

# IPM 6.610 HPE NonStop Shadowbase Gateway for SOLVNV Addendum

## Abstract

The HPE Shadowbase™ for NonStop Gateway product enables SOLVNV to compare Enscribe files over TCP/IP connections. Initially, SOLVNV only worked with NonStop systems networked via Expand.

## Product Version

HPE Shadowbase Version 6.610\T1122H06^AAP

## Supported Release Version Updates (RVUs)

This manual applies to all currently supported versions as described by the [Shadowbase Supported Versions Cross-Reference](#), unless otherwise noted by a superseding document. Alternatively, visit the <http://www.ShadowbaseSoftware.com> website for the list of supported platforms and databases.

## Part Number

nnnnnn-001

## Published

July, 2020

**Document History**

<b>Part Number</b>	<b>Product Version</b>
nnnnnn-001	6.610\T1122H06^AAP

**Published**  
July 2020

# Legal Notices

© Copyright 2020 Hewlett-Packard Enterprise Development Company, L.P.

## Legal Notice

Confidential Computer Software. Valid license from HPE required for possession, use or copying.

Consistent with FAR12.211 AND 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services or as specified in the applicable HPE license. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

Export of the information contained in this publication may require authorization from the U.S. Department of Commerce.

Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation

Intel, Pentium, and Celeron are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java is a registered trademark of Oracle and/or its affiliates.

All other trademarks and registered trademarks are acknowledged and are the property of their respective companies.

# Table of Contents

- Legal Notices ..... 3
- Table of Figures ..... ii
- What's New In This Manual ..... iii
- Notation Conventions..... iv
- SBGW ..... 6
  - Introduction..... 6
  - Prerequisites..... 9
  - Running Enscribe Compare with Gateways ..... 9
    - SHADPARM File ..... 12
  - SBGW Configuration Parameters ..... 12
  - Enabling SBGW..... 12
    - SBGW Parameters..... 13
  - Known Problems ..... 17
- Messages ..... 18
  - SBGW Error Messages ..... 18
  - SBGW Informational Messages ..... 27

# Table of Figures

Figure 1 – Original SOLVNV configuration with \$Receive .....	7
Figure 2 – SOLVNV using Gateways for TCP/IP communication .....	8

# What's New In This Manual

nnnnnn-001

- Initial release

# Notation Conventions

## Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

To preserve the modification timestamp, see....on page \_\_\_\_.

## General Syntax Notation

The following list summarizes the conventions for syntax notation and presentation in the HPE Shadowbase manual collection.

**UPPERCASE LETTERS.** Uppercase letters indicate keywords and reserved words; enter these items exactly as shown. Items not enclosed in brackets are required For example:

```
PARAM SBCMDFILE
```

**lowercase letters:** Represent variable entries to be supplied by user. For example:

```
<file name>
```

**Computer type.** Computer type letters within text indicate C and Open System Services (OSS) keywords and reserved words. Type these items exactly as show. Items not enclosed in brackets are required. For example

```
RUN LOADHELP
```

**Italic computer type.** Italic computer type letters within text indicate C and Open System Services (OSS) variable items that you supply. Items not enclosed in brackets are not required. For example:

```
pathname
```

**Brackets [ ]:** Enclose optional syntax. A vertically aligned group of items enclosed in brackets represents a list of selections from which one, or none, can be chosen. For example:

```
CKPTFNAME      [\system.] [$volume.] [subvolume.]  
filename
```

**Braces { }:** Enclose required syntax. A vertically aligned group of items enclosed in braces represents a list of selections from which exactly one must be chosen. For example:

```
CHECKCONFIG    { ON }  
                { OFF }
```

**Ellipses ...:** The enclosed syntax can be repeated a number of times. For example:

```
KEYVALUESTART < <string> | <byte> > [, <string> |  
<byte> ] ...
```

**Punctuation:** All punctuation and symbols other than those described above must be entered precisely as show For example:

```
error := NEXTFILENAME ( file-name ) ;
```

```
LISTOPENS SU $process-name. #su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example:

```
"{repetition-constant-list}"
```

**Item Spacing.** Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items:

```
$process-name.#su-name
```

**Line Spacing.** If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line this spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] LINE  
      [ , attribute-spec ] ...
```



# SBGW

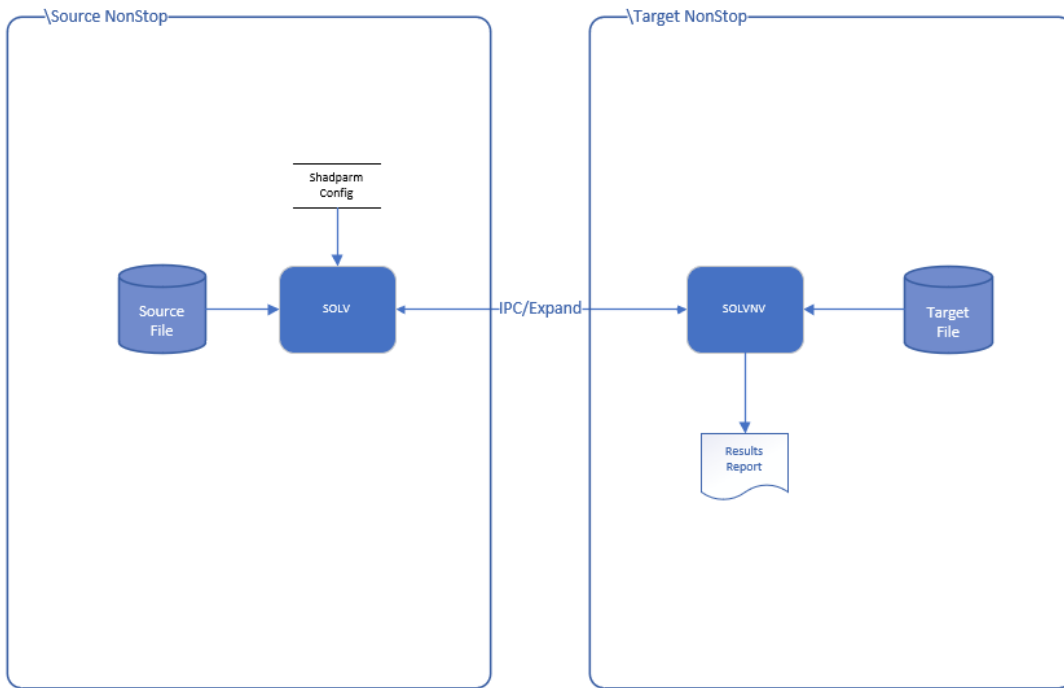
## Introduction

SBGW is a gateway process that enables SOLV and SOLVNV to work between two systems that are connected via TCP/IP network. Earlier releases of SOLV and SOLVNV required that HPE NonStop Expand exist between the two systems.

The SOLV process will read Enscribe records from a source file and prepare the data for 'SOLV Validation and Verification'. All the configuration for the compare is performed with SOLV. SOLV will forward all configuration information and the source data to SOLVNV.

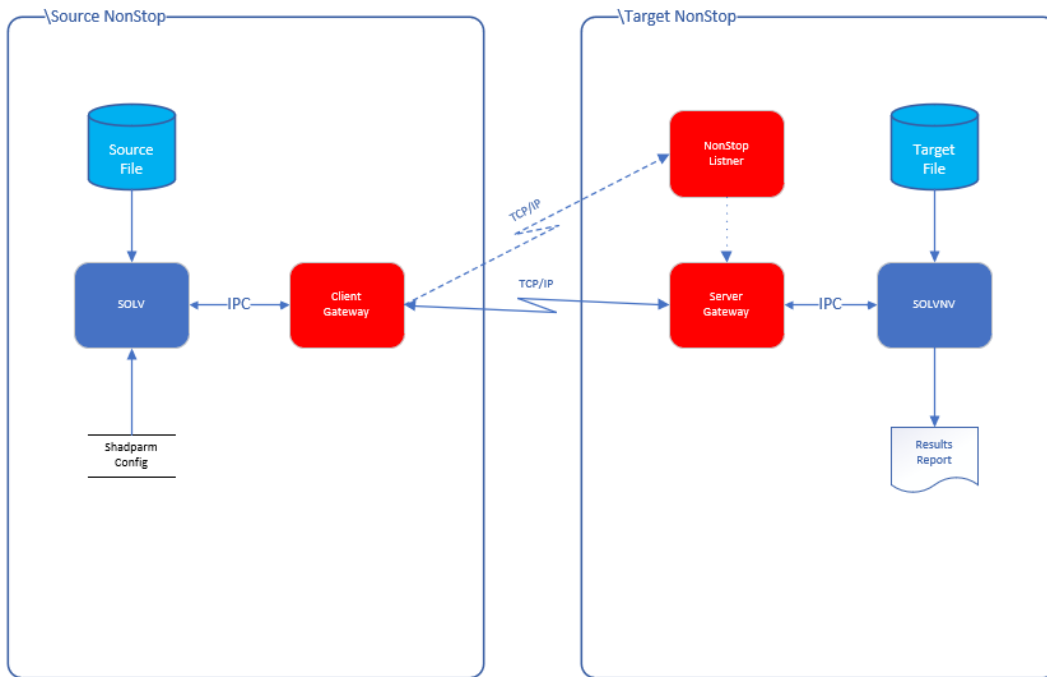
The SOLVNV process reads the target files and compares this data with the records received from SOLV. As SOLVNV is comparing the data between the source and target Enscribe files, it has an option to generate a report containing the results with different levels of detail. At the end of the run, SOLVNV will also send the final summary results to SOLV. These summary results are logged by SOLV to the configured logging location(s).

It is ideal for SOLV to run on the source system and SOLVNV to run on the target system. However, these two processes can be configured to run on either system when Expand exists. The following diagram depicts the ideal configuration when running an Enscribe compare across HPE NonStop Expand.



**Figure 1 – Original SOLVNV configuration with \$Receive**

With the introduction of SBGW, the Expand limitation has been eliminated. SOLV must now run on the source system along with the SBGW client process. These two processes will communicate locally via \$Receive interprocess messages. SOLVNV must now run on the target system along with the SBGW server process. Again, these two processes communicate via \$Receive interprocess messages. It is the two SBGW processes that communicate via TCP/IP. The following diagram depicts this new Enscribe compare configuration. Expand between the two systems is no longer required in this configuration.



**Figure 2 – SOLVNV using Gateways for TCP/IP communication**

SOLVCOM also has been enhanced with the capability to query each Gateway for statistics. SOLVCOM has always been capable of querying a running SOLV process and the 'marker' files. SOLVCOM can also be used at any time to adjust the dynamic parameters that were initially used to configure a running SOLV.

As with the original version of SOLV, the majority of the configuration is still performed at the source with SOLV. TACL parameters or an HPE Shadowbase SHADPARM ("ini file") file can be used to maintain and manage the configuration. SOLVNV will still generate the report details and SOLV will continue to display a summary of the final results.

An HPE NonStop Listener will need to be configured on the target system. All this listener needs to do is start the server (target) Gateway process. No other configurations are required on the target system as they will be forwarded by the SOLV process from the source system.

All the existing features in SOLV, like restart positioning, are still available with the product. No features available to SOLV have been removed as a result of adding the Gateway processes. The 'marker' files are still used. They will maintain a restart position during the comparison. Upon completion, the archive or history marker file will contain the comparison activity from the source file's perspective. Note that comparison results are not saved in the marker files. The marker-a file that is used for audited source file reading during a SOLV load is not used in compare for Enscribe.

## Prerequisites

The source system uses the SOLV and SBGW (client) processes. Using SOLVCOM for displaying statistics from a running SOLV, the marker files or a running gateway is optional. SOLVCOM can also dynamically adjust some of SOLV's configuration settings.

The target system uses the SOLVNV and SBGW (server) processes. SOLVCOM can be used to query the server gateway. SOLVCOM cannot query a running SOLVNV process. The target system also requires that an HPE NonStop Listener be configured and running.

Both the client and server Gateways use the same code base. They are simply configured to run as a client (default) or server. The executable code is only different when running on different platforms. For example, code 500 for TNS-X systems and code 800 for TNS-E systems.

## Running Enscribe Compare with Gateways

Follow these steps to configure Enscribe Compare with the TCP/IP gateway.

- 1 Configure and start a NonStop listener on the target system.
  - a. Add the following to an edit file named PORTCONF. The PORTCONF file is then supplied to the LISTNER process as a runtime argument when it is started.

```
<port-no> $<vol>.<subvol>.SBGW -s [-e<alt-ems-collector>] [-t]
```

For example:

```
55555 $DATA.SBINSTAL.SBGW -s
```

Where:

- 55555 is the port number (for the SBGW connection).
- \$DATA.SBINSTAL.SBGW is the location of the SBGW program that ships with Shadowbase.

- 's' is required. This tells the gateway to be started as a server. The default is to run in client mode and this is not correct for the server side.
  - The default ems collector is \$0 for when the '-e' option is not specified.
  - '-t' is for debug tracing, and should only be enabled when requested by Shadowbase Support. This will produce a substantial amount of EMS messages. It is suggested to configure a dedicated EMS collector when tracing is necessary.
- b. Start the listener. First add or alter the tcpip^process^name as appropriate.

```
> add define =TCPIP^PROCESS^NAME,class map,file $ZJ02B
```

Or

```
> alter define =TCPIP^PROCESS^NAME,class map,file $ZJ02B
```

In addition, set the LISTNER^HOST^IP TACL parameter in the same session.

```
> PARAM LISTNER^HOST^IP 10.1.50.80
```

These two settings will determine which TCP/IP address and process name will be used by the listener. When those are set, start the HPE NonStop LISTNER as follows:

```
> $SYSTEM.SYSnn.LISTNER/NAME $GWLIS,IN $ZHOME,OUT $ZHOME,TERM
$ZHOME,NOWAIT/$VOL.SUBVOL.PORTCONF
```

- 2 Configure and start SOLV for VNV on the source system. SOLV can be configured using TACL params or a SHADPARM file with sections.
  - a. The TCP/IP ipv4 address used below must be the address associated with the subnet address used in the TCPIP^PROCESS^NAME used in step 1 above. This address needs to be set in the GWCREMOTETCPIPADDRESS parameter in the steps that follow. The same hold true for the port that was used above. This same port must be set in the GWCREMOTETCPIPPORT parameter.

- b. This is a minimal TACL param configuration along with the process run command.

```
>PARAM SOLVRUNMODE          VV
>
>PARAM GWCLOCALTCPIPPROCESSNAME \SRC.$ZTC1
>PARAM GWCREMOTETCPIPADDRESS  10.1.50.80
>PARAM GWCREMOTETCPIPSPORT    55555
>
>PARAM VVMKRFILEPRE          \SRC.$DATA.GW.MKR
>PARAM VVPROGRAMFILENAME     \TGT.$DATA.SB.SOLVNV
>PARAM VVSOURCEFILENAME       \SRC.$DATA.GW.E1
>PARAM VVTARGETFILENAME       \TGT.$DATA.GW.E1
>
>RUN SOLV/NAME $SOLV/
```

The first parameter is specific to the SOLV process. The second group of parameters is specific to the gateway client process. And the last group is specific to the verification and validation logic in SOLV and also the SOLVNV process. Refer to the *HPE Shadowbase Compare for Enscribe Manual* for more information.

- c. This is the same minimal configuration. However, instead of using TACL params, a SHADPARM file is used.

Add the following to a file named SHADPARM

```
[MIN]
SOLVRUNMODE          =VV

GWCLOCALTCPIPPROCESSNAME=\SRC.$ZTC1
GWCREMOTETCPIPADDRESS  =10.1.50.80
GWCREMOTETCPIPSPORT    =55555

VVMKRFILEPRE          =\SRC.$DATA.GW.MKR
VVPROGRAMFILENAME     =\TGT.$DATA.SB.SOLVNV
VVSOURCEFILENAME       =\SRC.$DATA.GW.E1
VVTARGETFILENAME       =\TGT.$DATA.GW.E1
```

As with using TACL params, the first parameter is specific to the SOLV process. The second group of parameters is specific to the gateway client process. And the last group is specific to the verification and validation logic in SOLV and also the SOLVNV process.

Start SOLV using the following run command. SOLV will get the configuration parameters from the section named min in the

SHADPARM file. If a setting exists in both the SHADPARM file and is set as a TACL parameter, the TACL parameter will be used.

```
>RUN SOLV/NAME $GWSOL/-fSHADPARM -sMIN
```

## SHADPARM File

SOLV and the SBGW configuration parameters can be set multiple ways. They can be set with standard TACL parameters, entries in the labeled sections of a SHADPARM file or a combination of both. TACL parameters will always override entries in the SHADPARM file.

A line in an edit file is limited to roughly 230 characters. As an alternative to edit files, SOLV is also capable of reading text files (file code 180). Each line in a text file is limited to 32,767 characters. This limit is a program limit.

## SBGW Configuration Parameters

As noted earlier, SBGW is designed to be configured with TACL parameters or ideally, with parameters that are specified in a SHADPARM file. All SBGW parameters are initially processed by SOLV who in turn will start the client SBGW. SOLV will then forward all necessary configuration parameters to the client SBGW.

On the other hand, the server SBGW is minimally configured when it is started by the HPE NonStop Listener. Additional configuration values are forwarded from SOLV to the client SBGW and finally to the server SBGW.

## Enabling SBGW

Specify the SOLVRUNMODE parameter and set it to VV to run SOLV in the validation and verification mode. By default, SOLV will start SOLVNV and the communication between the two will use Guardian calls/interprocess messaging via Expand.

Specifying any of the required GWCxxx parameters will enable the use of the gateway for communication via TCP/IP.

```
GWCLOCALTCPIPPROCESSNAME  
GWCREMOTETCPIPADDRESS
```

## GWCREMOTETCPIPPORT

See the *HPE Shadowbase Compare for Enscribe Manual* for the description of these existing parameters used by SOLV, SOLVNV and the new gateway.

VVPROCESSCPU  
VVPROCESSNAME  
VVPROCESSPRIORITY  
VVPROGRAMFILENAME  
VVSERVERDEBUGONSTART

This is the list of the new client (source) SBGW parameters. The '\*' signifies which parameters are required.

GCDEBUGONSTART  
GCSEMPROCESSNAME  
GCIPCTIMEOUTSECONDS  
\*GCLOCALTCPIPPROCESSNAME  
GCPROCESSCPU  
GCPROCESSNAME  
GCPROCESSPRIORITY  
GCPROGRAMFILENAME  
\*GCREMOTETCPIPADDRESS  
\*GCREMOTETCPIPPORT  
GCTRACE

These are the new server (target) SBGW parameters

GWSDEBUGONCONNECT  
GWSSEMPROCESSNAME  
GWSIPCTIMEOUTSECONDS  
GWSTRACE

## SBGW Parameters

GCDEBUGONSTART                      ON | OFF

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

Use this parameter to have the client gateway process started in debug. Normally this setting would be enabled at the request of HPE support personnel.



**GWCEMSPROCESSNAME** <ems-process-name>

<u>Initial</u> <u>Version</u>	<u>Changed</u> <u>Version</u>	<u>Values</u>	<u>Default</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Alter</u>
6.610		\$0		-	-	NO

Set this parameter to the name of a valid EMS collector name if you wish to use something other than \$0 for the client gateway process.

**GWCIPTIMEOUTSECONDS** <timeout-in-seconds>

<u>Initial</u> <u>Version</u>	<u>Changed</u> <u>Version</u>	<u>Values</u>	<u>Default</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Alter</u>
6.610		900	(15 min)	0	32767	NO

This parameter is used to set the IPC timeout value for the client gateway.

**\*GWCLOCALTCPIPPROCESSNAME** <local-tcpip-process-name>

<u>Initial</u> <u>Version</u>	<u>Changed</u> <u>Version</u>	<u>Values</u>	<u>Default</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Alter</u>
6.610		-		-	-	NO

This required parameter sets the local tcp process name that the client gateway will open for the TCP/IP connection to the remote gateway. Set it to something like \$ztc0. See your network admin or use this command, '>scf info subnet \$\*.\*', to get a list of available subnets.

**GWCPROCESSCPU** <cpu-no>

<u>Initial</u> <u>Version</u>	<u>Changed</u> <u>Version</u>	<u>Values</u>	<u>Default</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Alter</u>
6.610		see description		0	15	NO

This optional parameter will be used to create the client gateway process in the specified CPU when it is created by SOLV. If not specified, then the same CPU where SOLV is running will be used.

**GWCPROCESSNAME** <process-name>

<u>Initial</u> <u>Version</u>	<u>Changed</u> <u>Version</u>	<u>Values</u>	<u>Default</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Alter</u>
6.610		see description		-	-	NO

This optional parameter will be used as the client gateway's process name when it is created by SOLV. If not supplied, the character 'C' will be appended to the end of the SOLV process name (or replace the last letter if the process name is already 5 characters long) and used in the process create.

**GWCPROCESSPRIORITY** <system-priority>

<u>Initial</u> <u>Version</u>	<u>Changed</u> <u>Version</u>	<u>Values</u> <u>Default</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Alter</u>
6.610		see description	1	199	NO

This optional parameter will be used to create the client gateway process with the specified priority when it is created by SOLV. If not specified, then the same priority as SOLV will be used.

**GWCPROGRAMFILENAME** <program-name>

<u>Initial</u> <u>Version</u>	<u>Changed</u> <u>Version</u>	<u>Values</u> <u>Default</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Alter</u>
6.610		see description	-	-	NO

This optional parameter sets the client gateway program name that is started by SOLV. If not supplied then the program name 'SBGW' in the SOLV subvolume will be used.

**\*GWCREMOTETCPIPADDRESS** <remote-ipv4-address>

<u>Initial</u> <u>Version</u>	<u>Changed</u> <u>Version</u>	<u>Values</u> <u>Default</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Alter</u>
6.610		none	-	-	NO

This required parameter specifies the remote TCP/IP address of the remote listener. A listener on the remote system must already be configured to start SBGW in server mode before starting the SOLV process.

**\*GWCREMOTETCPIPSPORT** <remote-port-number>

<u>Initial</u> <u>Version</u>	<u>Changed</u> <u>Version</u>	<u>Values</u> <u>Default</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Alter</u>
6.610		none	0	65535	NO

This required parameter specifies the remote port number of the listener. The remote listener must be configured to start SBGW in server mode. Ask your network admin or use '>SCF LISTOPENS PROVIDER \$\*.\*' on TNS-X or '>scf listopens process \$\*' on earlier NonStops to get a list of in-use ports on the system. This port must not be in use by any other process on the system.

GWCTRACE ON | OFF

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

Use this option to enable tracing. This is meant to be used when requested by HPE support. Note that this tracing send messages to EMS and will dramatically slow the compare.

GWSDEBUGONCONNECT <guardian-file-name>

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

Use this option to have the server gateway go into debug when the TCP/IP network session from the client gateway is started and initialized. This parameter is meant to be used by HPE support.

GWSEMSPROCESSNAME <ems-collector-process-name>

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
6.610		\$0		-	-	NO

Set this parameter to the name of a valid EMS collector name if you wish to use something other than \$0 for the server gateway process messages. Note that initially the server gateway will startup logging messages to \$0 or the ems collector specified in the run command in the PORTCONF file used to configure the LISTNER. This parameter is not activated by the server gateway until the initial handshake with the client gateway.

GWSIPCTIMEOUTSECONDS <number-of-seconds>

Initial Version	Changed Version	Values	Default	Minimum	Maximum	Alter
6.610		900	(15 min)	0	32767	NO

This parameter is used to set the IPC timeout value for the server gateway.

GWSTRACE ON | OFF

<u>Initial</u> <u>Version</u>	<u>Changed</u> <u>Version</u>	<u>Values</u>	<u>Default</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Alter</u>
6.610		OFF		OFF	ON	NO

Use this option to enable tracing. This is meant to be used when requested by HPE support. Note that this tracing send messages to EMS and will dramatically slow the compare.

## Known Problems

- 1) None at the time this document was published.

# Messages

## SBGW Error Messages

ERR 2136: SBGW: server, no accept address specified

Cause: The server gateway could not identify the remote ipv4 address during the session connection.

Effect: The server gateway will abend.

Recovery: Determine the TCP/IP network error that caused this situation and correct.

ERR 2136: get\_startup\_msg() returned an error.

Cause: A fatal error occurred trying to get the startup message from the system.

Effect: The program abends.

Recovery: Determine the cause of the invalid startup message and retry.

ERR 2136: Error <ReturnCode> returned by DEFINESETATTR adding or changing define <define\_name> as <define\_value>

Cause: A fatal error occurred trying to set define attributes. A define is used for setting the EMS collector process name internally and when starting child processes.

Effect: The program abends.

Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: Error <ReturnCode> in DEFINEADD changing define <define\_name> as <define\_value>

Cause: A fatal error occurred trying to set or add the value for an internal define. A define is used for setting the EMS collector process name internally and when starting child processes.

Effect: The program abends.

Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: call to msg\_ ->reply() returned <error-no>

Cause: A fatal error occurred in the IPC receive client module.

Effect: The program abends.

Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: <stmt> failed in <func>

Cause: A fatal error occurred in the IPC receive client module.

Effect: The program abends.

Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: ipc/rep/recv type:<num> len:<len> sess-seqno:<seqno> size:<size>; error <error-no> identified in <func>

Cause: A fatal error occurred in the IPC send server module.

Effect: The program abends.

Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: ipc/rep/recv type:<num> len:<len> sess-seqno:<seqno> vv-seqno<vvseqno> size:<size>; error <error-no> identified in <func>

Cause: A fatal error occurred in the IPC send server module.  
Effect: The program abends.  
Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: process timeout encountered in <func>

Cause: A fatal error occurred in the IPC send server module.  
Effect: The program abends.  
Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: <proc> returned <error-no> in <func>

Cause: A fatal error occurred calling the <proc> procedure in the specified function in the IPC send server module.  
Effect: The program abends.  
Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: error parsing initialization parameter <param-name> in <func>

Cause: A fatal error occurred in the IPC send server module.  
Effect: The program abends.  
Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: Process create <prog>/name <pname>,pri <pri>,cpu <cpu><debug>/ returned error <error>

Cause: A fatal error occurred in the IPC send server module creating the specified process.

Effect: The program abends.

Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: <file>:<func>:<line> server writeread completed with error <error> for msg#<number>

Cause: A fatal error occurred in the TCP receive server module writing a message.

Effect: The program abends.

Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: <file>:<func>:<line> socket::send() completed with error <error> for msg#<number>

Cause: A fatal error occurred in the TCP receive server module sending a message.

Effect: The program abends.

Recovery: Use the error code to determine the cause of the problem and retry.

ERR 2136: SBGW: Invalid address in run command, expected port.nnnnnn, received <address>

Cause: A fatal error occurred in the TCP receive server during startup. An invalid IPV4 and port encountered.

Effect: The program abends.

Recovery: Fix the address entered in the run command and retry.

ERR 2136: SBGW: socket::accept\_connection() returned <error> in <proc>

Cause: A fatal error occurred in the TCP receive server accepting a connection.

Effect: The program abends.



Recovery: Use the error code to determine the problem and retry.

ERR 2136: SBGW: socket::recv\_all() returned <error> in <proc>

Cause: A fatal error occurred in the TCP receive server while receiving a message.

Effect: The program abends.

Recovery: Use the error code to determine the problem and retry.

ERR 2136: SBGW: socket::recv\_all() completed with <error> in <proc> reading header. <bytes> bytes were received

Cause: A fatal error occurred in the TCP receive server while receiving the header portion of a message.

Effect: The program abends.

Recovery: Use the error code to determine the problem and retry.

ERR 2136: SBGW: socket::recv\_all() completed with <error> in <proc> reading payload. <bytes> bytes were received.

Cause: A fatal error occurred in the TCP receive server while receiving the payload portion of a message.

Effect: The program abends.

Recovery: Use the error code to determine the problem and retry.

ERR 2136: SBGW: socket\_.send() returned <error> in <proc> sending for msg#<number>

Cause: A fatal error occurred in the TCP receive server while sending a message.

Effect: The program abends.

Recovery: Use the error code to determine the problem and retry.

ERR 2136: SBGW: Header of TCPIP message is invalid in <proc>

Cause: A fatal error occurred in the TCP receive server while parsing a received header message.

Effect: The program abends.

Recovery: Determine if there was a potential network issue that could have caused the problem and retry.

ERR 2136: SBGW: length in header of TCPIP message is too short in <proc>

Cause: A fatal error occurred in the TCP receive server while parsing a received message.

Effect: The program abends.

Recovery: Determine if there was a potential network issue that could have caused the problem and retry.

ERR 2136: SBGW: Body of TCP/IP message is invalid in <proc>

Cause: A fatal error occurred in the TCP receive server while parsing a received message.

Effect: The program abends.

Recovery: Determine if there was a potential network issue that could have caused the problem and retry.

ERR 2136: SBGW: sequence number is invalid in <proc>. Expected <seqno>, received <myseqno>

Cause: A fatal error occurred in the TCP receive server while parsing a received message. A message sequence number is out of order.

Effect: The program abends.

Recovery: Determine if there was a potential network issue that could have caused the problem and retry.

ERR 2136: SBGW: socket\_.recv\_all() returned error <error> in <proc>  
reading for <type>, length=<length>

Cause: A fatal error occurred in the TCP receive server while receiving a message.

Effect: The program abends.

Recovery: Use the socket error to determine the cause of the network error and retry.

ERR 2136: SBGW: TCP/IP Connection return code <err> in <proc>;  
Local:<lc> Remote:<port>d.<host>

Cause: A fatal error occurred in the TCP receive server while trying to make a connection.

Effect: The program abends.

Recovery: Use the socket error to determine the cause of the network error and retry.

ERR 2136: SBGW: (buf.size() == 0) in <proc>

Cause: A fatal error occurred in the TCP receive server while trying to receive a message. This can occur when the remote process is stopped during a TCP/IP session.

Effect: The program abends.

Recovery: Determine if and why remote process stopped and retry.

ERR 2136: SBGW: header of message is invalid in <proc>

Cause: An invalid header message was encountered in the TCP send client module.

Effect: The program abends.

Recovery: Determine if there was a network error and retry.

ERR 2136: SBGW: length in header of message is too short in <proc>

Cause: An invalid header message was encountered in the TCP send client module.

Effect: The program abends.

Recovery: Determine if there was a network error and retry.

ERR 2136: SBGW: (buf.size() != length\_) in <proc>

Cause: An invalid message length was encountered in the TCP send client module.

Effect: The program abends.

Recovery: Determine if there was a network error and retry.

ERR 2136: SBGW: Body of message is invalid in <proc> (invalid sequence eye catcher

Cause: An invalid message header was encountered in the TCP send client module.

Effect: The program abends.

Recovery: Determine if there was a network error and retry.

ERR 2136: SBGW: Received TCPIP message out of sequence (no corresponding send), failure in <proc>

Cause: An invalid message sequence number was encountered in the TCP send client module.

Effect: The program abends.

Recovery: Determine if there was a network error and retry.

ERR 2136: SBGW: sock\_.send() returned error <err> in <proc>

Cause: An error occurred in the TCP send client module while trying to send a message.

Effect: The program abends.

Recovery: Use the error number to determine the cause of the network error and retry.

ERR 2136: SBGW: Bad remote TCPIP address encountered in <proc>

Cause: The TCP send client module encountered a bad remote TCPIP address during a connection.

Effect: The program abends.

Recovery: Determine the cause of the network error and retry.

ERR 2136: SBGW: sock\_.connect returned error <err> in <proc>

Cause: The TCP send client module encountered an error during a socket connection.

Effect: The program abends.

Recovery: Determine the cause of the network error and retry.

ERR 2136: SBGW: Read TCPIP header message, unexpected condition, failure in <proc>

Cause: An error occurred in the TCP send client module reading a TCPIP header message.

Effect: The program abends.

Recovery: Determine the cause of the network error and retry.

ERR 2136: SBGW: sock\_.recv\_all() returned error <error> in <proc>

Cause: The TCP send client module encountered an error receiving a TCPIP message.

Effect: The program abends.

Recovery: Use the error number to determine the cause of the network error and retry.

ERR 2136: SBGW: Read TCPIP data, unexpected condition, failure in <proc>

Cause: The TCP send client module encountered an unexpected condition receiving a tcPIP message.

Effect: The program abends.

Recovery: Determine the cause of the network error and retry.

## SBGW Informational Messages

ERR 2136: SBGW: <this-prog> Client(<this-pname>) IPC connected to <prog>(<pname>)

Cause: The program with the specified process name receive a \$RECEIVE connection from the specified program and program name.

Effect: Informational message only.

Recovery: None necessary.

ERR 2136: SBGW: ignoring ipc request from <program-name>

Cause: The program logging the message receive a \$RECEIVE open request from an unexpected program-name.

Effect: The process open request is ignored.

Recovery: None necessary. The user may want to investigate why the program is trying to open the gateway process.

ERR 2136: <program\_name> <Client|Server> <process\_name> started

Cause: Client or server program start message

Effect: Informational message only.

Recovery: None necessary.

ERR 2136: <program\_name> <Client|Server> (<process\_name>) TCP/IP connected to Remote:<ip\_address>

Cause: Successful connection message.

Effect: Information only.

Recovery: None necessary.

ERR 2136: SBGW: Disconnecting from <prog-fname>(<process-name>)

Cause: Normal disconnect message

Effect: Informational message only.

Recovery: None necessary.

ERR 2136: SBGW: Process create <prog>/name <pname>,pri <pri>,cpu <cpu><debug>/ returned error <error>

Cause: Process created successfully

Effect: Informational message only.

Recovery: None necessary.

ERR 2136: SBGW: socket::recv\_all() completed with no data, remote client closed connection. Shutting down

Cause: The TCP receive server received a message with zero length. This is usually encountered when the client session is normally torn down.

Effect: Informational message only.

Recovery: None necessary.

ERR 2136: SBGW: Gateway Server shutting down TCPIP connection

Cause: Normal shutdown message.

Effect: Information only.

Recovery: None necessary.

ERR 2136: SBGW: Client TCP/IP connection on <>(lcl) to <>( <>) on <>:<>(rmt)"

Cause: Successful TCP send client connection message. The local and remote tcpip addresses and ports are included in the message.

Effect: Information only.

Recovery: None necessary.

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