
- **Map File Names:** whether or not to apply name mapping when processing FUP commands
- **Allow Mapping Collisions:** whether or not SBFUP should still attempt to map file names despite a detected collision
- **Replicate On Errors:** whether or not SBFUP should replicate erroneous commands
- **Trace Messages:** whether or not SBFUP should log trace messages to the TRACEFILE
- **Session Trans Timeout:** The timeout value used for the session transaction that is used to match SBFUP's FUP process with the FUPSERVER's FUP process.
- **Allow Errors:** the current ALLOWERRORS setting
- **Current Error Count:** the current error count as it pertains to the ALLOWERRORS setting;
    - Note that if ALLOWERRORS is set to ON this field will display N/A; otherwise the current error count is displayed.
- **Total Error Count:** the total number of errors encountered during program session. This count will always display the total number of errors encountered, regardless of the ALLOWERRORS setting or use of SB_RESET_ERRORS.

SB_SET

Sets various SBFUP program settings for the current session.

```
SB_SET < setting name > < value >
```

The following is a list of settable options:

```
  ALLOWCOLLISIONS      ALLOWERRORS          CHECKMODE     CONFIGFILE
  CONTROLFILE          DEFMODE              ERRORS        INFO
  LOGFILE              MAP                  MAPFILE       REPLICATEONERRORS
  TRACE                TRACEFILE
```

SYSCALL

Executes TACL commands on the target system. This command is not executed locally or mapped, so any filenames specified must be valid for the target system. The IN and

OUT files in the SYSCALL must be set to $NULL unless a command input file and outfile are assigned.

```
SYSCALL { ( <command> ) }
        { <filename>    }
```

The following arguments are supported:

- **(<command>)** – a single TACL command enclosed in parentheses that is executed
- **<filename>** – a filename containing TACL commands that are executed

In order to execute SYSCALL commands, the user that created this SBFUP process must be validated. By default, only SUPER.SUPER is permitted to execute SYSCALL commands. If AUDMON_CONTROL_FILE is specified in SBFUP's config file, the user that issued the START AUD command and any users specified in the AUD ALLOWUSERID parameter are also permitted.

! (replay)

Re-executes previously entered commands.

```
! {              }
  { < number >   }
  { < -number > }
  { < text >     }
  { "< text >"   }
```

- If no argument is supplied, ! will try to execute the last command entered.
- If a < number > is supplied, ! will execute the < number >th command, if it still exists in the command history.
    - Note that the command history stores only the last 100 commands.
- If a < -number > is supplied, ! will execute the command located at an offset from the last command (i.e. if there are 12 commands in the command history, and -5 is the argument, SBFUP will try to re-execute the 7th command in the command history.)
- If < text > is supplied, SBFUP will look through the command history from the most recently executed command, to the first command, and find a line whose start matches the given < text > input, and execute that command again.
- If “< text >” is supplied, SBFUP will look through the command history from the most recently executed command, to the first command, and find a line that contains the given < text > input, and execute that command again.

? (peek)

Peeks at the given, previously executed command, displaying it on the screen.

```
? {              }
  { < number >   }
  { < -number > }
  { < text >     }
  { "< text >"   }
```

- If no argument is supplied, ? display the last-entered command
- If a < number > is supplied, ? will display the < number >th command, if it still exists in the command history.
    - Note that the command history stores only the last 100 commands.
- If a < -number > is supplied, ? will display the command located at an offset from the last command (i.e. if there are 12 commands in the command history, and -5 is the argument, SBFUP will display the 7th command in the command history.)
- If < text > is supplied, SBFUP will look through the command history from the most recently executed command, to the first command, and find a line whose start matches the given < text > input, and display that command.
- If "< text >" is supplied, SBFUP will look through the command history from the most recently executed command, to the first command, and find a line that contains the given < text > input, and display that command.

Considerations

Qualified Fileset Lists

Replicated commands that accept qualified fileset lists as a parameter are evaluated differently.
The following commands are evaluated like this:

- ALLOCATE
- DEALLOCATE
- DUPLICATE
- PURGE
- PURGEDATA
- RENAME
- LICENSE
- REVOKE
- GIVE
- SECURE

These commands are handled by evaluating the qualified fileset list and replicating the commands for each individual file separately.

For example, the entered command:

```
PURGEDATA $VOL.SUBVOL.ABC*
```

Files belonging to the qualified fileset list above:

```
$VOL.SUBVOL.ABC1
$VOL.SUBVOL.ABC2
$VOL.SUBVOL.ABCFILE
$VOL.SUBVOL.ABCFILE2
```

Resulting commands that get replicated to the target (assuming name mapping is occurring and sub-volume SUBVOL is mapped to SUBVOL2):

```
PURGEDATA $VOL.SUBVOL2.ABC1
PURGEDATA $VOL.SUBVOL2.ABC2
PURGEDATA $VOL.SUBVOL2.ABCFILE
PURGEDATA $VOL.SUBVOL2.ABCFILE2
```

Note that HPE Shadowbase will coordinate for each individual command sent to the target, so for a PURGEDATA command that expands to 10 individual purgedata commands, 10 individual instances of coordination will occur.

LIKE Parameter

The LIKE parameter for the CREATE and SET commands cause those commands to be handled differently. Instead of replicating the command directly, SBFUP will create an equivalent command after executing the command locally. Because of this, the filename specified in the LIKE parameter does not need to exist on the target system in order to replicate a CREATE LIKE or SET LIKE command.

REPLICATEONERRORS

The REPLICATEONERRORS setting determines whether SBFUP will replicate commands that resulted in an error on the source. There are several situations that will never replicate, regardless of this setting:

- The source command exceeds the control file max data size.
- The error on the source-side command is syntax-related.
- The ALLOWERRORS limit is exceeded from the last source-side error.
- The command is a CREATE command containing a LIKE parameter.

- A command that takes a fileset list as a source returns an error 11.

CONFIGURE ALLOW

The ALLOW option of the CONFIGURE command is not supported. The SBFUP parameter ALLOWERRORS and the consumer parameter FUPERRORACTION should be used instead to handle errors. If entered, the follow warning message will be displayed:

```
SBFUP 1> CONFIGURE ALLOW ABENDS ON
SBFUP WARNING:
SBFUP does not support CONFIGURE ALLOW.
The SBFUP command SB_SET ALLOWERRORS should be used instead.
```

The SBFUP Command Line Interface

SBFUP Prompts

Most of the prompts from SBSQLCI apply to SBFUP as well. The only SBSQLCI prompt that does not apply for SBFUP is the prompt when attempting to EXIT with an outstanding transaction.

PURGE Prompts

For PURGE commands, SBFUP has two different prompts depending on what is being purged. If multiple filesets are specified, they are prompted for separately.

```
SBFUP 1> PURGE (FILE, FILE?)
              CODE             EOF      LAST MODIF   OWNER RWEP   TYPE   REC BL
$VOL.SUBVOL
 FILE                            0           14:57
PURGE?

DO YOU WISH TO PURGE THE ENTIRE FILESET $VOL.SUBVOL.FILE?
( Y[ES], N[ONE], S[ELECT], F[ILES] )?
```

If a single file is specified, type Y or y to purge the file. Any other response results in the file not being purged.

If a fileset is specified, the possible responses include:

- **YES:** accept the prompt, and SBFUP will attempt to purge all files within the given qualified fileset
- **NO:** decline the prompt, the fileset will not be purged
- **SELECT:** individually prompt for each file within the qualified fileset
- **FILES:** view the full list of files potentially affected by the PURGE, and re-prompt

BREAK and EOF Key

Receiving the EOF signal during the standard prompt will cause SBFUP to gracefully shut down. Receiving the EOF signal during a PURGE prompt or a COPY without an in-file specified will cause SBFUP to return to the standard prompt like with FUP.

Receiving the BREAK key signal will cause SBFUP to move to the background and return the user to the TACL prompt, regardless of the current prompt SBFUP is on. A PAUSE command on the TACL prompt will return the user to the SBFUP session.

Interacting with DCR Through AUDCOM

STATUS AUD

The STATUS AUD command provides general information about the current HPE Shadowbase environment, including what objects are configured, under which AUDMON process, whether each object is running and in what state it's in, etc. With respect to the DCR subsystem:

For FUP DCR, the STATUS AUD command will display any FUP processes that the FUPSERVER consumer has spawned, as seen below:

```
NAME              PROCESS           TYPE  STATE      STATE CHANGE TIME
----------------  ----------------  ----  ---------  -------------------
  FEP-Y764        \VIV1.$KTF00      FUP   RUNNING    2022-08-22 14:05:24
```

STATUS CONS

The STATUS CONS command displays relevant status information for the given consumer(s), including whether certain files are open or closed, whether the consumer is running, how many busy transactions the consumer currently has, etc.

For FUP DCR, the STATUS CONS command will display a line denoting the given consumer is a FUPSERVER and its status. Below that, the command will display any FUP processes this consumer has spawned. In addition, a line will display showing the current COORDINATION status for the FUPSERVER consumer. Finally, a line will display for the FUPLOG file, showing whether it's open or closed, what the file is, and whether there was an open error, or write error (see highlighted):

```
CONSNAME          PROCESS           STATE      TRANS BUSY             MSGSEQ
----------------  ----------------  ---------  ----------  -----------------
CONS-DIR-KTFPC    \VIV1.$KTFPC      RUNNING             2                 37
  (FUPSERVER)                       PAUSED
  FEP-Y764        \VIV1.$KTF00      RUNNING
  COORDINATION     WAITING ON RESUMEDDL COMMAND
  ENTSEQCHKPTFILE  CLOSED
  EMS              OPENED  $0                                   [ERROR=0]
  IOTRACEFILE      CLOSED
  KREP PROCESS     CLOSED
  FUPLOG           OPENED  \VIV1.$KMT.DCR.FUPLOGD               [ERR o:0 w:0]
  OVFLQFILE        OPENED  \VIV1.$KMT.DCR.OKTFPC                [ERROR=0]
  REJECTFILE       CLOSED
  SBCMDFILENAME    CLOSED
  TIDFILE          OPENED  \VIV1.$KMT.DCR.TID2                  [SQLCODE=0]
  TMF-TFILE        OPENED  $TMP  [MAXTRANS=1000]
  TRACE            CLOSED
  TRACKTXFILE      OPENED  \VIV1.$KMT.DCR.TID2X                 [ERROR=1]
  TRANSLOG         CLOSED
```

Note that:

- A FUPSERVER consumer’s state can display one of two values: IDLE or PAUSED.
- The COORDINATION status will display WAITING ON RESUMEDDL COMMAND only when the collector is waiting on coordination to complete, and the associated FUPSERVER consumer is in the PAUSED state and waiting for RESUMEDDL; otherwise, this line does not display.

STATS CONS

The STATS CONS command display a wide array of information; statistics pertaining not only to regular audit data replication, but other facilities such as SOLV loading info, SQL lock info, commit wait time info, etc.

For DCR, the STATS CONS will display FUP DCR stats for the FUPSERVER consumer, shown below:

```
FEP SESSION SUMMARY INFO:
  CUR BUSY  MAX BUSY    TOT SESS  MIN CMND  MAX CMND  AVG CMND    TOT CMND
  --------  --------  ----------  --------  --------  --------  ----------
         0         0           0         0         0         0           0
     LAST START        MIN DUR       MAX DUR       AVG DUR       SESS ERRS
  ----------------  ------------  ------------  ------------    ----------
  0000-00-00 00:00  00:00:00.000  00:00:00.000  00:00:00.000             0
      LAST CMND      TOT CMND DUR  %CMND BUSY    CMND ERRS   CMND SKIPS
  ----------------  -------------  ----------   ----------   ----------
  0000-00-00 00:00  000:00:00.000        0.00            0            0
  ------------------------ LAST FUP COMMAND TEXT -------------------------
  …
  ------------------------ LAST FUP COMMAND REPLY ------------------------
  |…|…|
FEP ACTIVITY INFO (IN-PROGRESS):
  FEP PROCESS: \VIV1.$Y764  TYPE: FUP  %CMND BUSY:   0.00
    SESS START TIME     SESS DUR    TARGET PROCESS    CMND BUSY  TOT CMND
    ----------------  ------------  ----------------  ---------  --------
    0000-00-00 00:00  00:00:00.000  \VIV1.$KTF00          N             0
        LAST CMND       MIN DUR      MAX DUR      AVG DUR    TOT CMND DUR
    ---------------- ------------ ------------ ------------ -------------
    0000-00-00 00:00 00:00:00.000 00:00:00.000 00:00:00.000 000:00:00.000
    ----------------------- LAST FUP COMMAND TEXT -------------------------
    …
    ----------------------- LAST FUP COMMAND REPLY ------------------------
    |…|…|
```

The FUP DCR information is split into two sections: the FEP Session Summary Info and the FEP Activity Info. The FEP Session Summary Info contain the following statistics about the FUPSERVER consumer as a whole:

- **CUR BUSY:** The number of FUP sessions currently active
- **MAX BUSY:** The largest that CUR BUSY has been over time
- **TOT SESS:** The total number of sessions over time
- **MIN CMND:** For all sessions to date, the minimum command count
- **MIN CMND:** For all sessions to date, the maximum command count
- **AVG CMND:** Average command count per session
- **TOT CMND:** Total number of commands for all sessions
- **LAST START:** Local timestamp of the last time a session was started
- **MIN DUR:** The duration of the shortest lasting session
- **MAX DUR:** The duration of the longest lasting session
- **AVG DUR:** The average duration of a session
- **SESS ERRS:** The total number of FEP DCR errors
- **LAST CMND:** Local timestamp of the last executed command
- **TOT CMND DUR:** Total time the FUPSERVER spent executing commands
- **%CMND BUSY:** How much time has the FUPSERVER spent executing commands, as a percentage of time running
- **CMND ERRS:** Total number of FUP errors that have been encountered
- **CMND SKIPS:** Total number of FUP commands that have been skipped
- **LAST FUP COMMAND TEXT:** The most recent command sent to any FUP process

- **LAST FUP COMMAND REPLY:** The most recent reply received from a FUP process

The FEP Activity Info section contains information about a single FUP session and appears once for each active session.

- **FEP PROCESS:** The name of the SBFUP process that is associated with this FUP process
- **%CMND BUSY:** How much time has the FUPSERVER spent executing commands using this session, as a percentage of time running
- **SESS START TIME:** Local timestamp of when this FUP session was started
- **SESS DUR:** Duration of this session at this current time
- **TARGET PROCESS:** The name of the spawned FUP process
- **CMND BUSY:** "Y" if this session is currently executing a command, "N" otherwise
- **TOT CMND:** Total number of commands that this session has received
- **LAST CMND:** Local timestamp of the last command received
- **MIN DUR:** The duration of the shortest command to execute
- **MAX DUR:** The duration of the longest command to execute
- **AVG DUR:** The average duration of a command for this session
- **TOT CMND DUR:** The total time this session has spent executing commands
- **LAST FUP COMMAND TEXT:** The most recent command sent to this FUP process
- **LAST FUP COMMAND REPLY:** The most recent reply received from this FUP process

On startup, the target FUP process will automatically execute the command "CONFIG ALLOW ABENDS OFF". This command will affect the STATS output

Configuring HPE Shadowbase for FUP DDL Command Replication

Back-end Configuration via SBUNIDIR

FUP DCR Replication Parameters

PARAMETER NAME	DEFAULT VALUE
CONTROLFILE	None
CONTROLFILERECSIZE	LEGACY
COORDINATIONMSG	WAITWARN
COORDWAITINGWARNRATE	300 Seconds
FUPABENDACTION	STOP
FUPCOORDINATION	ON
FUPERRORACTION	PAUSE
FUPLOG	None – (No FUPSERVER logging if omitted)
FUPPRI	-1 (FUPSERVER's priority used)

FUPPROCESS	None (REQUIRED)
FUPPROGRAM	Default FUP program is run ($SYSTEM.SYS*nn*.FUP)
FUPSERVER	OFF if omitted (and DCR will not happen)

A more detailed overview of each parameter can be found in the *HPE NonStop Shadowbase Command Reference Manual*. Notes about each parameter related to DCR are as follows:

CONTROLFILE: Configures the location of the CONTROLFILE used to facilitate DDL Command Replication. The control file must be configured for the MPSERVER consumer in order for DCR to work.

CONTROLFILERECSIZE: Configures the size of the control file HPE Shadowbase generates automatically on start-up.

COORDINATIONMSG: Determines under what circumstances the collector will issue coordination-related EMS messages.

COORDWAITINGWARNRATE: Configures an interval of time that the collector waits between logging coordination messages to EMS.

FUPABENDACTION: Configures how the FUPSERVER consumer responds to FUP processes abending unexpectedly.

FUPCOORDINATION: Configures whether FUP DCR events trigger HPE Shadowbase coordination or not

FUPERRORACTION: Configures how the FUPSERVER consumer responds to FUP errors during FUP DCR events.

FUPLOG: Configures the location of the FUPSERVER consumer log file. This file will contain a log of what commands were received from the upstream SBFUP and Collector, how the underlying FUPSERVER commands executed / FUP process responded, etc.

FUPPRI: Configures the priority of the underlying FUP processes that the FUPSERVER consumer spawns.

FUPPROCESS: Configures the prefix of the process name of the FUP processes spawned by the FUPSERVER consumer. The process portion of the name may be up to 4 characters, including the $.

FUPPROGRAM: Configures the location of the desired FUP program file.

FUPSERVER: Configures whether the given consumer is an FUPSERVER consumer.

Front-end Configuration via Config File

Use the name "FUPDCR" to denote the section containing configuration parameters for SBFUP.

```
[FUPDCR]
.
.
.
```

SBFUP Configuration Parameters

The following configuration parameters alter different aspects of SBFUP. These parameters belong under the configuration file header [FUPDCR].

PARAMETER NAME	DEFAULT VALUE
ALLOWERRORS	ON
ALLOWCOLLISIONS	FALSE
AUDMONCONTROLFILE	None
CHECKMODE	FALSE
CONFIGFILE	<user's default sub-volume>.SBFPCSTM
CONTROLFILE	None – (will be prompted for a valid control file)
DEFAULTVOLUME	The volume from which SBFUP is started
DEFMODE	TRUE
ERRORS	TRUE
FUPOUT	STD_OUT
FUPPROCESSNAME	None – (system-generated process name used)
INFO	TRUE
LOGFILE	STD_OUT
MAPFILE	None – (no name mapping if omitted)
MAP	TRUE
REPLICATEONERRORS	FALSE
SESSIONTRANSTIMEOUT	0
SUPPRESSCTRLFILEWARNING	FALSE
TRACEFILE	<current sub-volume>.TRACE
TRACE	FALSE

The following is a more detailed overview of each parameter:

AUDMONCONTROLFILE <filename>

Initial	Changed			
Version	Version	Values Default	Value Specifications	Alter
7.000		None	Valid Guardian File Name	No

Specifies the location of the AUDMON Control File. If included, SBFUP will determine if the user is permitted to execute SYSCALL commands. The users allowed are the user that issued the START AUD command and any user specified by the AUD ALLOWUSERID parameter. SUPER.SUPER is always allowed to execute SYSCALL commands, even if this parameter is omitted.

FUPOUT <filename | STD_OUT>

Initial Version	Changed Version	Values Default	Value Specifications	Alter
7.000		TERMINAL	Valid Guardian File Name or the keyword STD_OUT	No

Configures the location to which FUP output should go. This parameter is optional and defaults to STD_OUT.

FUPPROCESSNAME [\NODE.]<$name>

Initial Version	Changed Version	Values Default	Value Specifications	Alter
7.000		System Generated	[\NODE.]<$name>	No

Configures the process name of the underlying FUP process spawned by SBFUP. This parameter is optional and if omitted, a system-generated process name will be used. Note that you can use the SB_SHOW command within SBFUP to see the process name of the underlying FUP process. If this parameter is supplied, and the given process name is already taken by another Guardian process, SBFUP will notify the user and use a system-generated process name instead.

SESSIONTRANSTIMEOUT < number >

Initial Version	Changed Version	Values Default	Value Specifications	Alter
7.000		0	-1 - 21474836	No

Configures the Auto Abort timer for the session transaction that is used to pair SBFUP’s FUP process with one of the FUPSERVER’s FUP processes. This effectively functions as a limit to how long a single SBFUP process can stay active at a time. A value of 0 will use the system configured Auto Abort timer. A value of -1 will cause the transaction to never time out. A value larger than 0 will cause the transaction to time out after that many seconds.

Specifying Configuration File Location

SBFUP will check the default location for the configuration file (<default subvol>.SBFPCSTM). This location and file name can be changed with the following:

1. Configuring the TACL define =SBFUPCONFIG with the full path to the file

```
== On the TACL command prompt
ADD DEFINE =SBFUPCONFIG, CLASS MAP, FILE $VOL.SUBVOL.FILE

== In SBUNIDIR script
SB_ADD TACL_DEFINE * =SBFUPCONFIG, CLASS MAP, FILE $VOL.SUBVOL.FILE
```

2. Running SBFUP and executing the SB_SET CONFIGFILE <name> command

```
SBFUP 1> SB_SET CONFIGFILE $VOL.SUBVOL.FILE;
Config file set to \NODE.$VOL.SUBVOL.FILE;
```

Note that in order to overwrite the current program settings, you must run SB_LOAD after setting the config file in this way.

```
SBFUP 1> SB_SET CONFIGFILE $VOL.SUBVOL.FILE;
Config file set to \NODE.$VOL.SUBVOL.FILE;

SBFUP 2> SB_LOAD
Loaded program options from \NODE.$VOL.SUBVOL.FILE;
```

FUP Configuration Quick-Start

The following section gives a quick demonstration of setting up a simple NonStop to NonStop DDL Command Replication environment over EXPAND. The demonstration sets up a unidirectional environment via SBUNIDIR and uses default values wherever it can in the configuration.

Configure HPE Shadowbase via SBUNIDIR

```
?tacl macro

#frame
#push sb_src_install sb_tgt_install
#set  sb_src_install \NODE1.$VOL1.SBINST
#set  sb_tgt_install \NODE2.$VOL2.SBINST

load /keep 1/[sb_src_install].SBUNIDIR

SB_SET SB_CONFIGURATION_TYPE DIRECT

SB_ADD SRCNODE \NODE1 [sb_src_install] \NODE1.$VOL1.SBDATA &
                      ~[#MYTERM~] [sb_src_install].SHADPASS  &
                      $SBCL 0 1 145 $SBMN 0 1 145

SB_ADD TGTNODE \NODE2 [sb_tgt_install] \NODE2.$VOL2.SBDATA

== FUPSERVER CONSUMER ==
SB_ADD CONS $FPCNS 1 0 145 \NODE2.$VOL2.SBDATA.TID1

SB_ADD CONS_PARAM $FPCNS * TRACKTXFILE \NODE2.$VOL2.SBDATA.TTX
SB_ADD CONS_PARAM $FPCNS * CONTROLFILE \NODE1.$VOL1.SBDATA.CTRL

== REQUIRED FUP DCR PARAMETERS ==

SB_ADD CONS_PARAM $FPCNS * FUPSERVER ON
SB_ADD CONS_PARAM $FPCNS * FUPPROCESS $FUP

== REGULAR CONSUMER
SB_ADD CONS $ENSCN 1 0 145 \NODE2.$VOL2.SBDATA.TID2

SB_ADD DBS DBS1 \NODE1.$VOL1.SUBVOL.ENSSRC $ENSCN &
                \NODE2.$VOL2.SUBVOL.ENSTGT - -

SB_GEN

#unframe
```

Configure SBFUP Input File

```
### $VOL.USER.SBFPCSTM – SBFUP CONFIGURATION FILE

[FUPDCR]
CONTROLFILE = \NODE1.$VOL1.SBDATA.CTRL
MAPFILE = \NODE1.$VOL1.SBDATA.MAP
```

Configure the SBFUP Name Mapping File

```
=== DCR NAME MAPPING FILE

MAP NAME $VOL1.SUBVOL.ENS* TO $VOL2.ENSRECS.*
```

Open Another Terminal Window and Run EMS to Monitor FUPSERVER

```
> LOAD /KEEP 1/ $VOL.SUBVOL.SBUNIDIR

.
.
.

> SB_EMS $FP*
```

Run HPE Shadowbase, then Start SBFUP and Perform DCR

```
> SRSTRTSB

Y
Y
.
.
.

> SBFUP

SBFUP 1>
```

New and Modified Commands

SBUNDROS SB_ADD SBMON_PARAM (new)

The SBMON_PARAM command overrides the default settings for the specified DOC Writer, TRS, DOC Cleaner, or Direct Writer. Overrides can be made to the SBMON's ADD command for the specified object, or overrides can be made to the default object parameters contained in the .ini file.

Alterations to the Consumer cannot be made with SBMON_PARAM.

Depending on the override specified, the override is added to the Guardian file *SBMON, the Guardian file *SPINI and OSS file shadparm.ini, or both.

The syntax for SB_ADD SBMON_PARAM is as follows:

```
SB_ADD SBMON_PARAM         <src-cons-pname|*> <doc-trs-dir-dcl|*>
                           <p-b|*> <param-name-and-value>

    where

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

       <doc-trs-dir-dcl>   specifies which process the override applies to.
                           Valid options are DOC (DOC Writer), TRS (Transaction
                           Replay Server), DIR (Direct Writer), and DCL (DOC
                           Cleaner). This specifier can be wildcarded with an
                           asterisk.

       <p-b>               specifies if the override applies to the primary DOC
                           or DCL, or the backup DOC or DCL. Valid options are P
                           for primary and B for backup. This specifier can be
                           wildcarded with an asterisk.

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

ESTATS (modified)

```
ESTATS [/out <listfile>] {[AUD] [<audmon-name>]
                             [,RESET]
                             [,INTERVAL  <num> {HRS}
                                               {MINS}
                                               {SECS}]
                             [,{COLLNAME [<audmon-name>.]<collname>}
                               {SERVERS} ]
                             [,COLLSTATS {OFF}
                                         {ALL}
                                         {INCLUDE}
                                         {EXCLUDE} ]
                             [,CONSSTATS {OFF}
                                         {ALL}
                                         {DBSSTATS}
                                         {FILESTATS} ]
                             [,BRIEF]
                             [,LISTCOUNT <num>]
                             [,SORT {NAME}
                                    {ACTIVITY}
                         {[DBS] {<dbs-name> | *}
                             [{,FILESTATS}
                              {,DBSSTATS}
                              {,ALL} ]
                             [,CONSNAME <consumer-name>]
                             [,COLLNAME <collector-name>] }
                             [{,MATRIX}
                              {,BRIEF}]
                             [,SERVERS]
                             [,LISTCOUNT <num>]
                             [,SORT {NAME}
                                    {ACTIVITY}

Displays extended resource usage and system performance statistics.
-----------------------------------------------------------------------------
```

BRIEF – If the BRIEF option is specified, the statistics are limited to one line per DBS or file. For ESTATS DBS, the BRIEF option cannot be used with the MATRIX option.

LISTCOUNT – This option will limit the ESTATS report to only show the first DBSes/files based on the value entered. The number entered must be greater than 0.

SORT – This option controls the order in which entries of a single consumer are output. With the NAME option, entries are sorted by DBS name or source file name. With the ACTIVITY option, entries are sorted by total number of events. By default, sorting by name is used.

SBUNDROS SB_ADD CONSUE (new)

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

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

The syntax for SB_ADD CONSUE is as follows:

```
SB_ADD CONSUE       <cons-pname> <cons-cpu-p> <cons-cpu-b> <cons-pri>

    where

       <cons-pname> is a unique process name used for the consumer(audconxn).

       <cons-cpu-p> is a valid CPU number for the source consumer’s primary
                    CPU.

       <cons-cpu-b> is a valid CPU number for the source consumer’s backup
                    CPU (use -1 for no backup).

       <cons-pri>   is the priority number of the consumer.
```

SBUNDROS SB_ADD CONSUE_PARAM (new)

The CONSUE_PARAM command overrides the default Consumer parameters for Consumers configured with the SB_ADD CONSUE command.

The syntax for SB_ADD CONSUE_PARAM is as follows:

```
SB_ADD CONSUE_PARAM <cons-pname | *> <param-name-and-value>

    where

       <cons-pname>           is an existing source Consumer process name. This
                              id can also be wild carded with a single star.

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

New and Modified SET AUD Parameters

None in this release.

New and Modified SET COLL Parameters

None in this release.

New and Modified SET CONS Parameters

FUPCOORDINATION	{ ON }
	{ OFF }

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

Determines whether FUP DCR events will trigger HPE Shadowbase coordination. This parameter is optional, and defaults to ON. Coordination occurs when an FUP DCR event is read from the audit trail, causing the collector to stop reading audit and wait for downstream consumers to drain (close target files, free resources, etc.). SET, CONFIGURE, and RESET commands do not cause coordination, even if FUPCOORDINATION is set to ON. Once drained, the FUPSERVER consumer processes the FUP DCR event that triggered the coordination, and upon completion, the collector resumes reading audit and sending new audit data to its downstream consumers. In most cases it is recommended to leave FUPCOORDINATION ON if processing FUP DCR events.

FUPERRORACTION	{ PAUSE }
	{ SKIP }
	{ SKIPNOLOG }
	{ STOP }

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

Determines how the FUPSERVER consumer responds to errors encountered while processing FUP DCR events. This parameter is optional and defaults to PAUSE if omitted. The parameter values are:

- **PAUSE:** wait until the user issues a RESUMEDDL command via AUDCOM
- **SKIP:** log a message to the log file (denoted by the FUPLOG parameter,) skip the erroneous FUP DCR event, and continue running
- **SKIPNOLOG:** skip the erroneous FUP DCR event and continue running
- **STOP:** log a message and abend

Note that while the FUPSERVER consumer is in the PAUSED state, the upstream collector is not reading audit, as it is waiting for the consumer to finish processing the DCR event. Periodic EMS messages will be logged as the FUPSERVER consumer continues to wait for a RESUMEDDL command.

FUPLOG { filename }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
6.800			None	n/a	n/a	NO

Configures the location of the FUPSERVER log file. This parameter has no default value and if not set, the FUPSERVER consumer will not log DCR-related informational messages.

FUPPRI { number }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
6.800			None	1	199	NO

Configures the priority of the underlying FUP processes spawned by the FUPSERVER consumer. This parameter is optional, and if omitted, the priority of the FUPSERVER consumer is used.

FUPPROCESS { processname-prefix }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
6.800			None	n/a	n/a	NO

Configures the prefix for the process names of the underlying FUP processes spawned by the FUPSERVER consumer whenever an SBFUP session is started. The process portion of the name may only be a maximum of 4 characters long including the $ sign. The actual process name will be the prefix, followed by a two-digit identifier starting at 00, and increasing by one for every additional spawned FUP process. This parameter is required, and must be specified in \NODE.$PROC format specifying the name of the target system.

FUPPROGRAM { filename }

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
6.800			$SYSTEM.SYS*nn*.FUP	n/a	n/a	NO

Configures the location of the FUP program file that the FUPSERVER consumer will use to spawn a FUP session whenever an SBFUP session is started. This parameter is optional, and if omitted, the FUPSERVER uses $SYSTEM.SYS*nn*.FUP. $SYSTEM.SYS*nn* is the subvolume containg the operating system image that is currently in use.

FUPSERVER { ON }
{ OFF }

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

Designates the given Consumer as a FUPSERVER consumer. FUPSERVER consumers are capable of processing FUP DCR events. At least one consumer within an HPE Shadowbase configuration must be configured with this parameter for FUP DCR to be enabled and work properly. It is recommended to configure a dedicated FUPSERVER consumer, and let other consumer(s) in the environment perform regular audit data replication.

FUPABENDACTION { STOP }
{ SKIPALL }

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

Determines how the FUPSERVER consumer responds to its spawned FUP processes abending unexpectedly. This parameter is optional and defaults to STOP if omitted. The parameter values are:

- **STOP:** The consumer abends. This will allow the FUP session to replay any SET commands before continuing replication.
- **SKIPALL:** The consumer continues running, but any future FUP commands meant for this session are logged and skipped.

IOTRACEFORCEFLUSH { ON }
{ OFF }

Initial	Changed				
Version	Version	Values Default	Minimum	Maximum	Alter
6.800		OFF	n/a	n/a	YES

If set to ON, every IOTRACE will force a flush to the IOTRACEFILE. The IOTRACEFORCEFLUSH parameter can be altered while the Consumer is running.

FILTEREMPTYTX { ON }
{ OFF }

Initial	Changed				
Version	Version	Values Default	Minimum	Maximum	Alter
6.800		OFF	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

KEYSPECIFIER <key-specifier>

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
4.090	6.800			-32768	32767	No

This parameter is used to identify the key-specifier associated with a target file alternate key file to be used when applying events. This will be made available to BASE24 handler code in the Consumer to aid in identifying a unique record in a target file. This parameter can also be used with an alternate key on Enscribe entry-sequenced files without the need for BASE24 processing to be enabled or a unique index on SQL/MP entry-sequenced table.

If this parameter is used without BASE24, the source file must not be auditcompressed and BEFOREVALUES must be ON.

The value can be entered as a 2 byte string delimited by double quotes (e.g., “A” or “I1” or “IX”, etc.) or can be specified as a number in the range -32768 to 32767 similar to how the key-specifier can be defined within the NonStop FUP utility.

> **Note**: By default, HPE Shadowbase uses the primary key value of the source record/row to locate the correct target record/row to process.

CREATELIKE <file name>

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

When replicating Enscribe create events, the CREATELIKE file's attributes are used instead of the source's. Any secondary partition volumes in the CREATELIKE file’s attributes will be used to create the new target file. Any alternate key files will be created on the target node, with the volume and subvolume of the corresponding CREATELIKE alt file. If CREATELIKEALTPREFIX is omitted or set to 0, the name part of the target alt files will match the corresponding source alt file names and the CREATELIKE file cannot have more alt files than the source file.

For example, consider a Shadowbase environment with the following DBS parameters:

```
SOURCEFILE \SRC.$DATA1.SRCSUB.TL*
TARGETFILE \TGT.$DATA1.TGTSUB.*
CREATELIKE \TGT.$DATA1.MAPSUB.TLYYMMDD
```

The CREATELIKE file is defined with the following attributes:

```
$DATA1.MAPSUB.TLYYMMDD
    ENSCRIBE
    TYPE E
    FORMAT 2
    EXT ( 100 PAGES, 100 PAGES )
    REC 4048
    BLOCK  4096
    ALTKEY ( "CR", FILE 0, KEYOFF 38, KEYLEN 30 )
    ALTKEY ( "PR", FILE 1, KEYOFF 0, KEYLEN 3 )
    ALTKEY ( "TR", FILE 2, KEYOFF 14, KEYLEN 24 )
    ALTFILE ( 0, $DATA1.TGTSUB.T0YYMMDD )
    ALTFILE ( 1, $DATA1.TGTSUB.T1YYMMDD )
    ALTFILE ( 2, $DATA1.TGTSUB.T2YYMMDD )
    PART ( 1, $DATAX, 100 PAGES, 100 PAGES )
    PART ( 2, $DATAY, 100 PAGES, 100 PAGES )
    PART ( 3, $DATAZ, 100 PAGES, 100 PAGES )
    AUDIT
    MAXEXTENTS 16
    BUFFERED
    OWNER 255,5
    SECURITY (RWEP): NCNC
```

While Shadowbase is running, a new file is created on the source system like so:

```
$DATA1.SRCSUB.TL230101
    ENSCRIBE
    TYPE E
    FORMAT 2
    EXT ( 100 PAGES, 100 PAGES )
    REC 4048
    BLOCK  4096
    ALTKEY ( "CR", FILE 0, KEYOFF 38, KEYLEN 30 )
    ALTKEY ( "PR", FILE 1, KEYOFF 0, KEYLEN 3 )
    ALTKEY ( "TR", FILE 2, KEYOFF 14, KEYLEN 24 )
    ALTFILE ( 0, $DATA1.SRCSUB.T0230101 )
    ALTFILE ( 1, $DATA2.SRCSUB.T1230101 )
    ALTFILE ( 2, $DATA3.SRCSUB.T2230101 )
    PART ( 1, $DATA1, 100 PAGES, 100 PAGES )
    PART ( 2, $DATA2, 100 PAGES, 100 PAGES )
    PART ( 3, $DATA3, 100 PAGES, 100 PAGES )
    AUDIT
    MAXEXTENTS 16
    BUFFERED
    OWNER 255,5
    SECURITY (RWEP): NCNC
```

When Shadowbase replicates this create event, the CREATELIKE file results in the following file created on the target:

```
$DATA1.TGTSUB.TL230101
    ENSCRIBE
    TYPE E
    FORMAT 2
    EXT ( 100 PAGES, 100 PAGES )
    REC 4048
    BLOCK  4096
    ALTKEY ( "CR", FILE 0, KEYOFF 38, KEYLEN 30 )
    ALTKEY ( "PR", FILE 1, KEYOFF 0, KEYLEN 3 )
    ALTKEY ( "TR", FILE 2, KEYOFF 14, KEYLEN 24 )
    ALTFILE ( 0, \TGT.$DATA1.TGTSUB.T0230101 )
    ALTFILE ( 1, \TGT.$DATA1.TGTSUB.T1230101 )
    ALTFILE ( 2, \TGT.$DATA1.TGTSUB.T2230101 )
    PART ( 1, $DATAX, 100 PAGES, 100 PAGES )
    PART ( 2, $DATAY, 100 PAGES, 100 PAGES )
    PART ( 3, $DATAZ, 100 PAGES, 100 PAGES )
    AUDIT
    MAXEXTENTS 16
    BUFFERED
    OWNER 255,5
    SECURITY (RWEP): NCNC
    DATA MODIF:   2 Feb 2023, 12:31
    CREATION DATE:   2 Feb 2023, 12:31
    LAST OPEN:   2 Feb 2023, 12:31
    FILE LABEL: 546 (13.4% USED)
    EOF: 0 (0.0% USED)
    EXTENTS ALLOCATED: 0
```

The filenames of the base file and all altfiles match the source file, but all the other attributes match the CREATELIKE file, including partition volumes and altfile volumes and subvolumes.

CREATELIKEALTPREFIX <number>

Initial	Changed					
Version	**Version**	**Values**	**Default**	**Minimum**	**Maximum**	**Alter**
6.800		0		0	7	No

When this parameter is set greater than 0, the target alt file names will be generated from the corresponding CREATELIKE alt file name and the source base file name. When used, the name will be the first <number> characters from the CREATELIKE alt file, followed by the rest of the source base file.

HASHBUCKETCOUNT <number>

Initial	Changed					
Version	**Version**	**Values**	**Default**	**Minimum**	**Maximum**	**Alter**
6.800		0		0	2147483647	No

This parameter is used with HASHBUCKETID in order to split replication of incoming events between multiple replication threads. This parameter represents the total number of replication threads that events for this DBS will be split between. If a source table is a part of multiple DBSes in the same consumer, this parameter must be the same for each DBS. This parameter is only supported when running SBCOLL.

Note: This feature cannot be used when replicating entry-sequenced files using the ENTRYSEQEXACT method. For entry-sequenced files, this will cause events to very likely be inserted into the target in a different order from the events in the source file. If ordering is important, using a KEYSPECIFIER on a unique alternate key is required for maintaining ordering.

HASHBUCKETID <number>

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
6.800		-1		-1	2147483647	No

This parameter is used with HASHBUCKETCOUNT in order to split replication of incoming events between multiple replication threads. This parameter represents which hash bucket this DBS corresponds to. If this parameter is set greater than -1 and HASHBUCKETCOUNT is set greater than 0, the collector will run a hashing algorithm on values from all columns belonging to the primary key of the source table in order to determine which bucket the event belongs to. The event will only be sent down replication threads configured as that bucket. If a source table is a part of multiple DBSes in the same consumer, this parameter must be the same for each DBS. This parameter must be less than HASHBUCKETCOUNT. This parameter is only supported when running SBCOLL.

Note: This feature cannot be used when replicating entry-sequenced files using the ENTRYSEQEXACT method. For entry-sequenced files, this will cause events to very likely be inserted into the target in a different order from the events in the source file. If ordering is important, using a KEYSPECIFIER on a unique alternate key is required for maintaining ordering.

```
FETCHSOURCE   {ON}
              {OFF}
              { ( ON [, fetchsource-option] ...)}
                fetchsource-option is:
                  one of the following warning options:
                        [MISSINGSRC]WARN
                        [MISSINGSRC]NOWARN (default)
                  or one of the following behavior options:
                        [MISSINGSRC]ABEND
                        [MISSINGSRC]SKIP (default)
                        [MISSINGSRC]SENDCMP
                  or one of the following image usage options:
```

[ALWAYSUSE]AUDIT (default)
[ALWAYSUSE]SOURCE
or one of the following short record warning options:
SHORTRECWARN
SHORTRECNOWARN (default)
or one of the following short record behavior options:
SHORTRECABEND
SHORTRECSKIP
SHORTRECSENDCMP
SHORTRECSENDSRC (default)
or one of the following force fetch behavior options:
FORCEFETCHON
FORCEFETCHOFF (default)

Initial Version	Changed Version	Values	Default	Value Specificatons	Alter
5.001	6.800	OFF		**Valid FETCHSOURCE options**	No

…

The force fetch behavior options determine when the consumer will fetch the full row/record from the source. If FORCEFETCHON is set, the consumer will fetch the row/record for every compressed update. If FORCEFETCHOFF is set, the consumer will only fetch the row/record if the uncompressed row/record is required.

New and Modified SOLV AND SOLVNV Parameters

None in this release.

New or Modified EMS Messages

2125	UNABLE TO OBTAIN PARTITION INFO FOR SOURCEFILE <filename>(<sourcefile>), FILE ERROR NO <error number>

Cause: The DBS ALLPARTITIONS parameter is set ON or the DBS PURGEDATAOPTIONS parameter includes MAPTOPRIPTN, but AUDMON had a file error while attempting to acquire partition information.

Effect: The DBS object is not added.

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

2185	CREATELIKE MUST BE SET IF CREATELIKEALTPREFIX IS SET

Cause: The DBS parameter CREATELIKEALTPREFIX is set greater than 0, but the DBS parameter CREATELIKE is not set ON.

Effect: The DBS object is not added.

Recovery: Set CREATELIKE ON or set CREATELIKEALTPREFIX to 0.

2095	FILEMNT: UNABLE TO MAP ALTKEY FILES FOR SOURCE <filename>, CREATELIKE <filename>, AND TARGET <filename>, TARGET FILE WILL NOT BE CREATED

Cause: The Consumer encountered an error while attempting to map the alt key file names while replicating an Enscribe create event.

Effect: The target file will not be created.

Recovery: Check your configuration.

2095	FILEMNT: FAILED TO CREATE <filename> FROM SOURCE FILE <filename> AND CREATELIKE FILE <filename>

Cause: The Consumer encountered an error while replicating an Enscribe create event.

Effect: The target file will not be created.

Recovery: Check your configuration.

2095	FILEMNT: UNABLE TO EXTRACT CREATELIKE NODE FROM <filename>; TARGET FILE <filename> WILL NOT BE CREATED

Cause: The Consumer was unable to determine the node name from the reported file.

Effect: The target file will not be created.

Recovery: Check your configuration.

2008	ATTEMPTED TO POSITION BY ALTKEY FOR ENTRY-SEQUENCED FILE UPDATE VIA KEY SPECIFIER [<key-spec>], BUT TARGET FILE [<file>] HAS NO ALTKEY WITH THAT KEY SPECIFIER.
2008	ATTEMPTED TO POSITION BY ALTKEY FOR ENTRY-SEQUENCED FILE UPDATE VIA KEY SPECIFIER [<keyspecifier>], BUT TARGET FILE [<file>] HAS NO ALTKEY WITH THAT KEY SPECIFIER. IF REPLICATING DYNAMICALLY CREATED FILES, ALSO CHECK THE TEMPLATE FILE [<file>].

Cause: The specified file does not have an altfile with the specified key specifier. If the DBS parameter MAPFROMSOURCE is ON, the second message will be displayed instead. In that case, the specified template file must also have an altfiles the with key specifier.

Effect: The consumer abends.

Recovery: Modify the KEYSPECIFIER DBS parameter to match one that your TARGETFILE has.

4546	QMGR <name> (<process>): QUEUE IS FULL DUE TO LONG TRANSACTION, ADD MORE DISKS (QVOLUME PARAMETER) OR FILES PER VOLUME (QFILESPERVOLUME)

Cause: The queue is full due to a long transaction. No more data can be added to the queue, the queue must be expanded.

Effect: The Queue Manager stops reading from the collector until there is more room in the queue files.

Recovery: Alter the QMGR and expand the space for the queue, either by adding disks or by increasing the number of files per volume. Note: increasing the extent size of the files will not have an effect in this case. The Queue Manager must create a new file to continue.

4546 QMGR <name> (<process>): QUEUE IS FULL DURING TRACKTX PREREAD, ADD MORE DISKS (QVOLUME PARAMETER) OR FILES PER VOLUME (QFILESPERVOLUME), OR ALTERNATELY PURGE THE TRACKTXFILE AND RESTART THE ENVIRONMENT IF POTENTIAL REPLAY OF DATA IS ACCEPTABLE

Cause: The queue filled up while reading the TRACKTX file to determine where to restart from. No more data can be added to the queue, the queue must be expanded.

Effect: The Queue Manager stops reading from the collector until there is more room in the queue files.

Recovery: Alter the QMGR and expand the space for the queue, either by adding disks or by increasing the number of files per volume. Note: increasing the extent size of the files will not have an effect in this case. The Queue Manager must create a new file to continue. Alternatively, purging the TRACKTXFILE and restarting the environment may resolve the problem, but may also cause data to replay.

2130 DBS <dbs> HAS BEFOREVALUES SET TO OFF WHILE REPLICATING AN ENTRY-SEQUENCED FILE. UNABLE TO REPLICATE VIA ALTKEY.
2130 DBS <dbs> HAS BEFOREVALUES SET TO OFF WHILE REPLICATING AN ENTRY-SEQUENCED FILE. UNABLE TO REPLICATE VIA INDEX.

Cause: The consumer is replicating an entry-sequenced file or table via altkey or index and the DBS parameter BEFOREVALUES is set to OFF.

Effect: The consumer abends.

Recovery: Set BEFORE.

2130 SOURCE FILE <source> IS AUDITCOMPRESSED AND ENTRY-SEQUENCED. UNABLE TO REPLICATE VIA ALTKEY.

2130 SOURCE FILE <source> IS AUDITCOMPRESSED AND ENTRY-SEQUENCED. UNABLE TO REPLICATE VIA INDEX.

Cause: The consumer is replicating an entry-sequenced file or table via altkey or index, but the source file is audit-compressed.

Effect: The consumer abends.

Recovery: Alter the source file to not be audit-compressed or change the KEYSPECIFIER parameter to not replicate via altkey or index.

2130 TARGET TABLE <table> DOES NOT HAVE UNIQUE INDEX WITH KEYTAG <keytag>.

Cause: The consumer is attempting to replicate a SQL/MP entry-sequenced table by unique index, but keytag specified using KEYSPECIFIER does not point to an index, or the index is non-unique.

Effect: The consumer abends.

Recovery: Set KEYSPECIFIER to the key tag of a unique index on that table.

2008 PULSEFILE I/O ERROR NO <error number> ON <file name> (<I/O type>) [UNDER TRANSACTION <trans id>]

Cause: An I/O related to PULSEFILE processing was unsuccessful due to the <error number> identified. <I/O type> will be one of: INSERT, UPDATE, DELETE, KEYPOSITION, READ, READLOCK, SETMODE, OPEN, BEGINTRANSACTION, or ENDTRANSACTION. If this error occurred under a transaction, the transaction id will be included.

Effect: AUDMON continues to run. The pulse feature may be disabled. See subsequent EMS messages.

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

New and Modified SOLV User/EMS Messages

None in this release.

New and Modified TACL Parameters

SBCOLLGENEXTRAZZSA

Initial	Changed					
Version	Version	Values	Default	Minimum	Maximum	Alter
6.800			1	0	1	No

By default, all errors that cause abends in AUDCOLLN will produce saveabend files. If set to 0, some error cases will no longer produce saveabend files.

```
PARAM SBCOLLGENEXTRAZZSA 0
```

SBSUSPENDUPDTRACE

Initial	Changed					
Version	Version	Values	Default	Minimum	Maximum	Alter
6.800			0	0	1	No

Controls the amount of EMS messages that get logged when SUSPENDUPD events are processed. If set to 1, each object will log a message when it receives the SUSPENDUPD request, as well as when it receives the acknowledgement.

```
PARAM SBSUSPENDUPDTRACE 0
```

SBPULSEUSEREXIT

Initial	Changed					
Version	Version	Values	Default	Minimum	Maximum	Alter
6.800			0	0	1	No

Controls if a user exit can be used on a DBS with the PULSEFILE parameter configured. If set to 0, setting USEREXITID and PULSEFILE on the same DBS causes a configuration error.

```
PARAM SBPULSEUSEREXIT 0
```

New and Modified User Information Messages

The new user information messages are:

1372	DBS COLLISIONHANDLER AND SBCMDFILE NOT ALLOWED FOR PULSEFILE

Cause: The DBS object COLLISIONHANDLER parameter must be 0 and SBCMDFILE must be OFF for a pulse file.

Effect: The DBS object is not added.

Recovery: Shut down HPE Shadowbase, correct the configuration, and restart HPE Shadowbase.

1400	DBS USEREXITID NOT ALLOWED FOR PULSEFILE

Cause: The DBS object USEREXITID parameter must be 0 for a pulse file.

Effect: The DBS object is not added.

Recovery: Shut down HPE Shadowbase, correct the configuration, and restart HPE Shadowbase.

New and Modified User Exit Functions

SBABENDNOW

This function is called from anywhere in order to safely abend without having to return from the user exit program. There are no parameters or arguments associated with this call.

Product	Availability	Calling Module Usage
HPE NonStop Server	Yes	All
Other Servers	No	N/A

Syntax for COBOL Programmers

```
ENTER C "SBABNEDNOW" IN USRXLIB.
```

Syntax for C Programmers

```
void SBABENDNOW(void)
```

Parameters

None.

Considerations

- This function does not return from being called.

SBTIMERSTART

The SBTIMERSTART function is used to start timers in a user exit. When the timer elapses, the entry point USRXTIMEOUT will be called.

Product	Availability	Calling Module Usage
HPE NonStop Server	Yes	All
Other Servers	No	N/A

Syntax for COBOL Programmers

```
ENTER C "SBTIMERSTART" IN USRXLIB.
    USING TIMEOUT, PARAM, TAG
    GIVING RETURN-CODE.
```

Syntax for C Programmers

```
short SBTIMERSTART(long long timeout, long param, long *tag)
```

Parameters

RETURN-CODE returned value

- Indicates whether call completed successfully or encountered an error:
  -2 = Cannot allocate a TLE, as all available TLEs are in use
  -1 = Invalid or inconsistent parameters
   0 = Successful and tag is returned

TIMEOUT input

- Time period in microseconds after which a timeout message is queued on $RECEIVE
- Must be greater than zero

PARAM input

- Will be passed into USRXTIMEOUT

TAG output

- Identifier associated with the timer

- Can be passed into SBTIMERSTOP in order to cancel the timer

Considerations

- Timers should only be started using this function.

SBTIMERSTOP

The SBTIMERSTOP function can be used to cancel timers that have been started by SBTIMERSTART

Product	Availability	Calling Module Usage
HPE NonStop Server	Yes	All
Other Servers	No	N/A

Syntax for COBOL Programmers

```
ENTER C "SBTIMERSTOP" IN USRXLIB.
    USING TAG
    GIVING RETURN-CODE
```

Syntax for C Programmers

```
short SBTIMERSTOP(long tag)
```

Parameters

RETURN-CODE returned value

- Indicates whether call completed successfully or encountered an error:
  -4 = Tag not found on timer list of $RECEIVE queue
  -3 = Tag not associated with SBTIMERSTART
  -2 = Tag is not owned by this process
  -1 = Tag is invalid
  0 = Successful

Considerations

- None.

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 G06.16 and above operating system documentation (or contact Shadowbase Support) for more information. The SBPAK file described below is delivered as a self-extracting file.

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

For G-based systems:

```
        FUP ALTER SBPAK, CODE 700
```

For H-based and J-based systems:

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

For L-based systems:

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

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

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

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

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.

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

IMPORTANT NOTES ABOUT USER EXITS:

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

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

HPE Release File Structure

If this release is obtained through HPE Scout, the downloaded file is a single self-extracting zip file that is intended to be run on a Windows PC. When the zip file has extracted itself, 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 self-extracting ZIP file or on the product DVD under the directory \T1122HO6-<SPR ID>. This file set contains the following files:

File or Directory	Description
\readme.txt	This file.
\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) 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.

2) 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.

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

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

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

After investigation of the problem, the error was subsequently reproduced with SQLCI and submitted to 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.

5) 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.

6) 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.

7) 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.

8) 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.

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

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

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

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

13) 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.

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

15) 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 }
```

16) 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.

17) 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.

18) 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.

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

20) 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.

21) 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.

22) 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.

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

24) 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.

25) 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.

26) 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.

27) 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.

28) 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.

29) 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.

30) 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.

31) 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.

32) 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.

33) 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).

34) 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 try and limit the use of these types of DBS specifications for BASE24.

35) 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).

36) 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)

37) 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.

38) 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
```

39) 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.

40) 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.

41) 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. It is recommended to avoid this by doing one of the following things:
    - 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

42) 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).

43) The QMGR occasionally abends on a Shadowbase restart with a message similar to the following:

```
19-06-26 10:56:03 \H2.$CJCQM       *GRAVIC.100.V66        004530 QMGR
  QMGR-SRC-CJCQM(\H2.$CJCQM): COLDSTART REQUIRED: UNABLE TO FIND FIRST
  EXPECTED EVENT IN THE RESTART IPC; ADT POSITION (AA000001,282391212); MAT
  POSITION (1, 282423416); MATCH MAT ONLY (FALSE); FIRST MAT POSITION IN IPC
  (1, 282423416); LAST MAT POSITION IN IPC (, )
```

Allowing Shadowbase to restart the QMGR will usually clear up this issue.

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) Due to existing volume recovery logic, the Queue Manager holds onto transaction abort records until it sees another event without the volume recovery flag set (typically an IUD event.) In some cases, when no IUD events are sent down the Queue Manager's thread for long enough (12 hours), the Queue Manager will clear these transactions and output an EMS message saying that transactions are being "age cleared." However, the Queue Manager still acknowledges to the collector that the abort event is stored, so the collector's restart point is NOT held back for the 12 hour

duration. In this scenario, either a subsequent replicated event without the volume recovery flag, or the eventual age clear successfully cleans up the transaction from memory without causing any issues.

50) 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).

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

52) 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%
    ```

53) 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.

54) 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.

55) In a replication environment with QMGRs configured, when the environment becomes idle and there are no events for it to replicate, the LAG value in the STATS QMGR and STATS CONS output may show large values or start to grow over time. This is expected because these LAG values are calculated when events are replicated. When there are no events to replicate, these large LAG values can be misleading and make it seem that the environment is not performing properly. As long as the LAG in the STATS COLL output is low, and the number of events queued in the QMGR is not increasing, there should be no cause for alarm in this case.

56) 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.

57) 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.

58) 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.

59) 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.

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

61) If the system time is changed use the SETTIME TACL command, AUDCOM may show incorrect times in STATS and STATUS output. A workaround for this is to set the SBJTS-BYPASS TACL parameter to 1.

62) 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.

63) 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. It is recommended to avoid this by doing one of the following things:
    - Not using a QMGR in the Shadowbase environment if MP DCR, FUP DCR, or DDL Replication with coordination will occur under the same AUDMON
    - Turning MPCOORDINATION OFF for MP DCR, turning FUPCOORDINATION OFF for FUP DCR, or turning DDL coordination off (SET DBS CREATES NOCOORDINATION) for DDL replication
    - Using TID files with TIDFILECONSTANT ON instead of TRACKTXFILE

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

65) 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.

66) Enscribe Compare may trap after restarting a compare job with iterations multiple times. This can be avoided by not using iterations or by completing the compare in fewer sessions.

67) If an SBMAP map block contains any assignments to a key column, the where clause for any updates or deletes will only contain the columns that are assigned to in the map block. The workaround is to add an assignment to the map block for all of the other key columns in the target.

68) When using the MPLOG consumer parameter with MP DCR, it is possible for the consumer to trap when first trying to write to an MPLOG file that already contains data. The workaround is to open the MPLOG file and insert a line before the first log message.

69) Use of UPDATEDUPLICATE ON when replicating an Enscribe file via the alternate key through the use of the KEYSPECIFIER parameter without BASE24HANDLERID processing is not supported. The Consumer will fail if this is attempted. UPDATEDUPLICATE ON may be used with the KEYSPECIFIER parameter when BASE24HANDLERID is also set, as well as with KEYSPECIFIER for replicating a unique index of a SQL/MP Table.

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