SOFTWARE RELEASE DOCUMENT (SOFTDOC)

Product:	HPE Shadowbase for Other Servers
Release:	Gravic Version 6.300 T1124-AAC (SB REPL/HP UX) T1125-AAD (SB REPL/SUSE LINUX) T1126-AAC (SB REPL/RHEL LINUX) T1127-AAC (SB REPL/SOLARIS X86) T1128-AAC (SB REPL/SOLARIS) T1129-AAC (SB REPL/MS WINDOWS) T1130-AAC (SB REPL/AIX)
Date:	August 31, 2017
Copyright Notice:	Copyright Gravic, Inc. 1995 – 2017 (www.gravic.com)
File Name:	IPM6300_other_servers.pdf

NOTE: This softdoc covers new features and corrected problems for HPE Shadowbase for Other Servers, Version 6.300. It is available as an Adobe 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.

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.

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 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 these 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 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. However, Gravic currently expects that all general availability releases up to and including the date of the last release will contain the deprecated features.

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

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

Special Notes for Version 6.300

1. Version 6.300 is a “general availability” (non-restricted TCF) release. The previous non-restricted TCF release is version 6.220 (HPE version AAB) for all servers except OSS, and version 6.230 (HPE version AAD) for HPE OSS systems.

2. Due to licensing changes introduced in Version 6.100, existing installations of Shadowbase prior to that version will require a new password file in order to run after the upgrade.

3. Version 6.300 contains support for SAP Sybase ASE databases as a target using ODBC for cached statements only. Version 6.300 does not support Execute Immediate (EI) statements for Sybase. If you are using EI for Sybase, you cannot upgrade to Version 6.300.

4. HPE Shadowbase for Other Servers now obfuscates configuration data (in particular the passwords) for objects when it is stored in the COLLCONFIG data file. Version 6.300 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.

5. 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.

New Features in Version 6.300

This section provides a summary of the features added to HPE Shadowbase for Other Servers products since the previous general availability release (V6.220 for all servers except HPE NonStop OSS; V6.230 for HPE NonStop OSS).

1. Support for DB2 Log Based Source replication is available. To use this, IBM MQ and IBM DB2 Data Event Publishing are required. See *DB2 Source Collection for DB2 Replication* for further details.

2. HPE Shadowbase for Other Servers supports additional columns for Audit Logging. These additional columns are currently only supported for DB2 Log Based Source collection. For a full updated list of the available Audit Log columns, see *Extended Audit Logging Support*.

3. This release provides a new mode for DOC reading that allows events to be read in committed order without using the Transaction DOC (TRANS DOC). Eliminating the TRANS DOC file provides a significant performance increase in DOC writing, particular for the small transactions that are typical for transaction processing systems.

   Note: this configuration does not work for DOC Writers that have multiple TCP/IP ports configured (multi-ported DOCs) or DOCs created by the OPCOL collectors. It is also unavailable for DOC Writers running on Solaris and Solaris X86 systems. See the DOC Writing without the TRANS DOC section later in the SOFTDOC for more information.

4. This release provides a new mode for DOC blocking that allows additional event records to be blocked (grouped) in a single 'physical' record, significantly increasing the performance of statement DOC writing and reading. For small transactions, this mode can reduce the number of I/O operations to the statement DOC by two thirds.

   See the DOC Blocking Version 2 section later in the SOFTDOC for more information.

5. DOC cleaning uses a new algorithm when Transaction Processing is disabled as it relied on information in the TRANS DOC to determine which files can be cleaned.

   See the DOC Cleaner Changes section later in the SOFTDOC for more information.

6. HPE Shadowbase Log Source for Oracle now supports replicating from Oracle Multi-tenant (Container) databases.

   See the Oracle Multi-Tenant (Container) Databases section later in the SOFTDOC for more information.

7. A new User Exit call has been added to allow user exits in a TRS to control target database commits. Using the API, the user exit can have the current transaction commit immediately after processing the current event or at the next source commit. The user exit can also specify that the TRS will stop after the next commit. See the description under SBSETCOMMITBATCH, later in this document, for more information.

8. You can now optionally override the HPE Shadowbase Log Source for Oracle configuration table names with the SHAD_OPCOLLECT_INCLUDE_TABLES_NAME, SHAD_OPCOLLECT_EXCLUDE_TABLES_NAME, SHAD_OPCOLLECT_INCLUDE_USERS_NAME, and SHAD_OPCOLLECT_EXCLUDE_USERS_NAME shadparm.ini parameters.

9. In non-Windows Open environments, the SBMON is able to more efficiently confirm that an object is not running. This results in better performance for the STATUS command and all other commands that must check for running Shadowbase objects. This behavior is toggled with the SHAD_PROCESS_PRECHECK shadparm.ini parameter.

DB2 Source Collection for DB2 Replication

Configuring for DB2 Source Collection

Prerequisites:

- IBM DB2 Data Event Publishing
- IBM MQ
- IBM DB2 Client Libraries

Required configuration components:

- Tables to be replicated must be configured with data capture changes
- An MQ Queue Manager with:
    - a minimum of three Queues (an admin queue, a restart queue, and an event queue)
    - a connection channel configured to allow the IP address and user configured for the HPE Shadowbase DB2 Source OPCOL to connect and GET messages using the MQI client library
    - A listener
- A Q Capture Schema
- A Queue Publishing Map using the delimited format with the following settings:
    - If the database is setup for multibyte characters, code page 1208 must be used for publishing delimited messages
    - The character string delimiter must be set to a single quote (‘)
- Q Publication(s) capturing the tables to be replicated

For information on configuring the individual IBM DB2 and IBM MQ components, see the IBM DB2 and IBM MQ documentation.

In addition to the DB2 and MQ components, a Shadowbase open collector must also be configured. To add a DB2 Source Collector, specify the new open collector executable name, sbdb2col. An example configuration of a DB2 Source Collector follows:

```
+add DBCOL
             Entering EXIT or exit as the input for any
         parameter will break out without adding a new object

Types of Shadowbase Open servers:
  #  SERVER TYPE           INPUT         OUTPUT
  -  --------------------  ------------  ------------
  1. Database of Change    STP TCP/IP    DOC
  2. Direct                STP TCP/IP    SQL database
  3. Transaction Replay    DOC           SQL database
  4. Open Collector        SQL database  DOC
  5. Listener              GUI TCP/IP    Commands
  6. Transaction Forward   DOC           STP TCP/IP
  7. Remote NSK Consumer   STP TCP/IP    NSK i/o
  8. Custom (User)         NA            NA
  9. DOC Cleaner           DOC           NA
 10. Log Server            DOC           NA
Please enter number : 4
Executable Name
sbmscol = MS SQL Server source,
sborcol = Oracle source,
sborlog = Oracle LOG source,
sbsycol = Sybase source.
sbdb2col = DB2 source.
(sbmscol,sborcol,sborlog,sbsycol,sbdb2col) : sbdb2col
Event record format
1 = SQL Statement
2 = Comma Delimited
3 = Cached SQL Statement
(1, 2, 3) : 3
Transaction Processing ([Y]/N)?
Write transaction boundaries to event file ([Y]/N)?
Start Sequence Number [1]:
Source database format [SQL92]
(MSSQL, ORACLE, SYBASE, SQL92) :
Database server name :
Source database : [master] DBNAME
User name : [sa] username
Password : pass
```

For each DB2 Source OPCOL object defined, the following shadparm.ini file parameters must be specified to define the desired IBM MQ Queue Manager and Queue to collect from:

- SHAD_DB2COLLECT_EVENT_QUEUE_NAME
- SHAD_DB2COLLECT_QUEUE_MANAGER_NAME

Furthermore, to connect to the IBM MQ Queue Manager, the DB2 Source OPCOL object uses the MQI client library. In order to configure the connection, either the MQSERVER environment variable or a client channel definition table can be used. For information on configuring the MQSERVER environment variable, or configuring a client channel definition table, see the IBM MQ documentation on Connecting IBM MQ MQI client applications to queue managers.

For a complete list of all of the new shadparm.ini parameters added in this release, many of which pertain to DB2 source collection, see the section *New & Modified shadparm.ini Parameters*.

Determining DB2 Capture Server Restart Position

In addition to the standard information tracked by all OPCOL objects, DB2 Source Open Collectors track and report the DB2 last commit log sequence number(LSN) of the last MQ message processed. This information is reported as part of the output of the SBMON LIST command. The last commit LSN processed is also reported in the log on startup, and is included in some messages and stats reporting.

In some cases, it may be desireable to restart the DB2 Q capture server at the point last processed by the OPCOL. For example, in the event that bad messages are mistakenly placed on the MQ queue by another application or by an incorrect publication, and the queue is subsequently cleared. In this case, the last commit LSN tracked by the OPCOL can be used to restart the DB2 Q capture server at the point where replication was stopped.

The last commit LSN is noted in bold in the example LIST command output below. If no messages have been processed yet, 0000:0000:0000:0000:0000 will be reported.

```
+list DB2SC
-------------------Object Parameters--------------------------------------------

            Name: DB2SC                      |       SourceName: DB2SC
     Server Type: Open Collector Server      |   Read Open DOCs: N
      Executable: sbdb2col                   | Transaction Proc: Y
    Event Format: Cached SQL Statement       |         EarlyAck: 0
          SeqNum: 188                        |   Source Cleanup: N
 Transaction Key: 0000000000000000           | Write Boundaries: Y
 Max TCP/IP Send: 65534                      |         Protocol: 4
       IPAddress:                            |    Num Consumers: 1
Port Address(es): TRANS
-------------------Replicate Parameters-----------------------------------------
-
    Event Format: SQL92                      |    LSN: 0000:0000:0001:f493:6083
     Server Name:
   Database Name: MYDSN
-------------------Processed Events---------------------------------------------

Insert  Update  Delete  Total
------- ------- ------- -------
      2       0       0      10
            Working.....Do Not Interrupt!!!!!
-------------------Replay/Forwarding Server(s)----------------------------------

 Stats At:2015-09-14:16:40:50                        Since: N/A
                   DOC Total     Processed        TID/  IUD/  DOC Time
 Name Type Status  Seq TIDs      TIDs      %Done  sec   sec   Remaining
----- ---- ------- --- --------- --------- ------ ----- ----- ---------
DB2TR TRS  stopped 188         2         2 100.00   0.0   0.0    N/A
```

```
********************************************************************************
```

MQ Message Reject Handling

Support has been added for handling MQ message rejects that could not be processed in different ways. A shadparm.ini parameter, SHAD_DB2COLLECT_MQ_MESSAGE_ERROR_ACTION, can be used to configure Shadowbase to skip the message and continue, skip the message and continue without logging a message, or to stop.

Extended Audit Logging Support

HPE Shadowbase for Other Servers Audit Logging now supports additional audit context columns for audit logging. The new audit log columns are:

- SHAD_DB2_MESSAGE_PUT_DATE
- SHAD_DB2_MESSAGE_PUT_TIME
- SHAD_DB2_MESSAGE_PUT_TIMESTAMP
- SHAD_SOURCE_SCHEMA
- SHAD_SOURCE_TABLE
- SHAD_DB2_TRANSID
- SHAD_DB2_COMMIT_LSN

An updated table of the available audit log columns is as follows :

Column Name	Oracle target Data Type	SQL Server / DB2 Data Type	Comments
SHAD_BATCH_GROUP	NUMBER(20) min. NOT NULL	BIGINT NOT NULL	This field is generated by the target system and is a derived 'timestamp like' value. Provides for a 'group by' query capability, as all events across all audit log tables for the same transaction will have the same value. Column definition must support 64-bit numeric values.
SHAD_BATCH_ID	NUMBER(20) min. NOT NULL	BIGINT NOT NULL	Contains the source system's TMF transaction id. Column definition must support 64-bit numeric values.
SHAD_EVENT_SEQ_NO	NUMBER(18) min. NOT NULL or INTEGER NOT NULL	INTEGER NOT NULL	This field is generated by the target system and is a sequentially ascending number (not necessarily serially ascending) of an event's ordinal position in the SHAD_BATCH_ID (TMF transaction) as provided by the source system. To review the order of the transaction's events from the source system, read along this column in ascending order across all audit log tables.
SHAD_EVENT_TYPE	CHAR(1)	CHAR(1)	I = Insert, U = Update, D = Delete

Column Name	Oracle target Data Type	SQL Server / DB2 Data Type	Comments
SHAD_EVENT_TIMESTAMP	DATE	DATETIME(Oracle or SQLServer) TIMESTAMP (DB2)	Source system's audit trail timestamp of when (approximately) the I/O was applied into the source database. This timestamp is stored and reported in GMT.
SHAD_ADT_PREFIX	CHAR(2)	CHAR(2)	Source node's audit trail prefix (for identifying the specific audit trail on the source node of where this event came from).
SHAD_ADT_SEQ_NUM	NUMBER (6) min. or INTEGER	INTEGER	Source nodes' audit trail sequence number (for identifying the specific audit trail on the source node of where this event came from).
SHAD_ADT_RBA	NUMBER(20) min.	BIGINT	Source nodes' RBA (for identifying the specific audit trail location on the source node of where this event came from). The RBA uniquely identifies this event's location in the source node's audit trail. Column definition must support 64-bit numeric values.
SHAD_DB2_MESSAGE_PUT_DATE	CHAR(7)	CHAR(7)	For DB2 source replication, the date that the message was put on the MQ Queue. The date includes the year and day of year in the format YYYYnnn.
SHAD_DB2_MESSAGE_PUT_TIME	CHAR(12)	CHAR(12)	For DB2 source replication, the time that the message was put on the MQ Queue. The time is in the format HHMMSSmmmmmm.
SHAD_DB2_MESSAGE_PUT_TIMESTAMP	DATE	DATETIME(Oracle or SQLServer) TIMESTAMP (DB2)	For DB2 source replication, message put timestamp for when the message was put onto the MQ Queue. Derived from the date and time from the delimited message header, reformatted into a timestamp.
SHAD_SOURCE_SCHEMA	VARCHAR(128)	VARCHAR(128)	The source schema name. Currently available only for DB2 source replication.
SHAD_SOURCE_TABLE	VARCHAR(128)	VARCHAR(128)	The source table name. Currently available only for DB2 source replication.
SHAD_DB2_TRANSID	CHAR(24)	CHAR(24)	The DB2 source transaction id in hex in the format hhhh-hhhh-hhhh-hhhh-hhhh.
SHAD_DB2_COMMIT_LSN	DATE	DATETIME(Oracle or SQLServer) TIMESTAMP (DB2)	The DB2 source transaction commit timestamp retrieved from the MQ delimited message header.

Additionally, user exit processing now allows for retrieval of audit context data from audit log enabled DOC files. Any available audit context column information can be retrieved even when audit logging data is not being applied.

Overview of DOC Writing and Reading Performance Enhancements

Several enhancements to DOC writing and reading have been made to significantly increase the performance of SQL/MX replication. These enhancements include:

- Enhancements to the DOC reading algorithm to allow replay of events in committed transaction order (the normal DOC replay order) without the use of a transaction DOC. Previous versions required both a transaction DOC and a statement (event) DOC for replay by the TRS and TFS processes.

- Enhancements to the DOC blocking algorithm (DOC Blocking Version 2) to enable BEGIN, COMMIT, and ABORT events to be combined with Insert, Update, and Delete events. This allows the events for a single short transaction to be written into a single DOC record, rather than requiring multiply I/O's in the previous algorithm.

- Enhancements to the DOC blocking algorithm to allow DOC blocking to span multiple IPCs from the source system.

- Enhancements to the SQL/MX configuration TACL macros SBUNDROS and SBBIDROS to support configuration of the new DOC writing parameters. Please see the section titled Configuring with the SBUNDROS and SBBIDROS Macros for more information.

These enhancements significantly reduce the number of I/O operations between the DOC writer and the file system, resulting in a significant improvement in DOC writing performance when taken together.

The table below shows the number of logical I/Os to the file system for a transaction that consists of an INSERT and an UPDATE operation. In the least efficient DOC writing mode (using the TRANS DOC and with no DOC blocking), 22 logical I/O operations are required for the transaction – 6 for the TRANS DOC and 16 for the EVENT DOC). In the most efficient mode, only 4 logical I/O operations are required.

Note that the table is referencing logical operations. DOC writing uses a library that implements a B-Tree structure on the DOC files and indexes. As a result, each logical operation may result in multiple I/O related calls to the file system as the DOC Writer reads and updates index and data blocks. These I/O-related calls are more efficient on Linux/Unix file system implementations than they are on the NonStop OSS file system implementation (eg, DMA into the file system structures vs message passing to a DP2 process). Hence, DOC writing was much slower in the OSS environment than it was in a 'normal' Linux/Unix environment.

MODE	Event	Transaction DOC Ops	Event DOC Ops

Prior Versions, No DOC Blocking	Begin	1 base insert + 2 index inserts	1 base insert + 3 index inserts
	Insert		1 base insert + 3 index inserts
	Update		1 base insert + 3 index inserts
	Commit	1 base update + 2 index updates	1 base insert + 3 index inserts
	Total	***6 logical I/O Operations***	***16 logical I/O Operations***
Prior Versions, DOC Blocking	Begin	3 inserts	1 base insert + 3 index inserts
	Insert		0 (combined with Update)
	Update		1 base insert + 3 index inserts
	Commit	3 updates	1 base insert + 3 index inserts
	Total	***6 logical I/O Operations***	***12 logical I/O Operations***
No Trans DOC, DOC Blocking	Begin		1 base insert + 3 index inserts
	Insert		0 (Combined with UPDATE)
	Update		1 base insert + 3 index inserts
	Commit		1 base insert + 3 index inserts
	Total		***12 logical I/O Operations***
No Trans Doc, DOC Blocking Version 2	Begin		0 (Combined with COMMIT)
	Insert		0 (Combined with COMMIT)
	Update		0 (Combined with COMMIT)
	Commit		1 base insert + 3 index inserts
	Total		***4 logical I/O Operations***

DOC Writing without the TRANS DOC

DOC Writing without the TRANS DOC is enabled by answering “N” to the “*Transaction Processing ([Y]/N)?”* prompt when configuring the DOC Writer through SBMON with the *ADD*, *EDIT*, or *EDIT!* command. This will cause the DOC Writer to create only the Event DOC related files.

Gravic suggests that all customers consider running in this new mode for DOC/TRS environments.

This configuration cannot be used for DOC Writers that have more than one TCP/IP port configured (multi-ported DOCs). It will also switch the DOC cleaner to use an algorithm that uses the DOC modified time to determine when to purge the DOC file.

There are some limitations to keep in mind when switching to this configuration:

- Previous versions of Shadowbase cannot replay data from DOCs created without Transaction Processing. However, previous versions can replay data from DOCs created by this version with Transaction Processing enabled.

- This version cannot replay data from previous versions of Shadowbase when DOC writing is configured without Transaction Processing.

- This version can replay data from DOCs created by this version with Transaction Processing enabled even when the configuration is altered to disable transaction processing.

Please review these limitations carefully when planning any fail-back strategy.

If you are upgrading an existing configuration and want to disable Transaction Processing, you have two choices:

1) Completely drain the existing DOCs created by the previous version and roll both the DOC writer and TRS / TFS objects to a new DOC. Then edit DOC Writer configuration to disable Transaction Processing.

2) Upgrade to the new version of Shadowbase and restart replication without disabling Transaction Processing. Continue to run replication until the DOCs created by the previous version of Shadowbase have been completely processed, and then modify the configuration to disable Transaction Processing.

If you want to retain the option of reverting back to the previous version of Shadowbase, do not disable Transaction Processing.

DOC Blocking Version 2

DOC Blocking Version 2 combines the BEGIN, COMMIT, and ABORT events with the INSERT, UPDATE, and DELETE events when writing to the DOC. In the first implementation of DOC blocking, the DOC Writer wrote separate Event DOC records for the BEGIN, COMMIT, and ABORT events – there were always at least three records written per transaction. For small transactions that fit in a single Event DOC record, the new version of DOC blocking will combine the events into just a single record containing all events.

The new version of DOC blocking is enabled setting the SHAD_DOC_BLOCKING parameter to 2.

There are a couple of items to note about the new version of DOC blocking:

- This version of Shadowbase can read DOC files created with any blocking mode, regardless of the setting of the DOC blocking parameter.

- Previous versions of Shadowbase cannot use DOC files created with DOC blocking version 2.

Note: If you do try to replay data blocked with the version 2 algorithm using a previous version of Shadowbase, the results are unpredictable. Data corruption and loss will likely occur.

Blocking Across Multiple IPCs

This version of Shadowbase introduces an advanced mode for DOC blocking that, in certain transaction profiles, can increase the effectiveness of blocking. The default mode of DOC blocking (for both version 1 and 2) will block events in the same IPC together but will not block events in different IPCs together. In this previous mode, the DOC Writer processes all of the events in the IPC by either writing them to the DOC (if the transaction is completed) or by caching them for further blocking. When the DOC Writer has processed all events in the IPC, it flushes any events left in its cache ensuring that it has written all data in the IPC to DOC.

The new mode eliminates the cache flush at the end of processing for the IPC. Instead, the events remain in cache either until the Event DOC record is full or until the transaction completes. This mode only guarantees that the DOC Writer writes events for completed transactions to the DOC prior to the completion of processing of an IPC, not all events.

The flushing mode is controlled by the SHAD_DOC_BLOCKING_FLUSH_CACHE parameter. This parameter should only be changed under the advice of Shadowbase, most users should use the default mode (flushing after each IPC enabled). See the SHAD_DOC_BLOCKING_FLUSH_CACHE parameter description for more information.

DOC Cleaner Changes

If Transaction Processing is disabled for DOC Writing, the DOC Cleaner will use file modification times of the Event DOC files to determine which files to keep and which files to purge. With Transaction Processing disabled for DOC Writing, the TRANS DOC file is no longer created or updated. The TRANS DOC files contained information the DOC Cleaner used to determine the oldest Event DOC that still contained data to be processed by TRS or TFS.

This information is no longer available to the DOC Cleaner. Instead, it uses the file modification times to determine which files it can safely delete. Specifically, it gets an approximation of the time the first event was written in the file by taking the modification of the previous DOC file. It will then use the file modification times of previous files to make sure it keeps the specified number of hours of data around.

The number of hours it keeps is specified by the *DOC Generations or Hours to Save* prompt when adding or editing the DOC Cleaner.

Configuring with the SBUNDROS and SBBIDROS Macros

The SBUNDROS and SBBIDROS TACL macros, shipped with the T1122H06_AAG (SB Repl/Guardian) SPR, Gravic version 6.320, have been enhanced to allow the new writing and blocking modes to be configured directly.

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

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

SB_ADD CONSOS (SBUNDROS Macro, DOCW/TRS Configuration)
Two optional parameters have been added to the end of the SB_ADD CONSOS command to specify transaction processing and the DOC blocking mode. Both should be specified or omitted.

The new syntax is:

```
sb_add consos <src_cons_pname> <src_cons_cpu_p> <src_cons_cpu_b> <src_cons_pri>
              <src_ip_pname_p> <src_ip_pname_b> <tgt_ip_addr_p>
              <tgt_ip_addr_b> <port> <tgt_ip_pname_p> <tgt_ip_pname_b>
              <tgt_pname_p> <tgt_pname_b> <tgt_cpu_p> <tgt_cpu_b> <tgt_pri>
              <db_catalog> <db_schema> <db_user> <db_password> <db_dsn>
              <trs_pname> <trs_cpu> <trs_pri> <dcl_pname_p>
              <dcl_pname_b> <dcl_cpu_p> <dcl_cpu_b> <dcl_pri> <trans_proc> <docblk ver>


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


Example:
> sb_add consos $gpjc1 0 1 131 $Zb03b $Zb02b 10.1.50.70 10.1.50.60 30100 &
$zb03b $Zb02b $GPDCP $GPDCB 1 0 132 catalog schema db.user pass &
```

```
TDM_Default_DataSource $GPJTR 1 133 $GPCLP $GPCLB 0 1 130 N 2
```

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

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

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

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

The new syntax is:

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

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

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

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

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

Oracle Multi-Tenant (Container) Databases

Introduction

HPE Shadowbase Log Source for Oracle now supports replicating from multi-tenant Oracle configurations. In a multi-tenant Oracle configuration (also known as a container database, or CDB), the Oracle database is configured with 0, 1 or many customer-created pluggable databases (PDBs) contained in a single Oracle Database instance. Each CDB has a root container, a seed PDM (Oracle template), and 0 or more user created PDBs.

A HPE Shadowbase Log Source for Oracle collector (OPCOL) collects events from the root container (CDB$ROOT) and can collect events for replication from only one PDB. If you want to replicate data from multiple PDBs, you will need to configure multiple OPCOLs, one for each database. You can, however, configure multiple OPCOLS to collect from the same PDB.

Figure 1 - Sample Container Database Configuration, below, shows a typical active / passive disaster recovery configuration. Shadowbase is replicating data from the CDB East node to the CDB West Node. In this case, Shadowbase is replicating two PDBs (Sales and Marketing), so there are two OPCOLs configured, one for each PDB. Note that Shadowbase collects the events from the root container on the source, but applies the events into the appropriate PDB on the target.

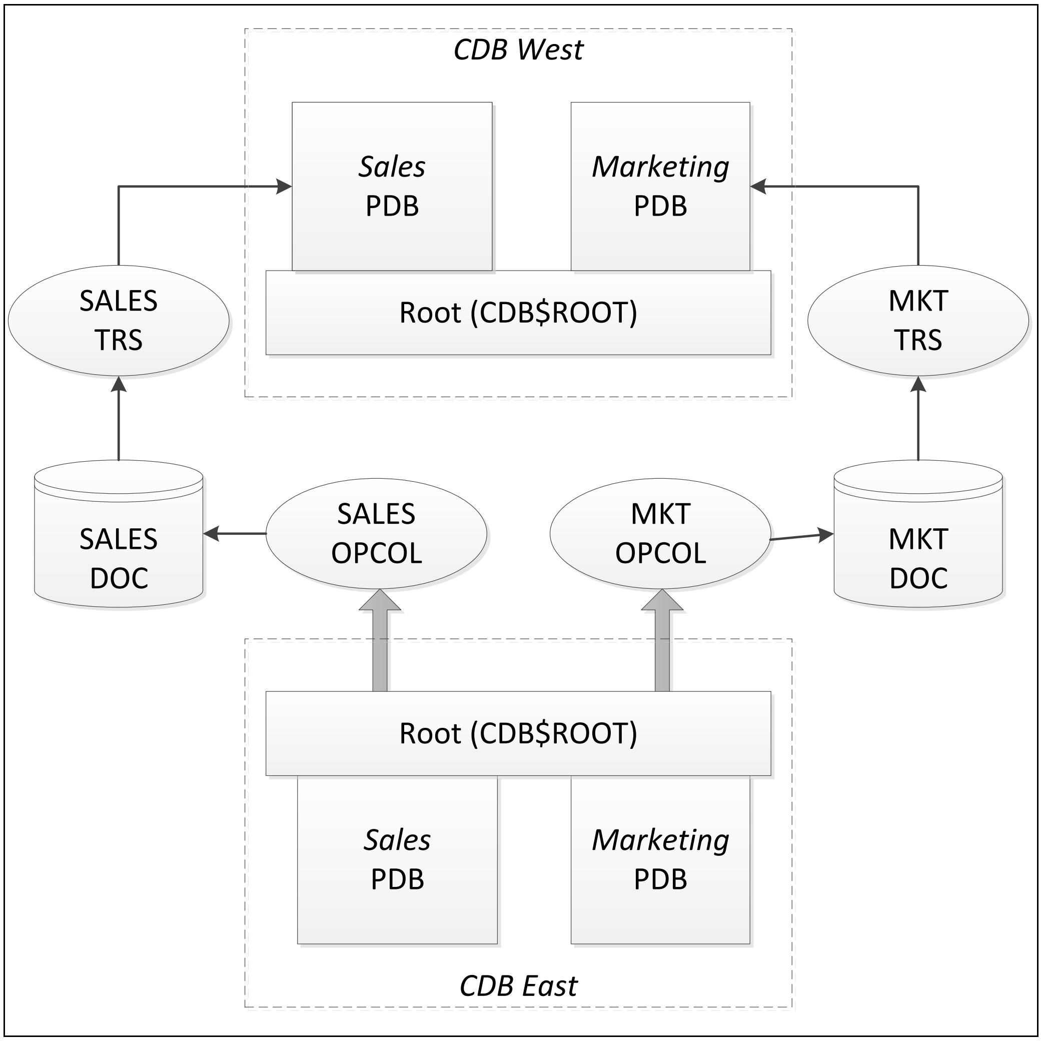

Figure 1 - Sample Container Database Configuration

There are a number of special considerations when configuring collection from a PDB which are discussed in the following sections. These include:

- Configuring the user and connection to the root database.
- Specifying the PDB to be replicated.
- Creating a link to the PDB
- Specifying the SHAD_USERS_INCLUDE, SHAD_USERS_EXCLUDE, SHAD_TABLES_INCLUDE, SHAD_TABLES_EXCLUDE location.

Configuring the Collector’s User

For Oracle Container Databases, additional considerations apply. The user must be an Oracle common user. Oracle common user names begin with “C##” or “c##”. Only Oracle common users can access the root database (CDB$ROOT) and V$LOGMNR data.

The user must also have access to the pluggable database where Shadowbase user and table configuration tables (SHAD_USERS_INCLUDE, SHAD_USERS_EXCLUDE, SHAD_TABLES_INCLUDE, SHAD_TABLES_EXCLUDE) reside. The root database cannot contain any user tables, hence those configuration tables must reside in a pluggable database. Usually, the tables will be created in the PDB that is being replicated, however, that is not a requirement.

The user also must have the following privileges granted:

- Connect to the CDB with the configuration tables.
- Create database link to the CDB with the configuration tables.
- Logmining privileges
- Reading container data

Below is a sample script used at Gravic to create test users to access CDBs. It creates a test user named c##qasource.

```
create user c##qasource container=all profile default identified by *****
default tablespace users
temporary tablespace temp
account unlock;
grant connect to c##qasource container=all;
grant create procedure to c##qasource container=all;
grant create session to c##qasource container=all;
grant create table to c##qasource container=all;
grant create trigger to c##qasource container=all;
grant create type to c##qasource container=all;
grant create library to c##qasource container=all;
grant create database link to c##qasource container=all;
grant select any table to c##qasource container=all;
grant select any dictionary to c##qasource container=all;
grant select any transaction to c##qasource container=all;
grant execute on DBMS_LOCK to c##qasource container=all;
grant execute on DBMS_UTILITY to c##qasource container=all;
grant execute on DBMS_LOCK to c##qasource container=all;
grant execute on UTL_FILE to c##qasource container=all;
grant execute_catalog_role to c##qasource container=all;
grant logmining to c##qasource container=all;
grant unlimited tablespace to c##qasource container=all;
alter user c##qasource set container_data=all container=current;
```

Table 3 - Sample Common User Configuration Script

Specifying the Pluggable Database (PDB)

Each Shadowbase for Oracle Log Source collector can collect events from only one Pluggable Database. This is done using the shadparm.ini parameter SHAD_OPCOLLECT_PDB_NAME. This parameter is required for any collector selecting events from a container database, even if the container database only contains one pluggable database. See SHAD_OPCOLLECT_PDB_NAME section for more information.

Creating a Link to the Pluggable Database

The Shadowbase for Oracle Log Source collector connects to the root of the Container Database to collect events. The root cannot contain any user tables, these must be in a

PDB. The collector uses several tables to specify which user and table events are collected (the SHAD_USERS_INCLUDE, SHAD_USERS_EXCLUDE, SHAD_TABLES_INCLUDE, and SHAD_TABLES_EXCLUDE tables) which must be defined in one of the PDBs. The collector needs a link to access the tables from the root.

You can specify the name of the link with the parameter SHAD_OPCOLLECT_PDB_LINK_NAME. If you do not specify anything, the collector will use "SBLINK_TO_<name>", where <name> is the name of the PDB specified by the SHAD_OPCOLLECT_PDB_NAME parameter. For example, if you are replicating from the SALES PDB, the collector will create a link called SBLINK_TO_SALES linking from the CDB root to the SALES database.

By default, the collector will recreate the link every time it restarts by dropping the link if it exists, and then creating it. You can override this behavior and pre-create the link with the parameter SHAD_OPCOLLECT_PDB_CREATE_LINK.

Specifying the Configuration Table Names for Container Databases

You do not have to specify the configuration table names for Oracle Container Databases. By default, it will use the standard names (SHAD_USERS_INCLUDE, SHAD_USERS_EXCLUDE, SHAD_TABLES_INCLUDE and SHAD_TABLES_EXCLUDE) from the replicated PDB. However, you can use the SHAD_OPCOLLECT_INCLUDE_TABLES_NAME, SHAD_OPCOLLECT_EXCLUDE_TABLES_NAME, SHAD_OPCOLLECT_INCLUDE_USERS_NAME and SHAD_OPCOLLECT_EXCLUDE_USERS_NAME shadparm.ini parameters to override the table names. If you do override the names, please note that the database link must be included in the parameter value to link to the correct pluggable database. Assuming for example, that you wanted to use the table SALES_USERS when replicating the SALES database, you would use:

```
SHAD_OPCOLLECT_INCLUDE_USERS_NAME = SALES_USERS@SBLINK_TO_SALES
```

The example above uses the default link name, you can also use different name using the SHAD_OPCOLLECT_PDB_LINK parameter. Note that the name needs to be appended to the table name with an at (@) sign.

Problems Corrected in Version 6.300

This section provides a summary of the problems corrected in HPE Shadowbase for Other Servers products since the previous general availability release (V6.220 for all servers except HPE NonStop OSS; V6.230 for HPE NonStop OSS).

1. The TRS encountered a segmentation fault from the c-Tree library when it attempted to log a timestamp for a c-Tree database error message to the CTSTATUS.FCS log file. The library will now log the error to the CTSTATUS.FCS log file and will return c-Tree error 527. While this error indicates that index files must be reloaded, our testing has shown that in most cases this is a transient error that occurs when the TRS or TFS is reading close to the end of file. By default, the TRS or TFS will stop when this error occurs, however, you can set the SHAD_RETRY_CTREE_ERROR_527 parameter to have it retried.

2. The version of libdb2.so.1 installed with HPE Shadowbase and loaded by Direct Writer and TRS objects was incompatible with DB2 v10.5. This resulted in an error message such as the following:

   ```
   Critical Error: Cannot load library (libDB2CacheDLL.so) -
   /home/shadtest/TEST/v6110L/bin/../lib/../lib/libdb2.so.1: undefined symbol:
   irc__get_msg
   ```

   The workaround was to rename, move, or delete the libdb2.so.1 file in the HPE Shadowbase lib/ folder and to use the version of libdb2.so.1 from the DB2 10.5 installation. This behavior has been changed to default to using the version of libdb2.so.1 from the DB2 installation on the system.

   There are three ways to use the DB2 libdb2.so.1 library on the system from the DB2 installation. This can be done by creating a symbolic link to it in the HPE Shadowbase lib/ folder, by having it in the default shared library search path, or by setting LD_LIBRARY_PATH to add its location to the shared library search path.

   The version of libdb2.so.1 that was used in previous versions is now installed in the lib/optional/ folder of the HPE Shadowbase installation. To use it instead, move it back into the HPE Shadowbase installation lib/ folder.

3. When the SHAD_OPCOLLECT_SCAN_FOR_NULL_BYTES was set to STOP (the default), SKIP, or SKIPNOLOG, the HPE Shadowbase Log Source for Oracle collector incorrectly changed single back slash characters (‘\’) into two back slash characters (‘\\’) in text fields. The other settings for the parameter (OFF, KEEP, or REPLACE) worked properly. This has been corrected, all modes will correctly handle ‘\’ characters.

4. HPE Shadowbase Log Source for Oracle incorrectly converted date fields in the primary key to NULL for delete events.

5. HPE Shadowbase Log Source for Oracle looped when CONTINUOUS_BOTH collection is specified and there are no archive logs. While looping, it issued the error messages:

   ```
   Error: Drec:01, SQLSTATE: 00000, SQLCODE: 1405, Msg: ORA-01405: fetched
   column value is NULL
   GetFirstLogSCN(): fault detected; rc[-1]
   ```

   Note that archiving must be enabled for CONTINUOUS_BOTH collection, however.

6. In the beta version of 6.300 (released as restricted TCD to some of Gravic's customers), sborlog (the HPE Shadowbase Log Source for Oracle collector) failed with message:

   ```
    Processing fault detected; sqlcode[904]; Err[ORA-00904: "RN": invalid
   identifier]
   ```

   when CONTINUOUS_ARCHIVE mode was configured.

7. HPE Shadowbase Log Source for Oracle collection now supports Oracle databases configured with multiplexed redo logs. In previous releases, the collector would fail without collecting any data.

8. The SQL Server source collector stopped collecting data when collecting lots of data that includes aborts. The failure also called the TRANS error log to flooded with the following message:

   ```
   Drec:01, SQLSTATE: 08S01, SQLCODE: 233, Msg: [Microsoft][SQL Server Native
   Client 11.0]Shared Memory Provider: No process is on the other end of the
   pipe.
   ```

9. In previous releases, HPE Shadowbase Log Source for Oracle collection did not collect events identified as belonging to the Oracle SYS user. These events will now be collected. See the SHAD_OPCOLLECT_ALLOW_SYS_USER parameter for more information.

10. The TRS did not correctly identify the ODBC driver for MySQL 5.6 on Linux, forcing the user to manually set the error codes for MySQL using the SHAD_ODBC_GENERIC_CODES parameter.

11. The non-interactive HPE Shadowbase objects (such as the DOC Writer, TRS, and TFS) left the standard input file (stdin) open for the lifetime of the process. They will now close stdin shortly after starting.

12. HPE Shadowbase objects could, in rare cases, get into a recursive call loop that ended with object creating a core dump with a stack overflow when a Log Server DOC is configured. HPE Shadowbase objects write error messages to both the text error log and the Log Server DOC when the log server DOC is enabled. If an error occurs while writing to the log server DOC, that error was being written to both the log file and the DOC, which could result in another error message, and so on.

Message logging has been modified to log errors pertaining to the Log Server DOC only to the log file.

13. SHAD_OPCOLLECT_INSERTS, SHAD_OPCOLLECT_UPDATES, and SHAD_OPCOLLECT_DELETES defaulted to 0 in all cases. If none of the parameters were not specified in the shadparm.ini file, the collector would shut down as no events where specified to be collected. This has been changed so that all will default to 1 if none are specified – that is, if you do not specify any of the parameters, INSERTs, UPDATEs, and DELETEs will all be collected.

14. In certain cases, with SHADTRS processing enabled and using Execute Immediate processing (which is now deprecated), the TRS could fail inserting into the SHADTRS table. This issue has been fixed.

15. The cached Direct Writer did not properly process the SHAD_MAX_CURSORS parameter. As a result, SHAD_MAX_CURSORS could only be left at its default setting, 50 cursors.

16. The cached Direct Writer did not properly close open cursors when the number of statements exceeded the cache size (50). This resulted in a memory / resource leak which eventually would cause the Direct Writer to shutdown when the number of unique statements being executed exceeded 50.

17. Changed the default value for SHAD_OPCOLLECT_ADJUST_START_SCN from 2 to 10.

18. STDIN and STDERR were not being closed properly when STDOUT is switched.

19. The following parameters were not being properly logged when used:
SHAD_DOC_CLEANUP_POLL
SHAD_DOCS_TO_SAVE
SHAD_DOC_CLEANUP_LEVEL
SHAD_EVENT_POLL
SHAD_FILE_KEEPALIVE
SHAD_EVENT_ORDER_LEVEL

20. A number of issues related to using SQL keywords such as TO_DATE and WHERE in column and table names have been correct. Most cases resulted either the object trapping or invalid SQL statements being generated. Fixes include:

    - Incorrect processing of insert events when the table name ended with "TO_DATE".

    - Incorrect conversion of update statements to insert statements when a column name contained "WHERE".

21. The TRS and TFS trapped without logging an error message when all of the following conditions occurred:

    - The DOC was configured as an Oracle database.
    - The table contained a date / time column.
    - User exit processing was enabled.
    - The target database (used by the TRS / TFS) was configured to be a non-Oracle database.

    The TRS and TFS have been modified to detect this condition and to log error messages indicating that this is an invalid configuration.

22. The TRS and TFS could trap without logging an error message if user exit processing was enabled and one of the column values was null. This issue has been fixed.

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 (V6.220).

ADD / EDIT / EDIT! Prompt Changes

Transaction Processing Prompt (DOC Writer)

```
Transaction Processing ([Y]/N)?
```

This determines whether the transaction DOC is used when writing to cached DOC files. If you want to use the transaction DOC, enter Y. Otherwise, enter N.

Prior to Version 6.230, the transaction DOC was required to allow the TRS or TFS to process the DOC's data. From Version 6.230 and beyond, the TRS and TFS can replay data from cached DOC in committed transaction order without the transaction DOC.

Omitting the transaction DOC provides a significant performance increase when accessing the DOC.

Note that you must still answer 'Y' to the Write Transaction Boundaries to Event File replay data with the TRS or TFS.

Notes:

1) Transaction processing off ('N') is only available to DOC Writers and is not available for source collectors.
2) Transaction processing off ('N') requires that Write Transaction Boundaries to Event File is enabled ('Y')

DOC Generations To Save Prompt (DOC Cleaner)

```
DOC Generations or Hours to Save [<num>]?
```

This determines the number of processed DOC Generations to save (if Transaction Processing is enabled for the associated DOC Writer) or the number of hours of data to save (if Transaction Processing is disabled for the associated DOC Writer). If Transaction Processing is disabled, you should set the number of hours to save to the length of longest expected transaction (rounded up to the nearest hour) plus the number of hours of processed data you want to save.

For example, if the longest expected transaction takes 90 minutes from begin to commit, and you want to keep at least 10 additional hours of data, you would set the value to 12 (2 hours for the transaction plus 10 hours of additional data).

Note that the prompt has been changed to reflect the additional meaning.

FILESTATS Command

This command will show file statistics for a DOC created by a DOC Writer or Open Collector. You must specify the DOC name, the starting sequence number, and the ending sequence number.

FILESTATS provides statistics for every existing DOC file within the specified range.

Example

```
+filestats
Object Name : JHDW1
Starting sequence number [1] : 275
Ending sequence number [999] : 280

Seq                                              Multi   Total    DataFile Ref
Num Begins Commit Aborts Inserts Updates Deletes OPs     Events   Size (K) Seq
--- ------ ------ ------ ------- ------- ------- ------- -------- -------- ---
275   6262   6262      0    6262    6262    6262    6262    18786     3071 N/A
276   6264   6264      0    6264    6264    6264    6264    18792     3071 N/A
277   6262   6262      0    6262    6262    6262    6262    18786     3071 N/A
278   6262   6262      0    6262    6262    6262    6262    18786     3071 N/A
279   6262   6262      0    6262    6262    6262    6262    18786     3071 N/A
280   6262   6262      0    6262    6262    6262    6262    18786     3071 N/A
    ====== ====== ====== ======= ======= ======= ======= ======== ========
Tot  37574  37574      0   37574   37574   37574   37574   112722    18426

++
```

Notes

The Multi Ops column represents the number of records containing multiple events.

The Ref Seq represents the DOC sequence number containing the first event for the oldest in transaction in the DOC. It is only applicable for DOCs created with Transaction Processing enabled.

The statistics are generated by reading the DOC files sequentially. There may be a significant delay the first time the statistics are generated for a DOC, as the entire DOC must be scanned. Once generated, the counts are stored in the DOC and only new records need to be read and processed. Subsequent requests are processed significantly more quickly.

FILESTATS! Command

The FILESTATS! command produces the same report as the FILESTATS command. However, instead of using the stored counts as a basis for the report, FILESTATS! command re-reads the entire DOC and re-counts the records. It can be used if you suspect the record counts from a previous FILESTATS command are corrupted.

FILESTATO Command

The FILESTATO uses the algorithm from previous releases for counting records. This uses a record type indicator in the DOC record to determine the counts for each column. For DOC Blocking Version 2, the record types changed making the counts significantly less useful. The new version of the FILESTATS command will also provide counts of the number insert, update and delete events in the DOC for DOCs that have DOC Blocking enabled.

However, the revised FILESTATS command may take longer to execute than the FILESTATO for large DOCs.

This table below documents the meaning of each of the count columns in the FILESTATO output.

Table 4 - FILESTATO Column Definitions

Column	No DOC Blocking	DOC Blocking v1	DOC Blocking v2
BEGIN	Number of transaction BEGIN events in DOC	Number of transaction BEGIN events in DOC	Number of records in the DOC that contain a BEGIN event, plus zero or more INSERT, UPDATE, DELETE events, but no COMMIT events.
COMMIT	Number of transaction COMMIT events in DOC	Number of transaction COMMIT events in DOC	Number of transaction COMMIT events in DOC. This is determined by counting the number of records that contain a COMMIT event – these records may also include BEGIN INSERT, UPDATE, and DELETE events.
ABORT	Number of transaction ABORT events in DOC		Number of transaction ABORT events in DOC. This is determined by counting the number of records that contain a COMMIT event – these records may also include BEGIN INSERT, UPDATE, and DELETE events.
INSERT		N/A	N/A
UPDATE		N/A	N/A
DELETE		N/A	N/A
MULTI-OPS	N/A	Number of records containing one or more INSERT, UPDATE, and DELETE events.	Number of the records containing one or more INSERT, UPDATE, and DELETE events, but no BEGIN, COMMIT or ABORT events.

Example

In the following example, all transactions in the DOCs fit into a single record. As a result, there are only counts under the COMMIT column. See the example under the FILESTATS for its output for the same DOCs.

```
+filestato
Object Name : JHDW1
Starting sequence number [1] : 275
Ending sequence number [999] : 280

Seq                                              Multi   Total    DataFile Ref
Num Begins Commit Aborts Inserts Updates Deletes OPs     Events   Size (K) Seq
--- ------ ------ ------ ------- ------- ------- ------- -------- -------- ---
275      0   6262      0       0       0       0       0        0     3071 N/A
276      0   6264      0       0       0       0       0        0     3071 N/A
277      0   6262      0       0       0       0       0        0     3071 N/A
278      0   6262      0       0       0       0       0        0     3071 N/A
279      0   6262      0       0       0       0       0        0     3071 N/A
280      0   6262      0       0       0       0       0        0     3071 N/A
    ====== ====== ====== ======= ======= ======= ======= ======== ========
Tot      0  37574      0       0       0       0       0        0    18426

+
```

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 (V6.220 for all servers except HPE NonStop OSS; V6.230 for HPE NonStop OSS).

SHAD_DB2COLLECT_EVENT_QUEUE_NAME

Initial Version:	V6.300	**Syntax:**	SHAD_DB2COLLECT_EVENT_QUEUE_NAME=<name>
Last Change:	--		
Default Value	None		
Other Servers	DB2 Log Based Open Collector		
Valid Settings	The name of the desired IBM MQ Queue.		

Description

SHAD_DB2COLLECT_EVENT_QUEUE_NAME specifies the name of the IBM MQ Event Queue to open and read messages from. The Open Collector opens this queue on the queue manager specified by the related parameter, SHAD_DB2COLLECT_QUEUE_MANAGER_NAME.

Notes

This parameter is required for DB2 Log Based Open Collector objects. It must be specified in the shadparm.ini file prior to startup of a DB2 Log Based Open Collector in order for Open Collection to observe the requested setting. Altering this parameter for an executing DB2 Log Based Open Collector object requires a restart of the Open Collector for those alterations to have an effect.

The queue specified should contain only messages published by an IBM DB2 Q capture server, and no other process may read destructively from it. As a result, only one DB2 Log Based Open Collector may use any given queue. Note that different DB2 Log Based Open Collectors may specify the same SHAD_DB2COLLECT_EVENT_QUEUE_NAME if the values of SHAD_DB2COLLECT_QUEUE_MANAGER_NAME are different.

Related Parameters

SHAD_DB2COLLECT_QUEUE_MANAGER_NAME

SHAD_DB2COLLECT_MQ_COLUMN_DELIMITER

Initial Version:	V6.300	**Syntax:**	SHAD_DB2COLLECT_MQ_COLUMN_DELIMITER=<character>
Last Change:	--		
Default Value	,		
Other Servers	DB2 Log Based Open Collector		
Valid Settings	A single ascii character or a single ASCII character value specified in hexadecimal (e.g. 0x2C) that corresponds to the column delimiter of the IBM DB2 Publishing Queue Map.		

Description

Specifies the column delimiter used for the messages published to the IBM MQ Queue as configured in the Publishing Queue Map for the IBM DB2 Q Capture

Server. This parameter is required if a column delimiter other than the default is being used. For non-printable characters, such as newline, the value is specified in hex.

Notes

This parameter must be specified in the shadparm.ini file prior to startup of a DB2 Log Based Open Collector in order for Open Collection to observe the requested setting. Altering this parameter for an executing DB2 Log Based Open Collector object requires a restart of the Open Collector for those alterations to have an effect.

SHAD_DB2COLLECT_MQ_RECORD_DELIMITER

Initial Version:	V6.300	**Syntax:**	SHAD_DB2COLLECT_MQ_RECORD_DELIMITER=<character>
Last Change:	--		
Default Value	0x0A (newline)		
Other Servers	DB2 Log Based Open Collector		
Valid Settings	A single ascii character or a single ASCII character value specified in hexadecimal (e.g. 0x0A) that corresponds to the column delimiter of the IBM DB2 Publishing Queue Map.		

Description

Specifies the record delimiter used for the messages published to the IBM MQ Queue as configured in the Publishing Queue Map for the IBM DB2 Q Capture Server. This parameter is required if a record delimiter other than the default is being used. For non-printable characters, such as newline, the value is specified in hex.

Notes

This parameter must be specified in the shadparm.ini file prior to startup of a DB2 Log Based Open Collector in order for Open Collection to observe the requested setting. Altering this parameter for an executing DB2 Log Based Open Collector object requires a restart of the Open Collector for those alterations to have an effect.

SHAD_DB2COLLECT_MQ_MESSAGE_ERROR_ACTION

Initial Version:	V6.300	**Syntax:**	SHAD_DB2COLLECT_MQ_MESSAGE_ERROR_ACTION=option
Last Change:	--		
Default Value	STOP		
Other Servers	DB2 Log Based Open Collector		
Valid Settings	One of STOP, SKIP, or SKIPNOLOG		

Description

This parameter specifies the behavior of the DB2 Source Collector (sbdb2col) when it encounters an error processing an MQ message. The options are:

- *STOP* (default behavior): collection will stop with an error message when an error occurs preventing an MQ message from being processed.

- *SKIP*: collection will log an error message and skip the MQ message when an error occurs preventing an MQ message from being processed.
- *SKIPNOLOG*: collection will skip the MQ message without logging an error when an error occurs preventing an MQ message from being processed.

Notes

This parameter must be specified in the shadparm.ini file prior to startup of a DB2 Log Based Open Collector in order for Open Collection to observe the requested setting. Altering this parameter for an executing DB2 Log Based Open Collector object requires a restart of the Open Collector for those alterations to have an effect.

SHAD_DB2COLLECT_QUEUE_MANAGER_NAME

Initial Version:	V6.300	**Syntax:**	SHAD_DB2COLLECT_QUEUE_MANAGER_NAME=<name>
Last Change:	--		
Default Value	None		
Other Servers	DB2 Log Based Open Collector		
Valid Settings	The name of the desired IBM MQ local Queue Manager.		

Description

Specifies the name of the IBM MQ Queue Manager used for Open Collection. The particular queue to use on this queue manager is specified by the related SHAD_DB2COLLECT_EVENT_QUEUE_NAME parameter.

Notes

This parameter is required for DB2 Log Based Open Collector objects. It must be specified in the shadparm.ini file prior to startup of a DB2 Log Based Open Collector in order for Open Collection to observe the requested setting. Altering this parameter for an executing DB2 Log Based Open Collector object requires a restart of the Open Collector for those alterations to have an effect.

Related Parameters

SHAD_DB2COLLECT_EVENT_QUEUE_NAME

SHAD_DB2COLLECT_SQL_SCHEMA_FETCH_ERROR_ACTION

Initial Version:	V6.300	**Syntax:**	SHAD_DB2COLLECT_SQL_SCHEMA_FETCH_ERROR_ACTION =option
Last Change:	--		
Default Value	STOP		
Other Servers	DB2 Log Based Open Collector		
Valid Settings	One of STOP or MQMESSAGEERROR		

Description

This parameter specifies the behavior of the DB2 Log Based Source Collector (sbdb2col) when it encounters an error processing an MQ message. The options are:

- *STOP* (default behavior): collection will stop with an error message when an error occurs while attempting to fetch SQL schema information for the source table.
- *MQMESSAGEERROR*: collection will take action based on the value of the SHAD_DB2COLLECT_MQ_MESSAGE_ERROR_ACTION parameter when an error occurs while attempting to fetch SQL schema information for the source table.

Notes

This parameter must be specified in the shadparm.ini file prior to startup of a DB2 Log Based Open Collector in order for Open Collection to observe the requested setting. Altering this parameter for an executing DB2 Log Based Open Collector object requires a restart of the Open Collector for those alterations to have an effect.

SHAD_DB2COLLECT_STATS_FREQUENCY

Initial Version:	V6.300	**Syntax:**	SHAD_DB2COLLECT_STATS_FREQUENCY=<num>
Last Change:	--		
Default Value	100000		
Other Servers	DB2 Log Based Open Collector		
Valid Settings	1 to 2147483646.		

Description

This parameter sets the number of statements in each DB2 Log Based Open Collector stats collection interval. Stats collection is only performed if SHAD_OPCOLLECT_TRACK_PERFORMANCE is enabled. Logging of the statistics is controlled by the SHAD_OPCOLLECT_LOG_STATS parameter.

Notes

This parameter must be specified in the shadparm.ini file prior to startup of a DB2 Log Based Open Collector in order for Open Collection to observe the requested setting. Altering this parameter for an executing DB2 Log Based Open Collector object requires a restart of the Open Collector for those alterations to have an effect.

Related Parameters

SHAD_OPCOLLECT_TRACK_PERFORMANCE
SHAD_OPCOLLECT_LOG_STATS

SHAD_DOC_APPLY_AUDIT_CONTEXT_TYPE

Initial Version:	V6.300	**Syntax:**	SHAD_DOC_APPLY_AUDIT_CONTEXT_TYPE=<num>
Last Change:	--		
Default Value	1 (NSK)		
Other Servers	Cached TRS		
Valid Settings	0 = ALL 1 = NSK 2 = DB2		

Description

The SHAD_DOC_APPLY_AUDIT_CONTEXT_TYPE parameter specifies a group of Audit Log Context columns to be used for Audit Log processing. The groups enabled the following columns:

0 (ALL) – All of the audit context columns are used.
1 (NSK) – The following audit context columns are used (appropriate to Audit Logging from a NSK source, and matching the columns supporting prior to release 6.300):

- SHAD_BATCH_GROUP
- SHAD_BATCH_ID
- SHAD_EVENT_SEQ_NO
- SHAD_EVENT_TYPE
- SHAD_EVENT_TIMESTAMP
- SHAD_ADT_PREFIX
- SHAD_ADT_SEQ_NUM
- SHAD_ADT_RBA

2 (DB2) – The following audit context columns are used (appropriate to Audit Logging from a DB2 source):

- SHAD_BATCH_GROUP
- SHAD_BATCH_ID
- SHAD_EVENT_SEQ_NO
- SHAD_EVENT_TYPE
- SHAD_EVENT_TIMESTAMP
- SHAD_DB2_MESSAGE_PUT_DATE
- SHAD_DB2_MESSAGE_PUT_TIME
- SHAD_DB2_MESSAGE_PUT_TIMESTAMP
- SHAD_DB2_SOURCE_SCHEMA
- SHAD_DB2_SOURCE_TABLE
- SHAD_DB2_TRANSID
- SHAD_DB2_COMMIT_LSN

The specified set of columns can then be altered by specifying parameters for individual columns. See the <Audit Log Data Columns> section of the shadparm.ini parameters documentation for details on specifying individual columns.

Notes

This parameter must be specified in the shadparm.ini file prior to startup of a Cached TRS in order for audit log processing to observe the requested setting. Altering this parameter for an executing Cached TRS object requires a restart of the Cached TRS for those alterations to have an effect.

Related Parameters

<Audit Log Data Columns>

<Audit Log Data Columns>

Initial Version:	v5.001	**Syntax:**	SHAD_BATCH_GROUP=<new_name>|OMIT
			SHAD_BATCH_ID=<new_name>|OMIT
			SHAD_EVENT_SEQ_NO=<new_name>|OMIT
			SHAD_EVENT_TYPE=<new_name>|OMIT
			SHAD_EVENT_TIMESTAMP=<new_name>|OMIT
			SHAD_ADT_PREFIX=<new_name>|OMIT
			SHAD_ADT_SEQ_NUM=<new_name>|OMIT
			SHAD_ADT_RBA=<new_name>|OMIT
			SHAD_DB2_MESSAGE_PUT_DATE=<new_name>|OMIT
			SHAD_DB2_MESSAGE_PUT_TIME=<new_name>|OMIT
			SHAD_DB2_MESSAGE_PUT_TIMESTAMP=<new_name>|OMIT
			SHAD_SOURCE_SCHEMA=<new_name>|OMIT
			SHAD_SOURCE_TABLE=<new_name>|OMIT
			SHAD_DB2_TRANSID=<new_name>|OMIT
			SHAD_DB2_COMMIT_LSN=<new_name>|OMIT

Last Change:	v6.300		
Default Value	<default column name>		
Other Servers	Cached TRS		
Database Types	Oracle, SQL Server		
Valid Settings	Either OMIT or any other valid DBMS column name to substitute for the default name.		

Description

Each of these parameters specifies the column name to replace the given audit log table’s column name with before constructing and applying an audit log event into the target DBMS.
For example:

```
[ALTRS]
SHAD_EVENT_SEQ_NO=AUDIT_SEQUENCE_NUMBER
```

Assuming ALTRS is writing audit log data into a target table, instead of using the default value of SHAD_EVENT_SEQ_NO as one of its column names, it will replace that column name with AUDIT_SEQUENCE_NUMBER in all of its audit log events applied to the target DBMS.

Alternatively, each of these columns can be omitted from the audit log events by setting the parameter for the desired column name to OMIT. This will drop the unwanted column value out of all audit log statements before applying them into the target DBMS.

Additionally, setting a value for one of these column name overrides the default inclusion or exclusion of that column as determined by the setting of the SHAD_DOC_APPLY_AUDIT_CONTEXT_TYPE parameter. Specifying a value other than OMIT will force inclusion of the column.

These parameters can be set for any combination of all fifteen (15) default audit log column names:

SHAD_BATCH_GROUP
SHAD_BATCH_ID
SHAD_EVENT_SEQ_NO
SHAD_EVENT_TYPE
SHAD_EVENT_TIMESTAMP
SHAD_ADT_PREFIX
SHAD_ADT_SEQ_NUM
SHAD_ADT_RBA
SHAD_DB2_MESSAGE_PUT_DATE
SHAD_DB2_MESSAGE_PUT_TIME
SHAD_DB2_MESSAGE_PUT_TIMESTAMP
SHAD_DB2_TRANSID
SHAD_DB2_COMMIT_LSN
SHAD_SOURCE_SCHEMA
SHAD_SOURCE_TABLE

Related Parameters

SHAD_DOC_APPLY_AUDIT_CONTEXT_TYPE

SHAD_OPCOLLECT_ADJUST_START_SCN

Initial Version:	v6.110G	**Syntax:**	SHAD_OPCOLLECT_ADJUST_END_SCN=<num>
Last Change:	V6.300		
Default Value	10		
Other Servers	OPLOG		
Valid Settings	0 - 2147483647		
Basic/Advanced	Basic		

Description

This parameter provides a work around to an issue in Oracle Log Mining by adjusting the start position of each collection cycle. Testing has indicated that Oracle will not reliably return records at the beginning of the collection range. This is issue is infrequent, but it can cause Oracle to return syntactically correct data, but with some of the actual column values incorrectly set to NULL. It can also cause records in the beginning of the collection range to be missed, or to have partial records return with no usable information.

The value specified by this parameter is used to start the collection cycle at an earlier point by subtracting the value from the actual start SCN to determine the start of collection. Events up to and including the one last collected in the previous cycle are skipped, thus ensuring that the data that is needed for the collection cycle is correct.

The default value for this parameter is 10.

SHAD_OPCOLLECT_ALLOW_SYS_USER

Initial Version:	V6.300	**Syntax:**	SHAD_OPCOLLECT_ALLOW_SYS_USER = <number>
Last Change:	--		
Default Value	1 (ON)		
Other Servers	Oracle Log Source Collector		
Valid Settings	0 =OFF (Events generated by the Oracle SYS users are excluded from collection) 1 =ON (Events generated by the Oracle SYS users are included in replication)		

Description

This parameter specifies whether insert, update, and delete events generated by the Oracle SYS user are included (SHAD_OPCOLLECT_ALLOW_SYS_USER=1) or excluded (SHAD_OPCOLLECT_ALLOW_SYS_USER=0).

Note that due to the way log mining works, setting this parameter to OFF will not necessarily exclude all SYS user events. The user field in the V$LOGMNR view is only populated if the initial connection to the database occurred within the collection period – otherwise, it is NULL and the event will be collected. Gravic strongly recommends in using the default setting for this parameter and collecting SYS user events.

SHAD_OPCOLLECT_EXCLUDE_TABLES_NAME

Initial Version:	V6.3000	**Syntax:**	SHAD_OPCOLLECT_INCLUDE_TABLES_NAME=<name>
Last Change:	--		
Default Value	SHAD_TABLES_INCLUDE		
Other Servers	OPCOL (SBORLOG)		
Valid Settings	<table_name> - Non-container databases <table_name>@<link_name> - Oracle Container databases		

Description

This parameter specifies the name of the table containing the list of table names that will be ***excluded*** from replication. By default, the HPE Shadowbase for Oracle Log Source collector will use the table SHAD_TABLES_EXCLUDE when the SHAD_OPCOLLECT_USE_EXCLUDE_TABLES parameter is set. This parameter overrides the default name.

Notes

- If you specify the SHAD_OPCOLLECT_EXCLUDE_TABLE_NAME, you do not also have to specify the SHAD_OPCOLLECT_USE_EXCLUDE_TABLES parameter. Specifying the name implies that it should be used by default.

- If the table is not in the database that the collector connects to, you must also include a link to the database in the name. The format of the name should be <tablename>@<linkname>, where <tablename> is the name of the table and <linkname> is the name of a link to the database containing the table. The most common usage for this form is for replicating from an Oracle Container Database. In this configuration, the collector connects to the root database which cannot contain any user tables. The configuration tables are typically contained in one of the Pluggable Databases in the container.

Related Parameters

SHAD_OPCOLLECT_USE_EXCLUDE_TABLES
SHAD_OPCOLLECT_PDB_LINK_NAME

SHAD_OPCOLLECT_EXCLUDE_USERS_NAME

Initial Version:	V6.3000	**Syntax:**	SHAD_OPCOLLECT_INCLUDE_USERS_NAME=<name>
Last Change:	--		
Default Value	SHAD_USERS_INCLUDE		
Other Servers	OPCOL (SBORLOG)		
Valid Settings	<table_name> - Non-container databases <table_name>@<link_name> - Oracle Container databases		

Description

This parameter specifies the name of the table containing the list of the list of users whose tables will be ***excluded*** from replication. By default, the HPE

Shadowbase for Oracle Log Source collector will use the table SHAD_USERS_EXCLUDE when the SHAD_OPCOLLECT_USE_EXCLUDE_USERS parameter is set. This parameter overrides the default name.

Notes

- If you specify the SHAD_OPCOLLECT_EXCLUDE_USERS name, you do not also have to specify the SHAD_OPCOLLECT_USE_EXCLUDE_USERS parameter. Specifying the name implies that it should be used by default.

- If the table is not in the database that the collector connects to, you must also include a link to the database in the name. The format of the name should be <tablename>@<linkname>, where <tablename> is the name of the table and <linkname> is the name of a link to the database containing the table. The most common usage for this form is for replicating from an Oracle Container Database. In this configuration, the collector connects to the root database which cannot contain any user tables. The configuration tables are typically contained in one of the Pluggable Databases in the container.

Related Parameters

SHAD_OPCOLLECT_USE_EXCLUDE_USERS
SHAD_OPCOLLECT_PDB_LINK_NAME

SHAD_OPCOLLECT_PDB_CREATE_LINK

Initial Version:	V6.3000	**Syntax:**	SHAD_OPCOLLECT_PDB_CREATE_LINK = value
Last Change:	--		
Default Value	1 (create link)		
Other Servers	OPCOL (SBORLOG)		
Valid Settings	0 (don’t create link) 1 (create link)		

Description

This parameter specifies if the Shadowbase for Oracle Log collector will recreate the link from the root database to the Oracle Pluggable Database (PDB) containing the configuration files used by the Shadowbase for Oracle Log collector to select the files and users included in the collection (SHAD_USERS_INCLUDE, SHAD_USERS_EXCLUDE, SHAD_TABLES_INCLUDE, SHAD_TABLES_EXCLUDE).

This parameter only applies to replication from container databases and is optional. If 1, or not specified, the collector will recreate the link to the database specified by the SHAD_OPCOLLECT_PDB_NAME at startup by dropping if it exists and then creating it. If 0, the link must exist prior to starting. You can specify the link name using the SHAD_OPCOLLECT_PDB_LINK_NAME parameter.

Notes

- The collector can only create a link to the database specified by SHAD_OPCOLLECT_PDM_NAME. If the configuration tables are stored in a different database, you must pre-create link and set this parameter to 0.

Related Parameters

SHAD_OPCOLLECT_PDB_NAME
SHAD_OPCOLLECT_PDB_LINK_NAME

SHAD_OPCOLLECT_INCLUDE_TABLES_NAME

Initial Version:	V6.3000	**Syntax:**	SHAD_OPCOLLECT_INCLUDE_TABLES_NAME=<name>
Last Change:	--		
Default Value	SHAD_TABLES_INCLUDE		
Other Servers	OPCOL (SBORLOG)		
Valid Settings	<table_name> - Non-container databases <table_name>@<link_name> - Oracle Container databases		

Description

This parameter specifies the name of the table containing the list of table names that will be ***included*** in replication. By default, the HPE Shadowbase for Oracle Log Source collector will use the table SHAD_TABLES_INCLUDE when the SHAD_OPCOLLECT_USE_INCLUDE_TABLES parameter is set. This parameter overrides the default name.

Notes

- If you specify the SHAD_OPCOLLECT_INCLUDE_TABLES_NAME parameter, you do not also have to specify the SHAD_OPCOLLECT_USE_INCLUDE_TABLES parameter. Specifying the name implies that it should be used by default.

- If the table is not in the database that the collector connects to, you must also include a link to the database in the name. The format of the name should be <tablename>@<linkname>, where <tablename> is the name of the table and <linkname> is the name of a link to the database containing the table. The most common usage for this form is for replicating from an Oracle Container Database. In this configuration, the collector connects to the root database which cannot contain any user tables. The configuration tables are typically contained in one of the Pluggable Databases in the container.

Related Parameters

SHAD_OPCOLLECT_USE_INCLUDE_TABLES

SHAD_OPCOLLECT_PDB_LINK_NAME

SHAD_OPCOLLECT_INCLUDE_USERS_NAME

Initial Version:	V6.3000	**Syntax:**	SHAD_OPCOLLECT_INCLUDE_USERS_NAME=<name>
Last Change:	--		
Default Value	SHAD_USERS_INCLUDE		
Other Servers	OPCOL (SBORLOG)		
Valid Settings	<table_name> - Non-container databases <table_name>@<link_name> - Oracle Container databases		

Description

This parameter specifies the name of the table containing the list of users whose tables will be ***included*** in replication. By default, the HPE Shadowbase for Oracle Log Source collector will use the table SHAD_USERS_INCLUDE when the SHAD_OPCOLLECT_USE_INCLUDE_USERS parameter is set. This parameter overrides the default name.

Notes

- If you specify the SHAD_OPCOLLECT_INCLUDE_USERS name, you do not also have to specify the SHAD_OPCOLLECT_USE_INCLUDE_USERS parameter. Specifying the name implies that it should be used by default.

- If the table is not in the database that the collector connects to, you must also include a link to the database in the name. The format of the name should be <tablename>@<linkname>, where <tablename> is the name of the table and <linkname> is the name of a link to the database containing the table. The most common usage for this form is for replicating from an Oracle Container Database. In this configuration, the collector connects to the root database which cannot contain any user tables. The configuration tables are typically contained in one of the Pluggable Databases in the container.

Related Parameters

SHAD_OPCOLLECT_USE_INCLUDE_USERS
SHAD_OPCOLLECT_PDB_LINK_NAME

SHAD_OPCOLLECT_PDB_LINK_NAME

Initial Version:	V6.3000	**Syntax:**	SHAD_OPCOLLECT_PDB_LINK_NAME= <name>
Last Change:	--		
Default Value	None		
Other Servers	OPCOL (SBORLOG)		
Valid Settings	PDB Name		

Description

This parameter specifies the name of the database link from the root database to an Oracle Pluggable Database (PDB) that contains the configuration files used by the Shadowbase for Oracle Log collector to select the files and users included in the collection (SHAD_USERS_INCLUDE, SHAD_USERS_EXCLUDE, SHAD_TABLES_INCLUDE, SHAD_TABLES_EXCLUDE).

This parameter only applies to replication from container databases and is optional. If not specified, the collector will use the link name “SBLINK_TO_<pdb_name>, where <pdb_name> is the name specified in the SHAD_OPCOLLECT_PDM_NAME parameter.

Related Parameters

SHAD_OPCOLLECT_PDB_NAME
SHAD_OPCOLLECT_PDB_CREATE_LINK

SHAD_OPCOLLECT_PDB_NAME

Initial Version:	V6.3000	**Syntax:**	SHAD_OPCOLLECT_PDB_NAME = <name>
Last Change:	--		
Default Value	None		
Other Servers	OPCOL (SBORLOG)		
Valid Settings	PDB Name		

Description

This parameter specifies the name of the Oracle Pluggable Database (PDB) that is being replicated by the Shadowbase for Oracle Log Collection Object. For example, if the Oracle Container Database (CDB) is called ORA12C and it contains two databases, SALES and MARKETING, you would specify SHAD_OPCOLLECT_PDB_NAME=SALES to replicate from the SALES database.

Related Parameters

SHAD_OPCOLLECT_PDB_LINK_NAME
SHAD_OPCOLLECT_PDB_CREATE_LINK

SHAD_LEGACY_QUOTE_PROCESSING

Initial Version:	V6.220	**Syntax:**	SHAD_LEGACY_QUOTE_PROCESSING = <number>
Last Change:	--		
Default Value	0 (OFF)		
Other Servers	TRS,TFS		
Valid Settings	0 =OFF (Legacy mode disabled, leading and trailing quotes are not stripped) 1 =ON (Legacy mode enabled, leading and trailing quotes are stripped)		

Description

This parameter specifies whether leading and trailing quotes are stripped off when SQL statements are parsed. When disabled (the default), leading and trailing quotes are not stripped from the data. When enabled, the leading and trailing quotes are stripped from the column data when the SQL is parsed.

Legacy quote processing needs to be disabled in most cached DOC configurations, such as when using the Collector for Oracle Log Source. Otherwise, columns whose data has leading and trailing quotes will have them incorrectly stripped off. However, execute immediate environments with batch processing enabled need to use the legacy mode, otherwise, the single quotes will be doubled.

SHAD_OPCOLLECT_QUOTED_STRING_VALUES

Initial Version:	V6.300	**Syntax:**	SHAD_OPCOLLECT_QUOTED_STRING_VALUES=<num>
Last Change:	--		
Default Value	0 (OFF)		
Other Servers	DB2 Log Based Open Collector		
Valid Settings	0 =OFF (Surrounding quotes and embedded escaping quotes are removed before writing to DOC) 1 =ON (Surrounding quotes and embedded escaping quotes are retained in values written to DOC)		

Description

This parameter specifies whether delimited string values collected from the MQ Queues are written to the DOC with quoted string values stripped or maintained.

When set to 0 (off), the default, the surrounding string delimiters and any escaping embedded quotes are removed. When set to 1 (on), the surrounding string delimiters and escaping embedded quotes collected from the MQ Queues and preserved in the values written into the DOC files.

SHAD_OPCOLLECT_THROTTLE_DELAY

Initial Version:	V6.300	**Syntax:**	SHAD_OPCOLLECT_THROTTLE_DELAY=<num>
Last Change:	--		
Default Value	0		
Other Servers	DB2 Log Based Open Collector		
Valid Settings	0 = Disabled 1 to 2147483646 milliseconds.		

Description

The SHAD_OPCOLLECT_THROTTLE_DELAY parameter specifies the amount of time in milliseconds for the OPCOL to delay for throttling purposes every SHAD_OPCOLLECT_THROTTLE_MESSAGE_LIMIT messages are processed. When set to 0, throttling is disabled.

Notes

This parameter must be specified in the shadparm.ini file prior to startup of a Log Based Open Collector in order for Open Collection to observe the requested setting. Altering this parameter for an executing Log Based Open Collector object requires a restart of the Open Collector for those alterations to have an effect.

Related Parameters

SHAD_OPCOLLECT_THROTTLE_MESSAGE_LIMIT

SHAD_OPCOLLECT_THROTTLE_MESSAGE_LIMIT

Initial Version:	V6.300	**Syntax:**	SHAD_OPCOLLECT_THROTTLE_MESSAGE_LIMIT=<num>
Last Change:	--		
Default Value	1		
Other Servers	DB2 Log Based Open Collector		
Valid Settings	1 to 2147483646 milliseconds.		

Description

The SHAD_OPCOLLECT_THROTTLE_MESSAGE_LIMIT parameter specifies the number of messages to process each interval before delaying when throttling is enabled. SHAD_OPCOLLECT_THROTTLE_DELAY is used in conjunction to enable or disable throttling and set the throttling delay.

Notes

This parameter must be specified in the shadparm.ini file prior to startup of a Log Based Open Collector in order for Open Collection to observe the requested setting. Altering this parameter for an executing Log Based Open Collector object requires a restart of the Open Collector for those alterations to have an effect.

Related Parameters

SHAD_OPCOLLECT_THROTTLE_DELAY

SHAD_PROCESS_PRECHECK

Initial Version:	V6.300	**Syntax:**	SHAD_PROCESS_PRECHECK=<num>
Last Change:	--		
Default Value	1 (ON)		
Other Servers	SBMON		
Valid Settings	0 = OFF (SBMON does not query the OS for matching running processes) 1 = ON (SBMON queries the OS for matching running processes)		

Description

The SHAD_PROCESS_PRECHECK parameter allows SBMON to query the operating system to test whether a matching process exists for a given Shadowbase object. If the process does not exist, SBMON can immediately rule out that the object is running without waiting for a timeout, resulting in a significant speedup for the STATUS command and all other commands that use it.

Notes

This parameter is not currently supported on Windows platforms.

SHAD_RETRY_CTREE_ERROR_527

Initial Version:	V6.300	**Syntax:**	SHAD_RETRY_CTREE_ERROR_527=<num>
Last Change:	--		
Default Value	0 (OFF)		
Other Servers	TRS, TFS		
Valid Settings	0 =OFF (cTree error 527 are not retried while reading the STMT DOC) 1 =ON (cTree error 527s are retried while reading the STMT DOC)		

Description

This parameter specifies whether the c-Tree error 527 is retried while reading from the STMT DOC. By default, it is not retried and the TRS or TFS will stop. We have found that in most cases, these errors are transient. Setting this parameter to on allows the TRS or TFS to recover faster than the normal method of stopping and restarting.

These errors seem more prevalent on systems with busy disks and network based file systems. Systems that have the DOC data on local drives do not encounter these errors as frequently.

When set to 0 (off), the default, the error is not retried. When set to 1 (on), the error is retried.

New User Exit Exception Entry Points

The USRXREJECT user exit function is entered by the HP Shadowbase for Other Servers OPCOL process upon encountering an MQ message that could not be processed.

USRXREJECT

USRXREJECT is called when a message or event could not be processed. This user exit entry point is supported for DB2 Source OPCOL objects for IBM MQ messages that could not be parsed for successful processing. When corrupt or rejected messages are encountered, USRXREJECT processing can be used along with the SBGETREJECT user exit API function to retrieve the rejected MQ message and perform any desired logging or other processing. In addition, the USRXREJECT entry point can be used along with the SHAD_DB2COLLECT_MQ_MESSAGE_ERROR_ACTION shadparm.ini parameter to control if the OPCOL object stops or skips the rejected message and continues processing.

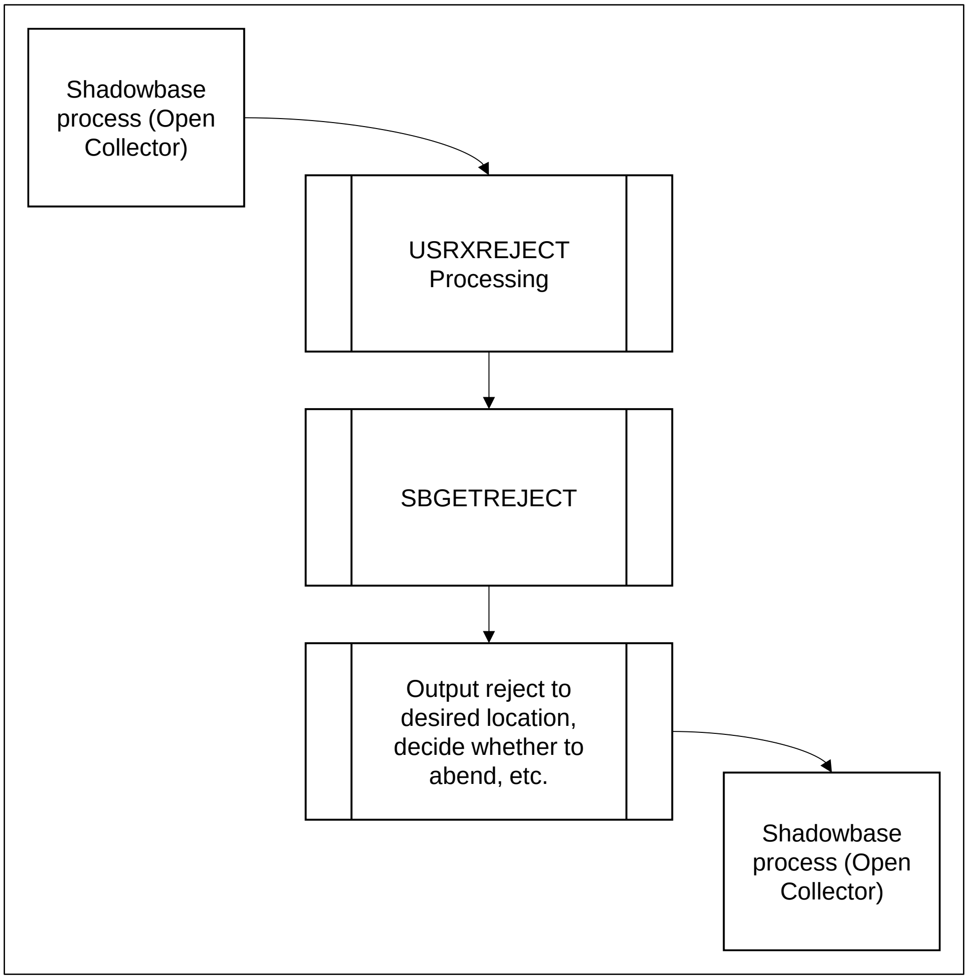
Shadowbase process (Open Collector)
USRXREJECT Processing
SBGETREJECT
Output reject to desired location, decide whether to abend, etc.
Shadowbase process (Open Collector)

Figure 2 - USRXREJECT Processing

USRXREJECT is entered by a HPE Shadowbase process (Open Collector) whenever a reject message that can not be processed is encountered. Currently, this means an IBM MQ message that could not be processed. Then, SBGETREJECT may be called to retrieve the corrupt or rejected message. The message may then be logged or otherwise processed by the user exit as desired (such as writing the rejected messages to a reject file). After USRXREJECT processing, control is returned to the HP Shadowbase Open Collector process which takes action based on the setting of the SHAD_DB2COLLECT_MQ_MESSAGE_ERROR_ACTION shadparm.ini parameter.

New User Exit API Functions

SBGETAUDITCONTEXT

This API is used in USRXPROCESS to retrieve the value for a specified audit context column via its column id defined in the AUDIT CONTEXT COLUMN values section of usrxlib.h.

Product	Availability	Calling Module Usage
HP NonStop Server	No	N/A
Other Servers	Yes	USRXPROCESS

Syntax for C Programmers

```
short SBGETAUDITCONTEXT(unsigned short column,
                        char *col_name,
                        short *name_len,
                        unsigned char *col_value,
                        short *value_len);
```

Parameters

return_code output

- indicates whether call completed successfully or encountered an error:
  -1 = Unknown column id specified
  0 = Audit context column information returned successfully

column input

- ID value of the audit context column to be retrieved. Accepted values defined in usrxlib.h.

col_name output

- returns the configured name for the audit context column

name_len output

- returns the length of the audit column name

col_value output

- returns the value of the requested audit context column

value_len output

- returns the length of the value returned for requested audit context column

Considerations

- SBGETAUDITCONTEXT is only available in the Other Servers environment.
- SBGETAUDITCONTEXT supports insert, update and delete events (begin, commit, abort events are not supported).
- The column name and value that are returned are not null terminated. The appropriate lengths are returned in the respective variables.
- Make sure that the *col_name* and *col_value* arguments are defined large enough to hold the maximum value that could be returned.
- If you attempt to retrieve a column value for a column id that is out of range, *return_code* will be set to -1 (unknown), and no other data will be returned.

C Example

```
short return_code = 0;
short col_id = 0;
short name_len = 0;
short val_len = 0;
char col_name[MAX_COLUMN_NAME];
char val[512];

/* assign desired audit context column identifier*/
col_id = ADT_CONTEXT_BATCH_GROUP;

return_code = SBGETAUDITCONTEXT (col_id,col_name,&name_len,val,&val_len);
```

SBGETREJECT

This API is used in USRXREJECT to retrieve the message or statement for the rejected event.

Product	Availability	Calling Module Usage
HP NonStop Server	No	N/A
Other Servers	Yes	USRXREJECT

Syntax for C Programmers

```
short SBGETREJECT(const char **reject,
                  unsigned short *length,
                  short *reject_type);
```

Parameters

return_code output

- indicates whether call completed successfully or encountered an error:
  -2 = invalid input detected (such as a NULL pointer)
  -1 = reject information not available
  0 = value returned

reject output

- Returns a pointer to the stored reject message value.

length output

- Returns the length in bytes of the reject message value.

reject_type output

- returns the reject message type. Reject message types are defined in usrxlib.h. Currently, only MQ reject messages are returned. These are

Considerations

- SBGETREJECT is only available in the Other Servers environment for retrieving MQ messages which could not be parsed.

C Example

```
short           return_code = 0;
short           reject_type = 0;
const char     *record      = NULL;
unsigned short  record_len  = 0;

return_code = SBGETREJECT (&record, &record_len, &);
```

SBSETCOMMITBATCH

This function is used from USRXPROCESS or USRXAFTERPROCESS in a TRS object to commit the current batch or transaction and to optionally stop the TRS after committing.

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

Syntax for C Programmers

```
short SBSETCOMMITBATCH(unsigned int options)
```

Parameters

RETURN-CODE returned value

- Indicates whether call completed successfully or encountered an error. Possible values:

-1	=	invalid option specified
0	=	successful

options input

A bit mask that specifying whether the current transaction should commit early and whether or not the TRS should stop after committing. See , below for the available option combinations.

Considerations

- SBSETCOMMITBATCH is only available in the Other Servers environment and only applies to the TRS.
- C defines for each option are defined in the usrxlib.h header file. It has the following defines:

COMMIT_BATCH_NOW	0x0002
COMMIT_BATCH_NEXT_COMMIT	0x0004
COMMIT_BATCH_STOP_AFTER_COMMIT	0x0001

Option	Value	Description
-	0	Normal processing will occur. 0 can be used to reset previously set options.
COMMIT_BATCH_STOP_AFTER_COMMIT	1	The TRS will stop after the current transaction in progress is committed. If transaction batching is enabled, it will stop after the batch is committed.
COMMIT_BATCH_NOW	2	If the event is an IUD event, the TRS will commit the current transaction immediately after the current event. The restart point will not be updated. If the source event is a COMMIT event, the transaction will be committed and the restart point will be updated.
COMMIT_BATCH_NOW + COMMIT_BATCH_STOP_AFTER_COMMIT	3	If the event is an IUD event, the TRS will commit the current transaction immediately after the current event and will then stop. The restart point will not be updated, so any events up to and including the current event will be reprocessed when the TRS restarts. If the source event is a COMMIT event, the transaction wlll commit, the restart point will be updated, and the TRS will stop.
COMMIT_BATCH_NEXT_COMMIT	4	The TRS will issue a commit when the next source COMMIT event is read. If the current event is a COMMIT, the batch will commit immediately. The restart point is updated.
COMMIT_BATCH_NEXT_COMMIT+ COMMIT_BATCH_STOP_AFTER_COMMIT	5	The TRS will issue a commit when the next source COMMIT event is read. If the current event is a COMMIT, the batch will commit immediately. The restart point is updated. After the batch is committed and the restart point updated, the TRS will stop.

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 (V6.220 for all servers except HPE NonStop OSS; V6.230 for HPE NonStop OSS).

Other Servers Process-Generated Messages

```
CONFIGURATION FAULT: Open Collector [<name>] configuration is invalid!
```

Cause: The Open Collector process detected an invalid configuration. Check for following messages with details on the reason for the invalid configuration.

Effect: The process logs any additional details and terminates.

Recovery: Correct the configuration based on details provided by the following message(s) and restart.

```
CONFIGURATION ERROR:
 SHAD_DB2COLLECT_QUEUE_MANAGER_NAME parameter is NOT SET
 and is REQUIRED
```

Cause: No value was specified for the SHAD_DB2COLLECT_QUEUE_MANAGER_NAME parameter. This parameter is required for DB2 Log Based Source Collection.

Effect: The process performs shutdown and terminates.

Recovery: Specify the name of the desired Queue Manager via the SHAD_DB2COLLECT_QUEUE_MANAGER_NAME parameter and restart.

```
CONFIGURATION ERROR: SHAD_DB2COLLECT_EVENT_QUEUE_NAME
 parameter is NOT SET and is REQUIRED
```

Cause: No value was specified for the SHAD_DB2COLLECT_EVENT_QUEUE_NAME parameter. This parameter is required for DB2 Log Based Source Collection.

Effect: The process performs shutdown and terminates.

Recovery: Specify the name of the desired IBM MQ Queue via the SHAD_DB2COLLECT_EVENT_QUEUE_NAME parameter and restart.

CONFIGURATION ERROR: Process performing shutdown

Cause: The Open Collector detected a configuration error prohibiting it from continuing. Check previous messages for details on the nature of the configuration error.

Effect: The process performs shutdown and terminates.

Recovery: Correct the configuration error based on details of the previous message(s) and restart.

SHAD_EVENT_POLL set to <number>

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

Effect: The process reports the condition, and continues.

Recovery: None, unless parameter setting unintended. A restart will be required if this is the case.

SHAD_FILE_KEEPALIVE set to <number>

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

Effect: The process reports the condition, and continues.

Recovery: None, unless parameter setting unintended. A restart will be required if this is the case.

SHAD_EVENT_ORDER_LEVEL set to <number>

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

Effect: The process reports the condition, and continues.

Recovery: None, unless parameter setting unintended. A restart will be required if this is the case.

SHAD_DOCS_TO_SAVE set to <number>

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

Effect: The process reports the condition, and continues.

Recovery: None, unless parameter setting unintended. A restart will be required if this is the case.

SHAD_DOC_CLEANUP_LEVEL set to <number>

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

Effect: The process reports the condition, and continues.

Recovery: None, unless parameter setting unintended. A restart will be required if this is the case.

SHAD_DOC_CLEANUP_POLL set to <number>

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

Effect: The process reports the condition, and continues.

Recovery: None, unless parameter setting unintended. A restart will be required if this is the case.

```
You have selected a non-cached (execute immediate) object which is currently
deprecated and may be removed in a future release.

Consider using <cached alternatives> instead.

Are you sure you want to choose this executable? (y/N) : <your confirmation>
```

Cause: A non-cached Shadowbase object was specified during object configuration in SBMON.

Effect: Configuration continues pending user confirmation.

Recovery: Consider using one of the listed cached alternatives and confirm your selection.

CLI/ODBC Messages

```
SQLDriverConnect(): fault detected; rc:= <reason code>
Using connection string[<string>]
DB2 Configuration Fault
Unable to continue; performing shutdown
```

Cause: The CLI/ODBC SQLDriverConnect function experienced a fault. See the reported reason code for the cause of the fault. Any additional available diagnostic messages will also be output.

Effect: The process terminates.

Recovery: Double check that the database, username, and password are correct. Correct any issues with the connection configuration and restart.

```
GetTableSchema(): SQLAllocHandle() fault detected; rc:= <return code>
Table being processed:= <table name>; For operation:= <function name>
```

Cause: The CLI/ODBC SQLAllocHandle function experienced a fault. See the reported reason code for the cause of the fault. Any additional available diagnostic messages will also be output.

Effect: The message cannot be processed. The process either skips the message and attempts to continue or terminates, depending on the configuration.

Recovery: Take appropriate action based on the reported return code and any logged diagnostic messages. Then, restart the process.

```
GetTableSchema(): SQLColumns() fault detected; rc:= <return code>
Table being processed:= <table name>
```

Cause: The CLI/ODBC SQLColumn function experienced a fault. See the reported reason code for the cause of the fault. Any additional available diagnostic messages will also be output.

Effect: The message cannot be processed. The process either skips the message and attempts to continue or terminates, depending on the configuration.

Recovery: Take appropriate action based on the reported return code and any logged diagnostic messages. Then, restart the process.

```
GetTableSchema(): SQLBindCol() for <column to be bound> fault detected; rc:=
 <return code>
Table being processed:= <table name>; For operation:= <function name>
```

Cause: The CLI/ODBC SQLBindCol function experienced a fault when attempting to bind the specified column of the results from the noted function. See the reported reason code for the cause of the fault. Any additional available diagnostic messages will also be output.

Effect: The message cannot be processed. The process either skips the message and attempts to continue or terminates, depending on the configuration.

Recovery: Take appropriate action based on the reported return code and any logged diagnostic messages. Then, restart the process.

```
GetTableSchema(): SQLFetch() fault detected; rc:= <return code>
Table being processed:= <table name>; For operation:= <function name>
```

Cause: The CLI/ODBC SQLFetch function experienced a fault when attempting to fetch results from the specified operation. See the reported reason code for the cause of the fault. Any additional available diagnostic messages will also be output.

Effect: The message cannot be processed. The process either skips the message and attempts to continue or terminates, depending on the configuration.

Recovery: Take appropriate action based on the reported return code and any logged diagnostic messages. Then, restart the process.

```
GetTableSchema(): SQLPrimaryKeys() fault detected; rc:= <return code>
Table being processed:= <table name>
```

Cause: The CLI/ODBC SQLPrimaryKeys function experienced a fault. See the reported reason code for the cause of the fault. Any additional available diagnostic messages will also be output.

Effect: The message cannot be processed. The process either skips the message and attempts to continue or terminates, depending on the configuration.

Recovery: Take appropriate action based on the reported return code and any logged diagnostic messages. Then, restart the process.

```
GetTableSchema(): SQLEndTran() fault detected; rc:= <return code>
```

Cause: The CLI/ODBC SQLEndTran function experienced a fault. See the reported reason code for the cause of the fault. Any additional available diagnostic messages will also be output.

Effect: The message cannot be processed. The process either skips the message and attempts to continue or terminates, depending on the configuration.

Recovery: Take appropriate action based on the reported return code and any logged diagnostic messages. Then, restart the process.

```
Drec<num>, SQLSTATE: <sql state>, SQLCODE: <sql code>, Msg: <message
 text>
```

Cause: A CLI/ODBC error condition has occurred and diagnostic information is available.

Effect: The diagnostic information is logged. Check previous/following messages for further information on the overall effect of the error condition.

Recovery: Take appropriate action based on the reported SQLSTATE, SQLCODE, and message text.

MQI Messages

```
MQGET[Destruct] expected message with message identifier [number] but no
 message found.
Message may have been removed by unexpected external source.
```

Cause: No messages remained in the MQ Queue when performing a destructive read after having previously read a message for browse access. Another process may be destructively reading from the queue.

Effect: The process reports the condition and terminates.

Recovery: Ensure only the DB2 Log Based Collector is reading destructively from the configured queue. To recover the missing data, restart the Q capture server at the reported position.

MQGET[Destruct] read message with message identifier [<identifier>]; expected [<expected identifier>]
Message may have been inserted or removed by unexpected external source.

Cause: When performing a destructive read after having previously read a message for browse access a message with a different message identifier was found. Another process may be destructively reading from the queue.

Effect: The process reports the condition and terminates.

Recovery: Ensure only the DB2 Log Based Collector is reading destructively from the configured queue and only the Q Capture server is putting messages to the queue in FIFO order. To recover the missing data, restart the Q Capture server at the reported position.

MQGET[Destruct] ended with reason code [<reason code>]; comp code [<completion code>]

Cause: A call to MQGET for destructive reading failed with the noted reason and completion codes

Effect: The process terminates.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

MQI MQCONN[Browse] failure; Reason Code [<reason code>];
QueueManager[<queue manager name>]

Cause: A call to MQCONN for the browse reading connection failed with the noted reason code

Effect: The process terminates.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

MQI MQOPEN[Browse] unable to open queue [<queue name>] in
QueueManager[<queue manager name>] for input,reason code[<reason code>]

Cause: A call to MQOPEN failed to open the specified queue for browse access with the noted reason code.

Effect: The process terminates.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

MQI MQOPEN[Browse] ended with reason code [<reason code>];
QueueManager[<queue manager name>] connection not established

Cause: A call to MQOPEN failed to open the specified queue for browse access with the noted reason code.

Effect: The process terminates.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

```
MQI MQCONN[Destroy] failure; Reason Code [<reason code>];
 QueueManager[<queue manager name>]
```

Cause: A call to MQCONN failed to connect to the specified queue manager for destructive access with the noted reason code.

Effect: The process terminates.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

```
MQI MQOPEN[Destroy] unable to open queue [<queue name>] in
 QueueManager[<queue manager name>] for input,reason code[<reason code>]
```

Cause: A call to MQOPEN failed to open the specified queue for destructive access with the noted reason code.

Effect: The process terminates.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

```
MQI MQOPEN[Destroy] ended with reason code [<reason code>];
 QueueManager[<queue manager name>] connection not established
```

Cause: A call to MQOPEN failed to open the specified queue for destructive access with the noted reason code.

Effect: The process terminates.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

MQGET[Browse] fault detected; Condition Code [<condition code>], Reason Code [<reason code>]

Cause: A call to MQGET for browse access failed with the noted condition and reason codes.

Effect: The process terminates.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

MQCLOSE[Browse] fault detected; Reason Code [<reason code>]

Cause: A call to MQCLOSE to close the connection for destructive reading failed with the noted reason code.

Effect: The process continues performing shutdown.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

MQCLOSE[Destruct] fault detected; Reason Code [<reason code>]

Cause: A call to MQCLOSE to close the browse connection failed with the noted reason code.

Effect: The process continues performing shutdown.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

MQDISC[Browse] fault detected; Reason Code <reason code>]

Cause: A call to MQDISC to disconnect the browse connection failed with the noted reason code.

Effect: The process continues performing shutdown.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

```
MQDISC[Destruct] fault detected; Reason Code [<reason code>]
```

Cause: A call to MQDISC to disconnect the destructive reading connection failed with the noted reason code.

Effect: The process continues performing shutdown.

Recovery: See the following message for a URL to lookup detailed information on the IBM MQI reason code and take appropriate action based on this.

```
For a detailed explanation of Reason Code [<reason code>], use the following
 URL[https://www.google.com/#q=%22<reason code search text>]
```

Cause: A MQI function returned a reason code other than one expected during normal operation. See the previous message for the MQI function in error.

Effect: The attempted action from the previous message failed and the process terminates.

Recovery: For additional details and recovery actions, paste the URL string into a browser to search for the pertinent IBM MQ documentation of the reason code.

Low-Level Shadowbase Messages

```
DB2SourceDLL::PrepDletIsrtStmts(<exception>):Failed to acquire 65536 bytes of
 dynamic memory.
```

Unable to continue; Performing Shutdown

Cause: An exception occurred while attempting to dynamically allocate memory.

Effect: The process terminates.

Recovery: Restart the process and check available system memory. If problem persists, contact Shadowbase Support.

Configured to stop on MQ message error; Performing Shutdown

Cause: The DB2 Log Based Source Collector was unable to parse a message read from the configured MQ Queue. See previous messages for details and message contents.

Effect: The process terminates.

Recovery: Check previous messages for further details on the cause of the problem. Resolve the problem and restart.

Error parsing delimited message header: Record only had <num> columns; 12 expected for message header

Cause: Fewer columns were found in the MQ Message record than the minimum expected number for message header information.

Effect: The message record is not processed. Depending on parameter settings, the process either terminates or skips the record and attempts to continue.

Recovery: Verify that only a Q Capture Server is feeding the configured MQ Queue using delimited messages with the correct delimiters. Correct any such issues found, remove any bad messages from the queue, and restart the Q capture server at the desired location.

Error parsing delimited message record header: delimited schema name (<name>) has shorter length than minimum expected length of 2

Cause: The header column value containing the schema name had shorter than expected length.

Effect: The message record is not processed. Depending on parameter settings, the process either terminates or skips the record and attempts to continue.

Recovery: Verify that only a Q Capture Server is feeding the configured MQ Queue using delimited messages with the correct delimiters. Correct any such issues found, remove any bad messages from the queue, and restart the Q capture server at the desired location.

Error parsing delimited message record header: delimited table name (<name>) has shorter length than minimum expected length of 2

Cause: The header column value containing the table name had shorter than expected length.

Effect: The message record is not processed. Depending on parameter settings, the process either terminates or skips the record and attempts to continue.

Recovery: Verify that only a Q Capture Server is feeding the configured MQ Queue using delimited messages with the correct delimiters. Correct any such issues found, remove any bad messages from the queue, and restart the Q capture server at the desired location.

Error parsing delimited message record header: Unknown record operation value (<value>)

Cause: The header column value containing the record operation type had an unknown value. Expected values are ‘ISRT’, ‘REPL’, or ‘DLET’.

Effect: The message record is not processed. Depending on parameter settings, the process either terminates or skips the record and attempts to continue.

Recovery: Verify that only a Q Capture Server is feeding the configured MQ Queue using delimited messages with the correct delimiters. Correct any such issues found, remove any bad messages from the queue, and restart the Q capture server at the desired location.

```
Error parsing delimited message: record had fewer column values than expected
Schema.Table: <schema name.table name>
Expecting <num> before column values; record only had <num> column values
 remaining
```

Cause: The message record contained fewer columns than expected based on the table schema read from the database.

Effect: The message record is not processed. Depending on parameter settings, the process either terminates or skips the record and attempts to continue.

Recovery: Verify that only a Q Capture Server is feeding the configured MQ Queue using delimited messages with the correct delimiters. If any table schema changes have occurred, corresponding changes must be made to the table publication. Correct any such issues found, remove any bad messages from the queue, and restart the Q capture server at the desired location.

```
Error parsing delimited message: record had fewer column values than expected
Schema.Table: <schema name.table name>
Expecting <num> after column values; record only had <num> column values
 remaining
```

Cause: The message record contained fewer columns than expected based on the table schema read from the database.

Effect: The message record is not processed. Depending on parameter settings, the process either terminates or skips the record and attempts to continue.

Recovery: Verify that only a Q Capture Server is feeding the configured MQ Queue using delimited messages with the correct delimiters. If any table schema changes have occurred, corresponding changes must be made to the table publication. Correct any such issues found, remove any bad messages from the queue, and restart the Q capture server at the desired location.

```
Error parsing delimited message: record had more column values than expected
Schema.Table: <schema name.table name>
Expected record to contain before and after image values for <num> columns
```

Cause: The message record contained more columns than expected based on the table schema read from the database.

Effect: The message record is not processed. Depending on parameter settings, the process either terminates or skips the record and attempts to continue.

Recovery: Verify that only a Q Capture Server is feeding the configured MQ Queue using delimited messages with the correct delimiters. If any table schema changes have occurred, corresponding changes must be made to the table publication. Correct any such issues found, remove any bad messages from the queue, and restart the Q capture server at the desired location.

c-Tree Error 527 Related Messages

```
ReadSTMTDsta(): Retrying DOC reading error 527; Retry #<n> of 7; <m>
records read successfully
```

Cause: The TRS / TFS encountered an error 527 while reading data for a transaction for replay. <m> c-Tree records were read prior to the error. The TRS / TFS will retry the error up to 7 times before stopping.

Effect: The error is retried.

Recovery: None needed, if the retry count of 7 is not exceeded. If it is exceeded, it is likely that the index files are corrupt and need to be reloaded. Use sbfile / REINDEX to correct the index files and restart.

```
ReadCommittedTransactions(): Retrying CTree error 527 reading commit records
in <doc>; retry <n> of 8
```

Cause: The TRS / TFS encountered an error 527 while reading data for a transaction for replay. The TRS / TFS will retry the error up to 8 times before stopping.

Effect: The error is retried.

Recovery: None needed, if the retry count is not exceeded. If it is exceeded, it is likely that the index files are corrupt and need to be reloaded. Use sbfile / REINDEX to correct the index files and restart.

Container Database Related Messages

```
<name> is a container database and SHAD_OPCOLLECT_PDB_NAME is not
 specified.
SHAD_OPCOLLECT_PDB_NAME is required to specify the PDB for
 replication. Correct the configuration and restart.
Performing Shutdown
```

Cause: The database specified for HPE Shadowbase for Oracle Log Source replication is a container database and the Oracle Pluggable Database (PDB) is not specified.

Effect: Configuration error, the collector stops.

Recovery: Correct the configuration by specifying the PDB name using the SHAD_OPCOLECT_PDB_NAME parameter in the shadparm.ini file. Note that each collector object can only collect from a single PDB in a pluggable database environment.

```
<name> is not a container database and SHAD_OPCOLLECT_PDB_NAME is
 specified.
Correct the configuration by removing the SHAD_OPCOLLECT_PDB_NAME
 entry and restart.
Performing Shutdown
```

Cause: The database specified for HPE Shadowbase for Oracle Log Source replication is a not container database and the Oracle Pluggable Database (PDB) is specified.

Effect: Configuration error, the collector stops.

Recovery: Correct the configuration by removing the PDB name using the SHAD_OPCOLECT_PDB_NAME parameter in the shadparm.ini file.

```
Select statement querying [GV$DATABASE] to see if it is a container database
 failed: sqlcode [<sqlcode>]; error[<error>]
Query: SELECT COUNT(*) FROM GV$DATABASE WHERE CDB='YES'
Performing Shutdown
```

Cause: The HPE Shadowbase for Oracle Log Source collector was not able to execute the query to determine if the database is a container database. The query failed with specified <sqlcode> and <error>.

Effect: The collector stops.

Recovery: Correct the error and restart. The most likely cause of the error is a permissions issue.

```
Select statement querying [sys_context] to see if we are logged into the root of a
 container database failed: sqlcode: [<sqlcode>]; error[<error>]
Query: SELECT COUNT(*) FROM DUAL WHERE
 SYS_CONTEXT('USERENV','CON_NAME')='CDB$ROOT'
Performing Shutdown
```

Cause: The HPE Shadowbase for Oracle Log Source collector was not able to execute the query to determine if it is connected to the root (CDB$ROOT) of the container database. The query failed with specified <sqlcode> and <error>.

Effect: The collector stops.

Recovery: Correct the error and restart.

```
The current connection to <name> is not to CDB$ROOT as required for
 replication
Correct the connection configuration and restart
Performing Shutdown
```

Cause: The database specified for HPE Shadowbase for Oracle Log Source replication is a container database but the collector is connected to one of the pluggable databases instead of the root database, as require.

Effect: Configuration error, the collector stops.

Recovery: Correct the connection information to connect to the root of the container database using SBMON. Restart.

```
Select statement querying [GV$PDBS] for the CON_UID for <pdb_name>
 returned no data: sqlcode [<sqlcode>]; error[<error>]
Either the pluggable database does not exist or the users CONTAINER_DATA
 attribute is not set
Verify that the SHAD_OPCOLLECT_PDB_NAME is correct and that the users
 CONTAINER_DATA attribute is set.
Performing Shutdown
```

Cause: The query to determine the UID for the PDB specified by the SHAD_OPCOLLECT_PDB_NAME parameter returned no data.

Effect: The collector stops.

Recovery: As noted in the messages, this is typically caused by one of the following:

- The SHAD_OPCOLLECT_PDB_NAME is incorrect and the PDB does not exist, or
- The Shadowbase user does not have the CONTAINER_DATA attribute set correctly.

Correct the issue and restart.

```
Select statement querying [GV$PDBS] for the CON_UID for <pdb_name> failed:
 sqlcode [<sqlcode>]; error[<error>]
```

```
Query: SELECT CON_UID INTO :my_con_uid FROM GV$PDBS WHERE
 NAME=<pdb_name>
Performing Shutdown
```

Cause: The HPE Shadowbase for Oracle Log Source collector was not able to execute the query to determine the UID of the pluggable database specified by the SHAD_OPCOLLECT_PDB_NAME parameter. The query failed with specified <sqlcode> and <error>.

Effect: The collector stops.

Recovery: Correct the error and restart.

```
DROP DATABASE LINK prepare failed: return [<return_code>]; sqlcode
 [<sqlcode>]; error[<error>]
Statement: DROP DATABASE LINK <link_name>
Link may not have been dropped
```

Cause: The prepare of the DROP DATABASE LINK statement failed.

Effect: The collector continues.

Recovery: Monitor collection to make sure it is working and the correct files have been selected. You can also pre-create the link and set SHAD_OPCOLLECT_PDB_CREATE_LINK=0 to avoid recreating the link every restart.

```
DROP DATABASE LINK failed: return [<return_code>]; sqlcode [<sqlcode>];
 error[<error>]
Statement: DROP DATABASE LINK <link_name>
Link may not have been dropped
```

Cause: Execution of the prepared DROP DATABASE LINK statement failed.

Effect: The collector continues.

Recovery: Monitor collection to make sure it is working and the correct files have been selected. You can also pre-create the link and set SHAD_OPCOLLECT_PDB_CREATE_LINK=0 to avoid recreating the link every restart.

```
Link <name> dropped, will be recreated
```

Cause: The database link <name> has been dropped successfully.

Effect: The collector continues.

Recovery: None required, information only.

```
Creating database link to PDB: CREATE DATABASE LINK <link-name>
 CONNECT TO <user-name> IDENTIFIED BY ******** USING ‘<pdb-name>
```

Cause: The collector is creating a link named <link-name> for the user <user-name> in to the pluggable database (PDB) <pdb-name> to access the configuration tables.

Effect: The collector continues.

Recovery: None required, information only.

```
CREATE DATABASE LINK prepare failed: return [<return_code>]; sqlcode
 [<sqlcode>]; error[<error>]
Performing Shutdown
```

Cause: The prepare of the CREATE DATABASE LINK statement failed. Note: the statement that failed will be listed immediately above in the log.

Effect: The collector stops.

Recovery: Correct the error condition and restart. You can also pre-create the link and set SHAD_OPCOLLECT_PDB_CREATE_LINK=0 to avoid recreating the link every restart.

```
CREATE DATABASE LINK failed: return [<return_code>]; sqlcode [<sqlcode>];
 error[<error>]
Performing Shutdown
```

Cause: Executing the prepared CREATE DATABASE LINK statement failed with the specified error codes. Note: the statement that failed will be listed immediately above in the log.

Effect: The collector stops.

Recovery: Correct the error condition and restart. You can also pre-create the link and set SHAD_OPCOLLECT_PDB_CREATE_LINK=0 to avoid recreating the link every restart.

```
Prepare of select statement querying [GV$INSTANCE] for version failed: sqlcode
 [<sqlcode>]; error[<error>]
Query: SELECT INSTANCE_NAME,VERSION FROM GV$INSTANCE
Performing Shutdown
```

Cause: The prepare of the SELECT statement to get the Oracle database version failed with the specified errors.

Effect: The collector stops.

Recovery: Correct the error condition and restart.

```
Select statement querying [GV$INSTANCE] for version failed: sqlcode
 [<sqlcode>]; error[<error>]
Query: SELECT INSTANCE_NAME,VERSION FROM GV$INSTANCE
Performing Shutdown
```

Cause: The HPE Shadowbase Collector for Oracle Log source was not able to query for the version of Oracle.

Effect: The collector stops.

Recovery: Correct the error condition and restart.

```
Select statement querying [GV$INSTANCE] returned invalid version string [%s],
 expected [n.n.n.n.n] format
Query: SELECT INSTANCE_NAME,VERSION FROM GV$INSTANCE
```

```
Performing Shutdown
```

Cause: The query to get the Oracle version number returned the version in an unexpected format.

Effect: The collector stops.

Recovery: It is likely that you are trying to replicate from an unsupported version of Oracle. Contact Shadowbase Support.

```
Unsupported Oracle database version <version>
Query: SELECT INSTANCE_NAME,VERSION FROM GV$INSTANCE
Performing Shutdown
```

Cause: You are trying to replicate from an unsupported version of Oracle.

Effect: The collector stops.

Recovery: You are trying to replicate from an unsupported version of Oracle. Contact Shadowbase Support.

```
Replicating from <database-name>; version [<version>] ([nn.nn])
```

Cause: Info message about the source database and version, printing during startup.

Effect: None.

Recovery: None required, informational only.

```
Select statement querying [<table-name>] failed: sqlcode <sql-code>];
 error[<error>]
Selection Criteria condition prevents proper change data collection operation
Performing Shutdown
```

Cause: The query to check for rows in the specified configuration table failed with the specified error. The <table-name> will be one of the configuration tables: SHAD_USERS_INCLUDE, SHAD_USERS_EXCLUDE, SHAD_TABLES_INCLUDE, or SHAD_TABLES_EXCLUDE.

Effect: The collector stops.

Recovery: Correct the issue that prevents the proper selection from the configuration table and restart.

```
<selection-parameter> enabled, but no rows exist in <table-name>
Selection Criteria condition prevents proper change data collection operation
Performing Shutdown
```

Cause: The configuration file (shadparm.ini) specified that the configuration table contains criteria for selecting the events, but there are no rows in the table. <selection-parameter> will be one of SHAD_OPCOLLECT_USE_INCLUDE_TABLES, SHAD_OPCOLLECT_USE_EXCLUDE_TABLES, SHAD_OPCOLLECT_USE_INCLUDE_USERS, or SHAD_OPCOLLECT_USE_EXLUDE_USERS. <table-name> will be the corresponding configuration table.

Effect: The collector stops.

Recovery: Either disable the parameter or add selection criteria to the specified table.

Statement Parsing Errors

```
GetColumnNames() : Error parsing SQL statement while searching for beginning
 of column list
GetColumnNames() : Error parsing SQL statement while searching for end of
 column name
GetColumnNames() : Error parsing SQL statement while searching for WHERE
 clause
GetColumnNames() : Error parsing SQL statement while searching for SET clause
GetColumnNames() : Error parsing SQL statement while searching for end of
 column name in UPDATE / SET clause
```

```
GetColumnNames() : Error parsing SQL statement while searching for end of
 column name in WHERE clause
GetColumnNames() : Error parsing SQL statement while searching for end of
 TO_DATE clause in WHERE clause

GetColumnNames() : Unable to find [<search string>] starting at index [<index>]
 in statement
Statement : [<statement>]
```

Cause: The TRS or TFS was not able to properly parse an SQL statement to determine the column names. One of the first seven messages will be displayed, indicated what processing the parsing was trying to do. The next message will indicate the string it was searching for and where in the statement it was starting its search. The final message provides the statement that failed.

Effect: The object stops.

Recovery: This error should not occur and indicates a flaw in the parsing logic. The offending table will likely need to be removed from replication and Shadowbase Support should be contacted a diagnosis of the issue and for a work-around.

```
ParseStement(): Error parsing SQL statement near [<index>] - missing TO_DATE
 clause
Statement: <statement>)
Verify that the source SQL format for DOC matches the target format
Performing shutdown
```

Cause: The TRS or TFS was not able to properly parse an SQL statement to determine the column names as it was not an Insert, Update, or Delete statement. The next message provides the statement that failed.

Effect: The object stops.

Recovery: We have seen this error occur when the source format for the DOC is Oracle but the target format configured for the TRS or TFS is non-Oracle. Verify that the source format for the DOC matches the format of TRS or TRS – if it doesn’t, correct the error and restart.

Known Problems Remaining

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. This issue will be addressed in an upcoming release.

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 collection does not support replication of tables with columns defined as TIMESTAMP WITH TIMEZONE or TIMESTAMP WITH LOCAL TIMEZONE.

    b. HPE Shadowbase Log Source for Oracle does not support replication of tables with CLOB or BLOB columns.

    c. 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.

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

    e. There is an issue handling delete operations that have a date as part of the primary key field. Contact Shadowbase Support if any of the tables to be replicated include a data field in the primary key.

    f. HPE Shadowbase Log Source for Oracle does not support collecting tables with table or column names that are reserved words or that require quoting to be resolved.

    g. 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.

    h. HPE Shadowbase Log Source for Oracle does not support collecting tables returned by Oracle LogMiner using the UNISTR function string.

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. The sample DOC reader code (SBDOCRD) for reading non-cached DOCS fails. Both a sample program that reads and prints the DOC, as well as source code to allow the user to process the data in the DOC. Both the sample program and code fail.

    This code has limited functionality and is deprecated. However, if you need a working copy of SBDOCRD, please contact Support.

11. 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.

12. 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.

13. When configuring HPE Shadowbase Log Source for Oracle collection with an Oracle password that is set to expire soon, the configuration fails. Oracle will issue a message when the initial connection is completed, which causes the script to fail. The sample below illustrates the issue:

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                      Shadowbase Open Server                        *
*                Oracle Log Based Source Collection                  *
*                   Configuration and Maintenance                    *
*                                                                    *
* Copyright 2014 Gravic, Inc.                                        *
* All Rights Reserved.                                               *
* SBSUPPORT@GRAVIC.COM                               WWW.GRAVIC.COM  *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                       M A I N - M E N U

     1. Install Open Collection Filtering Tables
     2. Remove  Open Collection Filtering Tables
     3. Install Shadlogpack Oracle Package
     4. Remove  Shadlogpack Oracle Package
     5. Add to Filtering Tables
     6. Update Filtering Tables
     7. Delete from Filtering Tables
     8. Dispaly Configuration Settings
     9. Check an Oracle User's Permissions
    10. V&V Filtering Tables
    11. V&V Supplemental Logging

     Enter your choice [1-10, Q to quit]:1


     A connection to an Oracle DBMS Instance has not yet been
established
     Please specify the following Oracle Connection Information

     User Name: qasource
      Password:
      Instance: ORA11R2
     ERROR:
ORA-28002: the password will expire within 7 days
Press any key to continue
```

The corrective action is to remove the condition that indicated in the warning message – in this case, to set a new password.

14. 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.

15. A configuration using a consumptive Direct Writer (a Direct Writer that is not connected to a database) connected to an HPE NonStop system as a source is not supported. If you need to use a consumptive process, you must either setup a configuration that replicates from the NonStop to a Doc Writer, and then uses a consumptive TRS; or use a Direct Writer that does connect to the database with your consumptive user exit.

16. 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.

17. 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.

18. SBFILE does not display text fields containing the string 'N', 'NU', 'NUL', or 'NULL' correctly – the enclosing single quotes are left off. For example, if TEST_TABLE has four varchar fields, SBFILE will display the statement:

    ```
    INSERT INTO TEST_TABLE (C1,C2,C3,C4) VALUES ('N', 'NU', 'NUL',
    'NULL')
    ```

    incorrectly as:

```
INSERT INTO TEST_TABLE (C1,C2,C3,C4) VALUES(N, NU, NUL, NULL)
```

Note: This is a display issue only. The data will be correctly applied to the database.

19. 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.

20. 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.

21. HPE Shadowbase Log Source for Oracle will generate an incorrect *UPDATE* statement if an update is performed against a table with only key columns defined and the update does not actually change any of the key columns. For example, consider the following SQL commands:

    ```
    create table test3 (kcol1 int not null,
                        kcol2 int not null,
                        primary key (kcol1,kcol2));

    insert into test3 values (9,2);

    update test3 set kcol1=9;
    ```

    Shadowbase will generate an invalid update statement for the third update – there will be no set values:

    ```
    UPDATE TEST3 SET  WHERE KCOL1='9' AND KCOL2='2';
    ```

    If the update statement modified one of the columns, Shadowbase would have processed the event as specified by the SHAD_OPCOLLECT_PRIKEY_UPDATE_OPTION.

If this occurs in your environment, there is a work-around. You can safely ignore UPDATE statements using DBS mapping on the target for tables with only key columns. The collector will either stop, skip, or convert updates that change column values into a delete followed by an insert, depending on the setting of the SHAD_OPCOLLECT_PRIKEY_UPDATE_OPTION. The only updates that will be stored in the DOC for processing by the TRS or TFS are ones that do not change any data, which do not need to be replicated.

22. 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.

23. The TRS fails without warning when replicating a column to Oracle that starts with “TO_DATE” if DBS Mapping is enabled. A workaround is to change the SQL type to SQL92.

24. 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.

25. 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.

26. 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 ‘% ‘;
```

27. 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.

28. There are issues with using versions of MySQL Connector ODBC drivers newer than 5.1. Incorrect data values are applied to the database, causing data corruption. The workaround is to use MySQL Connector/ODBC 5.1 to replicate to MySQL databases.

29. Special considerations are required when using a TFS as a relay server. 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.

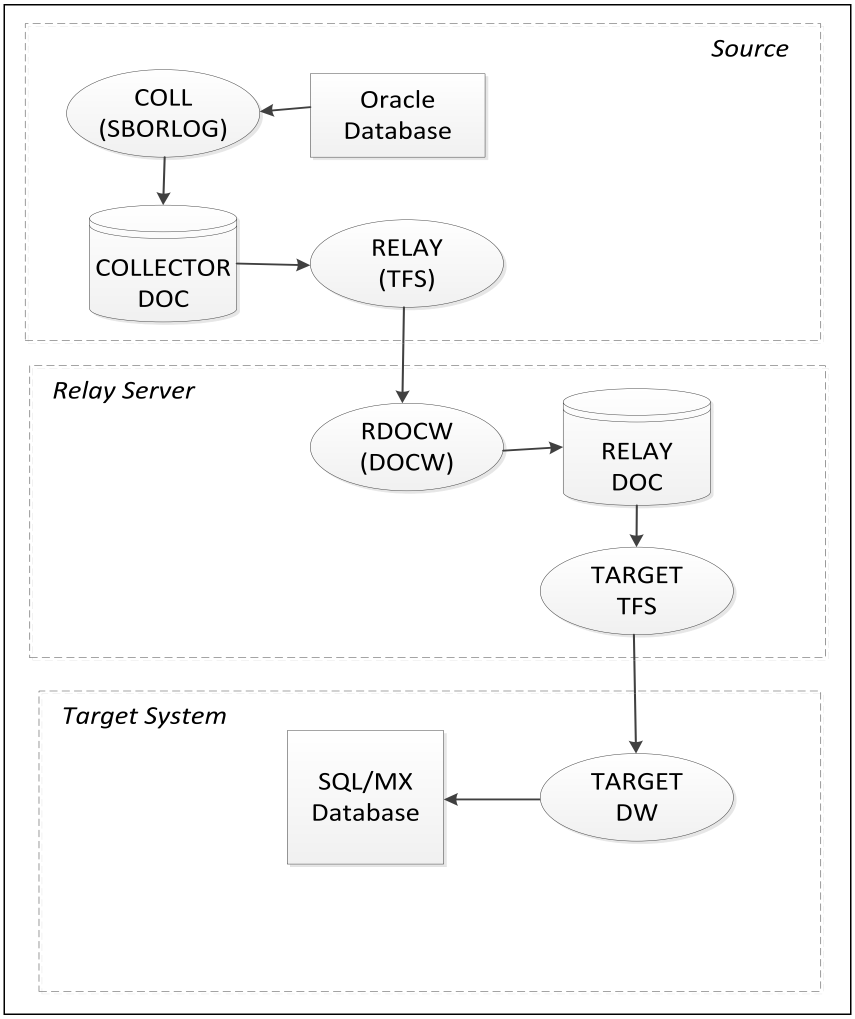

Figure 3 - Sample Relay Server Configuration

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 duplicated data from being replayed.

30. 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.

31. The new format DOC which eliminates the TRANS DOC is not compatible Solaris and Solaris x86 systems. Continue to use the old format DOC with trans processing enabled on Solaris and Solaris x86 systems.

Installation Instructions

Please follow the installation instructions included in the README.<platform>.<version>.TXT file that accompanies this release.

***** End of Document *****