SOFTWARE RELEASE DOCUMENT (SOFTDOC)

Product:	HPE Shadowbase for Other Servers
Release:	Gravic Version 6.800 T1129V06^AAI (SB REPL/MS WINDOWS)
Date:	June 7, 2023
Copyright Notice:	Copyright Gravic, Inc. 1995 – 2023 (www.gravic.com)
File Name:	IPM6800_other_servers.pdf

NOTE: This softdoc covers new features and corrected problems for HPE Shadowbase for Other Servers, Version 6.800. It is available as a PDF file (.PDF).

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

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 Features and Deprecated Features

This section provides a list of mature features ("***mature features***") and deprecated features ("***deprecated features***") for HPE Shadowbase for Other Servers. Gravic will continue to support ***mature features*** and will continue to include support for 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 Shadowbase for Other Servers, 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 Shadowbase for Other Servers, 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 Shadowbase for Other Servers

Feature	Description	Mature Release
Oracle Trigger Source	Oracle trigger source collectors (OPCOL) are being replaced by the Oracle log source collector (SBORLOG).	OSB 6.300 May, 2017
SQL Server Native Client	Objects that are reliant on SQL Server native client which Microsoft has labelled as obsolete 1. sbmswrtc 2. sbmssql	OSB 6.800 June, 2023

	These objects should be replaced with the generic ODBC executable types sbodbctrs and sbodbcdw.	
Oracle Call Interface (OCI)	OCI executables (sboracle and sborwtc) are being replaced by ODBC executables to allow for new and expanded features (such as ARRAY processing)	OSB 6.800 June, 2023

Table 2 - Deprecated Features in HPE Shadowbase for Other Servers

Feature	Description	Deprecated Release	Last Release Date
Execute Immediate Objects	The execute immediate (EI) mode of functions is deprecated and the execute immediate objects will be removed in a future release. Customers should take the announcement period to migrate their configurations to the Shadowbase cached objects that replace and supersede the EI functionality. The list of Shadowbase executable programs that use EI functionality are: 1. sbocons 2. sbmspre 3. sborapre 4. sbsypre 5. sbigdir 6. sbmswrt 7. sborcol 8. sbsywrt 9. sbigwrt 10. sborcol If you are using any of these executable programs (configured by SBMON), they will be retired on the date listed above.	OSB 6.300 May, 2017	May, 2019
SBDOCRD	The functionality in SBDOCRD has been replaced by SBFILE. SBDOCRD only supports execute immediate DOCs, which are also deprecated. Gravic supplied the source code for SBDOCRD to our customers for custom applications. If you are currently using SBDOCRD in a custom application, please contact Shadowbase Support for assistance migrating the application.	OSB 6.300 May, 2017	May, 2019
SQL92	The SQL database format SQL92 is deprecated and will be removed in a future update. This option is superseded by SQLSBV1 for DOC Writers and Direct Writers. SQL92 remains a current option for Open Collectors and Transaction Replay Servers in this release, but is expected to be deprecated for those objects as well in a future release.	OSB 6.600 April, 2020	April, 2022
UserExit Eclipse projects	The UserExit Eclipse projects have been deprecated in favor of the Makefiles shipped with the product for every platform but HPE OSS. If you are an existing customer that uses	OSB 6.300 May, 2017	May, 2019

	the Eclipse project please contact Support for assistance in migrating to a Makefile based workflow.		

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.

Important Note about Licenses to run Shadowbase

WARNING:

When moving from a version of Shadowbase for Other Servers software earlier than 6.700 to version 6.800, a new license file is required. Note that current license terms are still valid and remain in force. Contact the HPE License Manager (license.manager@hpe.com) or your local sales representative to request a new license file. DO NOT INSTALL Shadowbase for Other Servers software version 6.800 until a new license file has been obtained if you are upgrading from a version of Shadowbase for Other Servers software earlier than 6.700.

Special Notes for Version 6.800

1. Version 6.800 is a “general availability” (non-restricted TCF) release for Microsoft Windows Server systems. The previous non-restricted TCF release is version 6.703, which corresponds to HPE SPR version AAH for Microsoft Windows Server systems.

2. This document describes the changes since Version 6.703. If you are upgrading from a version of Shadowbase for Other Servers prior to that, please first see the version 6.703 software release document, IPM6703_other_servers, for a list of the changes in that version, as well as the softdocs for any additional intermediate versions since the version being upgraded.

3. **Due to licensing changes introduced in version 6.700, existing installations of Shadowbase prior to version 6.700 will require a new password file in order to run after the upgrade. For HPE Shadowbase customers, consult the HPE License Manager (license.manager@hpe.com) or your local sales representative for a new license to run this version.**

4. Immediately after install, sbmon should be opened and the LICENSE command should be run. This will perform some simple verification that the license file is valid for this version of HPE Shadowbase.

5. **For Shadowbase on OSS, a special file called “shadtype” will be provided along with each license file. This file should be placed in the $SHAD_BASE/data directory just like the shadpass.ini file. This file’s name should remain “shadtype” (all lowercase, no file extension.) This file is used to validate new NonStop system types that are newer than the Shadowbase release to ensure that the license is validated in order to allow replication to run properly.**

6. As of version 6.600, SAP Sybase ASE databases are supported as a target using ODBC for cached statements only. Execute Immediate (EI) statements for Sybase are no longer supported. If you are using EI for Sybase, you cannot upgrade to version 6.600 or later without converting to using cached statements. Contact Support for consultation if this pertains to you.

7. HPE Shadowbase for Other Servers now obfuscates configuration data (in particular the passwords) for objects when it is stored in the COLLCONFIG data file. This version can read the configuration records created by prior releases and will automatically store the information in obfuscated format when the record is saved. Once the information has been obfuscated, the record is no longer usable in prior versions of Shadowbase. If you are upgrading from a version prior to 6.300 and you want to maintain the ability to fall back to a prior release, you must keep a copy of the collconfig.dat and collconfig.idx files for the prior release.

   You can, for example, install the release in a new directory and copy the data directory from the old directory to upgrade. This will maintain both the binaries and the configuration files for the old release.

If you do need to recreate an obfuscated configuration with a prior release, you will need to drop and re-add the objects using the prior version of SBMON.

8. For Oracle databases, the SHAD_SQL_ERROR_* error processing parameters do not apply to database errors that occur while trying to read the table's schema to retrieve column related information. Instead, the SHAD_OCI_INCOMPLETE_SCHEMA specifies the action the Shadowbase process will take if it cannot read the table's schema. If, for example, you specify an error to skip using the SHAD_SQL_ERROR_EXCLUDE parameter, and that error occurs while reading schema, the SHAD_OCI_INCOMPLETE_SCHEMA will take precedence. If SHAD_OCI_INCOMPLETE_SCHEMA is set to SHUTDOWN, the process will stop even though the error is specified to be skipped.

9. The UserExit Eclipse projects have been deprecated in favor of the Makefiles shipped with the product for every platform but HPE OSS. If you are an existing customer that uses the Eclipse project please contact Support for assistance in migrating to a Makefile based workflow.

10. When building User Exits on Linux for x86_64, only binutils version 2.27 and later and GCC version 5 or later are supported due to an x86_64 psABI change that effected relocation information.

11. The consumptive TFS has been removed in its entirety, the current replacement is the consumptive TRS, if you have any issues with the migration please contact Support for assistance in migrating.

12. Sybase references were removed from both the sbinstall and sbfile scripts. Sbinstall would previously print what Sybase version was being used but now that is not relevant and could be confusing. Sbfile took a Sybase type as an argument with COLLCONFIG and RESETSBCONFIG commands.

13. The sbinstall script for windows has been updated so that it will now fail and inform the user if SHAD_BASE is not set correctly as a system variable. This may be due to it being set as a user level environment variable or not at all but this notification should make it far easier to find a fairly common issue in new environments.

14. All of the 3rd party vendored libraries shipped with Shadowbase are now located in the $SHAD_BASE/lib/optional directory. If any errors with missing libraries are encountered when attempting to start any Shadowbase objects, look for the missing library in this location and move it up a level into the $SHAD_BASE/lib directory and start the object again.

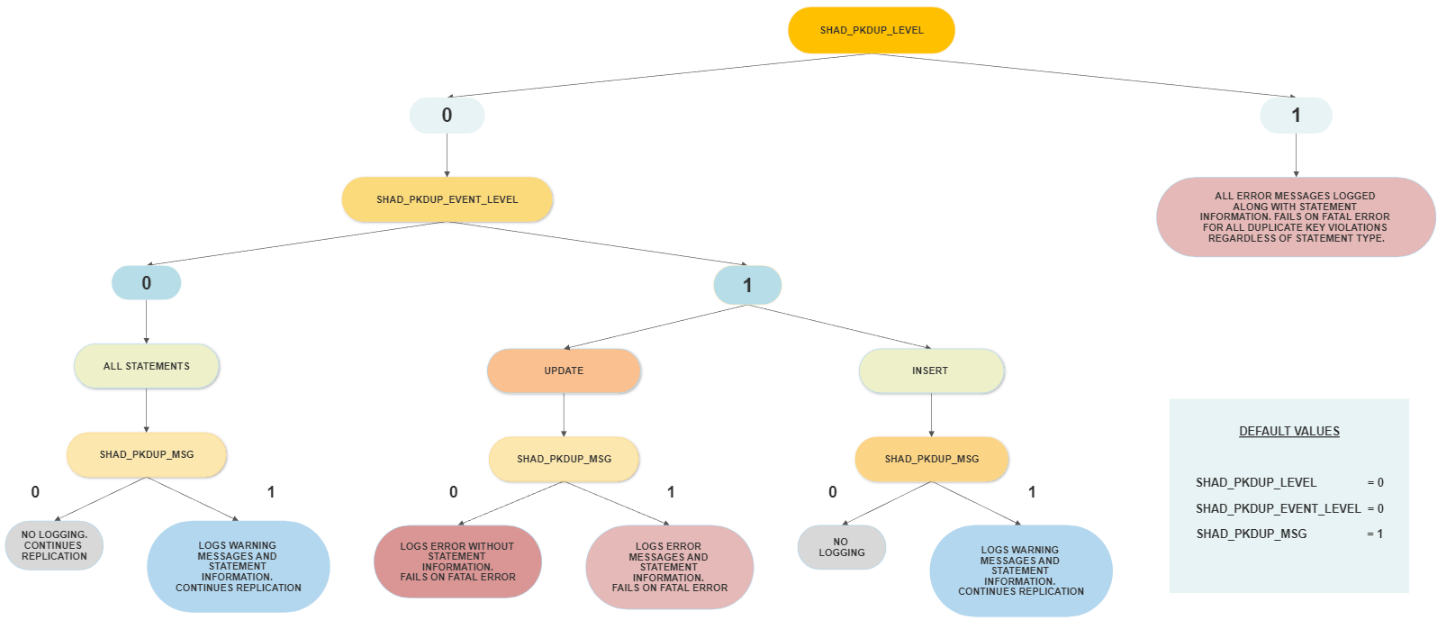
New Features, Problems Corrected, and Other Changes in Version 6.800 (since Version 6.703)

New Features in Version 6.800

This section provides a summary of the features added to HPE Shadowbase for Other Servers products since the previous general availability release (V6.703).

1. All SHAD_PKDUP related parameter settings have been corrected to show the proper logging information for any instance in which a fatal error occurs. Logging options for insert and update statements are now also properly differentiated. The logging pathways for various combinations of SHAD_PKDUP related parameters are depicted in the flowchart below. Additional details have been added to the relevant parameter descriptions as well; see the New & Modified shadparm.ini Parameters section for the updates on SHAD_PKDUP_MSG and SHAD_PKDUP_EVENT_LEVEL.

DUPLICATE KEY VIOLATION - PARAMETER INTERDEPENDENCE CHART


SHAD_PKDUP_LEVEL
0
1
SHAD_PKDUP_EVENT_LEVEL
ALL ERROR MESSAGES LOGGED ALONG WITH STATEMENT INFORMATION. FAILS ON FATAL ERROR FOR ALL DUPLICATE KEY VIOLATIONS REGARDLESS OF STATEMENT TYPE.
0
1
ALL STATEMENTS
UPDATE
INSERT
SHAD_PKDUP_MSG
SHAD_PKDUP_MSG
SHAD_PKDUP_MSG
0
1
0
1
0
1
NO LOGGING. CONTINUES REPLICATION
LOGS WARNING MESSAGES AND STATEMENT INFORMATION. CONTINUES REPLICATION
LOGS ERROR WITHOUT STATEMENT INFORMATION. FAILS ON FATAL ERROR
LOGS ERROR MESSAGES AND STATEMENT INFORMATION. FAILS ON FATAL ERROR
NO LOGGING
LOGS WARNING MESSAGES AND STATEMENT INFORMATION. CONTINUES REPLICATION
DEFAULT VALUES
SHAD_PKDUP_LEVEL = 0
SHAD_PKDUP_EVENT_LEVEL = 0
SHAD_PKDUP_MSG = 1

2. A new utility called “sblicinfo” is bundled into the Shadowbase release. This file is an executable program that can be used to collect information required for generating a license for that system. In order to use this utility, simply run the sblicinfo executable on the system that you want to collect information for. The program will create a new file called “<system name>.cache” in the same directory as the program. Send this .cache file to your Shadowbase sales representative when requesting a Shadowbase license.

3. Audit Log: The SHAD_EVENT_TIMESTAMP Audit Log column now contains six sub-second digits of precision down to the microsecond. Previous versions of Shadowbase only went down to the second. If you have any tables configured with Audit Log enabled, ensure the target column type is able to handle microsecond precision before upgrading Shadowbase.

4. A new array-processing based version of the TRS is now available. This TRS object will batch statements together where possible to reduce network traffic and improve performance. This can be particularly helpful in environments where the target database is remote from where the TRS is running, and there may be high latency in ODBC access to the database across the network. Refer to the Array-based Target Database Processing section for more information.

5. Database tracing for OCI and ODBC targets is now available. See the parameter description for SHAD_DB_TRACING_LEVEL for information on how to configure this feature. Depending on the set tracing level, each ODBC and OCI call will log a tracing message right before the call, as well as a message indicating success. Both messages for each call begin with a "DB Tracing:" tag, followed by the address of the database handle in parenthesis.

6. A latency tracking feature has been added for nonstop to OSB replication. When replicating from NSB to OSB, this feature will be accessible when running the LIST command on OSB DOCs/TRSs/DWs. For DOC/TRS configurations, the timestamp passed down from NSB will be used to show the latency between the source commit time and the time applied to a DOC, the latency between applying to the DOC and applying to the target database, and the latency between source commit time and TRS application into the target database time. For DW configurations, just the latency between the source commit time and the database application time will be displayed. Refer to the New & Modified SBMON Commands section for more details.

7. Multiple instances of Shadowbase can now be installed on a single Windows systems. Refer to the Changes for multiple instances on Windows section for more details.

8. The sbmon START, STOP and STATUS commands have been enhanced to allow wildcarding or grouping objects by type. See the New & Modified SBMON Commands section for more details.

9. A new option has been added to SHAD_TRACK_PERFORMANCE to include the cached statement text associated with each cursor whenever the stats are logged. Refer to the section on this parameter below for more details.

Array-based Target Database Processing

Shadowbase for Other Servers has historically processed events into a target database by applying the events into the tables by calling SQLExecute for each **individual** I/O. This performs well when the connection from Shadowbase into the target database is reasonably fast, regardless of whether the database is local to the system or remote on another system or SAN, etc., on the network.

With the increased popularity of cloud-based databases that are only remotely accessible, meaning the database is remote from the Shadowbase application, the latency in the connection from the client where Shadowbase is running to the target database in the cloud is much higher than a traditional target database on the network. As a result, the standard Shadowbase method of calling SQLExecute for each event can be prohibitively slow in these environments since the latency in the connection causes each call to take much longer.

In order to solve this problem, a new method of applying events into the target database is being introduced in version 6.800. This method uses ARRAY processing to apply many events at once into the target database with a single ODBC operation. In this new ARRAY method, as Shadowbase processes the events it receives, it saves them in memory and flushes them into the database periodically using a *bulk* style submission. By grouping many insert/update/delete events into a single ODBC operation, this allows Shadowbase to mitigate the high latency in the database connection and to replicate events sufficiently fast to keep up with the arrival rate.

As an example, consider the following transaction to be applied into a target database table by Shadowbase:

```
BEGIN
INSERT #1
INSERT #2
INSERT #3
INSERT #4
INSERT #5
INSERT #6
INSERT #7
INSERT #8
INSERT #9
INSERT #10
COMMIT
```

A traditional Shadowbase environment would apply each of these inserts individually into the database followed by a commit for a total of 11 calls into ODBC/the database. By switching to the array-based processing method, this would combine all insert operations into a single ODBC call to the database followed by the commit for a total of 2 calls into the database. This will be a great benefit for database connections with high latency, including all “cloud” interfaces that support an ODBC interface (e.g., Microsoft Azure, AWS Aurora, etc).

Just like traditional Shadowbase processing, the array-based method will use the same transaction boundaries based on the SHAD_BATCHTX… parameter settings for the environment. However, unlike traditional processing, the array-based method will sort all of the events within the transaction in order to maximize the efficiency when they are applied into the target database. As a result of this behavior, events within a transaction may not be applied into the target database in the same order they are received. However, the ordering of events applied to any specific row is preserved, and all of the expected events will be applied to the target database at the time commit is called. In other words, the correct ending values of all of the rows in the transactions that are batched together is preserved.

Statement Caching

Array-based processing is made possible with the help of the Shadowbase executable object caching events in memory and flushing them into the database as one ODBC operation whenever the object calls commit. For this reason, it is important that the SHAD_BATCHTX… parameters be enabled so that it allows the object to batch up small transactions received from the source system into fewer, larger, transactions that are applied on the target database. The larger the transactions applied to the target database, the fewer ODBC database operations are required which will help improve overall performance.

In order to balance memory usage while processing, a parameter called SHAD_MAX_CACHED_EVENTS is used to allow the object to periodically flush its cache into the target database if it hasn’t processed a commit yet. This prevents the Shadowbase object from potentially using up too much system memory while processing very large transactions. See that parameter definition for more details.

Statement Merging

When the array-based Shadowbase object receives two or more events in a single transaction for the same row of the target database, by default it will attempt to merge these events into a single event in order to further minimize the number of operations applied into the target.

Merging is enabled by default, but it can be disabled by use of the SHAD_MERGE_OPERATIONS shadparm.ini parameter. The statement merging behavior follows these rules:

- INSERT followed by an UPDATE: the merged event is an insert with column values from the UPDATE.

```

INSERT INTO TAB (KEY_COL, COL2, COL3) VALUES (1, ‘a’, ‘a’);
```

```
UPDATE TAB SET COL2=’z’ WHERE KEY_COL=1;
```

merge into:

```
INSERT INTO TAB (KEY_COL, COL2, COL3) VALUES (1, ‘z’, ‘a’);
```

- INSERT followed by a DELETE: both operations are removed.

```
INSERT INTO TAB (KEY_COL, COL2, COL3) VALUES (1, ‘a’, ‘a’);
DELETE FROM TAB WHERE KEY_COL=1;
```

merge into no events being applied to the target.

- UPDATE followed by a DELETE: the UPDATE operation is removed. Only the DELETE is performed.

```
UPDATE TAB SET COL2=’z’ WHERE KEY_COL=1;
DELETE FROM TAB WHERE KEY_COL=1;
```

merge into:

```
DELETE FROM TAB WHERE KEY_COL=1;
```

- UPDATE followed by an UPDATE: as long as one of the UPDATE operations contains all of the columns modified by the other UPDATE operation, the operations will be combined into a single UPDATE. For example:
    - Two UPDATEs modifying same columns will be merged (the column values will be the values in the second UPDATE statement).

```
UPDATE TAB SET COL2=’x’ WHERE KEY_COL=1;
UPDATE TAB SET COL2=’z’ WHERE KEY_COL=1;
```

merge into:

```
UPDATE TAB SET COL2=’z’ WHERE KEY_COL=1;
```

- If the first UPDATE modifies column 1 and 2, and the second UPDATE only modifies column 1, the UPDATEs will be merged. Column 1 will have the value from the second UPDATE, column 2 will have the value from the first UPDATE.

```
UPDATE TAB SET COL2=’x’, COL3=’x’ WHERE KEY_COL=1;
UPDATE TAB SET COL2=’z’ WHERE KEY_COL=1;
```

merge into:

```
UPDATE TAB SET COL2=’z’, COL3=’x’ WHERE KEY_COL=1;
```

- If the first UPDATE modifies column 2, and the second modifies columns 1 and 2, the UPDATEs will be merged. Both columns will have the values from the second UPDATE.

```
UPDATE TAB SET COL2=’x’ WHERE KEY_COL=1;
UPDATE TAB SET COL2=’z’, COL3=’z’ WHERE KEY_COL=1;
```

merge into:

```
UPDATE TAB SET COL2=’z’, COL3=’z’ WHERE KEY_COL=1;
```

- If the first UPDATE modifies column 1 and the second modifies column 2, the operations will not be merged.

```
UPDATE TAB SET COL2='x' WHERE KEY_COL=1;
UPDATE TAB SET COL3='z' WHERE KEY_COL=1;
```

will not be merged.

If the operations against the same row cannot be merged for any reason, or if statement merging is disabled, they will be applied against the row in the order in which they were received.

Again, note that setting the SHAD_BATCHTX… parameters allows more I/O operations to be merged, additionally reducing the overall number that actually need to be applied into the target database.

Statistics Reporting

The statistics displayed for array-based objects when SHAD_TRACK_PERFORMANCE is enabled is slightly different from those displayed for a traditional object. Here are the changes from the SHAD_STATS_FREQUENCY description of these statistics:

- Et: The total number of seconds for the EXECUTE statements. For the array-based object, this includes time spent caching and merging the statements in memory for later execution. Since these operations are cached in memory local to the process, this value is typically very low and may be 0 due to finite clock resolution.
- Ei: The total number of EXECUTE statements received by the object.
- Er: The EXECUTE statement execution rate (executions per second) calculated by Ei/Et. If Et is 0, -1 is displayed here.
- EAi: The total number of statements actually executed in the target database. If statement merging occurs, EAi may be less than Ei. It's also possible for this value to be 0.
- EAr: The rate at which statements are actually executed. This is calculated by dividing EAi by the total time for the SQL execute.

These final three statistics measure commit performance. These statistics are not related to any one particular SQL statement, but they will be reported on the first line of each SHAD_TRACK_PERFORMANCE errlog message:

- Ct: The total time spent issuing the SQL commit statement.
- Ci: The total number of commits issued.
- Cr: The commit rate calculated by Ci/Ct.

Here is a sample of what this errlog message will include. (For best viewing, copy/paste to a notepad with word wrap disabled.)

```
SC-idx |     Pt     |  Pi   |     Pr     |     Et     |    Ei    |     Er      |
EAi     |    EAr     |      Ct      |  Ci  |     Cr     |     S/sec    | DML |
Current Table Name            | Aggr Rate      | Prepared Statement
-------- ---------- ------ ---------- ---------- -------- ----------- -
---------- ---------- ----------- ------ ---------- ----------- -----
------------------------------ -------------- -------------------
    1        0.064      1     15.625      2.676    20000    7473.842
20000   7473.842        0.140     10     71.429     6944.444     I
BPRTTABL                             3819.710
```

Configuration

Configuring an object to use array-based processing is nearly identical to configuring an object for an ODBC target database. When adding the target database object with sbmon, the executable types that start with “sbarr…” should be used to allow Shadowbase to configure an array-based object instead of using the standard replication method. All other prompts when configuring the object should follow the same rules when configuring a normal replication object to connect into the target database.

Limitations

Array-based replication in version 6.800 has the following limitations. In this release:

- Array-based replication is only supported for SQL Server targets on Windows.
- Array-based replication can only be applied to TRS objects by using the executable type “sbarrtrs” when configuring the object. There is no Direct Writer support.
- SHADTRS inserts are processed using the traditional SQLExecute method rather than being batched together using the array-based method. Depending on the structure of the transactions being replicated, enabling SHADTRS processing may throttle the object to run as slowly as if it were a normal non-array-based object. SHADTRS processing should be disabled in this scenario to achieve higher performance.
- Insert Not Found (INF) processing is not supported. A new parameter called SHAD_UPDATE_NOT_FOUND has been added to handle the behavior when processing an update for a row that doesn’t exist, but it cannot convert the update to an insert and apply it to the target.
- SHAD_SQL_ERROR_INVALID_NUM and SHAD_SQL_ERROR_MAPTO processing is not supported.
- The WHERE clause used for all update and delete events must always include the same columns in every event replicated. Normally, this should match the primary key of the target table. If any of the target tables have no primary key, or if the WHERE clause in the updates and deletes does not match the primary key for any reason, this can be overridden by using the SHAD_KEY_COLUMNS_FILE parameter.
- If SHAD_TRACK_PERFORMANCE is enabled, the statistics can only be logged to the errlog when the object is issuing a commit to the target database. This may

cause the object to log its performance statistics to the errlog less frequently than specified by SHAD_STATS_FREQUENCY if there are transactions that are larger than the desired frequency.

- For certain SQL database errors, the individual event that caused the error cannot be determined, and therefore it is not logged to the errlog along with the failure message. In situations like this, it may be desirable to temporarily switch the object over from array-based processing to normal processing and start it again. The object will fail with the same error again, but this time it will provide the user with standard error details for the purpose of troubleshooting.
- An explicit ODBC DSN for the target database must be configured before starting the object.
- If SHAD_DB_TRACING is enabled, no information about the individual events applied to the target database will be logged.

Problems Corrected in Version 6.800

This section provides a summary of the problems corrected in HPE Shadowbase for Other Servers products since the previous general availability release (V6.703).

1. When SHAD_LOG_FILE_SPACE parameter was set to a value lower than the minimum, the info message detailing the default minimum would output a value double the SHAD_LOG_FILE_SPACE value when it should be double the SHAD_LOG_FILE_SIZE value. This has been corrected in 6.800.

2. CONTINUOUS_BOTH mode for Oracle log source will now work with DataGuard writing to source tables.

3. Cleaned up the error logging for DB2 when the nullable status of columns is unobtainable. Previously the message would be output multiple times but now it is only logged once.

4. SHAD_TRACK_PERFORMANCE output now uses local time rather than GMT

5. Removed unnecessary error logging when SHADTRS exists on startup.

6. Cleaned up an issue with TCP/IP message logging that would cause messages to include the prior TCP/IP messages each time it was output in the previous version.

7. Fixed logging of a DW error where a prepared statement was output without any preceding error text.

8. Fixed SHAD_MAX_DOC_CACHE_DEPTH behavior to properly adjust the max number of unique statements handled.

9. Fixed incorrect logging of the TCP receive buffer value.

10. Corrected an issue with SQL/MX sequence object replication where it would incorrectly expect the SQL/MX schema definition version to be 3700.

11. Fixed an issue where Log Server SHAD_REJECT events for SQL errors on INF events were not being generated properly.

12. There was an issue relating to the maximum statement cache depth of OSB Log Source objects where each Log Source Collector was limited to 1000 unique statements. A workaround to collect more than 1000 statements was to configure multiple Log Source collectors to keep each one under 1000 statements. This limit has been removed, and is now controlled by the SHAD_MAX_DOC_CACHE_DEPTH parameter.

13. Fixed an issue where an Oracle log source collector would fail when attempting to collect an event with over 600 binary NUL bytes.

14. Fixed an issue that prevented SQL Server trigger source from properly collecting events longer than 500 bytes.

New & Modified SBMON Commands

This section provides a description of the SBMON operational commands that are either new or modified since the previous general availability release (V6.703).

START <object_name> <object_class>

This command starts multiple processes. All objects that match both the provided object name and the object class if provided will be started.

The object name can begin or end with a wildcard(*) if starting multiple objects is required. A single '*' will match all objects.

If no object class is provided every object that matches the object name will be started. Running 'START *' will start all objects.

For example running 'START *ST DOC' will start all database of change objects that have a name that ends with 'ST'.

List of available object class names and the objects the correspond to:

- DOC : Database of Change
- DW : Direct Writer
- OPCOL : Open Collector
- LGSRV : Log Server
- TFS : Transaction Forwarding Server
- NSK : Remote NSK Consumer
- TRS : Transaction Replay Server
- CLNR : DOC Cleaner
- LSNR : Listener
- CSTM : Custom Object

STATUS <object_name> <object_class>

This command returns the status of multiple processes. All objects that match both the provided object name and the object class if provided will be polled for status.

The object name can begin or end with a wildcard(*) if retrieving the status of multiple objects is required. A single '*' will match all objects.

If no object class is provided every object that matches the object name will be polled. Running 'STATUS *' will status all objects.

For example running 'STATUS ST* OPCOL' will poll the status for all open collectors that have a name that starts with 'ST'.

List of available object class names and the objects the correspond to:

- DOC : Database of Change
- DW : Direct Writer
- OPCOL : Open Collector
- LGSRV : Log Server
- TFS : Transaction Forwarding Server
- NSK : Remote NSK Consumer
- TRS : Transaction Replay Server
- CLNR : DOC Cleaner
- LSNR : Listener
- CSTM : Custom Object

STOP <object_name> <object_class>

This command stops multiple processes. All objects that match both the provided object name and the object class if provided will be stopped.

The object name can begin or end with a wildcard(*) if stopping multiple objects is required. A single '*' will match all objects except for Listeners.

If no object class is provided every object that matches the object name will be stopped. Running 'STOP *' will stop all objects.

For example running 'STOP * TRS' will stop all Transaction Replay Server objects.

List of available object class names and the objects the correspond to:

- DOC : Database of Change
- DW : Direct Writer
- OPCOL : Open Collector
- LGSRV : Log Server
- TFS : Transaction Forwarding Server
- NSK : Remote NSK Consumer
- TRS : Transaction Replay Server
- CLNR : DOC Cleaner
- LSNR : Listener
- CSTM : Custom Object

LIST <object name>

The LIST <object name> output has been enhanced for DOC, TRS and DW objects to include information about the latency of the events being replicated. Here is a sample LIST output for a TRS object:

```
-------------------Object Parameters--------------------------------------------
            Name: TRS                       |       SourceName: DOC
```

```
     Server Type: Transaction Replay Server  |   Read Open DOCs: Y
      Executable: sbodbctrs                  |              INF: Y
    Event Format: Cached SQL Statement       |         EarlyAck: 0
          SeqNum: 004                        |   Source Cleanup: N
 Transaction Key: FF001C66AF350000           | Write Boundaries: Y
 Max TCP/IP Send: 65534                      |         Protocol: 2
       IPAddress:                            |    Num Consumers: 1
Port Address(es):
-------------------Replicate Parameters------------------------------------------
 Target Database Format: SQLSBV1
            Server Name: QA-WIN2022-1
        Target Database: QAT
          ODBC User DSN: QA-WIN2022-1
-------------------Processed Events---------------------------------------------
Insert  Update  Delete  Total
------- ------- ------- -------
      3       0       0       9
            Working.....Do Not Interrupt!!!!!
-------------------Latency Statistics-------------------------------------------
      ----------------------------------------------------
      | TRS CAUGHT UP, DISPLAYING MOST RECENT STATISTICS |
      ----------------------------------------------------
                   TRS Statistics:
Stats At:2023-06-13:18:06:20                        Since: 2023-06-13:18:05:17
        Latest TX                     Latest TX                Source to
     Source Commit TS               DOC Commit TS               DOC latency
---------------------------   ---------------------------   --------------------
2023-06-13:21:49:18.443286     2023-06-13:21:50:03.736000    00:00:45.292714

        Latest TX                    DOC to TRS             Source to TRS
       TRS Apply TS                Apply latency              Apply latency
---------------------------   -----------------------   ------------------------
2023-06-13:21:50:03.745000        00:00:00.009000        00:00:45.301714
*********************************************************************************
```

Noticed the new "Latency Statistics" section at the bottom. The new output is described below:

- `TRS CAUGHT UP…`: This will be logged if the TRS is at the end of the latest DOC sequence and has no more work to do. If the TRS was behind when it processed the last events out of the DOC, it's possible that the latency values may be high even though it has no more work to do. This message logged will help to confirm that the TRS is completely caught up even in this scenario.

- `Latest TX Source Commit TS`: The timestamp of when the TRS's latest event occurred on the source system (GMT).

- `Latest TX DOC Commit TS`: The timestamp of when the latest event was replicated into the DOC (GMT).

- `Source to DOC latency`: The difference in time showing how long it took for the event to get from the source system into the DOC.

- `Latest TX TRS Apply TS`: The timestamp of when the latest event was applied into the target database (GMT).

- `DOC to TRS Apply latency`: The difference in time showing how long it took for the event to be replicated into the target after it was present in the DOC.
- `Source to TRS Apply latency`: The difference in time showing how long it took for the event to get from the source system into the target database.

The LIST <DOC> output contains the same “Latest TX Source Commit TS”, “Latest TX DOC Commit TS” and “Source to DOC latency” fields as in the LIST <TRS> output.

The LIST <DW> output contains the same “Latest TX Source Commit TS”, “Latest TX TRS Apply TS” and “Source to TRS Apply latency” fields as in the LIST <TRS> output, but with using the term “Replay” instead of “TRS” since it’s a DW instead of a TRS.

Note that this information is only available when the source is an HPE NonStop Shadowbase collector process. For all other environments, these fields will show N/A for all values. N/A will also be used when first starting the Shadowbase objects when it hasn’t replicated any events and can’t measure the latency.

New & Modified shadparm.ini Parameters

This section provides a description of the parameters for the shadparm.ini configuration file that are either new or modified since the previous general availability release (V6.703).

SHAD_ALLOW_SERVICE_ERRORS

Initial Version:	V6.800	**Syntax:**	SHAD_ALLOW_SERVICE_ERRORS=<num>
Last Change:	--		
Default Value	1 (Allow)		
Other Servers	NTSBSERV		
Valid Settings	0: The ntsbserv Shadowbase service will shut down in the case of an error in windows system calls. 1: The ntsbserv Shadowbase service will log errors to the Windows event log and attempt continue operating.		
Basic/Advanced	Advanced		

Description

This parameter controls whether or not the ntsbserv Windows service for Shadowbase for Other Servers will quit upon receiving any error response from the Windows system calls involved in its operation. Setting the parameter to both 0 and 1 will allow errors to be reported to the Windows Event Viewer, however setting it to 0 will cause ntsbserv to shut down after logging issues, while 1 will allow it to keep running.

Notes

New error handling in the ntsbserv code started causing the ntsbserv service to crash under more strict conditions. This parameter was added to preserve the old behavior of tolerating bad Windows system call responses that don't immediately bring down the service.

SHAD_BASE

Initial Version:	V6.800	**Syntax:**	SHAD_BASE =<text>
Last Change:	--		
Default Value	--		
Other Servers	ALL		
Valid Settings	The full path to the installation of Shadowbase.		
Basic/Advanced	Basic		

Description

If SHAD_BASE is not set as an environment variable or SHAD_BASE_OVERRIDE is in use, then SHAD_BASE must be present in the shadparm.ini.

Notes

Prior to version 6.800 SHAD_BASE could only be set in the system environment rather than in shadparm.ini. The system environment method of setting SHAD_BASE is still valid, although on Windows this limits you to one installed instance per system, and you will need to reboot any time SHAD_BASE is changed.

Related Parameters

SHAD_BASE_OVERRIDE

SHAD_BASE_OVERRIDE

Initial Version:	V6.800	**Syntax:**	SHAD_BASE_OVERRIDE=<num>
Last Change:	--		
Default Value	0 (Off)		
Other Servers	ALL		
Valid Settings	0: Use the SHAD_BASE environment variable 1: Override the SHAD_BASE environment variable		
Basic/Advanced	Advanced		

Description

This parameter controls whether or not to use the SHAD_BASE environment variable. By default, the environment variable will be used if it is present. However, it may be helpful to override the SHAD_BASE environment variable setting if attempting to use a different instance of Shadowbase on the system. In order to do that, SHAD_BASE_OVERRIDE should be set to 1 in the shadparm.ini file located in the data directory of the location referenced by the SHAD_BASE environment variable, and the new instances of Shadowbase should have SHAD_BASE present in its shadparm.ini file.

Notes

This parameter is primarily meant to enable adding new Shadowbase instances on Windows without damaging the previous installation. Setting SHAD_BASE in shadparm.ini enables future upgrades without restarting the system when running on Windows.

If the SHAD_BASE environment variable points to an installation of Shadowbase older than V6.800, SHAD_BASE_OVERRIDE can still be set in that version's shadparm.ini file and it will be used correctly.

Related Parameters

SHAD_BASE

SHAD_DB_TRACING_LEVEL

Initial Version:	V6.800	**Syntax:**	SHAD_DB_TRACING_LEVEL=<num>
Last Change:	--		
Default Value	0 (Off)		
Other Servers	TRS | DW		
Valid Settings	0: No DB tracing is output 1: Only primary calls to ODBC and OCI are logged, without cached statement data values being displayed: execute calls (such as SQLExecute and OCIExecute), prepare calls (such as SQLPrepare and OCIPrepare), and commit/rollback/abort calls (such as SQLEndTran, OCITransCommit, and OCITransRollback) 2: Only primary calls to ODBC and OCI are logged, specified in level 1, with the addition of cached data values being reconstructed into their respective statements and displayed in prepare and execute calls. 3: All calls to ODBC and OCI are logged, without cached data values being displayed 4: All calls to ODBC and OCI are logged, with cached data values being displayed in statements		
Basic/Advanced	Advanced		

Description

This parameter controls the level to which tracing is output for both OCI and ODBC database calls within Shadowbase target objects. Depending on the set tracing level, each ODBC and OCI call will log a tracing message right before the call, as well as a message indicating success. Both messages for each call begin with a “DB Tracing:” tag, followed by the address of the database handle in parenthesis. Additionally, before a transaction is begun, and after it has been completed or rolled back, a tracing call will be logged with the transaction ID associated with the transaction being worked on being displayed.

Notes

This parameter is meant to be used for environment troubleshooting purposes, turning it on, especially to higher settings that log all calls, can result in a high frequency of disk access which can slow down replication throughput, as well as fill up logging files with potential noise. This parameter is not meant to be turned on in a production environment for extended periods of time, and is meant to be turned off when problems are diagnosed and resolved.

SHAD_DISPLAY_INF

Initial Version:	Pre v2.600	**Syntax:**	SHAD_DISPLAY_INF=<NUM>
Last Change:	V6.800		
Default Value	0 if INF is enabled, 1 if INF is disabled		
Other Servers	TRS | DW		
Valid Settings	0: Disable logging of insert not found messages. 1: Enable logging of insert not found messages.		
Basic/Advanced	Advanced		

Description

This parameter can be used to control whether Insert not found error messages are written to the logs.

SHAD_ENV

Initial Version:	V6.800	**Syntax:**	SHAD_ENV=<text>
Last Change:	--		
Default Value	--		
Other Servers	Windows Services		
Valid Settings	The unique name of this Shadowbase environment.		
Basic/Advanced	Advanced		

Description

This parameter defines the name for the current instance of Shadowbase for Windows service. When specified, it should be set to a simple text string like "inst1" for example.

Notes

This parameter must be unique for multiple Windows installations to function correctly on a single system. Otherwise the service will only function for the most recent installation.

This same name must also be used as an argument when running the sbinstall.bat script as part of the installation. Refer to the installation instructions for more details.

SHAD_KEY_COLUMNS_FILE

Initial Version:	v6.800	**Syntax:**	SHAD_KEY_COLUMNS_FILE=<filename>
Last Change:	--		
Default Value	N/A		
Other Servers	TRS (sbarrtrs)		
Valid Settings	Fully-qualified filename		
Basic/Advanced	Advanced		

Description

This parameter specifies a file that defines the key columns for one or more target tables during array-based processing. By default, the array-based object will read the key columns from the target SQL catalog and use those as the key columns for processing. This parameter can be used to override that behavior and to manually configure the key columns for one or more tables. This can be useful where either the target table has no primary key defined or an alternate set of columns should be used to identify each row in the table.

This parameter should be set to the location of the key columns definition file. This file is a comma-delimited text file and should be configured before starting the array-based object. Each line defines a single table's keys. The line starts with the table name (using the name as specified in the events being replicated into the object), followed by one or more column names to use for the primary key, separated by a comma. The following example defines the columns for four tables:

```
OINFTABL,AKEY
LONGTABLEX,FKEYXXXXXXXXXXXXXXXXXXXXXXXXXXYYYYYYYYYY
TESTTAB,PK
TESTTAB2,PK1,PK2,PK3
```

SHAD_LOCAL_CONTEXT

Initial Version:	V3.600	**Syntax:**	SHAD_LOCAL_CONTEXT =<text>
Last Change:	V6.800		
Default Value	--		
Other Servers	ALL		
Valid Settings	The path to the folder where HPE Shadowbase should keep its log, ID, and MSG files.		
Basic/Advanced	Advanced		

Description

In a clustered environment, this parameter specifies where HPE Shadowbase should keep the log files, ID files, and MSG files associated with the configured objects

Notes

Prior to version 6.800 SHAD_LOCAL_CONTEXT could only be set in the system environment rather than in shadparm.ini. The system environment method of setting SHAD_LOCAL_CONTEXT is still valid as long as SHAD_BASE is also set as an environment variable and SHAD_BASE_OVERRIDE is not used. If environment variables are used on Windows systems, a reboot will be required whenever they are changed.

It is recommended that this parameter be set to a local drive. If this parameter is set to a clustered drive, during a failover, if the clustered drive is no longer accessible, messages from the failed system may not be written to disk.

sbmon will only control the local node.

When this parameter is specified, before using SEM, a listener process must be configured for each node in the cluster.

Related Parameters

SHAD_BASE_OVERRIDE
SHAD_BASE

SHAD_LOG_FILE_SIZE

Initial Version:	v4.030	**Syntax:**	SHAD_LOG_FILE_SIZE=<num>
Last Change:	v6.800		
Default Value	100		
Other Servers	All		
Valid Settings	10 - 1000: Maximum log file size, in Megabytes.		
Basic/Advanced	Advanced		

Description

This parameter specifies the maximum file size in megabytes that a log file can consume. When the file size exceeds the specified limit, the log file instance is rolled and a new (empty) log file is created and opened for logging.

Notes

The size of a log file will slightly exceed the size specified in this parameter before the log roll starts.

Related Parameters

SHAD_LOG_FILE_SPACE
SHAD_LOG_ROLL_ON_STARTUP

SHAD_LOG_FILE_SPACE

Initial Version:	v4.030	**Syntax:**	SHAD_LOG_FILE_SPACE=<num>
Last Change:	V6.800		
Default Value	2000		
Other Servers	All		
Valid Settings	20 - 2000: Maximum space used by log files, in Megabytes.		
Basic/Advanced	Advanced		

Description

This parameter specifies the maximum file space in Megabytes that the current and rolled log file(s) can consume. When the used file space exceeds the specified limit, historical log file(s) (oldest-to-youngest) are removed until the over-all file space in use is reduced below the specified limit.

Notes

The actual minimum value of this parameter is determined by the maximum log file size set in SHAD_LOG_FILE_SIZE. This parameter's minimum value must provide enough space for at least two full logs to exist, otherwise it will be ignored.

When defined in the [GENERAL] section, the value set for this parameter applies for each object defined in the HPE Shadowbase configuration. Therefore, this parameter does not establish an aggregate maximum file space for logfiles for all objects defined within *sbmon*.

Related Parameters

SHAD_LOG_FILE_SIZE

SHAD_LOG_ROLL_ON_STARTUP

SHAD_MAX_CACHED_EVENTS

Initial Version:	v6.800	**Syntax:**	SHAD_MAX_CACHED_EVENTS=<num>
Last Change:	--		
Default Value	2000		
Other Servers	TRS (sbarrtrs)		
Valid Settings	1 – 100,000		
Basic/Advanced	Advanced		

Description

This parameter specifies the maximum number of events that will be cached in memory before the event cache is flushed to the database. This value should be greater than the number of events specified by the SHAD_BATCHTX_COUNT parameter which determines the minimum number of events processed before a commit is performed on the target database. SHAD_BATCHTX_COUNT defaults to 1000 and SHAD_MAX_CACHED_EVENTS defaults to 2000, so the defaults for both parameters are compatible with each other.

This parameter is primarily intended for situations where the source system has long-running batch transactions with many events. In that case, the commit will not be generated until the end of the batch transaction which would otherwise result in memory overflow or degraded performance due to memory pressure. By allowing the object to flush its cache into the target database, it will be able to handle any source transaction no matter how large it is.

SHAD_MERGE_OPERATIONS

Initial Version:	v6.800	**Syntax:**	SHAD_MERGE_OPERATIONS=<num>
Last Change:	--		
Default Value	1		
Other Servers	TRS (sbarrtrs)		
Valid Settings	0: Disable operation merging 1: Enable operation merging		
Basic/Advanced	Advanced		

Description

This parameter specifies whether or not the array-based object should merge operations on the same row within a transaction. Merging operations will result in the target database having all the correct rows by the time commit is called, but it may combine or remove some operations in the replication stream that are not required to get to the correct end result. This may make the replication object more efficient by reducing the number of events applied to the target.

It may be beneficial to disable this feature if there are any triggers or stored procedures in the target database that rely on every operation being applied rather than optimizing any of them out.

SHAD_PKDUP_EVENT_LEVEL

Initial Version:	v5.000 **Syntax:** SHAD_PKDUP_EVENT_LEVEL=<num>
Last Change:	--
Default Value	0
Other Servers	TRS | DW
Valid Settings	0: Insert and update events. 1: Insert events only (duplicate key errors on updates are treated like any other error; replication stops).
Basic/Advanced	Advanced

Description

This parameter provides a means to control the response to "Unique Constraint Violation" conditions with respect to the triggering event. This parameter operates in conjunction with SHAD_PKDUP_LEVEL which specifies the response (shutdown replication, continue replication), and SHAD_PKDUP_MSG which specifies the amount of error logging.

When SHAD_PKDUP_EVENT_LEVEL is set to 1, objects will IGNORE duplicate key violations for insert statements and FAIL on duplicate key violations for update statements. If not specified in the shadparm.ini file, SHAD_PKDUP_MSG will default to 1. In this case,
if SHAD_PKDUP_EVENT_LEVEL=1 and encounters a duplicate key violation, all error messages and statements will be logged in the error log. Please DO NOT OVERRIDE the SHAD_PKDUP_MSG=1 default when SHAD_PKDUP_EVENT_LEVEL=1. If SHAD_PKDUP_MSG is accidentally set to 0 when encountering a duplicate key violation that causes it to fail, SHAD_PKDUP_MSG must be changed to 1 in the shadparm.ini. After resetting SHAD_PKDUP_MSG, the object may be started again and will then include all error messages and statement logging.

Related Parameters

SHAD_PKDUP_LEVEL
SHAD_PKDUP_MSG

SHAD_PKDUP_MSG

Initial Version:	v3.601 **Syntax:** SHAD_PKDUP_MSG=<num>
Last Change:	--
Default Value	1
Other Servers	TRS | DW
Valid Settings	0: Disable logging of duplicate primary key warning messages. 1: Enable logging of duplicate primary key warning messages.
Basic/Advanced	Advanced

Description

This parameter can be used to control whether duplicate primary key warning messages are logged. This parameter also controls whether or not the failed statement is logged along with the warning.

Related Parameters

SHAD_PKDUP_EVENT_LEVEL
SHAD_PKDUP_LEVEL

SHAD_TRACK_PERFORMANCE

Initial Version:	v3.933	**Syntax:**	SHAD_TRACK_PERFORMANCE=<num>
Last Change:	--		
Default Value	0 (Off)		
Other Servers	DW | TRS		
Valid Settings	0: Disable performance statistic collection. 1: Enable performance statistic collection 2: Enable performance statistic collection with additional prepared statement information.		
Basic/Advanced	Advanced		

Description

This parameter controls whether performance statistics are written to the log file.

Notes

An additional level has been added to this parameter so that Shadowbase prints the associated prepare statements for each corresponding table and row of statistics in the log file.

SHAD_UPDATE_NOT_FOUND

Initial Version:	v6.800	**Syntax:**	SHAD_UPDATE_NOT_FOUND=<num>
Last Change:	--		
Default Value	0		
Other Servers	TRS (sbarrtrs)		
Valid Settings	0: Stop 2: Skip and log 3: Skip without logging		
Basic/Advanced	Advanced		

Description

This parameter specifies how the array-based object behaves when it detects that not all rows in a batch of updates were applied. This indicates that one or more of these rows do not exist in the target database. Due to limitations of the array-based processing, the standard Insert Not Found (INF) functionality is not available.

Depending on the batch size and ODBC implementation, the object may not have enough information to determine which rows were skipped. In this case, the TRS will log a message with as much information as is available.

New & Modified User Messages

This section provides a description of the important user messages that are either new or modified since the previous general availability release (V6.703).

```
SHAD_BASE not found. SHAD_BASE environment variable must be set -
 Exiting
```

Cause: On this system the installation directory cannot be programmatically determined.

Effect: The process reports the condition and exits.

Recovery: Set SHAD_BASE as an environment or remove the SHAD_BASE_OVERRIDE parameter and restart.

```
SHAD_BASE_OVERRIDE set to 1
SHAD_BASE environment variable will not be used.
```

Cause: User specified settings accepted for parameter. (informational)

Effect: The process reports the condition, and continues.

Recovery: Information only.

```
SHAD_BASE is not properly defined in shadparm.ini - Exiting
```

Cause: The shadparm.ini that was loaded does not have SHAD_BASE defined in it.

Effect: The process reports the condition and exits.

Recovery: Add SHAD_BASE to shadparm.ini and restart.

Known Problems Remaining in Version 6.800

1. There is a one-to-many relationship between the SSQLD000.dat file and the series of SSTMNCxxx.dat files within a given cached SQL statement DOC database. If the SSQLD000.dat or SSQLD000.idx files are deleted or otherwise modified, replication may fail. If the SSQLD000.dat file becomes unusable or is accidentally removed, contact support for assistance and resolution to this issue.

   Note: The above condition causes replication to the target database to fail. However, the target database is not adversely affected; target database corruption does not occur.

2. Audit Log: The Audit Log image column SHAD_EVENT_TIMESTAMP reflects the wall clock time in which the Shadowbase NonStop Consumer process replicated the event to the HPE Shadowbase for Other Servers DOC database. This column is meant to reflect the NonStop audit trail event timestamp. That is, this timestamp does not represent the events source database activity time, but rather the time the event was replicated to the Open Server DOC database. This issue will be changed in an upcoming Shadowbase NonStop release, such that the SHAD_EVENT_TIMESTAMP column will contain the time the event was recorded in the HPE NonStop system audit trail.

3. DOC Writer and Source Collector restarts the TRS/TFS even if the TRS/TFS was stopped by SBMON. When enabled, the DOC Writer and the Source Collector will monitor and restart TRS/TFS if it stops running. If a TRS/TFS was manually stopped by an SBMON STOP command (normal shutdown), the DOC Writer and or Source Collector will continue to restart the TRS/TFS instead of leaving it in a stopped state.

4. ***The SBMON ROLL command must not be used on actively replicating objects or DOC corruption may result***. DOC rolls generated internally by the DOCW or collector object are handled correctly. However, there is a risk that a DOC roll triggered by a user issuing the ROLL command may do so while the replication object is in a critical state. If a manual SBMON ROLL command is required, shut down the relevant DOC writing replication object(s) (e.g., OPCOL, DOC Writer) and all database user sessions for source collection objects prior to issuing the ROLL command.

5. Use of Reserved Words as target SQL Table Column Names. In particular, the following reserved words are not supported for HPE Shadowbase for Other Servers target replication:

   AND
   WHERE
   VALUES

6. The HPE Shadowbase for Other Servers, Oracle Log Based Source Collector (*sborlog*) has the following limitations:

   a. HPE Shadowbase Log Source for Oracle does not support replication of tables with the following column types:

      BLOB
      CLOB
      NCLOB
      BFILE

   b. Selection criteria for the source collection is limited to including/excluding table names and users. These selections can be combined to select all but a specified set of tables for a user (specify the user and the set of tables to exclude), all tables for all users except for a specified set of users (specify the tables to be included and the user or users to exclude), and a subset users and tables (specifying both the tables and users). However, more complex selections may require multiple SBORLOG processes to be configured. If, for example, Users 1 and 2 both have tables named A and B, and you want to collect data from User 1's table A and User 2's table B, you will need to configure two SBORLOG processes.

   c. Shadowbase Log Source for Oracle currently only collects DML events (Inserts, Updates, and Deletes).

   d. HPE Shadowbase Log Source for Oracle does not support collecting tables with table names that are reserved words.

   e. Detailed collection stats are not supported (shadparm.ini parameter SHAD_OPCOLLECT_LOG_STATS=3). If you specify SHAD_OPCOLLECT_LOG_STATS=3, no statistics will be collected.

7. When configuring triggered-based replication for Microsoft SQL Server on a Windows 2008 R2 or later server, the generated trigger scripts may not be written to the Shadowbase data directory due to permissions problems. SQL Server writes the trigger scripts to the directory. If the SQL Server user does not have the correct permissions, the configuration will fail. The SQL Server user also needs execute access to the bin directory within the installation to collect data.

   SQL Server needs Full Access (F), Object Inherit (OI) and Container Inherit (CI) permissions to the Shadowbase data and bin directories. If you set the permissions on the installation directory *(%shad_base%),* both directories will inherit the permissions. You can check and set these permissions logged on as an administrator by using *icacls* in the command prompt. To check permissions on the Shadowbase installation directory:

```
CD %shad_base%
icacls *
```

and look for the SQL Server owner. If needed, grant the correct permissions on the Shadowbase base directory before configuring triggered based replication:

```
CD %shad_base%
icacls /grant:R <user>:(OI)(CI)F /T
```

8. Under certain transaction profiles when replicating from Other Servers to HPE NonStop Guardian, the Consumer will stop with an EMS message error message (#2017):

   ```
   SBOS-TO-NSK COVERSION BUFFER OVERRUN, SET
   SHAD_REMOTE_MAX_EVENTS BETWEEN 100 TO 400
   IN SHADPARM.INI
   ```

   This typically occurs if there are many empty transactions (transactions with no associated database modifications) sent to the NonStop Consumer. If this occurs, set the SHAD_REMOTE_MAX_EVENTS parameter in SHADPARM.INI to between 100 to 400 events, e.g.:

   ```
   SHAD_REMOTE_MAX_EVENTS=200
   ```

9. On Windows, when configured to “roll on size”, the DOC will not always roll at the correct size and can exceed the specified roll size significantly. This is related to a file system size reporting issue. Hence, you may need to set the roll size tens of MB’s less than you otherwise would.

10. Due to limitations in configuration record sizes, DNS names cannot be longer than 20 characters. If the DNS name exceeds 20 characters, use the dotted IP address instead.

11. When replicating from the NonStop in a multi-ported DOC environment, the SUSPENDUPD/RESUMEUPD command ***cannot be used*** if the SHAD_TRANS_EXPECTED_ENDS parameter is set to a value greater than 1. The SHAD_TRANS_EXPECTED_ENDS parameter is not required in configurations where the NonStop Shadowbase is sending to a single multi-ported DOC Writer. If the configuration includes multiple DOC Writers and Direct Writers, SHAD_TRANS_EXPECTED_ENDS is a required parameter.

    Only one commit is sent for a SUSPENDUPD command. If SHAD_TRANS_EXPECTED_ENDS is greater than 1, the DOC Writer will leave the SUSPENDUPD command in an uncommitted state, preventing the DOC files from being removed by the DOC cleaner.

12. Due to shell incompatibilities, the configuration scripts for HPE Shadowbase Log Source for Oracle do not work on Solaris 10 and AIX. If you need to configure HPE Shadowbase Log Source for Oracle for Solaris 10 or AIX, please contact Shadowbase Support.

13. If the SQL Server Native Client version 10.0 is installed on Windows 2008, the TRS and Direct Writer will fail as they cannot load the SQL Server client DLLs. This issue is still under investigation. Two work-arounds exist: either install a different version of the SQL Server Native Client, such as version 11.0, or use set up an ODBC DSN (data source name) and use the generic Shadowbase ODBC process.

14. We have seen the Oracle database driver process's memory use grow over time for certain configurations of HPE Shadowbase Log Source for Oracle when using Oracle 11 and 12 databases. This issue seems to be tied to certain Oracle patch levels. If this occurs in your configuration, report the instance to Shadowbase Support; you may need to apply some Oracle patches. Note that there is a workaround using the SHAD_OPCOLLECT_RECYCLE_CONNECTION and the SHAD_OPCOLLECT_QUERY_EXECUTION_LIMIT parameters. Enabling the SHAD_OPCOLLECT_RECYCLE_CONNECTION will cause the database connection to Oracle to be closed and reopened after SHAD_OPCOLLECT_QUERY_EXECUTION_LIMIT queries. This also causes the Oracle database driver process to restart, releasing the memory.

    Setting SHAD_OPCOLLECT_RECYCLE_CONNECTION will enable the connection recycling. SHAD_OPCOLLECT_QUERY_EXECUTION_LIMIT defaults to 250 queries before the connection is recycled, you can reduce it (resulting in less memory usage but poorer throughput) or increase it (resulting in better throughput but more memory usage) as your needs dictate.

15. Shadowbase connects to Microsoft SQL Server databases using the default setting for the AutoTranslate ODBC parameter, which is on. If Shadowbase is running on a different server from the SQL Server database and the two servers are using different ANSI code pages, character data stored in char, varchar, and text fields will automatically be converted by the ODBC driver. The ODBC driver performs the conversion by converting the data to UNICODE based upon the Shadowbase server's ANSI code page and then back to character fields from UNICODE using the SQL Server database's code page.

    If you want to disable the conversion, you need to setup an ODBC database source connection (DSN) with AutoTranslate configured off and to configure Shadowbase to use the DSN instead of connecting directly.

16. On Solaris servers, you may have to specify the LD_LIBRARY_PATH in your environment to have Shadowbase start correctly. If Shadowbase fails to start and logs a message in the error log similar to:

```
2015-05-06 14:41:45 -[14406] Critical Error: Cannot load library
```

```
(libOCIEIDLL.so) - ld.so.1: shadowbase: fatal: libclntsh.so.11.1: open
failed: No such file or directory
```

you will need to set the LD_LIBRAY_PATH environmental variable to either $SHAD_BASE/lib or $ORACLE_HOME/lib.

17. HPE Shadowbase Log Source for Oracle does not detect schema changes to replicated tables and may use an incorrect schema if the table is updated while Shadowbase is running. Added columns may not be picked up, and deleted columns may result in Shadowbase stopping.

18. When DBS Mapping is enabled, the TFS is not sending fully qualified datetime data to the NonStop for UPDATE statements, which can result in conversion errors on the NonStop.

19. Internal testing uncovered a number of limitations on the size of columns, tables, and statements:

    a. There is a limit to the size of a row in the DOC database which limits the size of statements (for EI Docs), cached statements (for cached DOCS) and statement data (for cached DOCS) to approximately 56K bytes.

    b. Table names are limited to 80 characters.

    c. Column names are limited to 74 characters.

20. There is an issue configuring HPE Shadowbase Log Source for Oracle collection using the shadconfig configuration script if you enter an incorrect username/name/Oracle SID combination when specifying the connection information using Option 1. If you do not exit the script but instead correct the connection information, subsequent entries made to the SHAD_USERS_INCLUDE, SHAD_TABLES_INCLUDE, SHAD_USERS_EXCLUDE, and SHAD_TABLES_EXCLUDE table entries will include a trailing space for the name, which will cause the data selection criteria to be incorrect.

    If you suspect that the criteria may be incorrect, you can issue the following select command against the appropriate table:

```
SELECT * FROM SHAD_TABLES_INCLUDE
WHERE TABLE_NAME LIKE '% ';

SELECT * FROM SHAD_TABLES_EXCLUDE
WHERE TABLE_NAME LIKE '% ';

SELECT * FROM SHAD_USERS_INCLUDE
WHERE USERNAME LIKE '% ';

SELECT * FROM SHAD_USERS_EXCLUDE
WHERE USERNAME LIKE '% ';
```

The select should return no rows. You can correct the issue using the following update command against the appropriate table:

```
UPDATE SHAD_TABLES_INCLUDE
SET TABLE_NAME=RTRIM(TABLE_NAME)
WHERE TABLE_NAME LIKE '% ';

UPDATE SHAD_TABLES_EXCLUDE
SET TABLE_NAME=RTRIM(TABLE_NAME)
WHERE TABLE_NAME LIKE '% ';

UPDATE SHAD_USERS_INCLUDE
SET USERNAME=RTRIM(USERNAME)
WHERE USERNAME LIKE '% ';

UPDATE SHAD_USERS_EXCLUDE
SET USERNAME=RTRIM(USERNAME)
WHERE USERNAME LIKE '% ';
```

21. HPE Shadowbase Log Source for Oracle may incorrectly stop with the following messages:

```
*** Potential Data Loss Detected
*** SQL_REDO for last event processed in previous query does not match
current query
***  SCN [<scn>]
***  RBASQN [<rbasqn>]
***  RBABLK [<rbablk>]
***  RBABYTE [<byte>]
***  SSN [<ssn>]
***  Previous SQL_REDO[<sql statement>;]
***  Current  SQL_REDO[<sql statement>;]
***  SHAD_OPCOLLECT_RESET_SCN is DISABLED, stopping
***  Enable SHAD_OPCOLLECT_RESET_SCN to continue processing prior to restart
Performing shutdown
```

This can occur if your text columns have binary 0 (ASCII NUL) characters in them. To continue past this error, set the SHAD_OPCOLLECT_RESET_SCN=1 parameter in the SHADPARM.ini file.

22. Special considerations are required when using a TFS as a relay server. The figure below shows a sample configuration where a TFS is used as relay server. Events are replicated from the source Oracle database into a DOC using the log source collector. They are read from the DOC and send to a second DOC via the relay TFS and DOCW. Finally, they were read from the relay DOC and applied in SQL/MX using another TFS and a Direct Writer.

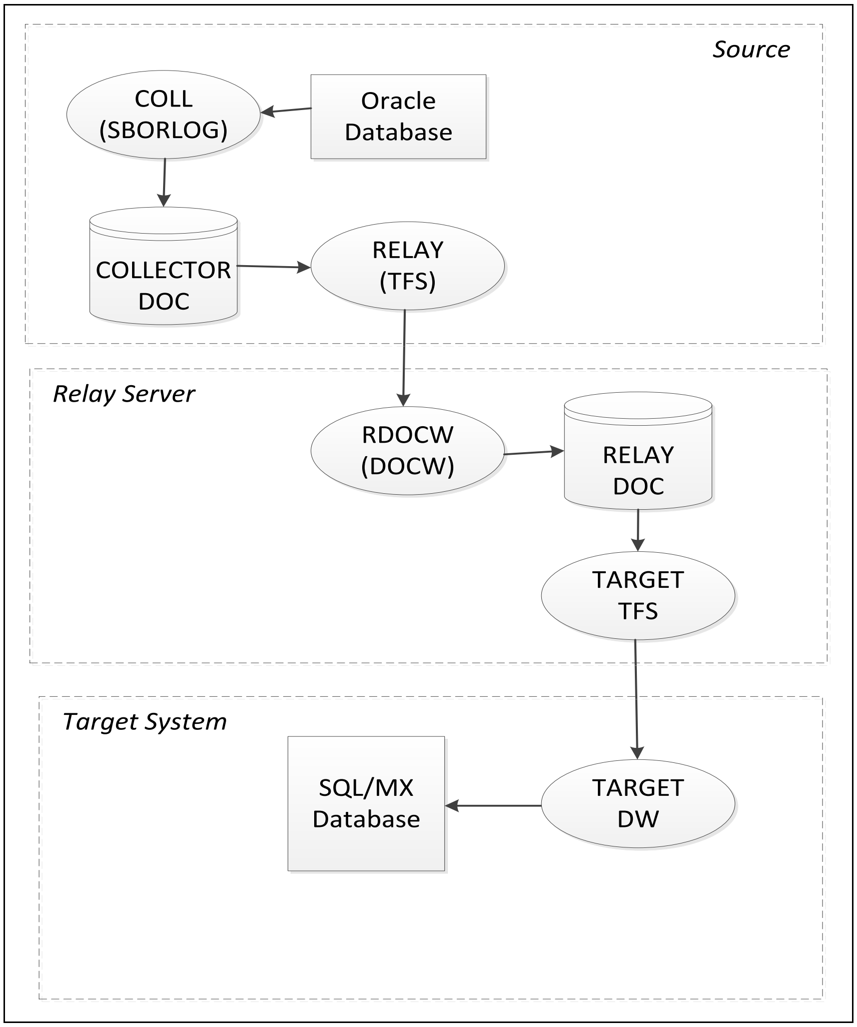


Figure 1 - Sample Relay Server Configuration

In a configuration such as this one, the relay TFS restarts must be coordinated with the collector restarts. Whenever the collector is restarted, the relay TFS should also be restarted to prevent duplicate data from being replayed.

23. There remains an issue with target replication into SAP HANA columns with the BOOLEAN data type. SAP HANA itself can accept literal string "TRUE" and "FALSE" values which it converts and stores as TINYINT, however HPE Shadowbase target replication currently only supports replicating numeric values into HANA BOOLEAN columns.

24. The new format DOC, which eliminates the TRANS DOC, is not compatible with Pulse processing. If you are planning on using Pulse processing, continue to use the old format DOC with trans processing enabled.

25. When replicating to MySQL, the ODBC library looks in /tmp by default for mysql.sock. Set the socket attribute in the odbc.ini section to the desired location, such as in the example below:

```
[myodbc64]
Description    = ODBC for MySQL
Driver         = /usr/lib64/libmyodbc5.so
Setup          = /usr/lib64/libodbcmyS.so
FileUsage      = 1
socket         = /var/lib/mysql/mysql.sock
```

Alternatively, make sure that the mysql.sock file is either located in, or is symlinked to /tmp/mysql.sock. Due to limitations of UnixODBC, the ODBC library that HPE Shadowbase uses to interface with ODBC databases on linux systems, the socket file must reside in the /tmp directory. You can find out the current location of the mysql.sock file by running `mysqladmin -p -u <user-name> variables` and finding the output under the ‘socket’ column. The socket file can then be symlinked to the /tmp directory with some version of the command `ln –s /var/lib/mysql/mysql.sock /tmp/mysql.sock`. The symlink may have to be recreated every time the system is rebooted or whenever the system cleans out the /tmp directory.

26. There is an issue in Shadowbase Doc Writer and Direct Writer support for SAP ASE target, also known as Sybase, such that users should discontinue use of SQL92 as the source SQL type and instead switch to SQLSBV1.

27. There is an issue with the Shadowbase Transaction Replay Server where a single TRS can only apply to a single database if the target database is MySQL. If you need to replicate into multiple MySQL databases you will need multiple TRS objects.

28. We have reports of a potential CTREE UpdateTrans error 101 occurring in a DOC Writer which causes a disconnection with the upstream consumer. In this situation, the Consumer and DOC Writer may continue to connect and disconnect. In this case, the workaround is to stop and restart the DOC Writer to get replication to continue as usual.

29. There is an issue on Windows 2016 where a TRS stopping can have Critical Errors that may cause problems upon the TRS restarting. This is caused by an issue between Shadowbase and Windows Services. The workaround for this is to set the Windows Service handling the TRS to restart itself after an error which will bring the TRS back up even if there were critical errors

30. There is an issue relating to the maximum statement cache depth of OSB Log Source objects where each Log Source Collector is limited to 1000 unique statements. A workaround to collect more than 1000 statements is to configure multiple Log Source collectors to keep each one under 1000 statements.

31. There are certain cases where when running a TRS or other HPE Shadowbase object from a service on Microsoft Windows Server 2019 or 2022, the errlog will be empty, however the object will be running fine. This problem is intermittent and nondeterministic. Restarting the object will correct the problem and cause the errlog to be recorded as expected.

32. There is an issue when running from an OPLOG to a TRS on the same system with high cache invalidation. This causes the TRS to abend once it has processed around 32 thousand transactions. The workarounds for this are to set SHAD_MAX_CURSORS to a higher value that prevents the need to remove and re-add cached statements (i.e. enough to hold all of the statements used by the TRS in cache), or to go from a source into a DOC then from the DOC to a TRS.

33. There is an issue attempting to configure a DB2 source collector on a Windows system to connect to an MQ Queue Manager on a remote server. This issue prevents the collector from establishing a connection to the Queue Manager. The workaround is to configure the DB2 source collector locally on the server where the MQ Queue Manager is configured.

34. Replication to DB2 columns with data types of BINARY, VARBINARY, and BLOB is not currently supported.

35. There is a constraint for Microsoft SQL Server tables called IGNORE_DUP_KEY that causes the database to suppress duplicate primary key errors. Avoid setting this constraint to `ON` for target replication to SQL Server as it can cause target replication objects to fail without a descriptive error message.

36. There is an issue with updates when replicating Oracle data of type LONG RAW. Inserts using that data work correctly but updating any data will result in any columns of LONG RAW type in that row to be cleared.

37. There remains an issue with the replication of MySQL columns with the type of BLOB. It is recommended that the target column type be changed to TEXT or another supported target character data type.

38. There is an issue with the replication of PostgreSQL columns with the type of BYTEA. It is recommended that an alternative target column type be chosen instead.

39. Pulse events do not currently function using DOC Blocking Level 2 due to a fault when inserting the event which causes the Pulse to be ignored. The workaround for this is to use DOC Blocking Level 1 or no DOC Blocking when Pulse events are required.

40. The Microsoft SQL Server ’SQL_VARIANT’ data type has limited support and has not been fully tested.

41. There is an issue with the Visual Studio project files that are intended to be used for building custom user exits on Windows. They will load in Visual Studio properly but for building the dll files you will need to run nmake using the CustomUE-MakeFile.mak file from an Administrator Command Prompt. Using Visual Studio to

build will create the files but result in an error at runtime stating that the entry point for USRXDLLINIT is missing.

42. Daily license checks will not be performed on DOCs waiting for a connection. If the connection is not established until after the check would have occurred it will be performed at the time of the connection. If a DOC is left in the waiting state this could cause the license to expire without any prior warning upon connection.

43. When replicating to Microsoft SQL Server with the data types BINARY, VARBINARY, and IMAGE, the data must be entered as hex values with no leading '0x'. A user exit will be needed to modify the values for this to work. Microsoft has also announced that the IMAGE data type will be deprecated in a future version of SQL Server and has begun encouraging using VARBINARY to store large data instead.

44. DBS mapping does not allow for $VOL.SUBVOL.TABLE syntax in the [DBS+] command. This may be a problem when replicating MP -> MP with syskey columns via ODBC/MX if the fully qualified $VOL.SUBVOL.TABLE table name is specified in the DBS PATHNAME. If DBS mapping is required, because an MP table has an MX reserved word as a column name, the TRS will fail.

    The workaround for this is to create a SQLMP ALIAS in the MX target environment to point to the target MP table and reconfigure the source NSB environment to use this alias value in the PATHNAME instead of the Guardian fully-qualified filename. The data will need to be re-collected from the source side and sent to the target OSS side again after the change is made, but that will allow the shaddbs file to have a normal table name in the [DBS+] line which will allow everything to work properly.

45. Replication to SQL/MX target table columns with data types of BINARY or VARBINARY are not currently supported.

46. Replication of fractional values in to MySQL TIME column types fails with a "Fractional truncation" error. This is due to an issue with certain versions of the ODBC MySQL driver. DATETIME and TIMESTAMP columns can handle up to 6 digits of precision appropriately.

    In addition, other MySQL ODBC driver versions cause Shadowbase to fail to replicate values into DATETIME target columns with a "Date overflow" error. For these issues and other similar MySQL issues, be sure to use the latest available ODBC driver.

47. Modifying the COLUMNS environment variable may cause the SBMON STATUS command to report objects as stopped incorrectly.

48. DB2 source events with BOOLEAN type are not processed by asnqcap and therefore cannot be replicated by Shadowbase.

49. When running Shadowbase on Windows in to a Hana or MySQL target if the first event processed results in a constraint violation it will be skipped along with any subsequent constraint violations until an event occurs that does not result in a violation. Once a single valid event has occurred constraint violations will be appropriately handled based on the PKDUP parameters.

50. There is an issue with the SBMON LIST command output for Direct Writers where the count of inserts, updates, deletes, and total transactions does not always update in sync with when transactions are processed. Replicating additional transactions will cause previously uncounted events and transactions to be added to this count.

51. When using SQLServer trigger source on Windows, the SHAD_BASE environment variable must be set to the current SHAD_BASE location. If it points to a previous install, the OPCOL EXTPR process will not start correctly. When set up correctly, non-trigger source environments use the proper SHAD_BASE whether it is set as an environment variable or not.

52. For array-based processing, DB tracing does not work on individual statements but rather for the full array.

53. SHAD_SKIP_DOC_OPEN_ERRORS does not work in environments with No Trans Docs. In order to get similar functionality, the TRS restart point will need to be advanced past the errors that need to be skipped.

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

To install Shadowbase, new steps have been added at the beginning to ensure that users acknowledge the license format change since the previous version. To start the installation, use the following instructions:

1. Copy the .zip file containing the software release onto the server where Shadowbase will be installed. If using FTP to move the files onto the server, be sure to use “binary” mode.
2. Unzip the .zip file on the server.
   a. Alternatively, unzip the .zip file on a PC first, then binary FTP all the contents to the server where Shadowbase will be installed.
3. Run the `` `run_first` `` script on the Shadowbase server.
   a. On Linux/Unix/OSS, run `` `. run_first` `` from a command prompt.
   b. On Windows, run `` `run_first.bat` `` from a command prompt.
4. The script will prompt for an acknowledgement of the license format change in this version. Say `` `Y` `` to the prompt to continue. If anything other than `` `Y` `` is entered, the run_first script will need to be run again.

After the license format change has been acknowledged, the installation can proceed using the normal instructions. To complete the installation and configuration of Shadowbase please see the HPE Shadowbase Other Servers Target Installation and Configuration Manual (for replication target installations) and the HPE Shadowbase Other Servers Source Installation and Configuration Manual (for replication source installations).

Changes for multiple instances on Windows

Previous versions of Shadowbase for Windows involved running a .msi installer to install the software on the system. This has been removed in favor of unzipping a zip file. Starting with this version of Shadowbase, to install on Windows, simply extract the contents of the .zip file to the desired install location. From this point on, the previous installation process works the same way for a single Shadowbase instance as before where the next step is to run sbinstall.bat.

This change also allows users to configure multiple instances of Shadowbase on a single Windows system. With previous versions of Shadowbase, there could only be one single active instance of Shadowbase on a Windows system since it relied on the SHAD_BASE system environment variable. Now there can be multiple instances of Shadowbase running simultaneously in different directories on the system.

In order to be able to have multiple instances of Shadowbase running simultaneously, there are some settings that must be configured in addition to the standard settings. The release package should first be unzipped into each of the locations where Shadowbase will run. The following shadparm.ini parameters must all be set before running sbinstall.bat.

- SHAD_BASE_OVERRIDE: If the SHAD_BASE environment variable is present, SHAD_BASE_OVERRIDE must be enabled in the shadparm.ini file located at the SHAD_BASE environment variable location.
- SHAD_ENV: This should be set to a unique identifier for each particular instance of Shadowbase on the system.
- SHAD_BASE: This should be set to location of this instance's Shadowbase installation.

Once these parameters are configured correctly, sbinstall.bat can be run for each instance. A single argument must be included in sbinstall.bat which needs to be identical to the SHAD_ENV value for each instance. For example, assume the following environment:

Old Shadowbase installation: C:\Shadowbase\v6600

System-wide SHAD_BASE environment variable set:

```
SHAD_BASE=C:\Shadowbase\v6600
```

We want to configure two new instances of Shadowbase v6800 and put them in the following locations:
C:\Shadowbase\v6800_A
C:\Shadowbase\v6800_B

The old Shadowbase version’s shadparm.ini file (C:\Shadowbase\v6600\data\shadparm.ini) must be modified to include the following:

```
[GENERAL]
SHAD_BASE_OVERRIDE=1
```

The first new Shadowbase v6800 instance’s shadparm.ini file (C:\Shadowbase\v6800_A\data\shadparm.ini) must include the following:

```
[GENERAL]
SHAD_BASE=C:\Shadowbase\v6800_A
SHAD_ENV=inst1
```

The second new Shadowbase v6800 instance’s shadparm.ini file (C:\Shadowbase\v6800_B\data\shadparm.ini) must include the following:

```
[GENERAL]
SHAD_BASE=C:\Shadowbase\v6800_B
SHAD_ENV=inst2
```

Now that all shadparm.ini files are complete, it’s time to run sbinstall.bat on the new instances. From C:\Shadowbase\v6800_A\bin, run:

```
sbinstall.bat inst1
```

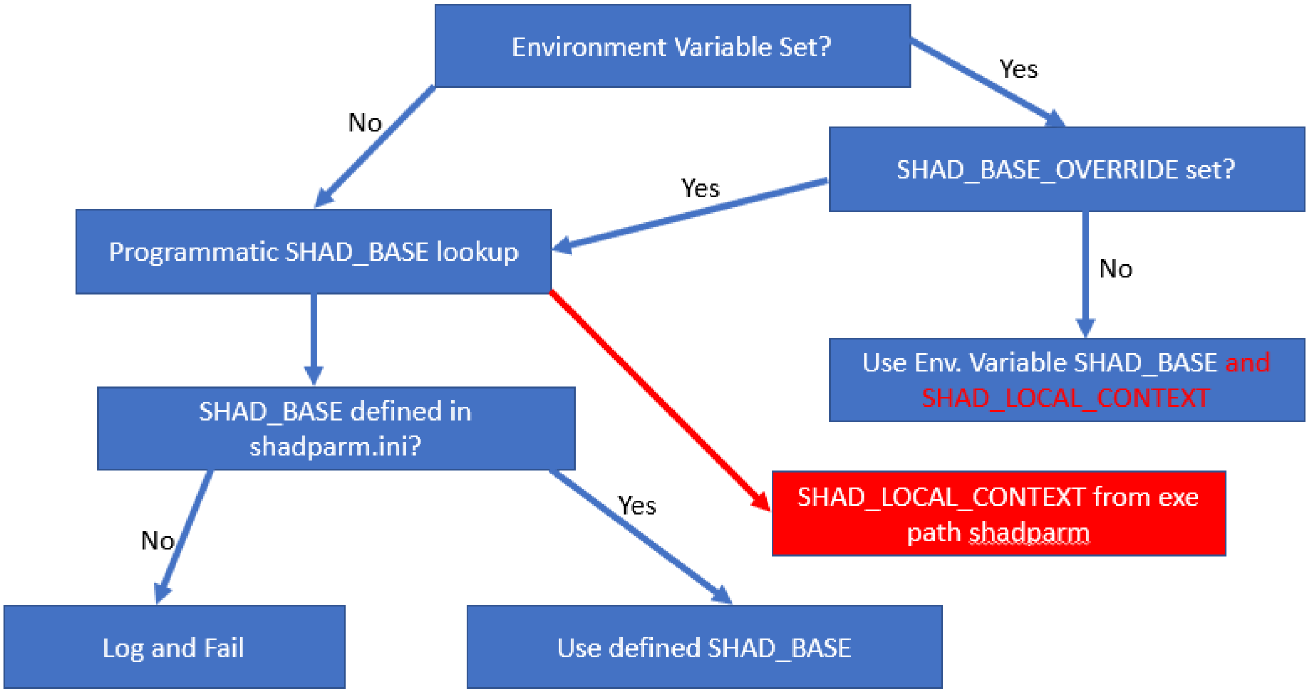
And from C:\Shadowbase\v6800_B\bin, run:

```
sbinstall.bat inst2
```

Note that the argument text of the sbinstall.bat command must match the SHAD_ENV parameter set in shadparm.ini for each instance. Otherwise the required services will be created using the parameter value but sbinstall won’t be able to start them. Without any argument provided, sbinstall.bat will behave in the same manner as in previous Shadowbase versions which won’t allow running multiple instances of Shadowbase on a single Windows system.

When SHAD_BASE_OVERRIDE is set, the logic will detect the initial installation directory by looking up the path of the executable being run. The shadparm.ini located in the installation data directory will be read to look up the SHAD_BASE parameter setting and use it as its ultimate value. SHAD_LOCAL_CONTEXT will also be read from the shadparm.ini located in the installation data directory before setting SHAD_BASE to the final value.

SHAD_BASE must be set in the shadparm.ini in the installation directory if the environment variables aren't being used. Often this will result in SHAD_BASE being set to the same path that the shadparm.ini is loaded from.


Environment Variable Set?
Yes
No
SHAD_BASE_OVERRIDE set?
Yes
Programmatic SHAD_BASE lookup
No
Use Env. Variable SHAD_BASE and SHAD_LOCAL_CONTEXT
SHAD_BASE defined in shadparm.ini?
SHAD_LOCAL_CONTEXT from exe path shadparm
Yes
No
Log and Fail
Use defined SHAD_BASE