SOFTWARE RELEASE DOCUMENT (SOFTDOC)

Product:	HPE Shadowbase for Other Servers
Release:	Gravic Version 6.230 T1123-AAD (SB REPL/OSS)
Date:	October 1, 2016
Copyright Notice:	Copyright Gravic, Inc. 1995 – 2016 (www.gravic.com)
File Name:	IPM6230_other_servers_OSS.pdf

NOTE: This softdoc covers new features and corrected problems for HPE Shadowbase for Other Servers, Version 6.230 for HP NonStop OSS only. It is available as an Adobe PDF file (.PDF). Copies of the PDF file reader can be freely downloaded from www.adobe.com

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.

Special Notes for Version 6.230

1. Version 6.230 is a restricted TCF release (HPE version AAD). The current non-restricted TCF release is version 6.220 (HPE version AAC) for OSS. This release applies to the T1123-AAD (SB REPL/OSS) component of HPE Shadowbase only; this component provides the target SQL/MX functionality.

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. HPE Shadowbase for Other Servers now obfuscates configuration data (in particular the passwords) for objects when it is stored in the COLLCONFIG data file (in other words, the information is no longer stored in 'clear text'). Version 6.230 can read the configuration records created by prior releases and will automatically store the information in obfuscated format when the record is saved/written. Once the information has been obfuscated, the record is no longer usable in prior versions. If you are upgrading from a version prior to 6.230 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 need an obfuscated configuration with a prior release, you will need to drop and re-add the objects using the prior version of SBMON.

New Features in Version 6.230

This section provides a summary of the features added to HPE Shadowbase for Other Servers products since the previous general availability release (Version 6.220). This release focusses on performance increases related to DOC file writing and replay in the OSS file system.

1. 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).

   See the *DOC Writing without the TRANS DOC* section later in the SOFTDOC for more information.

2. 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

3. The SBMON FILESTATS command has been modified to accommodate the new DOC blocking algorithm. FILESTATS will now display the Insert, Update, and Delete counts with DOC Blocking enabled.

   See *FILESTATS Command* section later in the SOFTDOC for more information.

4. 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.

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.

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.

Table 1 - Logical I/Os Per Transaction, 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.

Table 1 - Logical I/Os Per Transaction

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* section 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

Configuring DOC Writing with the TRANS DOC

The current SBUNDROS and SBBIDROS macros do not have an option to generate a configuration without Transaction Processing enabled. To configure DOC writing with the TRANS DOC, you need to edit the DOC writers after they have been configured. The basic steps are:

1) Create the scripts using either the SBUNDROS or SBBIDROS macros and running the sb_gen macro.

2) Configure the target system by running the generated xxSBMON script, where xx is the target node abbreviation.

3) On the target system, run SBMON and edit the DOC Writer objects. Answer ‘N’ to the *Transaction Processing ([Y]/N)?* prompt. You will need enter a password to save the change. By default, the TACL macros use ‘OPEN’ as the password.

You can then start the target using the normal scripts.

Note that you will have edit the DOC Writers after every time the xxSBMON scripts are run.

Configuring DOC Blocking Version 2

You can override the default value for doc blocking using the SB_ADD_CONSOS_PARAM macro to add a DOC Writer parameter to the SHADPARM.INI file. For example, to configure DOC blocking version 2 for all DOC Writers, you would add the following to your configuration:

SBUNDROS:

```
SB_ADD CONSOS_PARAM * DOC * SHAD_DOC_BLOCKING 2
```

SBBIDROS:

```
SB_ADD CONSOS_PARAM * * * DOC SHAD_DOC_BLOCKING 2
```

Problems Corrected in Version 6.230

This section provides a summary of the problems corrected in HPE Shadowbase for Other Servers products since the previous general availability release (V6.6.220).

1) The example user exit make file shipped with the OSS version of HPE Shadowbase for Other Servers references the release version of USRXDLLINIT.o library, which is not previously shipped with OSS. There is now a release version of the USRXDLLINIT.o shipped with OSS.

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.

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.

Table 2 - FILESTATO Column Definitions, below, documents the meaning of each of the count columns in the FILESTATO output.

Table 2 - 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).

Basic vs. Advanced Parameters

Shadparm.ini parameters can be broken into two categories: basic and advanced. Basic parameters are the ones most frequently set and changed by HPE Shadowbase users across various implementations. Advanced parameters are typically left to their default value and should not be changed unless advised to do so by HPE Shadowbase Support.

NOTE: Fully understand the proper use and potential side effects before setting either a basic or advanced parameter. In particular, do not set any of the advanced parameters before consulting your qualified HPE Shadowbase support specialist. Undesirable effects may result from the indiscriminate use of parameters, including the potential for data loss or target database corruption. Wherever possible, test parameter use in a non-production environment, including a copy of real production data, before deploying in a production environment.

SHAD_DOC_BLOCKING

Initial Version:	v3.930	**Syntax:**	SHAD_DOC_BLOCKING=<num>
Last Change:	V6.230		
Default Value	1		
Other Servers	DOC		
Valid Settings	0: Disable combining of DOC rows (No DOC blocking) 1: Enable combining of DOC IUD Event Rows (Doc Blocking, Version 1) 2: Enable combining of all DOC Event Rows (DOC Blocking, Version 2)		
Basic/Advanced	Advanced – See Basic vs. Advanced Parameters section		

Description

This parameter is used to control whether a single DOC row can contain multiple SQL statements from the same transaction. There are three variations of DOC blocking:

- Disabled (SHAD_DOC_BLOCKING=0): Each event is written to a separate record in the DOC.
- Version 1 (SHAD_DOC_BLOCKING=1): Insert, Update, and Delete events for a transaction may be combined in a single DOC record. BEGIN, COMMIT, and ABORT events are still written in individual records.
- Version 2 (SHAD_DOC_BLOCKING=2): All events for a transaction may be combined in a single DOC record.

Notes

Records with multiple, combined events are referred to as “Multi OPs” in *sbmon* filestats.

When using cached DOCs, statement data expansion can occur for in-bound data representing the ‘null’ state. In this case, the DOC database would store the character string "NULL" to represent the ‘null’ state, which would increase the total byte length by four for each column instance processed.

Version 2 of DOC Blocking was introduced in release 6.230. DOCs created with Version 2 of DOC Blocking are not compatible with releases prior to 6.230.

Related Parameters

SHAD_DOC_BLOCKING_FLUSH_CACHE

SHAD_DOC_BLOCKING_FLUSH_CACHE

Initial Version:	V6.230	**Syntax:**	SHAD_DOC_BLOCKING_FLUSH_CACHE=<num>
Last Change:	--		
Default Value	1		
Other Servers	DOC		
Valid Settings	0: Disable flushing of blocked events after every IPC. 1: Enable flushing of blocked events after every IPC.		
Basic/Advanced	Advanced – See Basic vs. Advanced Parameters section		

Description

This parameter controls how long the DOC Writer will cache data for DOC blocking prior to writing the record to the DOC. The previous algorithm for DOC blocking wrote blocked records after every IPC, which corresponds to SHAD_DOC_BLOCKING_FLUSH_CACHE=1, the default. Setting SHAD_DOC_BLOCKING_FLUSH_CACHE=0 disables this flushing. Data will not be written until either no more events can fit in the DOC record or an end transaction is received.

Notes

This parameter only takes effect with SHAD_DOC_BLOCKING=2 or greater.

This parameter is experimental and should only be set under the direction of Shadowbase Support.

Related Parameters

SHAD_DOC_BLOCKING

SHAD_DOC_BLOCKING_MAX_RECORD_LENGTH

Initial Version:	V6.230	**Syntax:**	SHAD_DOC_BLOCKING_MAX_RECORD_LENGTH=<num>
Last Change:	--		
Default Value	52,000 (OSS), 65,490 (Windows, Linux, Unix)		
Other Servers	DOC		
Valid Settings	1 – 52,000 (OSS); 1-65,490 (Windows, Linux, Unix)		
Basic/Advanced	Advanced – See Basic vs. Advanced Parameters section		

Description

This parameter specifies the maximum amount of data that will be blocked together in a DOC record before it is written. Note that an individual event may exceed this length, in which case the previously cached events will be as a record first, and the long event will be written in its own record.

Notes

This parameter only takes effect with SHAD_DOC_BLOCKING=2 or greater.

This parameter is experimental and should only be set under the direction of Shadowbase Support.

Related Parameters

SHAD_DOC_BLOCKING

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 (V6.101 for all servers except HPE NonStop OSS; V6.200 for HPE NonStop OSS).

Logged for Reading Committed Transaction Errors

```
ReadCommittedTransactions(): Corrupted (short) record for COMMIT record
 type read in DOC <DOC Name> at location <loc>; length read=<length>
```

Cause: The TRS or TFS read a statement record that was too short to process. This indicates that the specified DOC file contains corrupt data.

Effect: The TRS or TFS stops.

Recovery: Restart the TRS or TFS to see if it is a transient error. If not, the restart point will need to be adjusted to skip the transaction.

```
ReadCommittedTransactions(): CTree error <error> reading commit records
 in DOC <DOC Name>;
```

Cause: The TRS or TFS encountered a CTree error while reading data from the event DOC.

Effect: The TRS or TFS stops.

Recovery: Refer to the CTree error codes to determine the issue and the corrective action.

Logged for Reading Begin Records

```
FindSeqnoWithBeginRecord(): CTree error <error> reading event records
 in DOC <DOC Name> for transaction <transaction id>
```

Cause: The TRS or TFS encountered a CTree error while reading data from the specified event DOC while searching for the DOC with the BEGIN event for the specified transaction.

Effect: The TRS or TFS stops.

Recovery: Refer to the CTree error codes to determine the issue and the corrective action.

```
FindSeqnoWithBeginRecord (): Unable to find BEGIN event record for
 transaction %s; COMMIT record is in DOC %s%03.3d
```

Cause: The TRS or TFS could not find an event DOC containing the begin record for the specified transaction.

Effect: The TRS or TFS stops.

Recovery: Verify that the DOC files have not been prematurely purged. If the DOC files have been purged, you may need to reload the data from the source system.

Logged for SHAD_DOC_BLOCKING Parameter Errors

```
SHAD_DOC_BLOCKING setting invalid for Comma Delimited Objects; Using
 default:= 1; DOC Blocking Mode: ENABLED
```

Cause: An invalid SHAD_DOC_BLOCKING setting was specified in the SHADPARM.INI.

Effect: The blocking mode is set to DOC blocking version 1.

Recovery: None required.

Logged for File Modification Time Errors

```
GetFileModificationTime(): Unable to open file <file> to determine
 modification time. Errno=<error>

Or

GetFileModificationTime(): Unable to determine modification time for
 file file>. Errno=<error>
```

Cause: The process was not able to determine the file modification time for the specified file. It was either not able to open the file (first message), or the call to fstat for the modification failed (second message)

Effect: Varies, depending on why the modification was being requested. Subsequent messages in the log will provide additional information.

Recovery: Varies, depending on why the modification was being requested. Subsequent messages in the log will provide additional information.

Logged for DOC Version Errors

```
SetDocInfo(): Incorrect version for the event DOC file <DOC NAME>
```

Cause: The TRS, TFS, or SBFILE was not able to read the DOC file as it contains an unknown version. This is most likely caused by trying to process a new version of the DOC with an older version of the software.

Effect: The process stops.

Recovery: Use the correct version of the software to process the DOC.

```
SetDocInfo(): Reading in committed transaction order without using the
 TRANS doc file requested but not supported by this version
```

Cause: The TRS, TFS, or SBFILE tried to read from a DOC that did not have a TRANS doc file, but that was created before version 6.230. Version 6.230 changed the statement doc format to allow reading in commit order.

Effect: Program stops without processing the DOC files.

Recovery: Reload the DOC data in the correct format.

Logged for DOC Cleaner Errors

```
FileCleanupDocsByTime(): Unable to generate list of DOC files, will try
 later ...
```

Cause: The DOC Cleaner was not able to generate a list of DOC files to be checked for cleaning. Prior error messages in the log should provide the reason from the issue.

Effect: No DOC files are removed during this cycle of DOC cleaning. If the error is transient, the files will be cleaned on the next cycle.

Recovery: Check the cause of error and, if possible, correct. Otherwise, the DOC files will need to be deleted manually.

```
FileCleanupDocsByTime(): Unable to get file modification time for file
 %s, file will not be purged
```

Cause: A file error (logged earlier) prevented the DOC cleaner from determining the modification for the file.

Effect: The file will not be purged.

Recovery: If necessary, purge the file manually.

```
FileCleanupDocsByTime(): Unable to extract sequence number from file
 %s, file will not be purged
```

Cause: The specified file does not contain a valid sequence number.

Effect: The file is not purged.

Recovery: Purge the file manually, if needed.

```
FileCleanupDocsByTime(): Unable to find current TRS/TFS DOC seqno
 (%3.3d) in the list of DOC files. No files will be purged
```

Cause: The DOC cleaner could not find the DOC the TRS/TFS is currently processing.

Effect: No DOC files are purged.

Recovery: Purge DOC files manually. This may indicate an issue with the TRS/TFS; you may need to edit the TRS/TFS to point to an existing DOC file to allow it to run.

```
FileCleanupDocsByTime(): Delete of file %s failed.
```

Cause: The DOC cleaner could not delete the specified DOC file.

Effect: The file is not deleted. A prior message should indicate the reason the delete failed.

Recovery: Delete the file manually.

Logged for DOC Reading Errors

```
ReadByTxUsingEVENTDoc(): Reading from multi-ported DOC files is not
 supported for DOC files with transaction processing disabled
Performing shutdown
```

Cause: The TRS or TFS cannot read from multi-ported DOCs that have Transaction Processing disabled.

Effect: The TRS or TFS stops.

Recovery: Correct the DOC Writer configuration. You will need to roll to new DOCs prior to restarting. Any data in the existing DOCS will not be replayed. You may need to adjust the start position for the source system to re-collect the data in the existing DOCs.

```
ReadByTxUsingEVENTDoc(): Unable to position to restart transaction
 <trans id> in DOC <doc name> sequence #<seq no>
```

Cause: The TRS or TFS could not find the transaction specified by the restart point in the DOC file. Either the DOC or the restart point is corrupted.

Effect: The TRS or TFS stops.

Recovery: If the restart point is corrupted, edit the TRS or TFS and modify the DOC sequence number so the DOC is re-processed. If the DOC is corrupted, roll to a new DOC and restart the source system collection at the appropriate point to collect the missing data.

```
ReadByTxUsingEVENTDoc(): Error reading committed transactions from DOC
 <doc name> sequence number <seq no>
```

Cause: The specified DOC file is corrupt.

Effect: The TRS or TFS stops.

Recovery: Roll to a new DOC and restart collection on the source system to re-collect the missing data.

```
ProcessTxCommit(): Unable to find BEGIN record for transaction <trans
 id>; Commit record is in DOC <doc name> sequence number <seq no>
```

Cause: The TFS or TRS was not able to locate the BEGIN record for the specified transaction in the DOC files. It is likely that one or more DOC files were prematurely purged.

Effect: The TRS or TFS stops.

Recovery: Restart replication, modifying the start position on the source system to collect the missing data.

Logged for FILESTATS Errors

```
CountMultiOps(): Unexpected operation type [optype] at offset [offset]
 found while parsing MultiOp record; file may be corrupt.
```

Cause: SBMON found an unexpected operation type at the specified offset while parsing the events in a multi-op Event DOC record for a FILESTATS command. The operation is not counted in the total.

Effect: None.

Recovery: None required. The FILESTATS counts may be incorrect.

```
CountMultiOps(): Statement may be truncated, multi-op length is
 [<actual length>], expected at least [<expected length>]
```

Cause: SBMON found a short record while parsing the events in a multi-op Event record for FILESTATS

Effect: None.

Recovery: None required. The FILESTATS counts may be incorrect.

```
CountEventsInRecords(): CTree error [<error code>] reading statement
 while counting events, totals may be incorrect
```

Cause: SBMON encounter the CTree error while reading events for the FILESTATS.

Effect: None. The record is skipped.

Recovery: None required. The FILESTATS counts may be incorrect.

```
AddStatsRecord(): CTree error [<error code>] adding counts record to
 the statement db.
```

Cause: SBMON was not able to add the counts record to the Event DOC.

Effect: None

Recovery: None required.

Logged for LIST Command Processing Errors

```
get_tid_info():OpenTRANS(); Error: <error>; OPENING STRNSnnn.dat for
 Object: <DOC writer>
```

Cause: SBMON was not able open the specified TRANS DOC file while processing a LIST command.

Effect: Transaction processing statistics are not printed for the LIST command

Recovery: None required.

```
GetTxInfoFromStmtDoc(): CTree Error: <error>; reading SSTMCnnn.dat for
 Object: <DOC writer>
```

Cause: SBMON encountered the specified error reading the event DOC while processing a LIST command.

Effect: Transaction processing statistics are not printed for the LIST command.

Recovery: None required.

```
GetTxInfoFromStmtDoc(): Unable to find transaction <trans id> in
 SSTMCnnn.dat for Object: <DOC writer>
```

Cause: SBMON was not able to find the specified transaction while processing a LIST command.

Effect: Transaction processing statistics are not printed for the LIST command.

Recovery: None required.

Logged for Invalid Event Formats.

```
<Procedure Name>() : Invalid event format specified <event type>
 (<event type id>)
```

Cause: There is a configuration error. The specified procedure was called with an unsupported event type. This likely indicates that the DOC was created with later version of Shadowbase that has additional event types unknown to this version of Shadowbase.

Effect: The process stops.

Recovery: Correct the configuration.

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. 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
   ```

7. 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.

8. 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.

9. 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.

   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.

10. 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.

11. 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.

12. Pulse processing is not compatible with DOC writing with no TRANS DOC (transaction processing disabled). The pulse request never completes and shows as active, resulting in invalid pulse statistics. The output from a pulse request to a DOC Writer configured without transaction process is shown below:

```
=============================================================================
  NAME: CONS-OPN-PULC1    PROCESS: \VIV1.$PULC1      AUDMON: \VIV1.$PAUDM

    MOST RECENT PULSESTATS (MAXPULSESTATS = 5):
      -----------------------------------------------------------------------
      CONFIG PARAMETERS:         ADTSAMPLEDELAY=?        ADTXSAMPLEDELAY=?
      ADTXEOFREPEATS=?           ADTTHROTTLEDELAY=?      ADTTHROTTLELIMIT=?
      FASTSAMPLE=?               IPMBLOCKING=?           PULSEAUTOADJ=?
      TURBOMODE=?                TURBOWAITTIME=?         TURBOMAXEVENTS=?

      TGT PROCESS:        UNAVAILABLE                    TYPE:  NOT KNOWN YET
      CREATE TS:          2016-10-14 12:17:52.242862     STATE: ACTIVE
      COMPLETE TS:        UNAVAILABLE                    DIFF:  UNAVAILABLE
      CONS USRX CALLS:    0                              DUR:   00:00:00.000000
      -----------------------------------------------------------------------
      NUMBER OF PULSES FOR CONS CONS-OPN-PULC1:
        TOTAL DISPLAYED      1             TIMED OUT    0
        ACTIVE (INCOMPLETE)  1             INCOMPLETE   0
        IN BAND              0             OUT OF BAND  0
```

Installation Instructions

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

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