HP NonStop SSL Reference Manual

HP Part Number: 628203-009

Published: January 2014
Edition: HP NonStop SSL Reference Manual 1.8

H06.07 and subsequent H-series RVUs
J06.01 and subsequent J-series RVUs

Hewlett Packard Company
3000 Hanover Street
Palo Alto, CA 94304-1185

Contents

Preface

Who Should Read This Guide

This document is for system administrators who are responsible for configuring HP NonStop SSL to secure Telnet, FTP or middleware communication for ODBC, RSC and other protocols used by HP products.

Document History

Version 1.8

This version documents the changes and enhancements introduced with HP NonStop SSL AAI:

- New ODBC/MX IPv6 capabilities.
- Enhanced auditing, especially for FTPS and FTPC, see AUDITFILELAYOUT.
- Enhance EXPAND session setup plus configurable EXPAND encryption and compression.
- A few new SSLCOM commands (also see SSLCOM help output).
- Explicit warnings for usage of unsecure CIPHERSUITES.

Version 1.7

This version documents the changes introduced with HP NonStop SSL AAH

- Added documentation for newly introduced parameter ALLOWRENEGOTIATION
- Added documentation for new parameter DNSRESOLVERTCPIPPROCESSNAME

Version 1.6

- Added explicit warnings for usage of unsecure CIPHERSUITES.
- Added description of parameter DYNAMICROUTINGENABLEIPV6.

Version 1.5

This version documents the newly introduced IPv6 support and the corresponding parameters.

Version 1.4

This version clarifies the role of the remote proxy (RemoteProxy) in NonStop SSL. It is only supported for selected HP NonStop products.

Version 1.3

This version documents the support for configuring all available CIPHERSUITES. This feature is implemented starting with HP NonStop SSL version AAD.

- The new parameter HASHALGORITHMS has been documented.
- The changes in the TRUST parameter have been documented

Version 1.2

- The section about SSL Certificate Generation with OpenSSL was updated.
- The ODBC/MX install section was updated.

Version 1.1

This version documents the change in the CIPHERSUITES parameter. Preliminary support for additional ciphers was added and documented.

Version 1.0

This is the initial version of this manual.

Introduction

What is the Purpose of HP NonStop SSL?

HP NonStop SSL provides encryption of data which is sent or received by programs on HP NonStop servers over TCP/IP. It adds transport layer security to TCP/IP protocols without built-in support of SSL/TLS on HP NonStop, such as Telnet, FTP or ODBC.

HP NonStop SSL will run as a proxy server supporting the following modes of operation:

- [TELNETS] Acting as a secure proxy server for the NonStop TELSERV, to secure the communication between the NonStop system and a telnet client with built-in SSL, such as comForte's MR-Win6530, Crystal Point's OutsideView or Cail CTT.
- [PROXYS] Acting as a secure proxy server for plain TCP/IP servers acting as Server Gateways for Client/Server-Middleware, such as the HP NonStop RSC product, to secure the communication between the HP NonStop system and a client in conjunction with the RemoteProxy component included with HP NonStop SSL.

> **Note**: The usage of the NonStop SSL RemoteProxy component is supported for selected HP NonStop products only, including HP NonStop Remote Server Call (RSC/MP) and HP NonStop ODBC/MX.
> Further note that the NonStop RemoteProxy component does not support being run as a Microsoft Windows service.

- [PROXYC] Acting as a client proxy for plain TCP/IP client programs, to secure the communication between the NonStop Server and remote SSL-enabled server programs.
- [FTPS] Acting as a secure proxy server for plain FTP servers, such as the NonStop FTPSERV, to secure the communication between the NonStop system and a secure FTP client, such as MR-Win6530 or WS_FTP.
- [FTPC] Acting as a client proxy for the NonStop FTP client program, to secure the communication between the NonStop system and a SSL-enabled FTP server, such as the WS_FTP Server.
- [EXPANDS] Creating an SSL tunnel to secure EXPAND over IP lines.
- [ODBCMXS] Acting as a secure server proxy for the ODBC/MX protocol.

To support the above functions, HP NonStop SSL proxy processes can be started in different modes. These so-called "run modes" of a HP NonStop SSL proxy are listed in square brackets in the list above. Multiple HP NonStop SSL proxy processes can co-exist on a single NonStop system to support concurrent proxy services, as well as multiple TCP/IP processes.

The following table lists all run modes of an HP NonStop SSL proxy:

Run Mode	**Usage**
FTPC	FTP client proxy
FTPS	FTP server proxy
PROXYC	Generic SSL client proxy
PROXYS	Generic SSL server proxy
TELNETS	Secure Telnet proxy
EXPANDS	Secure EXPAND proxy
ODBCMXS	Secure ODBC/MX proxy

HP NonStop SSL Features

Support of SSL and TLS Protocol Standards

HP NonStop SSL uses SSL (Secure Socket Layer) in the TLS (Transport Layer Security) variant as standardized by the IETF in RFC 2246, to secure an application on the transport layer. SSL 2.0, SSL 3.0 and TLS 1.0 (SSL 3.1) are supported. It offers multiple configurable cipher suites with RSA key exchange with public key lengths up to 8192 bit (client authentication with max. 4096 bit RSA) and up to 256 bit AES for bulk encryption. Additionally Elliptic Curve Cryptography (ECC) with various available curves is supported.

Fault-Tolerance

HP NonStop SSL proxies can be configured as persistent processes, enabling automatic recovery from failures, such as CPU outages.

SSL-enabling for HP Client Components Running on Microsoft Windows Systems

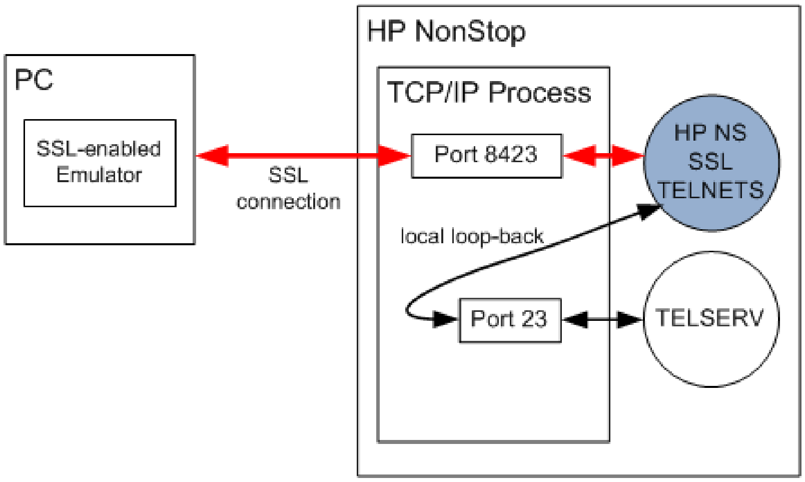
The RemoteProxy component included with HP NonStop SSL is used to enable SSL encryption for HP client components running on Microsoft Windows systems. Usage of the RemoteProxy component is supported for selected HP NonStop products only, including HP NonStop Remote Server Call (RSC/MP) and HP NonStop ODBC/MX. Additionally, the RemoteProxy can act as an SSL enabling LPD server proxy in order to secure LPD printing off the HP NonStop platform. Usage of the LPDS server mode is supported in combination with the Microsoft Windows platform only. Further note that the HP NonStop SSL RemoteProxy does not support being installed as a Windows service.

IPv6 Support

With HP NonStop SSL AAE, IPv6 support was introduced for the run modes whose plain counterpart protocols do support IPv6, in particular the PROXY and the FTP run modes. Please see section "IPv6 considerations" for details.

Secure Proxy for Telnet Access

HP NonStop SSL can be run as a proxy process to front-end TCP/IP servers accepting plain TCP connections, such as the NonStop TELSERV process, enabling secure communication to clients, which also support the SSL protocol. SSL capable clients are, for example, comForte's MR-Win6530 and J6530, Crystal Point's OutsideView and Cail's CTT.



HP NonStop SSL proxy front-ending the NonStop TELVERV process

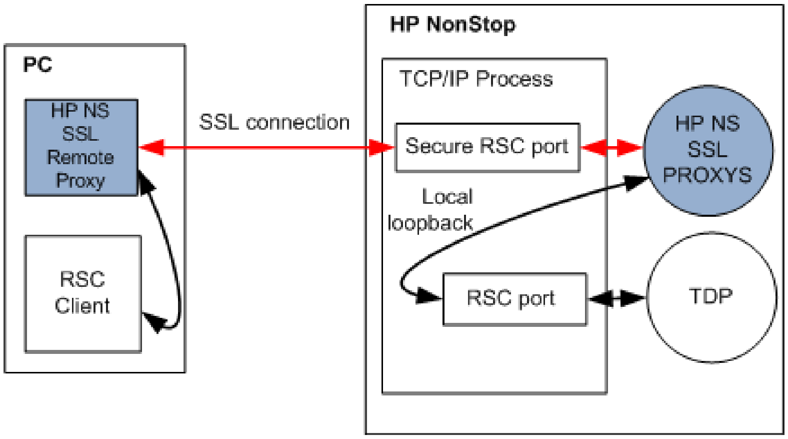
The HP NonStop SSL proxy will accept SSL connections from the network and "tunnel" them to a plain TCP server. Encrypted data received from the SSL client will be decrypted and forwarded to the server. Plain data received from plain TCP server will be encrypted and sent to the SSL client. For example, from the Telnet server's point of view, the proxy acts as a normal Telnet client, while from an SSL telnet client the HP NonStop SSL proxy authenticates the Telnet server and encrypts/decrypts the session's payload.

Typically, a HP NonStop SSL proxy will reside on the same IP process on the same system as the TCP server it tunnels the session to, which allows to create a "local loopback" session (a connection to "127.0.0.1" for IPv4, respectively "::1" for IPv6) for the unencrypted data. This avoids that any unencrypted data has to traverse the network. For a local loopback, the data is only being passed within the local TCP/IP stack.

One instance of a HP NonStop SSL proxy handles multiple SSL connections received on a single IP process and port number and tunnels them to a single target port. If multiple plain ports need to be secured, such as multiple Telnet Servers, a HP NonStop SSL process can be started for each plain TCP port.

Secure Proxy for Generic TCP/IP Client/Server Protocols

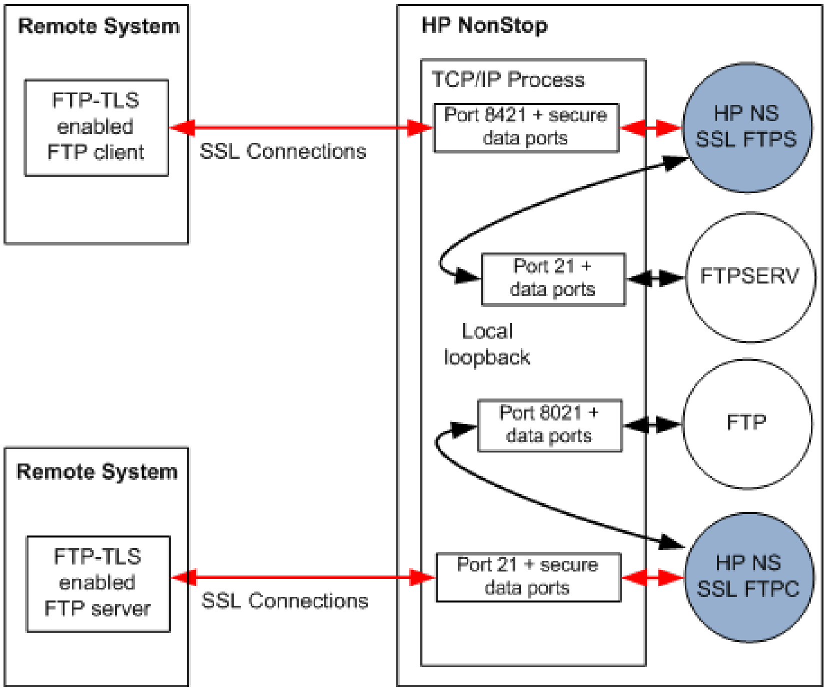
HP NonStop SSL adds encryption not only for Telnet but for any Client/Server protocol facilitating TCP sockets communicating over a single IP port. HP NonStop SSL can act as a secure proxy for the server or client side of the client/server communication. If required, the RemoteProxy component included with HP NonStop SSL can be used to enable SSL encryption for HP client components running on Microsoft Windows systems. Usage of the RemoteProxy component is supported for selected HP NonStop products only, including HP NonStop Remote Server Call (RSC/MP) and HP NonStop ODBC/MX. Additionally, the RemoteProxy can act as an SSL enabling LPD server proxy in order to secure LPD printing off the HP NonStop platform. Usage of the LPDS server mode is supported in combination with the Microsoft Windows platform only. Further note that the HP NonStop SSL RemoteProxy does not support being installed as a Windows service. For details regarding the differences between the HP NonStop SSL and comForte SecurCS please see HP SAW document "Differences between HP NonStop SSL (T0910) and SecurCS. " (mmr_ns-0102006).



HP NonStop SSL securing Remote Server Call (RSC) communication

Secure FTP Proxy

HP NonStop SSL can be run as a proxy process to front-end the NonStop FTPSERV or FTP process. With its SSL support, HP NonStop SSL will enable secure communication to FTP clients or servers, which support FTP over SSL/TLS according to RFC-2228. SSL capable FTP clients are, for example, MR-Win6530 or WS_FTP Pro from http://www.ipswitch.com/.



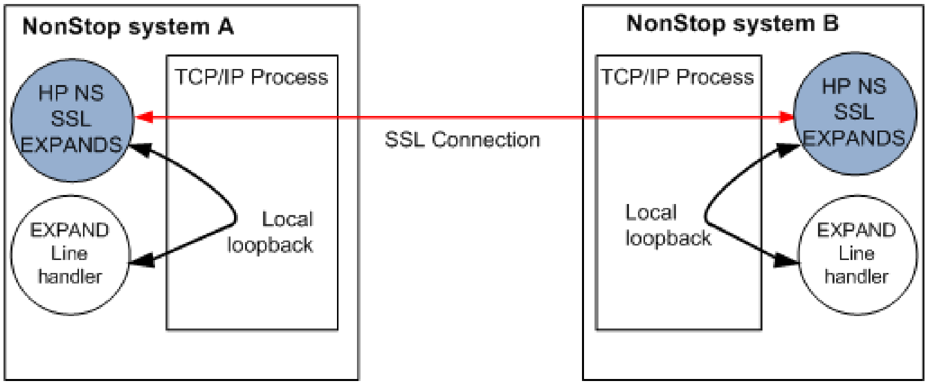
HP NonStop SSL secure FTP proxies front-ending standard FTP and FTPSERV

Acting as a proxy server, HP NonStop SSL will use secure FTP connections with the FTP partner and "tunnel" them to a plain FTP client or server.

The HP NonStop SSL FTPS proxy will intercept the communication on the FTP command socket to add encryption for both the command and data sockets. From the FTP server's or client's point of view the proxy acts as a normal FTP partner, while for the remote SSL FTP partner the HP NonStop SSL proxy acts as a RFC-2228 compliant secure FTP server or client.

Secure Proxy for EXPAND-over-IP

HP NonStop SSL running in EXPANDS mode encrypts EXPAND over IP traffic between two NonStop systems. It does so by creating a secure SSL session between the two systems as depicted in the following diagram:



HP NonStop SSL as a proxy for EXPAND over IP traffic

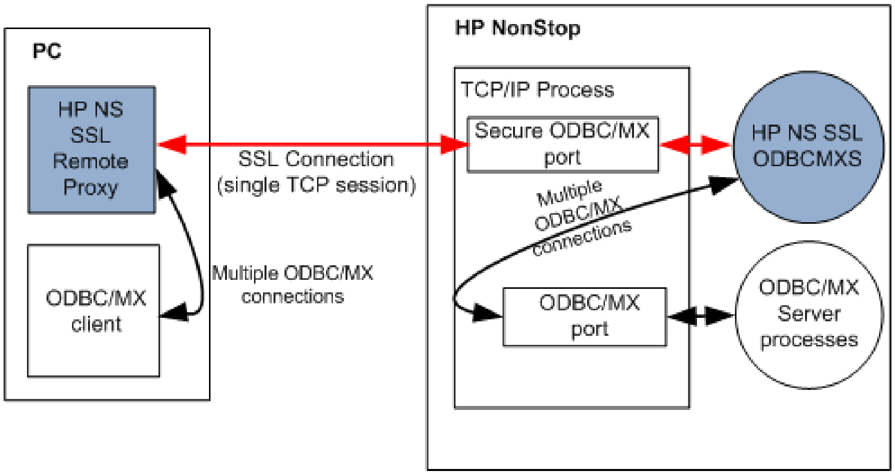
The EXPAND line handler will exchange UDP traffic with an instance of HP NonStop SSL running on the same NonStop system; the two HP NonStop SSL processes create an SSL TCP session between the two systems to forward the traffic.

Secure Proxy for ODBC Drivers

HP NonStop SSL can encrypt traffic between an ODBC driver (ODBC/MP, OBDC/MX, JDBC/MP and JDBC/MX) on client workstations and NonStop systems.

Since ODBC/MP only uses a single TCPIP session, it can be enabled for SSL as described under "Secure Proxy for Generic TCP/IP Client/Server Protocols".

In contrast, ODBC/MX, JDBC/MP and JDBC/MX use multiple TCP/IP sessions with different port numbers between a single client and the NonStop system. However in conjunction with the RemoteProxy component on the Windows client, HP NonStop SSL "tunnels" multiple sessions over a single one as shown in the following diagram:



HP NonStop SSL as a proxy for ODBC/MX traffic

The "tunneling" approach has the following benefits:

- It is firewall-friendly, as only a single port needs to be opened between the workstations and the clients.

- The configuration both of the HP NonStop SSL ODBCMXS process and the RemoteProxy is independent of the number of ports used by ODBC/MX.

Note: The ODBC/MX protocol supports IPv6 starting with release H06.26/J06.15, but running HP NonStop SSL in ODBCMXS mode is currently only valid with IPMODE IPv4.

Limiting Remote IP Addresses

HP NonStop SSL can be configured to allow only certain remote IP addresses. By default, HP NonStop SSL will allow connections from any IP address; this behavior can be changed by

1. Setting a "black list" of forbidden IP addresses or subnets using the DENYIP parameter.
2. Setting a "white list" of allowed IP addresses or subnets using the ALLOWIP parameter.

Note: the black list will take precedence over the white list: if an IP address is matching both lists, it will NOT be allowed.

For details, please refer to parameters "DENYIP" and "ALLOWIP" in the "Parameter Reference".

Installation

General Considerations

HP NonStop SSL is made available by HP with the purchase of the NonStop Operating System kernel for H Series and J Series NonStop platforms. HP NonStop SSL was introduced as SPR T0910 in H06.21/J06.10 and is not available on S-series. The files of the package are located on $SYSTEM.ZNSSSL.

HP NonStop SSL is not pre-installed or pre-configured. You have to install it depending on your requirements. A license file is not required.

The main executable file is named SSLOBJ, which can be run to create an SSL proxy process running in a specific run mode. While you can manually create SSL proxy processes with the TACL run command, it is recommended to create a persistent process under control of the Kernel subsystem.

For convenience, HP NonStop SSL includes a SETUP macro, which helps you create an initial configuration for a persistent proxy process in one of the available run modes. You may fine tune the configuration by editing the configuration files created by the SETUP macro.

Note: Specific attention needs to be paid to a proper SSL configuration. HP NonStop SSL is delivered with a set of sample SSL certificate and key files which are used by default. For a production installation, you should use your own SSL server certificate. Please refer to the "SSL Reference" chapter for details.
When replacing the certificate files delivered in $system.znsssl with production certificates they may be overwritten by DSM/SCM and restored to the original ones. Therefore it is recommended to place the production certificates in a separate volume and point to those files in a CONFIG2 configuration file.

The installation subvolume znsssl also contains a Tacl macro named CFWSADDR. This macro provides the real client IP address of a Visual Inspect session when connected to a NonStop SSL TELNETS process. The best way to install the CFWSADDR macro is to include it in the TACLLOCL file so it gets executed for every new TACL session started:

```
LOAD/KEEP 1/$SYSTEM.ZNSSSL.CFWSADDR
WSADDR
```

Note that the invocation of this macro is WSADDR, not CFWSADDR. The macro searches all process occurrences of the SSLOBJ file (NonStop SSL) and also the SWAP file (comForte SecurCS).

For securing some protocols, such as ODBC or RSC, you will also need to install the HP NonStop SSL RemoteProxy, which will enable SSL for the HP components running on a remote user workstation.

Note that usage of the RemoteProxy component is supported for selected HP NonStop products only, including HP NonStop Remote Server Call (RSC/MP) and HP NonStop ODBC/MX. Additionally, the RemoteProxy can act as an SSL enabling LPD server proxy in order to secure LPD printing off the HP NonStop platform. Usage of the LPDS server mode is supported in combination with the Microsoft Windows platform only. Further note that the HP NonStop SSL RemoteProxy does not support being installed as a Windows service.

IPv6 Considerations

With HP NonStop AAE, IPv6 support was introduced. The new parameter IPMODE was introduced for this purpose:

```
IPMODE {IPv4|IPv6|DUAL}
```

If not specified, the IPMODE parameter will default to IPv4. When IPMODE DUAL is specified, SSLOBJ will listen to both IPv4 and IPv6 with one single dual mode socket. In IPMODE DUAL, IPv4 addresses will be shown as mapped IP addresses with the corresponding prefix "::ffff:" for this purpose, e.g. "::ffff:10.10.10.110".

Note the following considerations in respect to IPv6 support:

- ODBC/MX in IPv6 mode is only supported by HP NonStop SSL release AAI (Jan 2014) or later.
- The SOCKS4 protocol does not support IPv6 by design, thus specifying SOCKS4 protocol parameter (SOCKSHOST, SOCKSPORT, SOCKSUSER) is only valid in IPMODE IPv4.
- Following recommendation of the IETF, IPv6 support for FTP was implemented according to RFC 2428 (EPSV, EPRT). The FOOBAR protocol (LPSV, LPRT) is not supported.
- IPMODE DUAL is not supported by design in runmode EXPANDS (use either IPMODE IPv4 or IPMODE IPv6 instead)
- By design when SSLOBJ is run in IPMODE DUAL, the TCP/IP stack must also support and be configured in DUAL mode.
- Setting INTERFACE or TARGETINTERFACE is not valid in IPMODE DUAL, since no bind address except the IPv6 ANY address '::' can handle both IPv4 and IPv6.
- IP version transition (4to6 or 6to4) is *not* one of the main intended application areas of SSLOBJ. Although IP version transition is partially possible with IPMODE DUAL, support for it is limited. Contact HP support for further information on supported setups.
- When specifying an IP address in IPv6 representation followed by a port number, the IPv6 address must be embraced by square brackets to avoid ambiguity. E.g. in FTPC mode the user name has to be specified as follows:

  ```
  john@[2001:db9::1421:51ab]:11013
  ```
- Same requirement for enclosing the IPv6 address with square brackets applies to specification of DENYIP and ALLOWIP values in CIDR format in order to avoid ambiguity with the potentially leading direction character (See parameter description of ALLOWIP/DENYIP for more details)

Starting an HP NonStop SSL Process

You can start a HP NonStop SSL process by issuing a TACL RUN command using the following syntax:

```
RUN SSLOBJ / runoptions / mode [ ; paramname paramvalue; ... ]
```

where

- *runoptions* are the standard Guardian RUN options, such as IN, CPU or TERM
- *mode* defines the run mode of the HP NonStop SSL process with the following valid keywords:

FTPC	**FTP client proxy**
FTPS	FTP server proxy
PROXYS	Generic SSL server proxy
PROXYC	Generic SSL client proxy
TELNETS	Secure Telnet proxy
EXPANDS	Secure EXPAND over IP proxy
ODBCMXS	Secure Proxy for ODBC/MX

- *paramname paramvalue*; ...
  is a list of HP NonStop SSL configuration parameter settings as described in the "Parameter Reference".

Note: When you start a HP NonStop SSL process in NOWAIT mode, make sure you have disabled logging to the home terminal. To do so, set the parameter LOGCONSOLE to *.

Installing a Secure Telnet Server Proxy

To encrypt Telnet sessions with the standard NonStop TELSERV process and an SSL-enabled Telnet client, you will need to perform the following steps:

1. On the NonStop server, start a HP NonStop SSL telnet server (TELNETS) proxy for the target TELSERV process.
2. On the workstation side, re-configure your telnet client to connect via SSL to the port number that the TELNETS proxy is listening on.

To install an HP NonStop SSL TELNETS proxy

1. Determine the Telnet server you want to install the secure proxy for and find out the TCP/IP process and port number it is listening on (usually 23).
2. Select a port number that will be used for SSL telnet connections (e.g. 8423).
3. At your TACL prompt, run the HP NonStop SSL SETUP macro:

   ```
   > VOLUME $SYSTEM.ZNSSSL
   > RUN SETUP
   ```

   Select "TELNET SERVER" as run mode and follow the installation instructions. Enter the port number of the TELSERV listening port as target port (e.g. 23) and the selected SSL telnet port as listening port (e.g. 8423).

   The SETUP macro will create a configuration file (e.g. TLNSCF0) and an SCF IN file for the installation as persistent process (e.g. TLNSIN0).
4. Edit the HP NonStop SSL TELNETS configuration file (e.g. TLNSCF0) to configure any additional parameters, if desired.

5. Install the TELNETS proxy persistent process, e.g.

```
> SCF /IN TLNSIN0/
```

6. Start the TELNETS proxy persistent process, e.g.

```
> SCF START PROCESS $ZZKRN.#SSL-TELNETS-0
```

7. Check the log file (configured in the configuration file) to verify the TELNETS proxy has started correctly, e.g.

```
> SHOWLOG TLNSLOG *
```

Verify that the log contains a message of the following pattern:

```
$TLNS0|06Jun10 21:42:15.82|20|secure-to-plain proxy started on target host 127.0.0.1,
target port 23, source port 8423
```

When logging with default log level 50, the last message of the log should then be similar to the following:

```
$ZTLNS0|29Jul12 16:31:29.37|30|-- PROXYS setup completed, starting to listen... --
```

To create a secure connection with a secure Telnet client

1. Configure your SSL Telnet client to connect to the address and port number the HP NonStop SSL secure telnet proxy listens for incoming connections. Make sure that the client has the SSL protocol enabled for the session.

Installing a Secure FTP Server Proxy

To encrypt FTP sessions with the standard NonStop FTP server and an FTP client with FTP-TLS (SSL) support, you will need to perform the following steps:

1. On the NonStop server, start an HP NonStop SSL ftp server (FTPS) proxy for the target FTP server.
2. On the remote system, configure your FTP client to connect via SSL to the port number that the FTPS proxy is listening on.

To install an HP NonStop SSL FTPS proxy

1. Determine the TCP/IP process and port number the NonStop LISTNER process it is listening for FTP sessions (usually 21).
2. Select a port number that will be used for FTP-TLS connections (e.g. 8421).
3. At your TACL prompt, run the HP NonStop SSL SETUP macro:

```
> VOLUME $SYSTEM.ZNSSSL
> RUN SETUP
```

Select "FTP SERVER" as run mode and follow the installation instructions. Enter the port number of the FTP listening port as target port (e.g. 21) and the selected FTP-TLS port as listening port (e.g. 8421).

The SETUP macro will create a configuration file (e.g. FTPSCF0) and an SCF IN file for the installation as persistent process (e.g. FTPSIN0).

4. Edit the HP NonStop SSL FTPS configuration file (e.g. FTPSCF0) to configure any additional parameters, if desired.
5. Install the FTPS proxy persistent process, e.g.

```
> SCF /IN FTPSIN0/
```

6. Start the FTPS proxy persistent process, e.g.

```
> SCF START PROCESS $ZZKRN.#SSL-FTPS-0
```

7. Check the log file (configured in the configuration file) to verify the FTPS proxy has started correctly, e.g.

```
> SHOWLOG FTPSLOG *
```

Verify that the log contains a message of the following pattern:

```
$FTPS0|18May10 20:22:51.63|20|FTP server proxy started on target host 127.0.0.1, target
port 21, source port 8421
```

When logging with default log level 50, the last message of the log should then be similar to the following:

```
$FTPS0|27Jul12 16:14:55.41|30|-- FTPS setup completed, starting to listen... --
```

To create a secure connection with an FTP-TLS enabled FTP client

1. Configure your FTP client to connect to the address and port number the HP NonStop SSL secure FTPS proxy listens for incoming connections. Make sure that the client has the FTP-TLS protocol enabled for the session.

Installing a Secure FTP Client Proxy

To encrypt FTP sessions with the standard NonStop FTP client and a FTP server, you will need to perform the following steps:

1. On the NonStop server, start a HP NonStop SSL ftp client (FTPC) proxy.
2. On the partner system, use a "FTP-TLS" compliant server to receive connections from the FTPC proxy, such as the WS-FTP Server from Ipswitch, Inc.

To send or receive files securely to/from the remote system, you will use the standard NonStop FTP client. You may also use an application that uses the NonStop FTP client API. Instead of connecting directly to the remote system, you will first connect to the HP NonStop SSL FTPC proxy. Using an extended user id that includes information on the host address and port number of the remote FTP system you will instruct the FTPC proxy to connect securely to the remote FTP server. From there on, you may proceed as with normal plain FTP to list directories, as well as to send or receive files.

To install an HP NonStop SSL FTPC proxy

1. Select a port number that the HP NonStop SSL FTPC proxy will use for plain connections from local FTP clients (e.g. 8021).
2. At your TACL prompt, run the HP NonStop SSL SETUP macro:

   ```
   > VOLUME $SYSTEM.ZNSSSL
   > RUN SETUP
   ```

   Select "FTP CLIENT" as run mode and follow the installation instructions. Enter the selected port number as listening port (e.g. 8421).

   The SETUP macro will create a configuration file (e.g. FTPCCF0) and an SCF IN file for the installation as persistent process (e.g. FTPCIN0).
3. Edit the HP NonStop SSL FTPC configuration file (e.g. FTPCCF0) to configure any additional parameters, if desired.
4. Install the FTPC proxy persistent process, e.g.

   ```
   > SCF /IN FTPCIN0/
   ```
5. Start the FTPC proxy persistent process, e.g.

   ```
   > SCF START PROCESS $ZZKRN.#SSL-FTPC-0
   ```
6. Check the log file (configured in the configuration file) to verify the FTPC proxy has started correctly, e.g.

   ```
   > SHOWLOG FTPCLOG *
   ```

   Verify that the log contains a message of the following pattern:

   ```
   $FTPC0|18May10 20:22:51.63|20|FTP client proxy started on source port 8021
   ```

   When logging with default log level 50, the last message of the log should then be similar to the following:

```
$FTPC0|29Jul12 16:38:40.45|30|-- FTPC setup completed, starting to listen... --
```

To create a secure FTP connection to a remote FTP-TLS server

1. Issue the following command at the command prompt:

   ```
   > FTP localhost 8021
   ```

   where

   - the first parameter denotes the local loopback address
   - the second parameter specifies the port number the HP NonStop SSL FTPC proxy is listening on

   The HP NonStop SSL FTP client mode welcome message will now be displayed. You will be prompted for user-id and password:

   ```
   FTP Client - T9552H02 - (10JUL2009) - COPYRIGHT TANDEM COMPUTERS INCORPORATED 20
   09
   Connecting to 127.0.0.1.........Established.
   220 HP NonStop SSL version T0910H01_19JUL2010 running in encrypting FTP client m
   ode
   Name (127.0.0.1:user):
   ```

2. At the user id prompt, enter the following data:

   ```
   <user id>@<remote address>[:<port>]
   ```

   where

   - <user id> is the user name valid to login to the remote secure FTP server.
   - <remote address> is the IP address or DNS name of the remote system where the secure FTP server is running on.

     Note: if the remote address is an IPv6 address it has to be surrounded by square brackets. (E.g. john@[fe80:aa::bb42]:12345)

   - <port> is the port number the remote FTP server is listening on. If omitted, 21 is used as a default.

   The connection should now be established, allowing you to list directories and transfer files securely:

   ```
   Name (127.0.0.1:user): tb@172.24.91.233
   331- original FTP server Welcome follows
   331- 220 NOTEBOOK_TB X2 WS_FTP Server 3.1.4 (3995038631)
   331- original FTP server reply to USER command follows
   331 Password required
   Password:*****
   230 user logged in
   ftp> dir
   200 command successful
   150 Opening ASCII data connection for directory listing
   drwxr-x---  2 tb       System            0 Oct  1 19:17 .
   drwxr-x---  2 tb       System            0 Oct  1 19:17 ..
   -rwxr-x---  1 tb       System          161 Dec 12 12:17 l1
   -rwxr-x---  1 tb       System          161 Dec 12 12:17 l2
   -rwxr-x---  1 tb       System          161 Dec 12 12:17 l3
   -rwxr-x---  1 tb       System      1447718 Dec 12 12:20 testfile
   226 transfer complete
   496 bytes received in  0.07 seconds ( 6.92 Kbytes/s)
   ftp>
   ```

Note: Starting HP NonStop SSL version AAE, an FTPC default host and FTPC default port can be specified by using the parameters TARGETHOST and TARGETPORT. The respective parameter values will be taken into account if the user does not specify the corresponding value - or - if HP NonStop SSL was configured to always use the values of TARGETHOST respectively TARGETPORT due to the additional parameter TARGETHOSTFORCE or TARGETPORTFORCE. Please see corresponding parameter description for details.

Installing a Secure Tunnel for RSC

To install an SSL tunnel for Remote Server Call (RSC) communication, you will need to perform the following steps:

1. On the NonStop server, install an HP NonStop SSL generic server proxy (PROXYS) process for the target TDP server process.
2. On the workstation, install the HP NonStop SSL RemoteProxy and configure it to route plain connections to the PROXYS process on the NonStop server.
3. Re-configure RSC to connect to the local RemoteProxy.

The following implementation instructions assume that you have RSC installed on your target NonStop system and workstation.

To install a HP NonStop SSL PROXYS process for RSC

1. Determine the RSC Transaction Delivery Process (TDP) you want to install the secure proxy for and find out the TCP/IP process and port number it is listening on. You may do this by examining the TDPCFG file for SET TCPIPPORT PROCESSNAME and ADD TCPIPPORT commands. You may also check the TCPIPPORT object with RSCCOM as in the following example:

   ```
   53> RSCCOM
   RSCCOM - TDP Configuration Manager - T9711D43 - (05NOV96) – System \SUPPORT
   Tandem TM Remote Server Call using technology from Cornerstone Software, Inc.
   Copyright (c) Cornerstone Software, Inc. 1991 - 1995. All rights reserved.
   1 (( open $zrsc
   Current TDP is \SUPPORT.$ZRSC - T9711D430 - (05NOV96)
   2 (( status tcpipport *
   Service (Port)    Status   Sessions   Last Event
   -------------------------------------------------------------------
   RSCTEST1 (2001)   Started     0       TCPIPPORT started. [ 6502 ]
   ```

2. Select a port number that will be used for SSL RSC connections (e.g. 7502)
3. At your TACL prompt, run the HP NonStop SSL SETUP macro:

   ```
   > VOLUME $SYSTEM.ZNSSSL
   > RUN SETUP
   ```

   Select "GENERIC SERVER" as run mode and follow the installation instructions. Enter the port number of the TDP server as target port (e.g. 6502) and the selected SSL RSC port as SSL listening port (e.g. 7502).

   The SETUP macro will create a configuration file (e.g. PXYSCF0) and an SCF IN file for the installation as persistent process (e.g. PXYSIN0).

4. Edit the HP NonStop SSL PROXYS configuration file (e.g. PXYSCF0) to configure any additional parameters, if desired.
5. Install the PROXYS proxy persistent process, e.g.

   ```
   > SCF /IN PXYSIN0/
   ```

6. Start the HP NonStop SSL PROXYS persistent process, e.g.

   ```
   > SCF START PROCESS $ZZKRN.#SSL-PROXYS-0
   ```

7. Check the log file (configured in the configuration file) to verify the PROXYS process has started correctly, e.g.

   ```
   > SHOWLOG PXYSLOG *
   ```

   Verify that the log contains a message of the following pattern:

   ```
   $PXYS0|06Jun10 21:42:15.82|20|secure-to-plain proxy started on target host 127.0.0.1,
   target port 6502, source port 7502
   ```

   When logging with default log level 50, the last message of the log should then be similar to the following:

   ```
   $PXYS0|29Jul12 16:31:29.37|30|-- PROXYS setup completed, starting to listen... --
   ```

To install and configure RemoteProxy for RSC

1. Download $SYSTEM.ZNSSSL.PROXYEXE in binary format to your RSC workstation, renaming it to PROXY.EXE.
2. On the RSC workstation, run PROXY.EXE to start the RemoteProxy installation program and follow the installation instructions.
3. Double-click on HP NonStop SSL RemoteProxy icon in your system tray. The "RemoteProxy" configuration window will be displayed.
4. Select "New" from the "Session" menu. The "Session Properties" dialog will be displayed.
5. In the "Protocol" field, select "Generic TCP/IP".
6. In the "Target Host" field, enter the IP address or host name where your PROXYS process is listening on your NonStop server.
7. In the "Target Port" field, enter the port number, you have specified as the listen port of your PROXYS process on the NonStop server.
8. In the "Local (Accepting) Port" field, enter the port number that RemoteProxy will use to listen for connections from your RSC client. The port number must not be in use by any other program or service on your client PC. For simplicity, you may want to use the same port number that the plain TDP server process is using on the NonStop server side, e.g. "6502" in the example above.
9. Start the RemoteProxy session by clicking on the "Start" button
10. If the start is successful, check the startup messages with the "View Log" command.

To configure RSC to connect via the RemoteProxy

1. On the RSC workstation, locate the PIPE.INI file that is used by HP Piccolo.
2. In the PIPE.INI file, add an entry for your relevant RemoteProxy session in the [Resolver] section. The entry itself assigns an alias host name (1st argument) for a connection over a specified protocol (2nd argument) to a given peer. To communicate with RemoteProxy "ip" has to be used as the protocol (2nd argument), followed by the local host name and the value you specified as "Local Port" in the "Session Properties" of the relevant RemoteProxy Session. For example, a valid entry (with local port = 6502) could be:

   ```
   [Resolver]

   myhost=ip:127.0.0.1.6502
   ```

3. To prevent that both RemoteProxy and Piccolo are using the same port (configured in step 2), add an additional entry in PIPE.INI in which you specify an unused port (e.g. 1277) to be used by Piccolo on the client. For instance, as follows:

   ```
   [NIF-mynifsock]

   ProgramFile=nifsock

   ServicePort=1277
   ```

4. On the RSC workstation, locate the RSC.INI file that is used by the RSC transport process.
5. Edit the RSC.INI file, add a "host_pipename" entry referring to the alias host name you chose in PIPE.INI in step 2. For example, a valid entry could be:

   ```
   host_pipename = RSC@myhost
   ```

6. Restart the RSC Transport Process.
7. You may use the RSCTEST program to test the secure RSC connection to the NonStop system.
8. You may check the successful creation of the session through the proxy by examining the messages with the "View Log" command in the "Session Properties" screen of the RemoteProxy.

To connect securely with your RSC client

1. After you have correctly configured the RSC.INI file and started the RemoteProxy session for RSC, use your RSC client like you did before to connect to the NonStop system.
2. You may check the successful creation of the session through the proxy by examining the messages with the "View Log" command in the "Session Properties" screen of RemoteProxy.

Installing a Secure Tunnel for ODBC/MP

Note: The configuration of HP NonStop SSL for ODBC/MX differs from the configuration for ODBC/MP. This section describes the configuration of HP NonStop SSL for ODBC/MP; please see the next section for the ODBC/MX configuration.

To implement HP NonStop SSL to encrypt an Open Database Connectivity ODBC/MP connection, you will need to perform the following steps:

1. On the NonStop server, install an HP NonStop SSL generic server proxy (PROXYS) process for the target ODBC server process.
2. On the workstation, install RemoteProxy and configure it to route plain ODBC/MP connections to the HP NonStop SSL PROXYS process.
3. Re-configure the ODBC/MP driver on your workstation to connect to RemoteProxy.

To install an HP NonStop SSL PROXYS process for ODBC/MP

1. Determine the ODBC/MP server process you want to install the secure proxy for and find out the TCP/IP process and port number it is listening on. We assume 8889 as port number here.
2. Select a port number that will be used for SSL ODBC/MP connections (e.g. 9889).
3. At your TACL prompt, run the HP NonStop SSL SETUP macro:

   ```
   > VOLUME $SYSTEM.ZNSSSL
   > RUN SETUP
   ```

   Select "GENERIC SERVER" as run mode and follow the installation instructions. Enter the port number of the ODBC/MP server as target port (e.g. 8889) and the selected SSL ODBC/MP port as SSL listening port (e.g. 9889).

   The SETUP macro will create a configuration file (e.g. PXYSCF0) and an SCF IN file for the installation as persistent process (e.g. PXYSIN0).
4. Edit the HP NonStop SSL PROXYS configuration file (e.g. PXYSCF0) to configure any additional parameters, if desired.
5. Install the PROXYS proxy persistent process, e.g.

   ```
   > SCF /IN PXYSIN0/
   ```
6. Start the HP NonStop SSL PROXYS persistent process, e.g.

   ```
   > SCF START PROCESS $ZZKRN.#SSL-PROXYS-0
   ```
7. Check the log file (configured in the configuration file) to verify the PROXYS process has started correctly, e.g.

   ```
   > SHOWLOG PXYSLOG *
   ```

   Verify that the log contains a message of the following pattern:

   ```
   $PXYS0|06Jun10 21:42:15.82|20|secure-to-plain proxy started on target host 127.0.0.1,
   target port 8889, source port 9889
   ```

   When logging with default log level 50, the last message of the log should then be similar to the following:

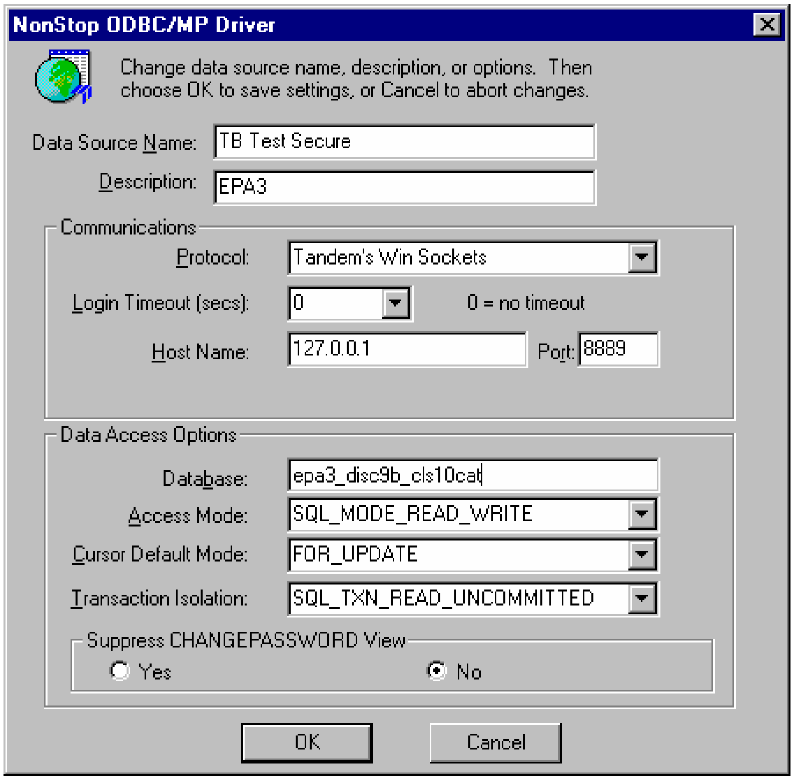
```
$PXYS0|29Jul12 16:31:29.37|30|-- PROXYS setup completed, starting to listen... --
```

To install and configure RemoteProxy for ODBC/MP

1. Download $SYSTEM.ZNSSSL.PROXYEXE in binary format to your OCBC/MP client workstation, renaming it to PROXY.EXE.
2. On the OCBC/MP client workstation, run PROXY.EXE to start the RemoteProxy installation program and follow the installation instructions.
3. Double-click on HP NonStop SSL RemoteProxy icon in your system tray. The "RemoteProxy" configuration window will be displayed.
4. Select "New" from the "Session" menu. The "Session Properties" dialog will be displayed.
5. In the "Protocol" field, select "Generic TCP/IP".
6. In the "Target Host" field, enter the IP address or host name where your PROXYS process is listening on the NonStop server.
7. In the "Target Port" field, enter the port number you have specified as the listen port of your PROXYS process on the NonStop server.
8. In the "Local (Accepting) Port" field, enter the port number that RemoteProxy will use to listen for connections from your ODBC/MP driver. The port number must not be in use by any other program or service on your client PC. For simplicity, you may want to use the same port number that the ODBC/MP server process is using on the NonStop server side, e.g. "8889" in this example.
9. Start the RemoteProxy session by clicking on the "Start" button.
10. If the start is successful, check the startup messages with the "View Log" command.

To configure the ODBC/MP driver to connect via the RemoteProxy

1. Navigate to the ODBC driver configuration dialog.
2. Reconfigure the Host Name to the local host address, e.g. "localhost", or "127.0.0.1"
3. Reconfigure the "Port" field to the port number you specified as "Local Port" in the "Session Properties" of the relevant RemoteProxy Session.
4. Assuming you used "8889" as "Local (Accepting) Port" in the RemoteProxy session configuration, your ODBC driver configuration dialog should look as follows:



Change data source name, description, or options. Then choose OK to save settings, or Cancel to abort changes.

Data Source Name: TB Test Secure

Description: EPA3

Communications

Protocol: Tandem's Win Sockets

Login Timeout (secs): 0   0 = no timeout

Host Name: 127.0.0.1   Port: 8889

Data Access Options

Database: epa3_disc9b_cls10cat

Access Mode: SQL_MODE_READ_WRITE

Cursor Default Mode: FOR_UPDATE

Transaction Isolation: SQL_TXN_READ_UNCOMMITTED

Suppress CHANGEPASSWORD View

○ Yes   ● No

Confirm the changes by clicking "OK".

5. You may use the NonStop Connectivity Tool to test the secure ODBC connection to the NonStop system.

6. You may check the successful creation of the session through the proxy by examining the messages with the "View Log" command in the "Session Properties" screen of the RemoteProxy.

To connect securely with your ODBC/MP client

1. After you have correctly configured your ODBC driver, use your ODBC client like you did before to connect to the NonStop system.

2. You may check the successful creation of the session through the proxy by examining the messages with the "View Log" command in the "Session Properties" screen of the RemoteProxy.

Installing a Secure Tunnel for ODBC/MX

Note 1: The configuration for ODBC/MX differs from the configuration for ODBC/MP. This section describes the configuration for ODBC/MX; please see the prior section for the configuration for ODBC/MP.
Note 2: NonStop ODBC/MX uses multiple port numbers to create connections between the ODBC/MX clients and the NonStop server. HP NonStop SSL is aware of that and "multiplexes" many connections over a single IP connection between the clients and the NonStop server. That has two benefits:
- only a single port needs to be open at the firewall.
- the configuration of HP NonStop SSL becomes easier.

To install HP NonStop SSL to encrypt an Open Database Connectivity ODBC/MX connection, you will need to perform the following steps:

1. On the NonStop server, install an HP NonStop SSL ODBC/MX server proxy (ODBCMXS) process for the target ODBC/MX server process.
2. On the workstation, install RemoteProxy and configure it to route plain ODBC/MX connections to the HP NonStop SSL ODBCMXS process.
3. Re-configure the ODBC/MX driver on your workstation to connect to RemoteProxy.

To Install an HP NonStop SSL ODBCMXS process for ODBC/MX

1. Determine the ODBC/MX server process you want to install the secure proxy for and find out the TCP/IP process and port number it is listening on. Note that ODBC/MX consists of multiple server processes; you should look for the port number of the ODBC/MX Association server. This is the MXCS port number you configure in the ODBC/MX client configuration (only!). We will assume a value of 18888 here.
2. Select a port number that will be used for SSL ODBC/MX connections, e.g. 28888
3. At your TACL prompt, run the HP NonStop SSL SETUP macro:

   ```
   > VOLUME $SYSTEM.ZNSSSL
   > RUN SETUP
   ```
4. Select "ODBC/MX SERVER" as run mode and follow the installation instructions.

   Enter the TCPIP process name for the subnet the ODBC/MX Association server runs on. Note that the SUBNET and TARGETSUBNET parameters will be set to the process name you provided here. Next, enter the listening port number as determined in (2) above for incoming SSL-encrypted ODBC/MX client connections. Note that you will not be prompted for a TARGETPORT because it will be determined automatically based on the client side configuration.

   Finally the SETUP macro will create a configuration file (e.g. ODBSCF0) and an SCF IN file for the installation as persistent process (e.g. ODBSIN0).
5. Edit the HP NonStop SSL ODBCMXS configuration file (e.g. ODBSCF0) to configure any additional parameters, if desired. Be careful with the additional parameter "TARGETHOST" that can be used to route outgoing traffic to another host.

   For security reasons, you should specify the "local loopback address" (127.0.0.1) as TARGETHOST since this avoids that unencrypted data traverses the network. The TARGETHOST parameter will default to "127.0.0.1" if omitted. In some cases it might be desired to handle incoming connections (originating e.g. from RemoteProxy) on a specific subnet and route the outgoing connections (to the ODBC/MX Association server) to another subnet. In that case you can set the SUBNET (incoming) and TARGETSUBNET (outgoing) parameter value to the respective process name. If TARGETSUBNET is omitted it defaults to the value of SUBNET.
6. Install the ODBCMXS proxy persistent process, e.g.

   ```
   > SCF /IN ODBSIN0/
   ```
7. Start the HP NonStop SSL ODBCMXS persistent process, e.g.

```
> SCF START PROCESS $ZZKRN.#SSL-ODBCMXS-0
```

8. Check the log file (configured in the configuration file) to verify the ODBCMXS process has started correctly, e.g.

   ```
   > SHOWLOG ODBSLOG *
   ```

   Verify that the log contains a message of the following pattern:

   ```
   $ODBS0|01Sep11 09:48:04.64|20|ODBC/MX server proxy started on target host 127.0.0.1,
   source port 28888, target port will be passed dynamically within client request.
   ```

   When logging with default log level 50, the last message of the log should then be similar to the following:

   ```
   $ODBS0|29Jul12 16:31:29.37|30|-- ODBCMXS setup completed, starting to listen... --
   ```

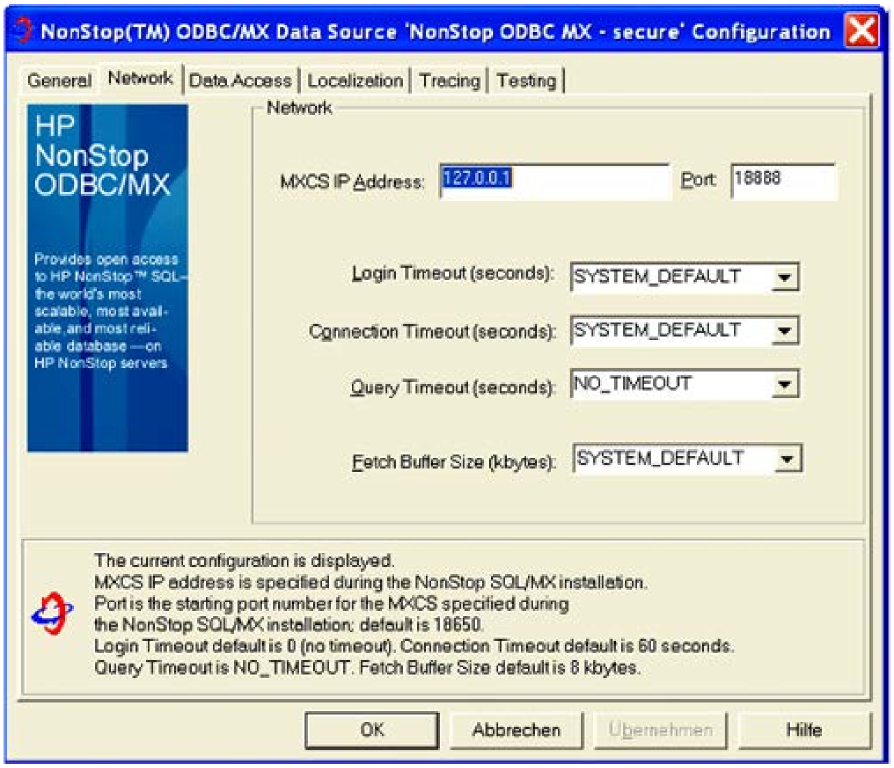
Note: Earlier versions of HP NonStop SSL might write out a "target port" with the above log message though it is not relevant for the setup.

To install and configure the RemoteProxy for ODBC/MX

1. Download $SYSTEM.ZNSSSL.PROXYEXE in binary format to your OCBC/MX client workstation, renaming it to PROXY.EXE.
2. On the OCBC/MX client workstation, run PROXY.EXE to start the RemoteProxy installation program and follow the installation instructions.
3. Double-click on HP NonStop SSL RemoteProxy icon in your system tray. The "RemoteProxy" configuration window will be displayed.
4. Select "New" from the "Session" menu. The "Session Properties" dialog will be displayed.
5. In the "Protocol" field, select "ODBC/MX Client".
6. In the "Target Host" field, enter the IP address or host name where your ODBCMXS process is listening on your NonStop server.
7. In the "Target Port" field, enter the port number that you have specified as the listen port of your ODBCMXS process on the NonStop server, (e.g. 28888).
8. In the "Local (Accepting) Port" field, enter the port number that RemoteProxy will use to listen for connections from your ODBC/MX driver. The port number must not be in use by any other program or service on your client PC. It must be the same port that the ODBC/MX server process is using on the NonStop server side, e.g. "18888" in this example.
9. Start the RemoteProxy session by clicking on the "Start" button
10. If the start is successful, check the startup messages with the "View Log" command.

To configure the ODBC/MX driver to connect via the RemoteProxy

1. Navigate to the ODBC/MX driver configuration dialog.
2. Navigate to the "Network" Tab within the dialog
3. Reconfigure the MXCS IP Address to the local host address, e.g. "localhost", or "127.0.0.1"
4. Reconfigure the "Port" field to the port number you specified as "Local Port" in the "Session Properties" of the relevant RemoteProxy Session.
5. Assuming you used "18888" as "Accepting Port" in the RemoteProxy session configuration, your ODBC driver configuration dialog should look as follows:



HP NonStop ODBC/MX

Provides open access to HP NonStop™ SQL– the world's most scalable, most available, and most reliable database —on HP NonStop servers

Network

MXCS IP Address: 127.0.0.1 Port: 18888

Login Timeout (seconds): SYSTEM_DEFAULT

Connection Timeout (seconds): SYSTEM_DEFAULT

Query Timeout (seconds): NO_TIMEOUT

Fetch Buffer Size (kbytes): SYSTEM_DEFAULT

The current configuration is displayed.
MXCS IP address is specified during the NonStop SQL/MX installation.
Port is the starting port number for the MXCS specified during
the NonStop SQL/MX installation; default is 18650.
Login Timeout default is 0 (no timeout). Connection Timeout default is 60 seconds.
Query Timeout is NO_TIMEOUT. Fetch Buffer Size default is 8 kbytes.

Confirm the changes by clicking "OK".

6. You may use the NonStop Connectivity Tool to test the secure ODBC connection to the NonStop system.
7. You may check the successful creation of the session through the proxy by examining the messages with the "View Log" command in the "Session Properties" screen of the RemoteProxy.

To connect securely with your ODBC/MX client

1. After you have correctly configured your ODBC/MX driver, use your ODBC client like you did before to connect to the NonStop system.
2. You may check the successful creation of the session through the proxy by examining the messages with the "View Log" command in the "Session Properties" screen of the RemoteProxy.

Installing an SSL Tunnel for EXPAND-over-IP Lines

Creating an SSL tunnel for an EXPAND-over-IP line requires running a HP NonStop SSL process in EXPANDS mode for the line handler on both sides of the connection. The configuration of the HP NonStop SSL processes can be easily derived from the existing line handler configuration of EXPAND-over-IP line. To enable the tunneling, only a single line handler attribute needs to be changed.

To install an SSL tunnel process for an EXPAND-over-IP line handler, you will need to perform the following steps on both NonStop servers connected by the line:

1. Install a HP NonStop SSL EXPAND proxy (EXPANDS) proxy process for the EXPAND line.
2. Reconfigure your EXPAND line configuration to activate the SSL tunnel for the EXPAND line.

Note: This section lists the basic installation instructions. For a production installation, please refer to "Load Balancing and Fault-Tolerance of EXPAND over SSL" in chapter "Configuration".

To install the HP NonStop SSL EXPANDS proxy

1. Determine the name of the EXPAND-over-IP line handler you want to secure.
2. At your TACL prompt, run the HP NonStop SSL SETUP macro:

   ```
   > VOLUME $SYSTEM.ZNSSSL
   > RUN SETUP
   ```

   Enter the name of the line handler when requested.

   The SETUP macro will create a configuration file (e.g. EXPSCF0) and an SCF IN file for the installation as persistent process (e.g. EXPSIN0).
3. Start the HP NonStop SSL EXPANDS persistent process, e.g.

   ```
   > SCF START PROCESS $ZZKRN.#SSL-EXPANDS-0
   ```
4. Check the log file (configured in the configuration file) to verify the EXPANDS process has started correctly, e.g.

   ```
   > SHOWLOG EXPSLOG *
   ```

   Verify that the log contains a message of the following pattern:

   ```
   $EXPS1|19May10 17:48:47.04|20|EXPAND proxy started (10.0.0.196:1280 <- 10.0.0.198:1280)
   ```

Note: These steps need to be performed on both systems connected over the EXPAND-over-IP line.

To activate the SSL tunnel for the EXPAND line

1. Using SCF, alter the configuration of the EXPAND line as follows:

   ```
   > ASSUME LINE <line>
   > ABORT
   > ALTER, DESTIPADDR 127.0.0.1
   > START
   ```
2. After the tunnel is properly configured on both sides, the HP NonStop SSL log file should contain messages of the following pattern:

   ```
   $EXPS |27Apr05 12:31:41.01|50|E1| tunnel connect succeeded, tunnel ready
   ```

   or

   ```
   $EXPS |27Apr05 12:37:26.78|50|E1| accepted tunnel connection, tunnel ready
   ```

   The EXPAND line should then show the "READY" state.

Note: Again, that change in the SCF configuration has to be done on both systems.

Configuration

Configuration Overview

HP NonStop SSL processes can be flexibly configured by a set of configuration parameters which can be specified by the following means:

- A configuration file
- PARAM commands
- startup command line parameters
- SSLCOM commands

The different options to specify a configuration for HP NonStop SSL allow system administrators to easily manage installations with multiple HP NonStop SSL processes running on multiple TCP/IP processes and ports as well as in different modes. For example, multiple HP NonStop SSL secure proxy processes with a an identical SSL configuration can share the same configuration file, while process-unique parameters such as proxy port, target host and port can be specified on the command line.

On startup, HP NonStop SSL parses the given configuration parameters sources. A single parameter may be specified in multiple sources, e.g. in the configuration file and on the startup command line. In this case, HP NonStop SSL will process parameters with the following precedence (highest to lowest):

1. PARAM parameter
2. Configuration file parameter
3. Startup line parameter

This means that a parameter given in the configuration file will override the value given for the same parameter on the startup line. Likewise, a parameter value given as PARAM command will override any value specified in the configuration file.

All parameters can be specified in any of the configuration parameter sources, with the following exceptions:

- The run mode of a HP NonStop SSL process is specified explicitly on the command line as first startup line parameter. This parameter defines if HP NonStop SSL acts as a secure Telnet server proxy, a secure FTP server proxy, or in any other supported mode of operation (see "Starting an HP NonStop SSL Process" for a complete list of run modes).
- The configuration file to be used as a parameter source can only be specified as a PARAM or startup line parameter, as it is meaningless in a configuration file itself.

Regardless which way they are specified, parameter names are case insensitive.

Additionally, a subset of configuration parameters can be changed at run time using SSLCOM commands (see chapter "SSLCOM Command Interface" for details).

The Configuration File

The configuration file is an edit type file which can be created and modified with a standard NonStop editor such as TEDIT. The name of the file that a HP NonStop SSL process should use as configuration source is passed to the program during startup.

The file contains entries of the form

```
parameter-name  parameter-value
```

Like in the standard TCP/IP configuration files, any lines starting with a "#" character are interpreted as comments. The following printout is the contents of the sample configuration file for running HP NonStop SSL telnet proxy:

```
# sample configuration file for a HP NonStop SSL secure telnet server proxy

#--------------------------------------------------------------------------------
# general settings

# TCP/IP process the web server runs on
SUBNET       $ZTC0

# SSL port telnet which HP NonStop SSL listens for incoming SSL emulator connections
PORT        4023
# TELSERV listening port the connections will be forwarded to
TARGETPORT    23

#--------------------------------------------------------------------------------
# log configuration
# set the level
LOGLEVEL 50
# enable console logging to $0
LOGCONSOLE $0
# additionally log to file
LOGFILE $DATA1.SSL.LOGTELS

#--------------------------------------------------------------------------------
# SSL configuration
# our server certificate and private key
SERVCERT      $DATA1.SSL.MYCERT
SERVKEY       $DATA1. SSL.PRIVKEY
SERVKEYPASS   myprivatepassword
# our server cert was issued by verisign
CACERTS       $DATA1. SSL.VERISIGN
# we only accept the strongest cipher suites with AES256
CIPHERSUITES 0.53,0.56,0.57
```

PARAM commands

HP NonStop SSL configuration parameters can be specified as PARAM commands as follows:

```
PARAM <parameter name> <parameter value>
```

All available HP NonStop SSL parameters can be specified as PARAM commands.

The following example demonstrates how to start a HP NonStop SSL telnet proxy listening on $ZTC03, port 8023, using PARAM commands:

```
> PARAM PORT 8023
> PARAM TARGETPORT 23
> PARAM SUBNET $ZTC03
> PARAM LOGFILE $DATA1.SSL.LOGTELS
> PARAM LOGCONSOLE *
> RUN SSLOBJ/ NAME $TELS, NOWAIT/ TELNETS
```

Startup Line Parameters

HP NonStop SSL configuration parameters can be passed on the startup line as follows (for a complete description of the RUN SSLOBJ see section "Starting an HP NonStop SSL Process"):

```
<parameter name> <parameter value>; <parameter name> <parameter value>; ...
```

The following example demonstrates how to start a multiple HP NonStop SSL proxies sharing the same SSLCONF configuration file:

```
> PARAM CONFIG SSLCONF
> RUN SSLOBJ /NAME $STN0, CPU 0, NOWAIT/ TELNETS; SUBNET $ZTC0; PORT 8023
> RUN SSLOBJ /NAME $STN1, CPU 1, NOWAIT/ TELNETS; SUBNET $ZTC1; PORT 8023
> RUN SSLOBJ /NAME $STN2, CPU 2, NOWAIT/ TELNETS; SUBNET $ZTC2; PORT 8023
> RUN SSLOBJ /NAME $STN3, CPU 3, NOWAIT/ TELNETS; SUBNET $ZTC3; PORT 8023
```

Parameter Reference

This section describes all available HP NonStop SSL parameters in alphabetical order. Note, that parameter names are case insensitive independently of the source.

Parameter Overview

The following table lists all available HP NonStop SSL parameters and their meanings:

Parameter	**Meaning**
ALLOWCERTERRORS	Allows selective overriding of certificate validation errors.
ALLOWIP	Limits allowed remote IP addresses.
ALLOWRENEGOTIATION	Controls whether SSL/TLS renegotiation is allowed.
AUDITASCIIONLY AUDITASCIIDUMPLENIN AUDITASCIIDUMPLENOUT AUDITCONSOLE AUDITLEVEL AUDITFILE AUDITFILELAYOUT AUDITFILERETENTION AUDITFORMAT AUDITMAXFILELENGTH	Control the creation of an audit file containing the remote FTP commands in run mode FTPS,FTPC or the socket activities in run modes PROXYS, PROXYC, ODBCMXS.
CACERTS	File names of a DER encoded X.509 CA certificates representing a certificate chain signing the certificate configured with the CLIENTCERT or SERVCERT parameter.
CIPHERSUITES	List of cipher suites that will be accepted by a secure HP NonStop SSL process. If omitted, default openssl cipher suites will be used.
CLIENTAUTH	Enforced client authentication when running as SSL server: a certificate signing the certificates the client is using for SSL client authentication
CLIENTCERT	File name of a DER encoded X.509 client certificate.
CLIENTKEY	The private key to be used for the client certificate.
CLIENTKEYPASS	Password for reading the (encrypted) private key file.
CONFIG	File name of a HP NonStop SSL configuration file.
CONFIG2	Allows the usage of a second configuration file with different security settings.
CONNECTIONINFOFORMAT	Specifies the default format for the output of the SSLCOM command "connections".

Parameter	**Meaning**
CONNECTIONINFOFORMATDETAILED	Specifies the default format for the output of the SSLCOM command "connections, detail".
CONTENTFILTER	Activates content-filtering in run modes TELNETS, PROXYS and PROXYC.
DENYIP	Limits allowed remote IP addresses.
DESTIPADDR DESTIPPORT	Sets the destination IP address and port for an EXPANDS tunnel.
DNSRESOLVERTCPIPPROCESSNAME	Can be used to explicitly set a TCP/IP process name for DNS resolution.
DONOTWARNONERROR	Log selected errors with LOGLEVEL 20 rather than as WARNING.
DYNAMICROUTINGENABLEIPV6	Causes the expected separator between the target host IP address and the target port to be a pipe symbol ('|') instead of a colon (':'). To be used only with ROUTINGMODE D (dynamic).
EXPANDCOMPRESSION	Controls if compression is used in Expand mode.
EXPANDENCRYPTION	Can be used to disable encryption in EXPAND mode.
FTPALLOWPLAIN	Allows plain FTP traffic when HP NonStop SSL is run in FTPS mode.
FTPCALLOW200REPLY	Sets compatibility for older FTP/TLS servers when run in FTPC mode.
FTPLOCALDATAPORT	Controls the value of the local port on the NonStop system of the data connection in FTPC mode with PASSIVE set to TRUE.
FTPMAXPORT	The maximum port number HP NonStop SSL will use for FTP data connections.
FTPMINPORT	The minimum port number HP NonStop SSL will use for FTP data connections.
HASHALGORITHMS	Configures which hash algorithms are used for the fingerprint validation of the server.
HEAPSIZELIMIT	Controls the maximum heap size to be used by the process.
INTERFACE	Controls the IP address HP NonStop SSL will bind to for connections made to HP NonStop SSL.
IPMODE	Specifies the TCP/IP mode (IPv4/IPv6/Dual) HP NonStop SSL will run in.
KEEPALIVE	Specifies if keep alive messages are sent to TCP/IP sockets.
LOGCONSOLE	Determines if log messages are written to a console.
LOGEMS	Determines if log messages are written to EMS.
LOGFILE	Determines if log messages are written to a file.
LOGFILERETENTION	Controls the number of log files kept after rollover occurs.
LOGFORMAT	Controls the format of the log messages that are written to the console or log file.
LOGFORMATCONSOLE	Controls the format of the log messages that are written to the console.
LOGFORMATEMS	Controls the format of the log messages that are written to EMS.
LOGFORMATFILE	Controls the format of the log messages that are written to a log file.
LOGLEVEL	Determines which messages will be written the log file.
LOGLEVELCONSOLE	Allows setting a different log level for LOGCONSOLE output.
LOGLEVELEMS	Allows setting a different log level for LOGEMS output.
LOGLEVELFILE	Allows setting a different log level for LOGFILE output.
LOGLEVELTCPCONNECTMESSAGE	Controls the log level used to log an incoming TCP/IP connection.
LOGMEMORY	Allows regular logging of HP NonStop SSL’s memory usage to the log output.
LOGMAXFILELENGTH	Controls the maximum size of the log file.

Parameter	Meaning
MAXSESSIONS	Limits the number of parallel connections in run modes PROXYS, PROXYC, TELNETS.
MAXVERSION	Maximum admissible SSL/TLS protocol version.
MINVERSION	Minimum admissible SSL/TLS protocol version.
PASSIVE	Sets the direction of the data socket connections in FTPC mode.
PEERCERTCOMMONNAME	For verification of remote certificates.
PEERCERTFINGERPRINT	For verification of remote certificates.
PORT	The port the HP NonStop SSL server listens on for incoming connections.
PTCPIPFILTERKEY	Sets the filter key to enable round robin filtering.
ROUTINGMODE	Controls how SSLOBJ routes traffic to the target, either static (default) or dynamic (usually not needed).
SERVCERT	File name of a DER encoded X.509 server certificate.
SERVKEY	The private key to be used for the server certificate.
SERVKEYPASS	Password for reading the (encrypted) private key file.
SLOWDOWN	Adds delay to processing resulting in slower encryption/decryption with less CPU usage.
SOCKSHOST SOCKSPORT SOCKSUSER	Configure HP NonStop SSL as SOCKS Version 4 client in run modes FTPC or PROXYC.
SRCIPADDR SRCIPPORT	Sets the source IP address and port for an EXPANDS tunnel.
SUBNET	The name of the TCP/IP process HP NonStop SSL should listen on for connections.
SSLCOMSECURITY	Restricts the execution of SSLCOM commands.
TARGETHOST	The IP address or name of the host that connections should be routed to.
TARGETINTERFACE	Controls the IP address HP NonStop SSL binds to for outgoing connections.
TARGETPORT	The port number that connections should route routed to.
TARGETSUBNET	The name of the TCP/IP process HP NonStop SSL should use for outgoing connections.
TCPIPHOSTFILE	Sets the DEFINE = TCPIP^HOST^FILE.
TCPIPNODEFILE	Sets the DEFINE = TCPIP^NODE^FILE.
TCPIPRESOLVERNAME	Sets the DEFINE = TCPIP^RESOLVER^NAME.
TCPNODELAY	Activates RFC1323 on all sockets.
TRUST	When running as SSL client: list of trusted CA or server certificate files or fingerprints.

ALLOWCERTERRORS

Use this parameter to allow selective overriding of certificate validation errors.

Parameter Syntax

```
ALLOWCERTERRORS number1 [, number2, ...]
```

Arguments

number

comma-separated list of certificate errors which HP NonStop SSL should ignore. The error numbers are defined in the OpenSSL sources used for HP NonStop SSL (see Considerations).

Considerations

- **Warning:** The usage of this parameter may compromise the security of your configuration. Use only as workaround and with care.
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.

The following table lists error numbers and names as defined in the OpenSSL sources:

Error Name	**Error number**
X509_V_ERR_UNABLE_TO_GET_ISSUER_CERT	2
X509_V_ERR_UNABLE_TO_GET_CRL	3
X509_V_ERR_UNABLE_TO_DECRYPT_CERT_SIGNATURE	4
X509_V_ERR_UNABLE_TO_DECRYPT_CRL_SIGNATURE	5
X509_V_ERR_UNABLE_TO_DECODE_ISSUER_PUBLIC_KEY	6
X509_V_ERR_CERT_SIGNATURE_FAILURE	7
X509_V_ERR_CRL_SIGNATURE_FAILURE	8
X509_V_ERR_CERT_NOT_YET_VALID	9
X509_V_ERR_CERT_HAS_EXPIRED	10
X509_V_ERR_CRL_NOT_YET_VALID	11
X509_V_ERR_CRL_HAS_EXPIRED	12
X509_V_ERR_ERROR_IN_CERT_NOT_BEFORE_FIELD	13
X509_V_ERR_ERROR_IN_CERT_NOT_AFTER_FIELD	14
X509_V_ERR_ERROR_IN_CRL_LAST_UPDATE_FIELD	15
X509_V_ERR_ERROR_IN_CRL_NEXT_UPDATE_FIELD	16
X509_V_ERR_OUT_OF_MEM	17
X509_V_ERR_DEPTH_ZERO_SELF_SIGNED_CERT	18
X509_V_ERR_SELF_SIGNED_CERT_IN_CHAIN	19
X509_V_ERR_UNABLE_TO_GET_ISSUER_CERT_LOCALLY	20
X509_V_ERR_UNABLE_TO_VERIFY_LEAF_SIGNATURE	21
X509_V_ERR_CERT_CHAIN_TOO_LONG	22
X509_V_ERR_CERT_REVOKED	23
X509_V_ERR_INVALID_CA	24
X509_V_ERR_PATH_LENGTH_EXCEEDED	25
X509_V_ERR_INVALID_PURPOSE	26
X509_V_ERR_CERT_UNTRUSTED	27
X509_V_ERR_CERT_REJECTED	28
X509_V_ERR_SUBJECT_ISSUER_MISMATCH	29
X509_V_ERR_AKID_SKID_MISMATCH	30
X509_V_ERR_AKID_ISSUER_SERIAL_MISMATCH	31
X509_V_ERR_KEYUSAGE_NO_CERTSIGN	32
X509_V_ERR_UNABLE_TO_GET_CRL_ISSUER	33
X509_V_ERR_UNHANDLED_CRITICAL_EXTENSION	34

Error Name	Error number
X509_V_ERR_KEYUSAGE_NO_CRL_SIGN	35
X509_V_ERR_UNHANDLED_CRITICAL_CRL_EXTENSION	36
X509_V_ERR_APPLICATION_VERIFICATION	50

Default

If omitted, HP NonStop SSL will work normally (all certificate validation errors are treated as such and connection attempts will fail)

Example

```
ALLOWCERTERRORS 10
```

This will temporarily allow expired certificates.

ALLOWIP

Use this parameter to specify which remote IP addresses are to be allowed to establish sessions ("white list").

Note: With HP NonStop SSL AAE, the parameter syntax for specifying subnets has been changed to using Classless Interdomain Routing (CIDR) format in order to prevent ambiguous subnet specification and simplify usage, especially with IPv6 entries.

Parameter Syntax

```
ALLOWIP [direction]range
```

Arguments

direction

Optional character specifying realm on which rules shall be applied

- A = Apply rules on incoming connections only
- C = Apply rules on outgoing connections only
- B = Apply rules on all connections only (*default*)

range

One or more Classless Interdomain Routing (CIDR) format entries specifying an IP subnet or a single host IP address. Entries have to be separated by comma. The network suffix can be left out for host entries (/32 or /128 will be assumed then). IPv6/DUAL entries have to be specified in square brackets. Entry types and the corresponding CIDR format:

- IPv4 address: 10.1.2.196 (/32 is assumed)
- IPv4 subnet : 10.2.0.0/16
- IPv6 address: [abcd:1111::ab00] (/128 is assumed)
- IPv6 subnet : [abcd::ef00/120]
- DUAL address: [::ffff:172.0.0.28] (/128 is assumed)
- DUAL subnet : [::ffff:172.1.1.0/104]

Considerations

- See section "Limiting Remote IP Addresses" for the concept of remote IP filtering.
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.

- Backwards compatibility to the former syntax is preserved, however in the mid-term ALLOWIP and DENYIP should be changed to using CIDR format.

Default

If omitted, HP NonStop SSL will use * to allow all remote IP addresses.

Example

```
ALLOWIP 10.0.1.0/24, 10.0.2.0/24, 172.22.22.42


ALLOWIP A[abcd::ef00/120] ,  [abcd:1111::ab00] , [::ffff:172.1.1.0/104]
```

ALLOWRENEGOTIATION

This parameter can be used to allow or disallow SSL/TLS Session renegotiation.

Parameter Syntax

```
ALLOWRENEGOTIATION TRUE | FALSE
```

Arguments

FALSE

Renegotiation will not be permitted. If the peer tries to initiate a renegotiation, the corresponding session will be closed and a warning including the detailed session information will be issued to the log.

TRUE

Renegotiation will be allowed and performed when the peer initiates a corresponding renegotiation request.

Default

By default, a value of FALSE will be used, i.e. starting with HP NonStop SSL AAH renegotiation is not permitted by default anymore (see also under Considerations).

Considerations

Prior to the introduction of this parameter (pre HP NonStop SSL AAH), renegotiation requests were allowed to be performed. This behavior was changed with the introduction of this parameter in HP NonStop SSL release AAH due to security concerns. Even though renegotiation was intended to add security (refresh cryptographic parameters), it can be misused to launch a Denial of Service (DoS) attack: a rogue SSL/TLS client can initiate renegotiation in an endless loop, which leads to permanent high CPU load on the server side of the connection. With a few connections doing renegotiation over and over, the server side can be kept in a state where it will be too busy to respond to new requests anymore (DoS). Due to this risk, starting with NonStop SSL AAH, renegotiation will not be allowed by default anymore. However, in case renegotiation is needed, it can be enabled again by explicitly setting this parameter to TRUE.

AUDITASCIIONLY

Use this parameter to define how HP NonStop SSL writes raw data to the audit log.

Parameter Syntax

```
AUDITASCIIONLY TRUE | FALSE
```

Arguments

TRUE

Data will be dumped in ASCII format; binary values with coded character will be represented as <hh> where hh is the hexadecimal representation of the binary value.

`FALSE`

Data will be dumped as full hex dump. This consumes a lot of resources but provides the most complete view.

Default

By default, a value of TRUE will be used

Considerations

- Audit messages will depend on the run mode – see parameter AUDITLEVEL for details
- See also parameters AUDITASCIIDUMPLENIN and AUDITASCIIDUMPLENOUT to control how much data is dumped.

AUDITASCIIDUMPLENIN

Use this parameter to define how many bytes of the incoming messages are written to the audit log when AUDITASCIIONLY is set to TRUE.

Parameter Syntax

```
AUDITASCIIDUMPLENIN -1 | n
```

Arguments

`-1`

means that each incoming message will be fully dumped.

n

means that only the first *n* bytes of each incoming message will be dumped.

Default

By default, a value of -1 will be used

See also

AUDITASCIIONLY, AUDITASCIIDUMPLENOUT

AUDITASCIIDUMPLENOUT

Use this parameter to define how many bytes of outgoing messages are written to the audit log when AUDITASCIIONLY is set to TRUE.

Parameter Syntax

```
AUDITASCIIDUMPLENIN -1 | n
```

Arguments

`-1`

means that each outgoing message will be fully dumped.

n

means that only the first *n* bytes of each outgoing message will be dumped

Default

By default, a value of -1 will be used

See also

AUDITASCIIONLY

AUDITCONSOLE

Use this parameter to define if and to what console device HP NonStop SSL audit messages are written to.

Parameter Syntax

```
AUDITCONSOLE * | % | $0 | auditdevice
```

Arguments

`*`

means that no audit messages are written to a console

`%`

means that audit messages are written to the home terminal of the HP NonStop SSL process

`$0`

audit messages are written to $0

`auditdevice`

audit messages are written the given device.

Default

By default, audit messages will be not be written to a device ("*")

Example

```
AUDITCONSOLE $DEV.#SUBDEV
```

Considerations

- Audit messages will depend on the run mode – see parameter AUDITLEVEL for details.

See also

AUDITFILE, AUDITLEVEL, AUDITFORMAT

AUDITFILE

Use this parameter to define if and to what file HP NonStop SSL audit messages are written to.

Parameter Syntax

```
AUDITFILE * | filenameprefix
```

Arguments

`*`

means that no audit messages are written to a file

`filenameprefix`

the prefix of the audit message file set. The actual audit file names are constructed from *filenameprefix* appended by a number controlled by the AUDITFILERETENTION parameter.

Default

By default, no audit messages are written to a file ("*")

Considerations

- Audit messages will depend on the run mode – see parameter AUDITLEVEL for details

See also

AUDITCONSOLE, AUDITLEVEL, AUDITFORMAT

AUDITFILELAYOUT

Use this parameter to control the layout format of the audit file. In particular, this parameter can be used to enable writing audit in CSV (comma separated values) format for easy subsequent processing in other tools like Excel, SQL, etc..

Parameter Syntax

```
AUDITFILELAYOUT layout
```

Arguments

layout

must be one of

- ORIGINAL - original pipe symbol separated audit format.
- CSV - comma separated values format (NonStop SSL AAI and higher).

Default

By default, AUDITFILELAYOUT will be set to CSV.

AUDITFILERETENTION

Use this parameter to control how many audit files HP NonStop SSL keeps when audit file rollover occurs.

Parameter Syntax

```
AUDITFILERETENTION n
```

Arguments

`n`

number of audit files to keep

Default

By default, 10 files are kept.

Considerations

- a minimum of 10 is enforced for that parameter
- See "Logfile/Auditfile Rollover" in chapter "Monitoring" for details on logfile rollover.

See also

AUDITMAXFILELENGTH, AUDITFILE

AUDITFORMAT

Use this parameter to control the format of audit messages that are written to the console or audit file.

Parameter Syntax

```
AUDITFORMAT format
```

Arguments

format

a number representing a bit mask controlling the format options. Please see parameter LOGFORMAT for the bit mask.

Default

The default log format is 93 (date, time, milliseconds, process ID and log level)

Example

Display date, time, and milliseconds only:

```
AUDITFORMAT 13
```

Display date, time only:

```
AUDITFORMAT 5
```

Considerations

- Audit messages will depend on the run mode – see parameter AUDITLEVEL for details

See also

AUDITCONSOLE, AUDITFILE, AUDITLEVEL

AUDITLEVEL

Use this parameter to control what audit messages are written to the audit console or file.

Parameter Syntax

```
AUDITLEVEL detail
```

Arguments

detail

a number representing the detail level.

Default

The default audit level is 50.

Considerations

- Audit messages are written only for the following run modes: PROXYS, PROXYC, ODBCMXS, FTPS.
- The following table describes how to set AUDITLEVEL for the various run modes.

Audit Level	**Run Modes TELNETS,PROXYS,PROXYC,ODBCMXS**	**Run Mode FTPS**
10	Startup of HP NonStop SSL	Startup of HP NonStop SSL
30		Logon of user
50	Network events (connect, disconnect)	FTP operations
60		Network events (connect, disconnect)
80	Data flowing through HP NonStop SSL: byte count only	

Audit Level	Run Modes TELNETS,PROXYS,PROXYC,ODBCMXS	Run Mode FTPS
90	Data flowing through HP NonStop SSL: full byte dump (see parameter AUDITASCIIONLY for details)	

- For PROXYS, PROXYC and ODBCMXS, we recommend 50 for basic auditing and 99 for extended auditing including full traffic log.

Note: If set to 99, all data flowing through the network is dumped to the audit log. This could include confidential data or passwords so make sure to properly secure the audit log files.

- For FTPS and FTPC mode, we recommend 50 for normal auditing.

See also

AUDITCONSOLE, AUDITFILE, AUDITFORMAT

AUDITMAXFILELENGTH

Use this parameter to control the maximum size of the audit file.

Parameter Syntax

```
AUDITMAXFILELENGTH length
```

Arguments

length

a number representing the maximum audit file length in kilobytes.

Max. 40.000 (~40 MB)

Min 100

Default

The default length is 20 000 KB.

Considerations

- After the current audit file reaches the maximum size, a log rollover will occur. Please see section "Logfile/Auditfile Rollover" in chapter "Monitoring" for details on logfile rollover.

See also

AUDITFILE, AUDITLEVEL

CACERTS

Use this parameter to specify a certificate chain validating the server or client certificate given by the SERVCERT or CLIENTCERT parameter.

Parameter Syntax

```
CACERTS file1 [, file2, ...]
```

Arguments

file1, file2, ...

the designated files are DER encoded X.509 CA certificates.

Default

If omitted, HP NonStop SSL will search for a single "CACERT" file on the default subvolume.

Example

```
CACERTS $DATA1.SSL.MYCA, $DATA1.SSL.MYROOTCA
```

Considerations

- The first file on the list must contain a certificate signing the given server certificate. Subsequent files must contain certificates that sign the previous certificate in the list.
- During SSL handshake, the certificate chain will be sent along with the client or server certificate to the SSL communication partner
- If a value of * is used for CACERTS, it will be assumed that the client or server certificate is self-signed.
- A CA certificate for testing purposes is delivered as CACERT file on the HP NonStop SSL installation subvolume to enable quick start installation. This test CA certificate signs the test server certificate contained in SERVCERT or CLIENTCERT.

See also

SERVCERT, CLIENTCERT, SSLCOM SSLINFO, SSLCOM RELOAD CERTIFICATES

CIPHERSUITES

Use this parameter to specify which cipher suites are admissible for a HP NonStop SSL process.

Parameter Syntax

```
CIPHERSUITES suite [, suite, ...]
```

Arguments

`suite`

specifies a cipher suite. Currently the following cipher suites can be explicitly configured:

Specifier	RFC Algo Name	OpenSSL Name	KEX	Enc	Mac
0.1	TLS_RSA_WITH_NULL_MD5	NULL-MD5	RSA	NULL	MD5
0.2	TLS_RSA_WITH_NULL_SHA	NULL-SHA	RSA	NULL	SHA
0.3	TLS_RSA_EXPORT_WITH_RC4_40_MD5	EXP-RC4-MD5	RSA_EXPORT	RC4_40	MD5
0.4	TLS_RSA_WITH_RC4_128_MD5	RC4-MD5	RSA	RC4_128	MD5
0.5	TLS_RSA_WITH_RC4_128_SHA	RC4-SHA	RSA	RC4_128	SHA
0.6	TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5	EXP-RC2-CBC-MD5	RSA_EXPORT	RC2_CBC_40	MD5
0.7	TLS_RSA_WITH_IDEA_CBC_SHA	IDEA-CBC-SHA	RSA	IDEA_CBC	SHA
0.8	TLS_RSA_EXPORT_WITH_DES40_CBC_SHA	EXP-DES-CBC-SHA	RSA_EXPORT	DES40_CBC	SHA
0.9	TLS_RSA_WITH_DES_CBC_SHA	DES-CBC-SHA	RSA	DES_CBC	SHA

Specifier	RFC Algo Name	OpenSSL Name	KEX	Enc	Mac
0.10	TLS_RSA_WITH_3DES_EDE_CBC_SHA	DES-CBC3-SHA	RSA	3DES_EDE_CBC	SHA
0.17	TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	EXP-EDH-DSS-DES-CBC-SHA	DHE_DSS_EXPORT	DES40_CBC	SHA
0.18	TLS_DHE_DSS_WITH_DES_CBC_SHA	EDH-DSS-DES-CBC-SHA	DHE_DSS	DES_CBC	SHA
0.19	TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	EDH-DSS-DES-CBC3-SHA	DHE_DSS	3DES_EDE_CBC	SHA
0.20	TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	EXP-EDH-RSA-DES-CBC-SHA	DHE_RSA_EXPORT	DES40_CBC	SHA
0.21	TLS_DHE_RSA_WITH_DES_CBC_SHA	EDH-RSA-DES-CBC-SHA	DHE_RSA	DES_CBC	SHA
0.22	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	EDH-RSA-DES-CBC3-SHA	DHE_RSA	3DES_EDE_CBC	SHA
0.23	TLS_DH_anon_EXPORT_WITH_RC4_40_MD5	EXP-ADH-RC4-MD5	DH_anon_EXPORT	RC4_40	MD5
0.24	TLS_DH_anon_WITH_RC4_128_MD5	ADH-RC4-MD5	DH_anon	RC4_128	MD5
0.25	TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA	EXP-ADH-DES-CBC-SHA	DH_anon_EXPORT	DES40_CBC	SHA
0.26	TLS_DH_anon_WITH_DES_CBC_SHA	ADH-DES-CBC-SHA	DH_anon	DES_CBC	SHA
0.27	TLS_DH_anon_WITH_3DES_EDE_CBC_SHA	ADH-DES-CBC3-SHA	DH_anon	3DES_EDE_CBC	SHA
0.47	TLS_RSA_WITH_AES_128_CBC_SHA	AES128-SHA	RSA	AES_128_CBC	SHA
0.50	TLS_DHE_DSS_WITH_AES_128_CBC_SHA	DHE-DSS-AES128-SHA	DHE_DSS	AES_128_CBC	SHA
0.51	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE-RSA-AES128-SHA	DHE_RSA	AES_128_CBC	SHA
0.52	TLS_DH_anon_WITH_AES_128_CBC_SHA	ADH-AES128-SHA	DH_anon	AES_128_CBC	SHA
0.53	TLS_RSA_WITH_AES_256_CBC_SHA	AES256-SHA	RSA	AES_256_CBC	SHA
0.56	TLS_DHE_DSS_WITH_AES_256_CBC_SHA	DHE-DSS-AES256-SHA	DHE_DSS	AES_256_CBC	SHA
0.57	TLS_DHE_RSA_WITH_AES_256_CBC_SHA	DHE-RSA-AES256-SHA	DHE_RSA	AES_256_CBC	SHA
0.58	TLS_DH_anon_WITH_AES_256_CBC_SHA	ADH-AES256-SHA	DH_anon	AES_256_CBC	SHA
0.65	TLS_RSA_WITH_CAMELLIA_128_CBC_SHA	CAMELLIA128-SHA	RSA	CAMELLIA_128_CBC	SHA
0.68	TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA	DHE-DSS-CAMELLIA128-SHA	DHE_DSS	CAMELLIA_128_CBC	SHA
0.69	TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA	DHE-RSA-CAMELLIA128-SHA	DHE_RSA	CAMELLIA_128_CBC	SHA
0.70	TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA	ADH-CAMELLIA128-SHA	DH_anon	CAMELLIA_128_CBC	SHA

Specifier	RFC Algo Name	OpenSSL Name	KEX	Enc	Mac
0.132	TLS_RSA_WITH_CAMELLIA_256_CBC_SHA	CAMELLIA256-SHA	RSA	CAMELLIA_256_CBC	SHA
0.135	TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA	DHE-DSS-CAMELLIA256-SHA	DHE_DSS	CAMELLIA_256_CBC	SHA
0.136	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA	DHE-RSA-CAMELLIA256-SHA	DHE_RSA	CAMELLIA_256_CBC	SHA
0.137	TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA	ADH-CAMELLIA256-SHA	DH_anon	CAMELLIA_256_CBC	SHA
0.98	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA	EXP1024-DES-CBC-SHA	RSA_EXPORT1024	DES_CBC	SHA
0.99	TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA	EXP1024-DHE-DSS-DES-CBC-SHA	DHE_DSS_EXPORT1024	DES_CBC	SHA
0.100	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA	EXP1024-RC4-SHA	RSA_EXPORT1024	RC4_56	SHA
0.101	TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA	EXP1024-DHE-DSS-RC4-SHA	DHE_DSS_EXPORT1024	RC4_56	SHA
0.102	TLS_DHE_DSS_WITH_RC4_128_SHA	DHE-DSS-RC4-SHA	DHE_DSS	RC4_128	SHA
0.128	TLS_GOSTR341094_WITH_28147_CNT_IMIT	GOST94-GOST89-GOST89	GOSTR341094	28147_CNT	IMIT
0.129	TLS_GOSTR341001_WITH_28147_CNT_IMIT	GOST2001-GOST89-GOST89	GOSTR341001	28147_CNT	IMIT
0.130	TLS_GOSTR341094_WITH_NULL_GOSTR3411	GOST94-NULL-GOST94	GOSTR341094	NULL	GOSTR3411
0.131	TLS_GOSTR341001_WITH_NULL_GOSTR3411	GOST2001-NULL-GOST94	GOSTR341001	NULL	GOSTR3411
0.150	TLS_RSA_WITH_SEED_CBC_SHA	SEED-SHA	RSA	SEED_CBC	SHA
0.153	TLS_DHE_DSS_WITH_SEED_CBC_SHA	DHE-DSS-SEED-SHA	DHE_DSS	SEED_CBC	SHA
0.154	TLS_DHE_RSA_WITH_SEED_CBC_SHA	DHE-RSA-SEED-SHA	DHE_RSA	SEED_CBC	SHA
0.155	TLS_DH_anon_WITH_SEED_CBC_SHA	ADH-SEED-SHA	DH_anon	SEED_CBC	SHA
192.1	TLS_ECDH_ECDSA_WITH_NULL_SHA	ECDH-ECDSA-NULL-SHA	ECDH_ECDSA	NULL	SHA
192.2	TLS_ECDH_ECDSA_WITH_RC4_128_SHA	ECDH-ECDSA-RC4-SHA	ECDH_ECDSA	RC4_128	SHA
192.3	TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA	ECDH-ECDSA-DES-CBC3-SHA	ECDH_ECDSA	3DES_EDE_CBC	SHA
192.4	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	ECDH-ECDSA-AES128-SHA	ECDH_ECDSA	AES_128_CBC	SHA
192.5	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	ECDH-ECDSA-AES256-SHA	ECDH_ECDSA	AES_256_CBC	SHA
192.6	TLS_ECDHE_ECDSA_WITH_NULL_SHA	ECDHE-ECDSA-NULL-SHA	ECDHE_ECDSA	NULL	SHA
192.7	TLS_ECDHE_ECDSA_WITH_RC4_128_SHA	ECDHE-ECDSA-RC4-SHA	ECDHE_ECDSA	RC4_128	SHA

Speci-fier	RFC Algo Name	OpenSSL Name	KEX	Enc	Mac
192.8	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	ECDHE-ECDSA-DES-CBC3-SHA	ECDHE_ECDSA	3DES_EDE_CBC	SHA
192.9	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	ECDHE-ECDSA-AES128-SHA	ECDHE_ECDSA	AES_128_CBC	SHA
192.10	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	ECDHE-ECDSA-AES256-SHA	ECDHE_ECDSA	AES_256_CBC	SHA
192.11	TLS_ECDH_RSA_WITH_NULL_SHA	ECDH-RSA-NULL-SHA	ECDH_RSA	NULL	SHA
192.12	TLS_ECDH_RSA_WITH_RC4_128_SHA	ECDH-RSA-RC4-SHA	ECDH_RSA	RC4_128	SHA
192.13	TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA	ECDH-RSA-DES-CBC3-SHA	ECDH_RSA	3DES_EDE_CBC	SHA
192.14	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA	ECDH-RSA-AES128-SHA	ECDH_RSA	AES_128_CBC	SHA
192.15	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	ECDH-RSA-AES256-SHA	ECDH_RSA	AES_256_CBC	SHA
192.16	TLS_ECDHE_RSA_WITH_NULL_SHA	ECDHE-RSA-NULL-SHA	ECDHE_RSA	NULL	SHA
192.17	TLS_ECDHE_RSA_WITH_RC4_128_SHA	ECDHE-RSA-RC4-SHA	ECDHE_RSA	RC4_128	SHA
192.18	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	ECDHE-RSA-DES-CBC3-SHA	ECDHE_RSA	3DES_EDE_CBC	SHA
192.19	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDHE-RSA-AES128-SHA	ECDHE_RSA	AES_128_CBC	SHA
192.20	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	ECDHE-RSA-AES256-SHA	ECDHE_RSA	AES_256_CBC	SHA
192.21	TLS_ECDH_anon_WITH_NULL_SHA	AECDH-NULL-SHA	ECDH_anon	NULL	SHA
192.22	TLS_ECDH_anon_WITH_RC4_128_SHA	AECDH-RC4-SHA	ECDH_anon	RC4_128	SHA
192.23	TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA	AECDH-DES-CBC3-SHA	ECDH_anon	3DES_EDE_CBC	SHA
192.24	TLS_ECDH_anon_WITH_AES_128_CBC_SHA	AECDH-AES128-SHA	ECDH_anon	AES_128_CBC	SHA
192.25	TLS_ECDH_anon_WITH_AES_256_CBC_SHA	AECDH-AES256-SHA	ECDH_anon	AES_256_CBC	SHA

Default

If omitted, NonStop SSL will use all default OpenSSL ciphers, i.e. currently:

ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA:DHE-RSA-CAMELLIA256-SHA:DHE-DSS-CAMELLIA256-SHA:AECDH-AES256-SHA:ECDH-RSA-AES256-SHA:ECDH-ECDSA-AES256-SHA:AES256-SHA:CAMELLIA256-SHA:ECDHE-RSA-DES-CBC3-SHA:ECDHE-ECDSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA:AECDH-DES-CBC3-SHA:ECDH-RSA-DES-CBC3-SHA:ECDH-ECDSA-DES-CBC3-SHA:DES-CBC3-SHA:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:DHE-RSA-CAMELLIA128-SHA:DHE-DSS-CAMELLIA128-SHA:AECDH-AES128-SHA:ECDH-RSA-AES128-SHA:ECDH-ECDSA-AES128-SHA:AES128-SHA:CAMELLIA128-SHA:ECDHE-RSA-RC4-SHA:ECDHE-ECDSA-RC4-SHA:AECDH-RC4-SHA:ECDH-RSA-RC4-SHA:ECDH-ECDSA-RC4-SHA:RC4-SHA

Example

```
CIPHERSUITES 0.53,0.47
```

Considerations

- Please note that the default CIPHERSUITES are subject to change in order to make sure that only the most secure ciphers are used by default.
- When running as an SSL client, CIPHERSUITES specifies the cipher suites that should be allowed in order of preference (favorite choice first). During the SSL handshake, HP NonStop SSL will present the list of cipher suites to the SSL server. The server will then select a cipher suite from the list, or, if no acceptable choices are presented, return a handshake failure alert and close the connection.
- When running as an SSL server, HP NonStop SSL will select the first cipher from the list presented by the client during the handshake that matches a cipher from CIPHERSUITES.

Notes:

- If you trace the client handshake with a tool like wireshark, you will see one additional cipher with hex representation 0x00FF. This is not an actual cipher but a hint for the server that the client supports secure renegotiation. Please see section 4 of http://tools.ietf.org/html/draft-ietf-tls-renegotiation-01 in case you like more details on that.

WARNINGS!

- Do NOT use ADH ciphers unless you know exactly what you are doing! ADH ciphers DO NOT include authentication, thus they are vulnerable to Man-in-the-Middle attacks! Strongly not recommended!
- The cipher suites 0.1 and 0.2 will NOT encrypt the traffic, they will only authenticate the partners and provide message integrity checking. Please only use if encryption is not required.

CLIENTAUTH

Use this parameter to enforce SSL client authentication when running as SSL server. The CLIENTAUTH parameter specifies a file (or a set of files) containing certificates. The certificate(s) will be sent to the client during connection setup. The client will reply with its own client certificate which must be signed by one of the certificates configured with the CLIENTAUTH parameter.

Parameter Syntax

```
CLIENTAUTH * | file1 [, file2, ...]
```

Arguments

`*`

No certificate request will be sent to the client

`file1, file2, ...`

DER encoded X.509 CA certificate(s) which sign the certificate to be sent by the SSL client to HP NonStop SSL. If the SSL client cannot send such a certificate, the connection setup will fail.

Default

If omitted, '*' is used and HP NonStop SSL will not enforce SSL client authentication when running as SSL server.

Example

```
CLIENTAUTH $DATA1.SSL.CACERT
```

CLIENTCERT

Use this parameter to specify the client certificate that HP NonStop SSL should use to authenticate itself to an SSL server.

Parameter Syntax

```
CLIENTCERT * | file
```

Arguments

`*`

SSL client authentication is deactivated.

`file`

Guardian file name of a DER encoded X.509 client certificate.

Default

If omitted or set to *, HP NonStop SSL will not authenticate itself to the SSL server.

Example

```
CLIENTCERT $DATA1.SSL.CLNTCERT
```

Considerations

- This parameter only applies to the run modes PROXYC and FTPC, it will be ignored in other run modes
- A client certificate for testing purposes is delivered as CLNTCERT file on the HP NonStop SSL installation subvolume to enable quick start installation.
- Client certificates received by a CA such as VeriSign or Thawte in BASE64 format must be converted to DER format (e.g. with the OpenSSL tools) before they can be used with HP NonStop SSL.
- The client certificate must match the private key file specified by CLIENTKEY.

See also

CLIENTKEY, CLIENTKEYPASS, "Requesting the SSL Client to Present a Client Certificate" in chapter "SSL Reference"

CLIENTKEY

Use this parameter to specify the file containing the private key associated with the public key contained in the client certificate configured by CLIENTCERT.

Parameter Syntax

```
CLIENTKEY file
```

Arguments

`file`

file name of a DER encoded PKCS-8 encrypted private key file with PKCS-5 algorithm identifiers.

Default

If omitted, HP NonStop SSL will search for a "CLIENTKEY" file on the default subvolume.

Example

```
CLIENTKEY $DATA1.SSL.MYKEY
```

Considerations

- This parameter only applies to the run modes PROXYC and FTPC, it will be ignored in other run modes
- The private key data in the file is password encrypted. For HP NonStop SSL to be able to decrypt the file, the correct password must be specified by the CLIENTKEYPASS parameter.
- A private key file for testing purposes is delivered as "CLNTKEY" file on the HP NonStop SSL installation subvolume to enable quick start installation. This private key file matches the test client certificate delivered as "CLNTCERT". The password for the CLNTKEY file is "test".

See also

CLIENTCERT, CLIENTKEYPASS

CLIENTKEYPASS

Use this parameter to specify the password for the file containing the private key associated with the public key given in the client certificate.

Parameter Syntax

```
CLIENTKEYPASS password
```

Arguments

password

the password or pass phrase to decrypt the private key file. The password string may contain spaces. However, leading or trailing spaces will be ignored.

Default

If omitted, HP NonStop SSL will try "test" as password.

Example

```
CLIENTKEYPASS my private passphrase
```

Considerations

- This parameter only applies to the run modes PROXYC and FTPC, it will be ignored in other run modes
- The default password ("test") enables quick start installation with the "CLIENTKEY" public key file delivered with HP NonStop SSL.

See also

CLIENTCERT, CLIENTKEY

CONFIG

Use this parameter to specify a configuration file for a HP NonStop SSL process.

Parameter Syntax

```
CONFIG file
```

Arguments

file

the name of the configuration file.

Default

If omitted, HP NonStop SSL will not use a configuration file.

Example

```
CONFIG $DATA1.SSL.SSLCONF
```

Considerations

- This parameter can only be specified as PARAM or on the startup line. It is not valid within a configuration file.
- Parameters specified in the configuration file can be overwritten by PARAM or startup line settings.

CONFIG2

Use this parameter to specify a second configuration file for a HP NonStop SSL process.

Parameter Syntax

```
CONFIG2 file2
```

Arguments

file2

the name of the second configuration file.

Default

If omitted, HP NonStop SSL will not use a second configuration file.

Example

```
CONFIG2 $DATA1.HP SSL.SSLCONF2
```

Considerations

- Having a second configuration file allows to store the pass phrases in a separate file with higher security settings
- The second configuration file has precedence over the first one
- This parameter can only be specified as PARAM or on the startup line. It is not valid within a configuration file.
- Parameters specified in the configuration file can be overwritten by PARAM or startup line settings.

CONNECTIONINFOFORMAT

Use this parameter to specify the output format for the SSLCOM command "connections".

Parameter Syntax

```
CONNECTIONINFOFORMAT format
```

Arguments

format

specifies the format to be used. Valid values are:

- o EXTENDED : designates the new default output format introduced with HP NonStop SSL AAE for connection info not exceeding 80 characters and consistency between IPv4 and IPv6
- o ORIGINAL : designates the format as used before which, for longer IPv4 and especially for IPv6 addresses exceeds the usually available window width

- o CSV : designates output as comma-separated values, primarily targeted to simplify automated parsing of the output.

Default

Starting with HP NonStop SSL AAE, the default format will be EXTENDED. Prior to that it was ORIGINAL, but not configurable.

EXAMPLE

```
CONNECTIONINFOFORMAT ORIGINAL
```

Considerations

- Both the ORIGINAL and the EXTENDED format are primarily targeted for human readers and are subject to change. Please do not use these when intending to parse the output programmatically but use format CSV for that instead.

CONNECTIONINFOFORMATDETAILED

Use this parameter to specify the output format for the SSLCOM command "connections, detail"

Parameter Syntax

```
CONNECTIONINFOFORMATDETAILED format
```

Arguments

format

specifies the format to be used. Valid values are:

- o EXTENDED : designates the new default output format introduced with HP NonStop SSL AAE for connection info not exceeding 80 characters and consistency between IPv4 and IPv6.
- o ORIGINAL : designates the format as used before which, for longer IPv4 and especially for IPv6 addresses exceeds the usually available window width.
- o CSV : designates output as comma-separated values, primarily targeted to simplify automated parsing of the output.

Default

Starting with HP NonStop SSL AAE, the default format will be EXTENDED. Prior to that it was ORIGINAL, but not configurable.

EXAMPLE

```
CONNECTIONINFOFORMAT ORIGINAL
```

Considerations

- Both the ORIGINAL and the EXTENDED format are primarily targeted for human readers and are subject to change. Please do not use these when intending to parse the output programmatically but use format CSV for that instead.

CONTENTFILTER

Use this parameter to configure a text file with rules which will be applied to all incoming messages in run modes TELNETS, PROXYS and PROXYC. If a message does not match the rule set, the connection will be terminated and the message will be discarded.

Parameter Syntax

```
CONTENTFILETER * | file
```

Arguments

`*`

no filtering.

`file`

The filename of the rule set file.

Default

If omitted, HP NonStop SSL will use a value of * (no filtering).

Example

```
CONTENTFILTER CFILTER
```

Considerations

- The value of the parameter can be changed without stopping HP NonStop SSL using the SSLCOM command SET CONTENTFILTER file.
- The following example shows the syntax of the filter rules. This example will only allow messages starting with "<A" or "<B" and ending with ">" to pass the filter.

```
#################################################################################
# file to define content filter rules
# empty lines or lines starting with '#' are ignored
#################################################################################

#################################################################################
# example file based on the following requirements:
#
# the following two are valid messages (double quotes are *not* part of msg)
#   "<ABC>"
#   "<BBC>"
#
# the following two are *not* valid messages
#   "<CCC>"    - does not start with "<A" or "<B"
#   "text"     - does not start with "<"
#################################################################################

#################################################################################
# msg delimiters (required)
# used to define a "message" as part of the byte stream
# all bytes are ASCII values represented as decimal numbers
#################################################################################
# start with < sign = 3C hex = 60 dec
msgstartbyte 60
# end with > sign = 3E hex = 62 dec
msgendbyte 62

#################################################################################
# list of regular expressions, in double quotes
# (at least one required)
#
# note that the engine implements "traditional unix regular expressions"
# see
#  en.wikipedia.org/wiki/Regular_expression#Traditional_Unix_regular_expressions
# for details
#
# regular expressions are combined using an implicit "logical or"
# a message matching any single regular expression will pass
# a message matching no regular expression will fail
# at least one regular expression must be present
#################################################################################
# allow any message starting with "<A"
regexp "^<A."
```

```
# allow any message starting with "<B"
regexp "^<B."
```

DENYIP

Use this parameter to specify which remote IP addresses are to be forbidden to establish sessions ("black list").

Parameter Syntax

```
DENYIP [direction]range
```

Arguments

direction

Optional character specifying realm on which rules shall be applied

- A = Apply rules on incoming connections only
- C = Apply rules on outgoing connections only
- B = Apply rules on all connections only (*default*)

range

One or more Classless Interdomain Routing (CIDR) format entries specifying an IP subnet or a single host IP address. Entries have to be separated by comma. The network suffix can be left out for host entries (/32 or /128 will be assumed then). IPv6/DUAL entries have to be specified in square brackets. Entry types and the corresponding CIDR format:

- IPv4 address: 10.1.2.196 (/32 is assumed)
- IPv4 subnet : 10.2.0.0/16
- IPv6 address: [abcd:1111::ab00] (/128 is assumed)
- IPv6 subnet : [abcd::ef00/120]
- DUAL address: [::ffff:172.0.0.28] (/128 is assumed)
- DUAL subnet : [::ffff:172.1.1.0/104]

Considerations

- See section "Limiting Remote IP Addresses" (in chapter "Introduction") for the concept of remote IP filtering
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.
- Backwards compatibility to the former syntax is preserved, however in the mid-term ALLOWIP and DENYIP should be changed to using CIDR format.

Default

If omitted, HP NonStop SSL will use an empty entry, respectively *DEFAULT* to not forbid any remote IP addresses.

Example

```
DENYIP 10.0.1.0/24, 10.0.2.0/24, 172.22.22.42


DENYIP A[abcd::ef00/120] ,  [abcd:1111::ab00] , [::ffff:172.1.1.0/104]
```

DESTIPADDR, DESTIPPORT

Use these parameters to for the configuration of an HP NonStop SSL EXPANDS process.

Parameter Syntax

```
DESTIPADDR ip-address
DESTIPPORT port
```

Arguments

ip-address

specifies the IP address of the remote end of the EXPAND line.

port

specifies the port number of the remote end of the EXPAND line.

Example

```
DESTIPADDR 10.0.0.13; DESTIPPORT 1202


DESTIPADDR fe80:aa::eeff:1023 ; DESTIPPORT 1202
```

Considerations

- The parameters should be set exactly to the original equivalent parameter values of the EXPAND line handler, as shown by the SCF INFO LINE DETAIL command.
- The parameters are ignored with any run mode other than EXPANDS.

See also

SRCIPADDR, SRCIPPORT

DNSRESOLVERTCPIPPROCESSNAME

Use this parameter to explicitly set the name of the TCP/IP process that should be used for DNS resolving

Parameter Syntax

```
DNSRESOLVERTCPIPPROCESSNAME processname
```

Arguments

processname

specifies the NonStop process name that is to be used for DNS resolving in HP NonStop SSL

Considerations

In most cases there is no need to explicitly set this parameter since a corresponding program logic will dynamically determine the TCP/IP stack to use for DNS resolution.

Example

DNSRESOLVERTCPIPPROCESSNAME $ZTC2

DONOTWARNONERROR

Use this parameter to log selected errors with LOGLEVEL 20 rather than as WARNING. By default, all errors on sockets result in a WARNING being displayed in the HP NonStop SSL log. Using this parameter, a log message with LOGLEVEL 20 will be issued instead for the configured error numbers.

Parameter Syntax

```
DONOTWARNONERROR ErrorList
```

Arguments

ErrorList

specifies a list of comma-separated error numbers

Default

If omitted, HP NonStop SSL will use an empty entry.

Example

```
DONOTWARNONERROR 4120
```

Considerations

- The example shown will yield in error 4120 ("Connection reset by remote") generating a log message with LOGLEVEL 20 rather than a WARNING.

DYNAMICROUTINGENABLEIPV6

This parameter is only valid in ROUTINGMODE D (dynamic routing) and must be used when IPv6 addresses are to be used as the dynamic targets. The parameter will cause the expected separator between the target host IP address and the target port to be a pipe symbol ('|') instead of a colon (':'). Please see parameter ROUTINGMODE for more details.

Parameter Syntax

```
DYNAMICROUTINGENABLEIPV6 {TRUE|FALSE}
```

Default

FALSE

Example

```
DYNAMICROUTINGENABLEIPV6 TRUE
```

See Also

ROUTINGMODE

EXPANDCOMPRESSION

This parameter controls if compression is used in Expand mode. Compression can improve the throughput of Expand connections. An additional license is required to enable compression. Contact HP if interested.

Parameter Syntax

```
EXPANDCOMPRESSION {TRUE|FALSE*}
```

Default

If omitted, the default will be FALSE, i.e. no Expand compression will be used.

Example

```
EXPANDCOMPRESSION TRUE
```

Considerations

- For compression to work, both SSLOBJ processes across an EXPAND connection must be version AAI or later. It is possible to have only one side of the connection use compression.
- Compression can be enabled/disabled while the SSLOBJ process is running by using the corresponding SSLCOM command SET EXPANDCOMPRESSION {ON|OFF}

EXPANDENCRYPTION

This parameter controls whether encryption is used for EXPAND connections. For obvious security reasons it is not recommended to disable the by default active EXPAND encryption.

Parameter Syntax

EXPANDENCRYPTION {TRUE*|FALSE}

Example

EXPANDENCRYPTION FALSE

Considerations

- This parameter should be only used where speed of transfer is very important and encryption is *not* required.
- The parameter value can **not** be changed via SSLCOM during runtime.
- The parameter value needs to be identical for both SSLOBJ processes for a given EXPAND line across two systems.

FTPALLOWPLAIN

Use this parameter to specify whether HP NonStop SSL will allow unencrypted FTP sessions when running in FTPS mode.

Parameter Syntax

`FTPALLOWPLAIN` *`boolean`*

Arguments

`boolean`

If set to TRUE or 1 or Yes, HP NonStop SSL will allow unencrypted traffic.

Default

If omitted, HP NonStop SSL will *not* allow unencrypted traffic

Example

```
FTPALLOWPLAIN TRUE
```

Considerations

- This parameter is relevant only if HP NonStop SSL is running in the FTPS mode.

FTPCALLOW200REPLY

Use this parameter to specify whether HP NonStop SSL will allow an illegal "200" response to the AUTH TLS command sent to the remote FTP/TLS server.

Parameter Syntax

```
FTPCALLOW200REPLY boolean
```

Arguments

boolean

if set to TRUE or 1 or Yes, HP NonStop SSL will allow the illegal response.

Default

If omitted, HP NonStop SSL will *not* allow the illegal 200 response.

Example

```
FTPCALLOW200REPLY TRUE
```

Considerations

- This parameter is relevant only if HP NonStop SSL is running in the FTPC mode.
- The FTP/TLS specification requires a "234" reply code to the AUTH TLS command. This parameter has been added to support some older FTP/TLS server with HP NonStop SSL in FTPC mode.

FTPLOCALDATAPORT

Use this parameter to specify how HP NonStop SSL will pick the local data port for the data connection in FTPC mode with PASSIVE set to true.

Parameter Syntax

```
FTPLOCALDATAPORT number
```

Arguments

number

0 for “pick a random port” or any specific port number

Default

If omitted, a value of 0 will be used.

Example

```
FTPLOCALDATAPORT 20
```

Considerations

- This parameter is relevant only if HP NonStop SSL is running in the FTPC mode with PASSIVE set to TRUE
- Choosing a value other than zero will be firewall-friendly. However, this can result in errors if the remote FTP server does not choose random data ports itself.

FTPMAXPORT

Use this parameter to specify the maximum port number HP NonStop SSL will use for FTP data connections

Parameter Syntax

```
FTPMAXPORT number
```

Arguments

number

The maximum port number HP NonStop SSL will use for FTP data connections

Default

If omitted, HP NonStop SSL will use a value of 41000

Example

```
FTPMAXPORT 22000
```

Considerations

- This parameter is relevant only if HP NonStop SSL is running in the FTPS or FTPC mode.
- Together with the parameter FTPMINPORT it controls the values HP NonStop SSL assigns for the FTP data sockets. You can change this value to make sure that the FTP data connections will not interfere with other TCP/IP services on your system.

FTPMINPORT

Use this parameter to specify the minimum port number HP NonStop SSL will use for FTP data connections

Parameter Syntax

```
FTPMINPORT number
```

Arguments

number

the minimum port number HP NonStop SSL will use for FTP data connections

Default

If omitted, HP NonStop SSL will use a value of 40000

Example

```
FTPMINPORT 20000
```

Considerations

- This parameter is relevant only if HP NonStop SSL is running in the FTPS or FTPC mode.
- Together with the parameter FPTMAXPORT it controls the values HP NonStop SSL assigns for the FTP data sockets. You can change this value to make sure that the FTP data connections will not interfere with other TCP/IP services on your system.

HEAPSIZELIMIT

This parameter can be used to set the maximum heap size that will be used by the process

Parameter Syntax

```
HEAPSIZELIMIT value_in_bytes
```

Arguments

value_in_bytes

The value of the maximum heap to be used in bytes.

Default

The default heap size limit is set to in 367001600 (=350 MiB). See Considerations for further explanation.

Considerations

SSLOBJ regularly checks its heap usage. At the point the heap usage reaches 85% of HEAPSIZELIMIT, newly incoming connections will be rejected until the heap usage has decreased again below 85% of HEAPSIZELIMIT.

In theory the heap could grow as large as the available main memory, however there are various limiting factors. In particular the TCP6SAM and CIPSAM TCP/IP stacks set their QIO segment at the location around 400MiB, thereby limiting the general heap usage to that value. In case the process reaches that value, the process will abend with a heap exhaustion error.

In the default settings about 100MiB headroom (400MiB-350MiB*85%) for existing connections is preserved for safety reasons.

INTERFACE

Use this parameter to specify the IP address HP NonStop SSL should use for local binding on incoming connections.

Parameter Syntax

```
INTERFACE ip-address
```

Arguments

`ip-address`

the IP address to bind to or “*” for none

Default

If omitted, HP NonStop SSL will use the value of “*” and bind to no specific IP address

Example

```
INTERFACE 10.0.0.197


INTERFACE [2001:db9::1421:51ab]
```

Considerations

- The parameter is relevant for the following run modes: PROXYS (incoming socket), PROXYC (incoming socket), FTPS (control listening socket being connected to from remote FTP client), FTPC (control listening socket being connected to from local NonStop FTP client)
- Use this parameter to control which IP address HP NonStop SSL binds to for incoming connections.
- If a host name rather than an IP address is used to configure INTERFACE, name resolution will take place only once during startup. If name resolution fails, HP NonStop SSL will terminate during startup.
- The parameter is invalid in IPMODE DUAL since no bind address except the IPv6 ANY address '::' can handle both IPv4 and IPv6.

See also

TARGETINTERFACE

HASHALGORITHMS

Use this parameter to define which hash algorithms are used when verifying the SSL server side based on its fingerprint.

Parameter Syntax

```
HASHALGORITHMS hashAlgorithm [, hashAlgorithm , ...]
```

Arguments

hashAlgorithm

Name of hash algorithm that should be used. If the parameter is explicitly set, at least one hash algorithm has to be given.

Valid hash algorithms names are:

- MD5 *
- SHA1 *
- RIPEMD160
- SHA256
- SHA384 **
- SHA512 **
- WHIRLPOOL

* You should not use this algorithm since it was cryptographically broken.

** Hash Algorithm not available on S-Series system

Default

By default, WHIRLPOOL as one of the two most secure algorithms is used (the other one is SHA512)

Example

```
HASHALGORITHMS WHIRLPOOL,SHA256
```

Considerations

- Cryptographic operations, such as hashing operations, do cost CPU cycles. When e.g. running as an FTP client proxy, fingerprint validation with hashing will be involved in every initial connection establishment process. Therefore consider choosing only one or two algorithms out of the list.
- Make sure to activate the algorithms actually used in the fingerprints specified in the TRUST parameter.
- Do not use MD5 or SHA1 anymore! These algorithms are cryptographically broken.

See also

TRUST

KEEPALIVE

Use this parameter to specify if TCP keep alive messages should be activated for established connections.

Parameter Syntax

```
KEEPALIVE mode
```

Arguments

mode

- 1 (on) for sending keep alive messages
- 0 (off) no messages are sent

Default

By default, keep alive messages are sent (1).

LOGCONSOLE

Use this parameter to define if and to what console device HP NonStop SSL log messages are written to.

Parameter Syntax

```
LOGCONSOLE * | % | $0 | logdevice
```

Arguments

`*`

means that no log messages are written to a console

`%`

means that log messages are written to the home terminal of the HP NonStop SSL process

`$0`

log messages are written to $0

`logdevice`

log messages are written the given device (e.g. $DEV.#SUBDEV)

Considerations

- The LOGLEVEL parameter controls what messages are produced by HP NonStop SSL.
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.

Default

By default, log messages will be written to the home terminal ("%")

See also

LOGEMS, LOGFILE, LOGLEVEL

LOGEMS

Use this parameter to define if HP NonStop SSL log messages are written to EMS.

Parameter Syntax

```
LOGEMS collector | *
```

Arguments

`*`

means that no log messages are written to EMS.

`collector`

means that log messages are written to the collector with that name.

Default

By default, no log messages are written to EMS ("*").

Example

```
LOGEMS $0
```

Considerations

- The LOGLEVEEMS parameter controls what messages are produced by HP NonStop SSL.
- The LOGFORMATEMS parameter controls the log message format.
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.
- If the EMS collector cannot be opened during startup, HP NonStop SSL will terminate. If the EMS collector cannot be opened after changing it through SSLCOM, the old collector will stay active

See also

LOGLEVELEMS, LOGFORMATEMS, LOGMAXFILELENGTH, LOGFILERETENTION

LOGFILE

Use this parameter to define if and to what file HP NonStop SSL log messages are written.

Parameter Syntax

```
LOGFILE * | filenameprefix
```

Arguments

`*`

means that no log messages are written to a file

`filenameprefix`

the prefix of the log file set. The actual log file names are constructed from *filenameprefix* appended by a number controlled by the LOGFILERETENTION parameter.

Default

By default, no log messages are written to a file ("*")

Considerations

- The LOGLEVELFILE parameter controls what messages are produced by HP NonStop SSL.
- The LOGFORMATFILE parameter controls the log message format.
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.
- See section "Logfile/Auditfile Rollover" in chapter "Monitoring" for details on logfile rollover.

See also

LOGLEVELFILE, LOGFORMATFILE, LOGMAXFILELENGTH, LOGFILERETENTION

LOGFILERETENTION

Use this parameter to control how many log files HP NonStop SSL keeps when logfile rollover occurs

Parameter Syntax

```
LOGFILERETENTION n
```

Arguments

n

number of log files to keep

Default

By default, 10 files are kept.

Considerations

- a minimum of 10 is enforced for that parameter
- See section "Logfile/Auditfile Rollover" in chapter "Monitoring" for details on logfile rollover.

See also

LOGMAXFILELENGTH, LOGFILE

LOGFORMAT

Use this parameter to control the default format the log messages.

Parameter Syntax

```
LOGFORMAT format
```

Arguments

format

a number representing a bit mask controlling the following format options:

bit 1 (decimal 1)	Date
bit 2 (decimal 2)	header (log messages a pre-fixed with "[log]")
bit 3 (decimal 4)	Time
bit 4 (decimal 8)	Milliseconds
bit 5 (decimal 16)	Process ID (name or PIN)
bit 7 (decimal 64)	Log Level of Message

Default

If omitted, 93 is used as format (date, time, milliseconds, process ID and log level).

Considerations

- If no value is set for the parameters LOGFORMATCONSOLE or LOGFORMATFILE, they will inherit their value from the parameter LOGFORMAT.
- If LOGFORMATCONSOLE, LOGFORMATFILE, LOGFORMATEMS are set with a value, the parameter of LOGFORMAT becomes meaningless.

See also

LOGFORMATCONSOLE, LOGFORMATEMS, LOGFORMATFILE

LOGFORMATCONSOLE

Use this parameter to control the format of the log messages that are written to the console.

Parameter Syntax

```
LOGFORMATCONSOLE format
```

Arguments

format

a number representing a bit mask controlling the following format options:

bit 1 (decimal 1)	Date
bit 2 (decimal 2)	header (log messages a pre-fixed with "[log]")
bit 3 (decimal 4)	Time
bit 4 (decimal 8)	Milliseconds
bit 5 (decimal 16)	Process ID (name or PIN)
bit 7 (decimal 64)	Log Level of Message

Default

If omitted, the console log format is derived from LOGFORMAT.

Example

Display date, time, and milliseconds only:

```
LOGFORMATCONSOLE 13
```

Display date, time only:

```
LOGFORMATCONSOLE 5
```

See also

LOGFORMAT, LOGFORMATEMS, LOGFORMATFILE

LOGFORMATEMS

Use this parameter to control the format of the log messages that are written to EMS.

Parameter Syntax

```
LOGFORMATEMS format
```

Arguments

format

a number representing a bit mask controlling the following format options:

bit 1 (decimal 1)	Date
bit 2 (decimal 2)	header (log messages a pre-fixed with "[log]")
bit 3 (decimal 4)	Time
bit 4 (decimal 8)	Milliseconds
bit 5 (decimal 16)	Process ID (name or PIN)
bit 7 (decimal 64)	Log Level of Message

Default

If omitted, the EMS log format is derived from LOGFORMAT.

Example

Display date, time, and milliseconds only:

```
LOGFORMATEMS 13
```

Display date, time only:

```
LOGFORMATEMS 5
```

See also

LOGFORMAT, LOGFORMATCONSOLE, LOGFORMATFILE

LOGFORMATFILE

Use this parameter to control the format of the log messages that are written to the log file.

Parameter Syntax

```
LOGFORMATFILE format
```

Arguments

`format`

a number representing a bit mask controlling the following format options:

bit 1 (decimal 1)	Date
bit 2 (decimal 2)	header (log messages a pre-fixed with "[log]")
bit 3 (decimal 4)	Time
bit 4 (decimal 8)	Milliseconds
bit 5 (decimal 16)	Process ID (name or PIN)
bit 7 (decimal 64)	Log Level of Message

Default

If omitted, the file log format is derived from LOGFORMAT.

Example

Display date, time, and milliseconds only:

```
LOGFORMATFILE 13
```

Display date, time only:

```
LOGFORMATFILE 5
```

See also

LOGFORMAT, LOGFORMATCONSOLE, LOGFORMATEMS, LOGFILE

LOGLEVEL

Use this parameter to control the default logging level.

Parameter Syntax

```
LOGLEVELCONSOLE detail
```

Arguments

`detail`

a number representing the detail level

Default

If omitted, a level of 50 is used.

Considerations

- If no value is set for the parameters LOGLEVELCONSOLE, LOGLEVELEMS, or LOGLEVELFILE, they will inherit their value from the parameter LOGLEVEL.
- If LOGLEVELCONSOLE, LOFLEVELEMS, and LOGLEVELFILE are all set with a value, the parameter of LOGLEVEL becomes meaningless.

See also

LOGLEVELCONSOLE, LOGLEVELEMS, LOGLEVELFILE

LOGLEVELCONSOLE

Use this parameter to control what messages are written to the log console.

Parameter Syntax

```
LOGLEVELCONSOLE detail
```

Arguments

detail

a number representing the detail level

Default

If omitted, the console log level is derived from LOGLEVEL.

Considerations

- Different log levels can be used for the outputs to LOGCONSOLE, LOGLEVELEMS, and LOGFILE.
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.

See also

LOGCONSOLE, LOGLEVEL, LOGFORMATCONSOLE

LOGLEVELEMS

Use this parameter to control what messages are written to EMS.

Parameter Syntax

```
LOGLEVELEMS detail
```

Arguments

`detail`

a number representing the detail level

Default

If omitted, the EMS log level is derived from LOGLEVEL.

Considerations

- Different log levels can be used for the outputs to LOGCONSOLE, LOGLEVELEMS, and LOGFILE.
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.

See also

LOGEMS, LOGLEVEL, LOGFORMATEMS

LOGLEVELFILE

Use this parameter to control what messages are written to the log file.

Parameter Syntax

```
LOGLEVELFILE detail
```

Arguments

detail

a number representing the detail level

Default

If omitted, the console file level is derived from LOGLEVEL.

Considerations

- Different log levels can be used for the outputs to LOGCONSOLE, LOGLEVELEMS, and LOGFILE.
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.

See also

LOGFILE, LOGLEVEL, LOGMAXFILELENGTH, LOGFORMATFILE, LOGFILERETENTION

LOGLEVELTCPCONNECTMESSAGE

Use this parameter to control the log level used to log the event of a new connection being established. Usually there is no need to specify this parameter explicitly.

Parameter Syntax

```
LOGLEVELTCPCONNECTMESSAGE detail
```

Arguments:

detail

a number representing the detail level

Default:

If omitted, the event of a new connection being established will be logged with the default level 50.

LOGMAXFILELENGTH

Use this parameter to control the maximum size of a log file.

Parameter Syntax

```
LOGMAXFILELENGTH length
```

Arguments

length

a number representing the maximum log file length in kilobytes in the range of 100 to 40000 (~40MB).

Default

The default length is 20000.

Considerations

- After the current file reaches the maximum size a log rollover will occur. Please see section "Logfile/Auditfile Rollover" in chapter "Monitoring" for details on logfile rollover.

See also

LOGFILE, LOGFILERETENTION

LOGMEMORY

Use this parameter to have HP NonStop SSL log memory usage information output in regular intervals.

Parameter Syntax

```
LOGMEMORY number_of_io’s
```

Arguments

number_of_io’s

a number representing after how many I/O operations HP NonStop SSL will send its memory usage to the log output

Default

The default is 0 meaning that memory usage will not be logged

Considerations

- Use to have an easy correlation between memory usage of HP NonStop SSL and events in the log output. Do not use if memory usage of HP NonStop SSL is not of interest for you.
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.

MAXSESSIONS

Use this parameter to limit the number of concurrent connections in run modes TELNETS, PROXYS, PROXYC and PROXY.

Parameter Syntax

```
MAXSESSIONS max
```

Arguments

max

the number of allowed concurrent sessions or 0 for unlimited.

Default

If omitted, HP NonStop SSL will use a value of 0 (no limits).

Example

```
MAXSESSIONS 100
```

Considerations

- If the number of allowed sessions is reached, any further connection request will be rejected and a warning will be written to the log file.
- The current number of connections is displayed in the STATUS command of SSLCOM.

MAXVERSION

Use this parameter to define the maximum admissible SSL/TLS protocol version.

Parameter Syntax

```
MAXVERSION version
```

Arguments

version

an SSL/TLS version number. Currently, the supported values are:

- 2.0: SSL 2.0
- 3.0: SSL 3.0
- 3.1: SSL 3.1 / TLS 1.0

Default

The default for this parameter is "3.1" (i.e. SSL 3.1 / TLS 1.0).

See also

MINVERSION

MINVERSION

Use this parameter to define the minimum admissible SSL/TLS protocol version.

Parameter Syntax

```
MINVERSION version
```

Arguments

version

an SSL/TLS version number. Currently supported values are:

- 2.0: SSL 2.0
- 3.0: SSL 3.0
- 3.1: SSL 3.1 / TLS 1.0

Default

The default for this parameter is "3.1"

Considerations

- For security reasons, it is recommended to use the latest version of the TLS protocol as standardized by the IETF (3.1). This requires setting MINVERSION to "3.1".

See also

MAXVERSION

PASSIVE

Use this parameter to define the direction of the data socket connection in FTPC mode

Parameter Syntax

```
PASSIVE mode
```

Arguments

mode

1 for passive mode, 0 for active mode.

Default

The default for this parameter is 1 (passive mode enabled).

Considerations

- This parameter is only relevant in the FTPC run mode of HP NonStop SSL
- In FTP, the data socket connection request can be made by the FTP client ("passive mode") or by the FTP server ("active mode"). The best choice for your environment depends on the capabilities of the FTP server you are communicating with and on your firewall settings.
- HP NonStop SSL in FTPS mode currently only supports passive mode, therefore to interact with HP NonStop SSL in FTPS mode, make sure to set the PASSIVE parameter to 1 for HP NonStop SSL running in FTPC mode.

PEERCERTCOMMONNAME

Use this parameter to enforce verification of the content of remote certificates presented to HP NonStop SSL.

Parameter Syntax

```
PEERCERTCOMMONNAME commonname
```

Arguments

commonname

the expected common name of the remote certificate.

Default

The default for this parameter is '*' which means the content will not be verified.

Examples

```
PEERCERTCOMMONNAME tandem1.mycompany.com
```

Considerations

- This parameter should not be used together with the parameter PEERCERTFINGERPRINT as behavior may be unpredictable then.
- If other than '*', the actual common name of the remote certificate will be compared against the content of the parameter.

- If the actual value of the common name in the remote certificate is part of the value configured in the parameter, it will be accepted. This allows configuring a list of common names.
- If the matching fails, the connection will be rejected.

PEERCERTFINGERPRINT

Use this parameter to enforce verification of the leaf certificate of the remote peer. In server runmodes this parameter is used to verify the fingerprint of the client certificate, in client runmodes it is used to verify the fingerprint of the server certificate.

Parameter Syntax

```
PEERCERTFINGERPRINT * | sha1-fingerprint
```

Arguments

`*`

No fingerprint verification.

`sha1-fingerprint`

the expected sha1 fingerprint of the remote certificate.

Default

The default for this parameter is '*' which means the fingerprint of the remote leaf certificate (i.e. client or server cert, depending on runmode) will not be verified.

Examples

```
PEERCERTFINGERPRINT da39a3ee5e6b4b0d3255bfef95601890afd80709
```

Considerations

- This parameter does not adhere to the HASHALGORITHMS parameter (yet), instead fingerprints should be given in SHA1 format.
- This parameter should not be used together with the parameter PEERCERTCOMMONNAME as behavior may be unpredictable then.
- If other than '*', the actual fingerprint of the remote server certificate will be compared against the value of the parameter.
- If the actual value in the certificate is part of the value configured in the parameter, it will be accepted. This allows configuring a list of fingerprints or common names.
- Fingerprints will be compared both as MD5 and SHA1 hashes, however for security reasons you should not use MD5 anymore.
- If the matching fails, the session will be rejected.

PORT

Use this parameter to specify the port number a HP NonStop SSL server should listen for incoming connections.

Parameter Syntax

```
PORT number
```

Arguments

`number`

the decimal number of a TCP/IP port.

Default

The default for this parameter depends on the HP NonStop SSL run mode:

TELNETS	11011 (*)
PROXYS	11011 (*)
PROXYC	11012 (*)
FTPS	11013 (*)
FTPC	11014 (*)

Considerations

- If operating as a secure server, HP NonStop SSL will only accept SSL connections on the specified port.
- Starting HP NonStop SSL to listen on a port number <=1024 requires SUPER group access.
- The ICANN manages a list of "well-known" port numbers for various protocols (see http://www.iana.org/assignments/port-numbers). Most run modes of HP NonStop SSL can not be mapped against this list with certainty, those run modes are marked with an asterisk (*). The default ports for those run modes were chosen from an "unassigned" port range (11002-11110)
- The choice for the PORT value in your environment will depend on the applications already running on your NonStop systems and the ports they use as well as your firewall configuration.
- You can specify a comma-separated list of multiple ports; see section "Multiple SSL Tunnels in a Single Process" for details.

PTCPIPFILTERKEY

Use this parameter to specify a filter key to enable round robin filtering with Parallel Library TCP/IP or TCP/IPV6.

Parameter Syntax

```
PTCPIPFILTERKEY password | *
```

Arguments

password

a password serving as a key to enable round robin filtering for multiple instances of HP NonStop SSL servers listening on the same port. The password will override the value of the DEFINE =PTCPIP^FILTER^KEY, which may have been passed to HP NonStop SSL at startup.

*

No filter key will be set. However, any DEFINE =PTCPIP^FILTER^KEY passed to HP NonStop SSL at startup will remain in effect.

Default

The default for this parameter is *.

Considerations

- Use this parameter to enable round robin filtering for multiple HP NonStop SSL servers configured as persistent processes with older release of the Kernel subsystem which did not support configuring DEFINEs.

ROUTINGMODE

The ROUTINGMODE parameter can be used in run modes PROXYC and PROXYS and is used to define in what way incoming connections shall be forwarded. By default this happens statically (the "S" stands for static), i.e. you have to specify the target to which connections are forwarded at the moment HP NonStop SSL is started.

The second possible value for ROUTINGMODE is "D" which stands for dynamic routing. In that case the first network packet sent to HP NonStop SSL needs to contain the destination IP address and port on the NonStop system in dotted decimal notation, preceded by a "D" and followed by a binary zero.

For IPv6 the additional parameter DYNAMICROUTINGENABLEIPV6 has to be set to TRUE. This will change the expected separator between the IP address and the port to be a pipe symbol ('|') which then has to be used for both IPv4 and IPv6 connections.

Example:

```
D10.0.0.198:8888<binary zero>
```

or, if DYNAMICROUTINGENABLEIPV6 is set to TRUE:

```
D10.0.0.198|8888<binary zero>
```

respectively

```
Dfe80:abcd::4711|8888<binary zero>
```

In this case the address will be taken as the target to which the connection shall be forwarded. This dynamic routing feature is only needed in really rare cases, so usually there is no need to touch this parameter.

Parameter Syntax

```
ROUTINGMODE S | D
```

Arguments:

S

Static routing is used

D

Dynamic routing is used

Default

If omitted, "S" is used and HP NonStop SSL will use static routing.

Example

```
ROUTINGMODE D
```

SERVCERT

Use this parameter to specify the server certificate HP NonStop SSL should use to authenticate itself to an SSL client.

Parameter Syntax

```
SERVCERT file
```

Arguments

file

Guardian file name of a DER encoded X.509 server certificate.

Default

If omitted, HP NonStop SSL will search for a file "SERVCERT" on the default subvolume.

Example

```
SERVCERT $DATA1.SSL.MYCERT
```

Considerations

- A server certificate for testing purposes is delivered as SERVCERT file on the HP NonStop SSL installation subvolume to enable a quick start installation.
- The server certificate must match the private key file specified by SERVKEY.

See also

SERVCERT, SERVKEY, SSLCOM SSLINFO, SSLCOM RELOAD CERTIFICATES

SERVKEY

Use this parameter to specify the private key file for an HP NonStop SSL server.

Parameter Syntax

```
SERVKEY file
```

Arguments

file

the file name of a DER encoded PKCS-8 encrypted private key file with PKCS-5 algorithm identifiers.

Default

If omitted, HP NonStop SSL will search for a "SERVKEY" file on the default subvolume.

Example

```
SERVKEY $DATA1.SSL.MYKEY
```

Considerations

- The private key data in the file is password encrypted. For HP NonStop SSL to be able to decrypt the file, the correct password must be specified by the SERVKEYPASS parameter.
- A private key file for testing purposes is delivered as "SERVKEY" file on the HP NonStop SSL installation subvolume to enable quick start installation. This private key file matches the test server certificate delivered as "SERVCERT". The password for the SERVKEY file is "test".

See also

SERVCERT, SERVKEYPASS, SSLCOM SSLINFO, SSLCOM RELOAD CERTIFICATES

SERVKEYPASS

Use this parameter to specify the password for the private key file.

Parameter Syntax

```
SERVKEYPASS password
```

Arguments

password

the password or pass phrase to decrypt the private key file. The password string may contain spaces. However, leading or trailing spaces will be ignored.

Default

If omitted, HP NonStop SSL will try "test" as password.

Example

```
SERVKEYPASS my private passphrase
```

Considerations

- The default password ("test") enables a quick start installation with the "SERVKEY" public key file delivered with HP NonStop SSL.

See also

SERVCERT, SERVKEYPASS, SSLCOM SSLINFO, SSLCOM RELOAD CERTIFICATES

SLOWDOWN

Use this parameter to make HP NonStop SSL use less CPU cycles for encryption. This will result in a decrease of possible throughput.

Parameter Syntax

```
SLOWDOWN <ticks>
```

Arguments

ticks

After each I/O operation, HP NonStop SSL will call the Guardian System Procedure DELAY with the value of <ticks>. A higher value will decrease both throughput and CPU usage of HP NonStop SSL.

Default

If omitted, SLOWDOWN will be 0 and HP NonStop SSL will consume all available CPU resources.

Example

```
SLOWDOWN 1
```

Considerations

- In most installations, the default value of 0 should be acceptable
- The parameter is mostly intended for use with the FTPC or FTPS modes of HP NonStop SSL. Setting SLOWDOWN to values between 1 and 5 will significantly reduce CPU usage but will also make the time a file transfer will take higher.
- The impact of HP NonStop SSL high volume data encryption/decryption can also be influenced by the priority of the HP NonStop SSL process. However, if it is desirable to run HP NonStop SSL at a higher priority than the target plain servers/clients, the SLOWDOWN can be used to limit the impact of the cryptographic operations.
- The best value for your environment will depend both on your hardware and requirements.

SOCKSHOST, SOCKSPORT, SOCKSUSER

Use these three parameters to make HP NonStop SSL act as a SOCKS Version 4 client in the run modes FTPC or PROXYC.

Note: SOCKS4 by design does not support IPv6, i.e. SOCKSHOST, SOCKSPORT and SOCKSUSER are only valid in IPMODE IPV4. SOCKS is a protocol that relays TCP sessions at a firewall host to allow application users transparent access across the firewall. For more information about SOCKS, please see http://en.wikipedia.org/wiki/SOCKS.

Parameter Syntax

```
SOCKSHOST sockshost
```

```
SOCKSPORT socksport
```

```
SOCKSUSER socksuser
```

Arguments

sockshost

the hostname or IP address of the host running the SOCKS-Version 4 enabled firewall. A value of * indicates that the SOCKS protocol will not be used.

socksport

the listening port of the host running the SOCKS-Version 4 enabled firewall.

socksuser

the SOCKS user name to be used to authenticate against the SOCKS server.

Default

If omitted, HP NonStop SSL will use a value of * for SOCKSHOST meaning the SOCKS protocol will not be used.

Example

```
SOCKSHOST 172.3.5.99
SOCKSPORT 1911
SOCKSUSER sockstest
```

Considerations

- In run mode PROXYC the value of TARGETPORT will still be required to determine the final host to connect to.
- In run modes FTPC the final host to connect to will be configured by adding it to the user name just as when not using SOCKS.

SRCIPADDR, SRCIPPORT

Use these parameters to for the configuration of an HP NonStop SSL EXPANDS process.

Parameter Syntax

```
SRCIPADDR ip-address
```

```
SRCIPPORT port
```

Arguments

ip-address

specifies the IP address of the local end of the EXPAND line.

port

specifies the port number of the local end of the EXPAND line.

Example

```
SRCIPADDR 10.0.0.12; SRCIPPORT 1202


SRCIPADDR 2001:db9::1421:51ab ; SRCIPPORT 1202
```

Considerations

- The parameters should be set exactly to the original equivalent parameter values of the EXPAND line handler, as shown by the SCF INFO LINE DETAIL command.
- The parameters are ignored with any run mode other than EXPANDS.

See also

DESTIPADDR, DESTIPPORT

SUBNET

Use this parameter to specify the TCP/IP process HP NonStop SSL should listen on for incoming connections.

Parameter Syntax

```
SUBNET tcpip-process-name
```

Arguments

tcpip-process-name

the name of an existing TCP/IP process on your system.

Default

If omitted, the HP NonStop SSL process will be bound to "$ZTC0".

Example

```
SUBNET $ZTC03
```

Considerations

- If you added a DEFINE =TCPIP^PROCESS^NAME to the TACL environment you use to start SSLOBJ, this setting will override the SUBNET parameter.
- If you use TCPIPV6 and want to share identical ports across multiple HP NonStop SSL processes, you need to add an identical DEFINE to all instances sharing that port as in the following example (please refer to the HP NonStop manual " TCP/IPv6 Configuration and Management Manual ", section 3, subsection "Monolithic Listening Model" for more details):

  ```
  ADD DEFINE =PTCPIP^FILTER^KEY, class map, file A1234
  ```

- If running in IPMODE DUAL, the specified subnet must support both IPv4 and IPv6.

SSLCOMSECURITY

Use this parameter to restrict the execution of SSLCOM commands.

Parameter Syntax

```
SSLCOMSECURITY boolean
```

Arguments

boolean

if set to TRUE, "sensitive" SSLCOM commands can only be executed by
a) a member of the SUPER group
b) the user under which the SSLOBJ process in running

Default

The default for this parameter is FALSE.

Example

```
SSLCOMSECURITY TRUE
```

Considerations

- The following commands are considered sensitive:
  - all SET commands
  - LOGMESSAGE, ROLLOVER LOGFILE and RELOAD CERTIFICATES

TARGETINTERFACE

Use this parameter to specify the IP address HP NonStop SSL should use for local binding of outgoing connections.

Parameter Syntax

```
TARGETINTERFACE ip-address
```

Arguments

ip-address

the IP address to bind to or “*” for none.

Default

If omitted, HP NonStop SSL will use the value of “*” and bind to no specific IP address

Example

```
TARGETINTERFACE 10.0.0.197


TARGETINTERFACE 2001:db9::1421:51ab
```

Considerations

- The parameter is relevant for the following run modes: PROXYS (outgoing socket), PROXYC (outgoing socket), FTPS (control socket connecting to FTPSERV), FTPC (control socket connecting to remote FTP server)
- Use this parameter to control which IP address HP NonStop SSL binds to for outgoing connections.
- If a host name rather than an IP address is used to configure TARGETINTERFACE, name resolution will take place only once during startup. If name resolution fails, HP NonStop SSL will terminate during startup.
- The parameter is invalid in IPMODE DUAL since no bind address except the IPv6 ANY address '::' can handle both IPv4 and IPv6.

See also

INTERFACE

TARGETHOST

Use this parameter to specify the IP host a HP NonStop SSL proxy server should route connections to.

Parameter Syntax

```
TARGETHOST ip-address
```

Arguments

Ip-address

the IP address of the target host.

Default

If omitted, the HP NonStop SSL proxy route connections to the "local loopback address" ("127.0.0.1").

Example

```
TARGETHOST 192.45.23.3
```

Considerations

- If the target server process runs on the same TCP/IP process (SUBNET) you should use the "local loopback address" ("127.0.0.1"). This is recommended for proxy servers, as it avoids that unencrypted data has to traverse the network.
- Starting with HP NonStop SSL AAE this parameter is not ignored anymore in run mode FTPC but can be used to specify the default target host in case none is given in the FTPC user command.
- You can specify a comma-separated list of multiple target hosts; see section "Multiple SSL Tunnels in a Single Process" for details.

See also

TARGETHOSTFORCE

TARGETHOSTFORCE

This FTPC only parameter can be used in combination with TARGETHOST to force the override of the targethost in the FTPC user command. HP NonStop SSL will use the TARGETHOST (if set) in FTPC to default to a certain host if none is given in the actual user command. If TARGETHOSTFORCE is specified in addition, the value of TARGETHOST will always be taken as host to connect to, no matter what the user actually specifies in the FTPC user command.

Parameter Syntax

```
TARGETHOSTFORCE {TRUE|FALSE}
```

Default

FALSE

Example

```
TARGETHOSTFORCE TRUE
```

See also

TARGETHOST, TARGETPORTFORCE

TARGETPORT

Use this parameter to specify the port number an HP NonStop SSL process should route connections to.

Parameter Syntax

```
TARGETPORT number
```

Arguments

number

the decimal number of the target TCP/IP port.

Default

If omitted, the HP NonStop SSL proxy will try route connections to the well known telnet port (23).

Example

```
TARGETPORT 1023
```

Considerations

- Starting with HP NonStop SSL AAE this parameter is not ignored anymore in run mode FTPC but can be used to specify the default port number in case none is given in the FTPC user command.
- You can specify a comma-separated list of multiple target ports; see section "Multiple SSL Tunnels in a Single Process" for details.

See also

TARGETPORTFORCE

TARGETPORTFORCE

This FTPC only parameter can be used in combination with TARGETPORT to force the override of the target port in the FTPC user command. HP NonStop SSL will use the TARGETPORT (if set) in FTPC to default to a certain port if none is given in the actual user command. If TARGETPORTFORCE is specified in addition, the value of TARGETPORT will always be taken as port to connect to, no matter what the user actually specifies in the FTPC user command.

Parameter Syntax

```
TARGETPORTFORCE {TRUE|FALSE}
```

Default

```
FALSE
```

Example

```
TARGETPORTFORCE TRUE
```

See also

TARGETPORT, TARGETHOSTFORCE

TARGETSUBNET

Use this parameter to specify the TCP/IP process a HP NonStop SSL process should use for outgoing connections.

Parameter Syntax

```
TARGETSUBNET tcpip-process-name
```

Arguments

tcpip-process-name

the name of an existing TCP/IP process on your system.

Default

If omitted, the HP NonStop SSL process will use same TCP/IP process which is configured for incoming connections (SUBNET parameter).

Example

```
TARGETSUBNET $ZTC03
```

Considerations

- If you added a DEFINE =TCPIP^PROCESS^NAME to the TACL environment you use to start SSLOBJ, this setting will override the TARGETSUBNET parameter.
- If running in IPMODE DUAL the TARGETSUBNET must support both IPv4 and IPv6 (even for IPv4 connections).

TCPIPHOSTFILE

Use this parameter to specify the value of the DEFINE=TCPIP^HOST^FILE value.

Parameter Syntax

```
TCPIPHOSTFILE hostfile | *
```

Arguments

hostfile

a hostfile to be used for DNS name resolution. The hostfile will override the value of the DEFINE =TCPIP^HOST^FILE, which may have been passed to HP NonStop SSL at startup.

*

No hostfile will be set. However, any DEFINE =TCPIP^HOST^FILE passed to HP NonStop SSL at startup will remain in effect.

Default

The default for this parameter is *.

Considerations

- See the HP NonStop manual for details of the usage of the DEFINE =TCPIP^HOST^FILE.

TCPIPNODEFILE

Use this parameter to specify the value of the DEFINE=TCPIP^NODE^FILE value.

Parameter Syntax

```
TCPIPNODEFILE nodefile | *
```

Arguments

nodefile

a node file to be used for DNS name resolution. The node file will override the value of the DEFINE =TCPIP^NODE^FILE, which may have been passed to HP NonStop SSL at startup.

*

No node file will be set. However, any DEFINE =TCPIP^NODE^FILE passed to HP NonStop SSL at startup will remain in effect.

Default

The default for this parameter is *.

Considerations

- See the HP NonStop manual for details of the usage of the DEFINE =TCPIP^NODE^FILE.

TCPIPRESOLVERNAME

Use this parameter to specify the value of the DEFINE =TCPIP^RESOLVER^NAME value.

Parameter Syntax

```
TCPIPRESOLVERNAME resolver | *
```

Arguments

resolver

a resolver to be used for DNS name resolution. The resolver will override the value of the DEFINE =TCPIP^RESOLVER^NAME, which may have been passed to HP NonStop SSL at startup.

*

No resolver will be set. However, any DEFINE =TCPIP^RESOLVER^NAME passed to HP NonStop SSL at startup will remain in effect.

Default

The default for this parameter is *.

Considerations

- See the HP NonStop manual for details of the usage of the DEFINE =TCPIP^RESOLVER^NAME.

TCPNODELAY

Use this parameter to specify whether RFC1323 will be activated on all sockets which HP NonStop SSL controls.

Parameter Syntax

```
TCPNODELAY boolean
```

Arguments

boolean

If set to TRUE or 1 or Yes, HP NonStop SSL will activate RFC1323.

Default

If omitted, HP NonStop SSL will *not* activate RFC1323.

Example

```
TCPNODELAY TRUE
```

Considerations

- If this parameter is set to true, HP NonStop SSL sets a socket option TCP_NODELAY when initializing sockets. This can help speed up throughput – please see RFC1323 and the HP NonStop "TCP/IP programming manual" for details.

TRUST

Use this parameter to specify a list of trusted CAs when running as SSL client.

Parameter Syntax

```
TRUST hashalgorithm:fingerprint [,hashalgorithm:fingerprint, ...]
```

or

```
TRUST certificate [, certificate, ...]
```

Arguments

hashalgorithm:fingerprint

the trusted CA certificate’s fingerprint generated with the hash algorithm 'hashalgorithm'.

certificate

the trusted CA certificate in PKCS-8 DER encoded format

Default

If omitted, HP NonStop SSL will not check the TLS/SSL partner’s certificate chain.

Examples

```
TRUST
WHIRLPOOL:85A8DAF0D76139154335C46E5E53C5A175CC1BDB8B7D80716CF19A93EDB75046F4BDD9BCDC005DAA5433D2D
BCE47AF0D4A2C9EB6DDBD1F94EF166308EA47FE73,
SHA256:1F4F7E0A6E1E92DDD6D5411C371C100B74DD7D32EAE7F447486AA4DAC5F43056

TRUST rootcert
```

Considerations

- The TRUST parameter can be specified in two ways: either by specifying the fingerprints of the CA certificates or by specifying a filename containing the full certificate in DER encoding. The two formats cannot be mixed.
- By default, the WHIRLPOOL hash algorithm - one of the currently strongest hash algorithms - is used. Therefore you should also specify fingerprints with their WHIRLPOOL hash. If you do want to use other hash algorithms, you have to use the HASHALGORITHMS parameter. Note that only fingerprints will be used for which the respective hash algorithm is marked as active (by including it in the HASHALGORITHMS parameter).
- If the remote SSL server is sending the complete certificate chain, the two forms of specifying the trusted CAs do not differ in functionality. Some SSL servers do not send the complete certificate chain during the handshake; for those servers the missing signing certificate(s) should be specified with the "certificate" syntax of the parameter.
- The parameter can be changed at run time using SSLCOM, please see chapter "SSLCOM Command Interface" for details.
- Due to the edit file length restriction of 255 characters, there are certain limitations for the number of fingerprints you can use in the configuration file. The following shows a table for the assumption that all fingerprints use the same algorithm. In general 5 characters of the line are required for the "TRUST". In addition to the actual fingerprint length the characters required for the <FingerprintName:> and the separator have to be considered ("add on"). Numbers in round brackets represent the number in case the old fingerprint format which is only available for SHA1 and MD5 is used.

Algorithm	Fingerprint Length	Add On	Max Fingerprints in Config
MD5*	32	5(0)	6 (7)
SHA1*	40	6(0)	5 (6)
RIPEMD160	40	11	4
SHA256	64	8	3
SHA384	96	8	2
SHA512	128	8	1
WHIRLPOOL	128	11	1

Of course you can mix fingerprints, thus if you have a WHIRLPOOL fingerprint specified, one SHA384 or one SHA256, or two RIPEMD160 fingerprints still fit within the given 255 characters.

See also

```
HASHALGORITHMS
```

Advanced Configuration Topics

Multiple SSL Tunnels in a Single Process

A single HP NonStop SSL process can listen on multiple ports at once and forward them to different IP addresses/port numbers. The following parameters are global to a single HP NonStop SSL instance:

- SUBNET
- TARGETSUBNET
- run mode

The following three parameters can be supplied as comma-separated lists:

- PORT
- TARGETPORT
- TARGETHOST

In case a comma-separated list is found, HP NonStop SSL will match the individual entries to create tuples (PORT, TARGETPORT, and TARGETHOST). Incoming connections on each PORT will then be forwarded to the matching TARGETPORT and TARGETHOST.

As an example, if you want to forward

- connections coming in on port 1023 to port 1023 on host Host23
- connections coming in on port 1024 to port 1024 on host Host24

you would start HP NonStop SSL as follows:

```
RUN HP NonStop SSL PROXYS; PORT 1023,1024; TARGETPORT 23,24; TARGETHOST Host23,Host24
```

Fault-tolerant Configuration

HP NonStop SSL services can be configured as persistent processes under control of the kernel subsystem, enabling automatic recovery from failures, such as CPU outages. The SETUP macro included with the package will guide you through the process of creating a persistent process (see chapter "Installation" for details).

Note: HP NonStop SSL cannot be run as a non-stop process. However, this is not required to achieve non-stop availability. Running as a non-stop process would not add value, as TCP sessions are reset upon CPU takeover. Non-stop availability is achieved with HP NonStop SSL as a persistent process which is automatically restarted upon failures.

Load Balancing and Fault-Tolerance of EXPAND over SSL

Using the EXPAND multi-line or multi-CPU path feature, it is possible to distribute the CPU load generated by the SSL encryption of the EXPAND traffic across multiple CPUs. Having multiple EXPAND SSL lines connecting systems will also provide fault-tolerance against CPU and other failures. If an EXPAND line goes down due to a HP NonStop SSL EXPANDS process terminating for any reason, the traffic will be redirected over the remaining lines.

EXPAND Multi-Line versus Multi-CPU Paths

The choice between Multi-Line or Multi-CPU paths (SUPERPATH) is influenced by the nature of the traffic between the systems, as well as the load-balancing and fault-tolerance goals to be achieved.

Multi-Line and Multi-CPU paths over SSL differ in the following aspects:

- CPU consumption
  Since Multi-CPU paths have a separate LH process for each line, the HP NonStop SSL processes can be configured to use the same CPU, reducing message-system hops between CPUs for the Loopback communication, resulting in a lower CPU consumption.
- Load-balancing
  A Multi-CPU path will assign a particular line to any pair of communicating processes. Hence, if a single pair of communicating process is generating a high traffic load, such as a FUP DUP or an RDF Extractor/Replicator, this traffic will burden a single CPU.
  Multi-Line paths will distribute traffic evenly across all available lines, independently of the number and CPUs of the processes communicating over EXPAND. Load will also be re-distributed dynamically and transparently, if a CPU of a HP NonStop SSL EXPANDS process is heavily loaded by processes with a higher priority. Hence, bandwidth can be preserved, even if the HP NonStop SSL processes run at a low priority to avoid impact on critical application processes.
- Fault-Tolerance
  With Multi-CPU paths, a single line is assigned to a communication link between a requestor and a server. If this line goes down, a communication error will be reported to the requestor, and the communication link will have to be re-established.
  A failure of a single line with a Multi-line path will be completely transparent to the application and the traffic will be re-routed automatically.
- Throughput
  The highest maximum throughput can be achieved with Multi-CPU paths. Measurements showed a throughput of up to 1,5 MB/s per CPU for FESA/100Mbit connected systems, with a linear scalability for multiple requestor/server pairs running in different CPUs (e.g. 6MB/s 1with 4 pairs).
  Multi-line paths have a lesser maximum throughput, as all traffic is handled by a single LH process. Measurements have shown a throughput of 1,4 MB/s for FESA/100Mbit connected systems with a single requestor server pair and a total maximum throughput of about 3 MB/s with multiple pairs.

Optimizing Throughput

The following configuration properties & setup can impact the overall throughput over an EXPAND over SSL path:

- LIF DataForwardCount (DFC) and DataForwardTime (DFT)

  Reducing the values DFC and DFT can increase the throughput for an EXPAND over SSL line. Setting DFT and DFC to the smallest possible values will minimize response time.

- CPU selection of HP NonStop SSL EXPANDS processes with multi-line paths

  Starting a HP NonStop SSL EXPANDS line process in primary CPU of a EXPAND line handler process handling multi-line path can severely decrease the overall throughput. For an optimal performance even in case of a takeover of the line handler backup, it is recommended to run the HP NonStop SSL EXPANDS processes in CPUs not used by the LH process.

Multi-Line Path Installation Sample

The following sample configuration illustrates how to optimize throughput, distribute CPU load and achieve fault-tolerance.

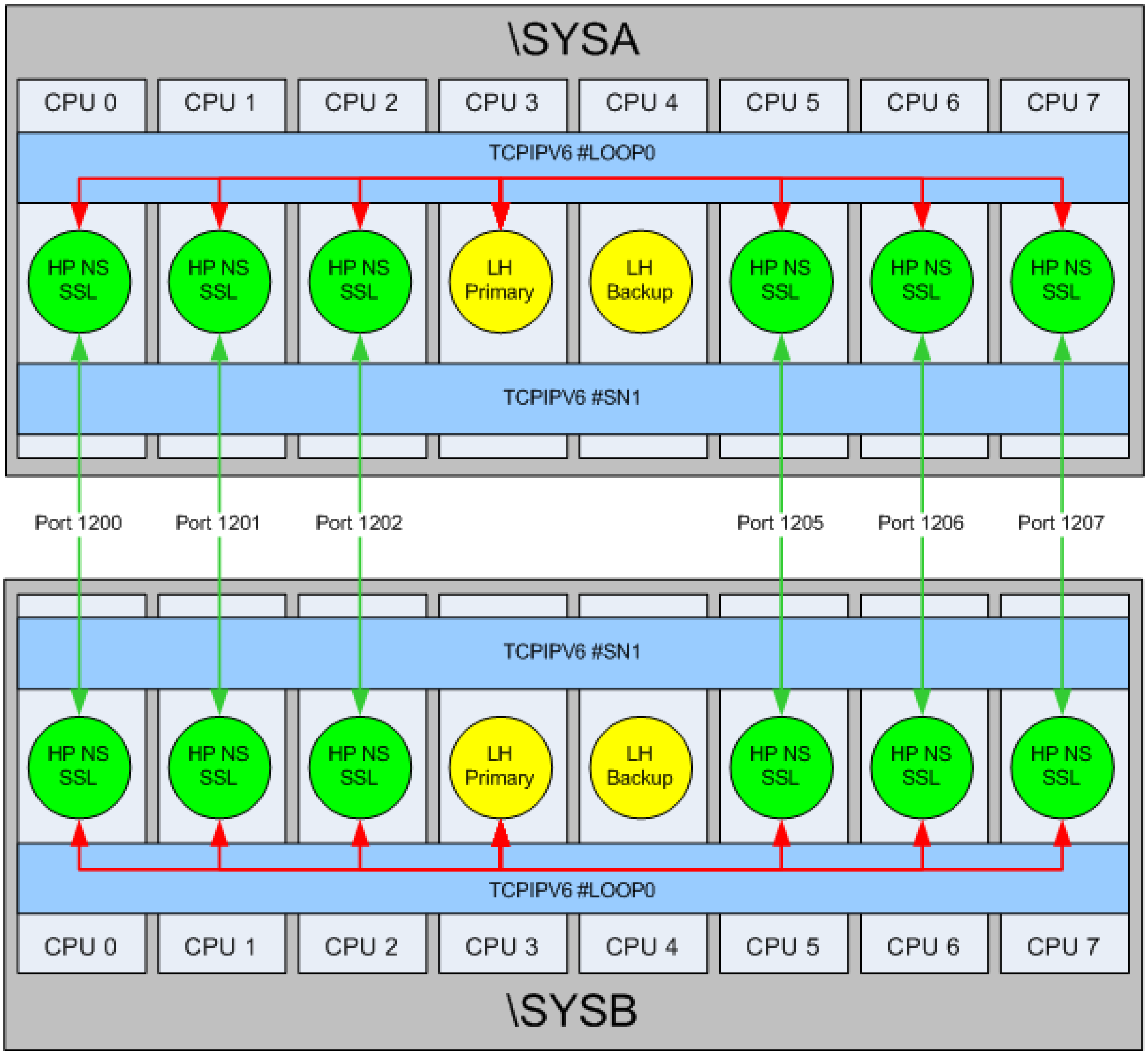
Assumptions:

- \SYSA and \SYSB to be connected over EXPAND SSL

- Systems have 8 CPUs each
- TCPIPv6

Configuration:

The following figure shows a complete setup:



The following steps have been performed for the above setup:

1. An Expand Multi-Line path was created on each system.
   - 2 CPUs were selected for the LH primary and backup.
   - To distribute SSL CPU load over the remaining CPUs, 6 lines were created for the path.
   - A unique port number was selected for each line (SRCIPPORT and DESTIPPORT can be identical).
   - DESTIPADDR of all lines was set to the loopback address (127.0.0.1).
2. Six HP NonStop SSL EXPANDS persistent processes were created on both systems.
   - A different CPU was selected for each SSL process.

- The SSL tunnel was associated to the line using the same SRCIPPORT and DESTIPPORT parameters as in the line configuration.
- The DESTIPADDR parameter of the HP NonStop SSL EXPANDS processes was set to the remote system's IP address.

Monitoring

Overview

HP NonStop SSL writes log and audit messages to a terminal, to a file, or to EMS. This is controlled by the parameters LOGCONSOLE, LOGFILE and LOGEMS for log messages and AUDITCONSOLE, AUDITFILE and AUDITEMS for audit messages. Messages can be written to any combination of those three “targets” (i.e. a single one, two of them, all of them, none of them). By default, log and audit messages are neither written to EMS nor to a log file.

Most parameters mentioned in this chapter can be configured both during startup as well as once HP NonStop SSL is running already. In the latter case, the parameters can be changed by using SSLCOM (see chapter "SSLCOM Command Interface" for details).

What is a log message?

A log message is issued by HP NonStop SSL for informational purposes, as a warning, or to indicate a fatal condition, which cannot be corrected automatically.

What is an audit message?

A audit message is issued by HP NonStop SSL for security-relevant events, such as network event (connect, disconnect), or FTP operations

Why are there three different target devices?

There are three different devices which to messages can be logged, i.e. a terminal, a file, or EMS. Operators may choose their favorite location for being alerted.

For productive installation, it is recommended to either have HP NonStop SSL log events to a file (LOGFILE, LOGFORMATFILE, LOGLEVELFILE) or to EMS (LOGEMS, LOGFORMATEMS, LOGLEVELEMS).

Log levels of these three devices can be different, i.e. can be written independently from each other.

What is a log/audit level?

A log or audit level is a number assigned to a every message in order to indicate its importance or grade of detail information. In general, a higher log or audit level for a given message indicates less importance or detail. While log or audit levels of individual messages cannot be changed, it can be controlled which levels will be displayed at all through the LOGLEVELxxx or AUDITLEVELxxx parameters.

Log and Audit Level Recommendations

The log level can be chosen individually for each log device through the parameters LOGLEVELFILE, LOGLEVELEMS and LOGCONSOLE. Depending on the device, it may be desirable to see different kind of log messages. The following table gives an indication of what “severity” individual log levels relate to:

Log Lever	Meaning
Level 0	fatal errors.
Up to level 10	only warnings.
Up to level 30	On Startup, HP NonStop SSL issues a whole set of log messages. Those will document the current version and the settings which were used to start the HP NonStop SSL process. The messages only occur once at startup.
Up to level 50	normal log messages like “close by remote client”, etc.
Up to level 89	messages only needed for trouble-shooting.
Starting from level 90	only messages to analyze extreme problems.

See the appendix for a detailed list of log messages and warnings issued by HP NonStop SSL.

Please refer to the AUDITLEVEL parameter description for recommendations and considerations for setting the audit level.

Customizing the Log and Audit Format

HP NonStop SSL allows customizing the appearance of the log or audit messages to a certain extent. For example, you may add the current date to the log message header. Please refer to the AUDITFORMATEMS, AUDITFORMATCONSOLE, AUDITFORMATFILE, LOGFORMATCONSOLE, LOGFORMATEMS, and LOGFORMATFILE parameter descriptions for details.

Using SHOWLOG to View a Log File

HP NonStop SSL processes may be configured to write log files to disk (see parameter LOGFILE). For performance reasons, those log files are created as unstructured files. While the program is running, the log file is kept open. However, it may be concurrently opened for viewing. To convert the unstructured file into a readable format, the SHOWLOG tool is supplied. Invoking SHOWLOG without arguments will display a brief syntax summary:

```
20> showlog
comForte SHOWLOG log file converter Version T9999A05_16Apr2009_comForte_SHOWLOG_
0022
usage: SHOWLOG <log file> [<process_one_line file>] [<start>] [<end>]
   <log file>      | the input log file to be converted
   <process_one_line file>  | file to write to, default is '*' meaning the home
terminal
   <start>         | either byte offset from beginning OR
                     timestamp in format "ddmmmyy HH:MM:SS.TT" (example 30Jan07 2
1:01:59.07)
   <end>           | either number of bytes after beginning OR
                     timestamp in format "ddmmmyy HH:MM:SS.TT" (example 30Jan07 2
1:01:59.07)


---examples---
SHOWLOG logfile                                   whole log file written to home
 terminal

SHOWLOG logfile logedit 10000 1000                1000 bytes starting at offset
10000
                                                  written to EDIT file logedit
```

```
SHOWLOG logfile * "30Jan07 20:00" "30Jan07 21:00" messages in timeframe to home
terminal

4>
```

If SHOWLOG is run with only the name of the log file as first runtime argument, it will dump the complete log file to the home terminal. The byte offset within the log file will be displayed regularly; this allows you to limit the output of SHOWLOG to certain sections of the log file as shown below.

```
3> SHOWLOG FTPCLOG
comForte SHOWLOG log file converter Version T9999A05_16Apr2009_comForte_SHOWLOG_0022
starting at binary offset 0
---processing in-file 'ftpclog'
$FCMH |23Jun10 12:43:09.91| 5|HP SSLOBJ version T0910H01_15Jun2010_HP_1059
$FCMH |23Jun10 12:43:09.92|10|using OpenSSL 1.0.0 29 Mar 2010 - see http://www.o
penssl.org
$FCMH |23Jun10 12:43:09.92|10|config  file: '$DATA1.T0910.FCMHCF'
$FCMH |23Jun10 12:43:09.92|10|runtime args: 'FTPC; SUBNET $ZSAM1; PORT 4021; CON
FIG $DATA1.T0910.FCMHCF'
$FCMH |23Jun10 12:43:09.93|20|--------- start settings for Logging  -----------
$FCMH |23Jun10 12:43:09.93|20| process name is $FCMH
$FCMH |23Jun10 12:43:09.94|20| trace file is '$DATA1.T0910.FTPClog' ('*' means n
one)
$FCMH |23Jun10 12:43:09.94|20| max file length 20480000 bytes, length-check ever
y 0 writes
$FCMH |23Jun10 12:43:09.94|20| console is '*' ('*' means none, '%' means home te
rminal)
$FCMH |23Jun10 12:43:09.95|20| global maximum level is 50, maximum dump length i
s 112
$FCMH |23Jun10 12:43:09.95|20|--------- end settings for Logging -------------
$FCMH |23Jun10 12:43:09.96|10|log level for console is 50
$FCMH |23Jun10 12:43:09.96|10|log level for logfile is 50
$FCMH |23Jun10 12:43:09.96|10|log level for EMS     is 10
$FCMH |23Jun10 12:43:09.97|10|global log max level     is 50
$FCMH |23Jun10 12:43:09.97|10|global trace max level     is -1
$FCMH |23Jun10 12:43:09.98|30|starting collecting of random data
$FCMH |23Jun10 12:43:13.12|10|collection of 64 bytes random data finished
$FCMH |23Jun10 12:43:13.14|30|dumping configuration:
[def  ] ALLOWCERTERRORS      <none>
[def  ] ALLOWIP              <*>
...
[def  ] TRUST                <*>
$FCMH |23Jun10 12:43:13.15|50|TCP_NODELAY is off
$FCMH |23Jun10 12:43:13.15|30|--- creating new SSL client context ---
$FCMH |23Jun10 12:43:14.78|50|minv=30, maxv=31, meth=SSLv23_client_method(), ssl
_options=0x81010fff
$FCMH |23Jun10 12:43:14.78|50|OpenSSL cipherstring 'RC4-MD5:DES-CBC3-SHA:RC4-SHA
:'
$FCMH |23Jun10 12:43:14.79|30|no client certificates are configured
$FCMH |23Jun10 12:43:14.79|30|trusted fingerprints are <*>
$FCMH |23Jun10 12:43:14.80|30|--- new SSL client context built
$FCMH |23Jun10 12:43:14.80|20|parameter SUBNET was evaluated
$FCMH |23Jun10 12:43:14.80|20|TCP/IP process is $ZSAM1
$FCMH |23Jun10 12:43:14.81|20|SSL buffer size is:      SSLBUF=       13000
$FCMH |23Jun10 12:43:14.81|20|Socket buffer sizes are: SOCKETSNDBUF=     0, SOCK
ETRCVBUF=     0
$FCMH |23Jun10 12:43:14.82|20|TCP buffer sizes are:    TCPSNDBUF=    16384, TCPR
CVBUF=    16384
$FCMH |23Jun10 12:43:14.82|20|FTP client proxy started on source port 4021
$FCMH |23Jun10 12:45:21.51|50|F1|--> connection from client established
$FCMH |23Jun10 12:45:21.52|50|F1|<-- sending proxy FTP Welcome
$FCMH |23Jun10 13:05:00.64|50|P1|> client 127.0.0.1:4021<--127.0.0.1:4890 closed
 connection
$FCMH |23Jun10 13:05:00.65|50|P1|> closing server :<?>:

---
---Byte offset is 24144
---
```

```
---
--- EOF reached, done
---
```

The second runtime argument can be used to create a new EDIT file containing the log file contents. The following example shows how to convert the whole log file into an edit file (note that this can take some time for large files):

```
42> showlog pxyslog logedit
comForte SHOWLOG log file converter Version T9999A05_16Apr2009_comForte_SHOWLOG_
0022
starting at binary offset 0
starting at offset 0
writing out-file 'logedit'
---processing in-file 'pxyslog'

---
--- EOF reached, done
---
43> fi logedit
$data1.ssl
              CODE              EOF   LAST MODIFIED  OWNER  RWEP   PExt   SExt
logedit        101          5506688 23Jun2010 13:05 110,110 aaaa      4     16
44>
```

The third and last runtime arguments can be used to limit the part of the file which is converted. This is helpful for the viewing large log files. The following example shows dumping a large log file. Only a limited number of log messages (totaling 10000 bytes) after a given offset (5000000) are shown:

```
33> run showlog telslog * 5000000 10000
comForte SHOWLOG log file converter Version T9999A05_16Apr2009_comForte_SHOWLOG_
0022
dumping at most 10000 bytes
---processing in-file 'telslog'

(output not shown here)

---
---finishing dump of file before end-of-file
---

---done 34>
```

Rather than using byte offsets, SHOWLOG can also use timestamp as filters for which parts of the log file to display. The command

```
SHOWLOG logfile * "30Jan07 20:00" "30Jan07 21:00"
```

will only display log messages between the two given timestamps.

Note: By using '*' as the second runtime argument the output is written to the home terminal. When using the byte offset parameter or the byte offset parameter and length parameter, the out file parameter must be entered as well.

Viewing File Contents from OSS

The log or audit files created by SSH2 are unstructured files and can be viewed from OSS with standard OSS tools such as *more* or *tail*. Standard OSS filter tools such as *grep*, *awk*, or *wc* can also be applied. This allows users to make use of the powerful Unix syntax for doing text processing.

Logfile/Auditfile Rollover

When logging to a file, HP NonStop SSL uses round-robin to switch to a new file. Logfile rollover applies both to auditing (to the file configured with the AUDITFILE parameter) as logging (to the file configured with the LOGFILE parameter).

A logfile rollover occurs when the logfile is greater than the size configured in the parameter LOGMAXFILELENGTH or when the audit file is greater than the size configured in the parameter AUDITMAXFILELENGTH.

HP NonStop SSL will round-robin over at least 10 files. The number of files can be configured using the LOGFILERETENTION (or AUDITFILERETENTION) parameter.

Archive files created during rollover will be created by appending a number to the log file name. The number of digits of the number appended will be calculated depending on the number of files to keep.

With LOGFILERETENTION set to 10 (the default value), the archive files for a LOGFILE of SSLLOG will be called SSLLOG0, SSLLOG1, ... SSLLOG9. With LOGFILERETENTION set to 1000, the archive files for a LOGFILE of SLOG will be called SLOG000, SLOG001, ... SLOG999.

SSLCOM Command Interface

Using SSLCOM, you can:

- get an overview of the status of a HP NonStop SSL process
- list sessions which are currently open and obtain detailed information about single sessions (limited to certain run modes)
- view and change the following parameters (please refer to the "Parameter Reference" for the meaning of the parameters):
    - ALLOWCERTERRORS
    - ALLOWIP
    - CONNECTIONINFOFORMAT[DETAILED]
    - CONTENTFILTER
    - DENYIP
    - EXPANDCOMPRESSION
    - LOGCONSOLE
    - LOGEMS
    - LOGFILE
    - LOGFORMATCONSOLE
    - LOGFORMATFILE
    - LOGFORMATEMS
    - LOGLEVELFILE
    - LOGLEVELCONSOLE
    - LOGLEVELEMS
    - LOGMEMORY
    - MAXSESSIONS
(only in applicable run modes)
    - TRUST
(only in run modes ending with a "C" and in run mode EXPANDS)
- execute the following additional commands
    - LOGMESSAGE
    - RELOAD CERTIFICATES

- SSLINFO

Usage of SSLCOM: a Sample Session

The usage of SSLCOM is similar to the HP PATHCOM program. You connect to an existing HP NonStop SSL instance using the OPEN command, then you issue commands against that instance of HP NonStop SSL. The HELP command will give you a brief overview of the supported commands.

The following example session illustrates how to:

1. Start SSLCOM and connect to a HP NonStop SSL instance running with the process name "$TELS"
2. Use the STATUS command to view the current status of HP NonStop SSL
3. Use the SHOW command to view the current settings of LOGLEVEL, LOGCONSOLE, LOGFILE and LOGMEMORY
4. Use the SET command to change the value of the LOGLEVEL parameter.

```
15> SSLCOM $TELS
GFTCOM^H16^06FEB03
OPEN $ TELS
% status
status
-----------------------------------------------------------
HP NonStop SSLOBJ version T9999G06_15Sep2003_comForte_SSLD_S40_1031
-----------------------------------------------------------
Startup configuration:
[def  ] ALLOWIP              <*>
[def  ] CACERTS              <CACERT>
[def  ] CIPHERSUITES         <0.4,0.10,0.5>
[def  ] DELAYRECEIVE         <0>
[def  ] DENYIP               <>
[def  ] LICENSE              <LICENSE>
[par  ] LOGCONSOLE           <*>
[run  ] LOGFILE              <lproxysl>
[def  ] LOGFORMAT            <76>
[def  ] LOGLEVEL             <50>
[def  ] LOGMAXDUMP           <100>
[def  ] LOGMAXFILELENGTH     <20000>
[def  ] LOGMEMORY            <0>
[def  ] MAXVERSION           <3.1>
[def  ] MINVERSION           <3.0>
[run  ] PORT                 <32005>
[def  ] RANDOMFEED           <64>
[def  ] SERVCERT             <SERVCERT>
[def  ] SERVKEY              <SERVKEY>
[def  ] SERVKEYPASS          <??11??>
[def  ] SLOWDOWN             <0>
[def  ] SUBNET               <$ZTC0>
[def  ] TARGETHOST           <127.0.0.1>
[run  ] TARGETPORT           <65023>
[def  ] TARGETSUBNET         <$ZTC0>
[def  ] TESTWRONGDATASOCKET  <0>

-----------------------------------------------------------
PROXYS mode
    active sessions right now:          3
    maximum number of active sessions:  25
-----------------------------------------------------------
current heap size: 2506752
current mem pages: 115
-----------------------------------------------------------
-----------------------------------------------------------
Root Certificate Info:
MD5 fingerprint   <4DFF502FD33EB41911ACE1943DB3DCCA>
SHA-1 fingerprint <A71418323DDCD3140460125D3321503EB2356FE9>
```

```
------------------------------------------------------------
% show
```

```
show
LOGLEVEL     50
LOGFILE      lproxysl
LOGCONSOLE   *
LOGMEMORY    0
% set loglevel 30
set loglevel 30
log level was set to 30
% exit
exit
16>
```

Supported Commands

The following commands are supported:

- OPEN <processname>: connects to an instance of HP NonStop SSL running. The process name may also be supplied as runtime parameter as shown in the example above.
- HELP: lists supported commands.
- STATUS: shows current status. This includes the display of the following information:
  - The startup configuration of HP NonStop SSL.
  - The current configuration of HP NonStop SSL. The current configuration will differ from the startup configuration when SET commands have been used from within SSLCOM to change values.
  - In run modes ending with an "S", the fingerprint of the root certificate will be displayed.
  - The number of sockets as well as the CPU ms used by HP NonStop SSL will be displayed.
- SHOW: shows current values of parameters which can be altered using SSLCOM.
- SET <parameter> <value>: changes a parameter.
- SSLINFO: displays the local certificate chain when HP NonStop SSL is running as SSL server.
- RELOAD CERTIFICATES: changes the server certificate chain at run time.
- CONNECTIONS [, DETAIL]: display on overview of the current open connections of HP NonStop SSL.
- CONNECTIONS, STATS: displays an extended usage statistics for the run modes PROXYS and PROXYC. This statistic will yield information on how many different remote IP addresses are connecting to HP NonStop SSL.
- INFO CONNECTION: displays detailed information about a single connection.
- RENEGOTIATE CONNECTION: forces SSL key renegotiation for a single connection.
- LOGMESSAGE <level> <text>: a log message with the level and text specified will be generated. This allows testing the current log settings.
- ROLLOVER LOGFILE: a log file rollover will be enforced regardless of the current size of the log file.
- ROLLOVER AUDITFILE: an audit file rollover will be enforced regardless of the current size of the audit file.
- CONTENTFILTER "<filename>": refreshes content filter file.
- CONTENTTEST "<string>": tests <string> against the current content filter rules.

- PROCESSINFO: displays some details about CPU and memory usage.
- STATISTICS [,RESET]: displays status and additional statistics for some run modes. When used with RESET option, resets all statistics counters.

Multiple commands can be concatenated with semicolons in-between.

The CONNECTION Commands

In the run modes TELNETS, PROXYS, PROXYC, FTPS and FTPC, HP NonStop SSL will have a set of TCP/IP connections open during normal operation. The number of open connections can vary between zero and several hundred. With the commands described in the following sections, HP NonStop SSL can display information about the connections.

CONNECTIONS

The CONNECTIONS command displays an overview of all currently open connections handled by HP NonStop SSL. The following example shows the output of HP NonStop SSL running in TELNETS mode with three proxy connections handled by HP NonStop SSL:

CONNECTIONINFOFORMAT EXTENDED (default starting HP NonStop SSL AAE):

```
% connections
connections
+-----+------------------------------------------------------------------------+
| Port| Connection Information                                                 |
+-----+------------------------------------------------------------------------+
| 6828| Incoming peer  : 192.168.113.4:37638                                   |
|     | Incoming local : 10.0.0.194:11011                                      |
|     | Outgoing local : 127.0.0.1:6828                                        |
|     | Outgoing peer  : 127.0.0.1:23                                          |
+-----+------------------------------------------------------------------------+
| 6829| Incoming peer  : 192.168.113.4:37640                                   |
|     | Incoming local : 10.0.0.194:11011                                      |
|     | Outgoing local : 127.0.0.1:6829                                        |
|     | Outgoing peer  : 127.0.0.1:23                                          |
+-----+------------------------------------------------------------------------+
| 6830| Incoming peer  : 192.168.113.4:37641                                   |
|     | Incoming local : 10.0.0.194:11011                                      |
|     | Outgoing local : 127.0.0.1:6830                                        |
|     | Outgoing peer  : 127.0.0.1:23                                          |
+-----+------------------------------------------------------------------------+
+--------------------------------- END ----------------------------------------+
%
```

CONNECTIONINFOFORMAT ORIGINAL (default before HP NonStop SSL AAE):

```
% connections
connections
| Port|--------remote connection-------------|----------local connection---------|
| 6831|10.0.0.194:11011<--192.168.113.4:37706|127.0.0.1:6831-->127.0.0.1:23      |
| 6832|10.0.0.194:11011<--192.168.113.4:37707|127.0.0.1:6832-->127.0.0.1:23      |
| 6833|10.0.0.194:11011<--192.168.113.4:37708|127.0.0.1:6833-->127.0.0.1:23      |
%
```

CONNECTIONINFOFORMAT CSV:

```
% connections
connections
```

```
Port,Local Conn. Local IP, Local Conn. Local Port,Direction,Local Conn. Remote IP,
Local Conn. Remote Port, Direction, Remote Conn. Local IP, Remote Conn. Local Port,
Remote Conn. Remote IP, Remote Conn. Remote Port
6837,10.0.0.194,11011,<--,192.168.113.4,37814,127.0.0.1,6837,-->,127.0.0.1,23
6838,10.0.0.194,11011,<--,192.168.113.4,37815,127.0.0.1,6838,-->,127.0.0.1,23
6839,10.0.0.194,11011,<--,192.168.113.4,37817,127.0.0.1,6839,-->,127.0.0.1,23
%
```

Note: The first column contains the local port of the connection. This number is used to access an individual session with the INFO CONNECTION or RENEGOTIATE CONNECTION commands.

CONNECTIONS, DETAIL

The CONNECTIONS, DETAIL command displays the list of connection with some additional information to each line.

CONNECTIONINFOFORMATDETAILED EXTENDED (default since HP NonStop SSL AAE):

```
% connections, detail
connections, detail
+-----+-----------------------------------------------------------------------+
| Port| Connection Information                                                |
+-----+-----------------------------------------------------------------------+
| 6837| Incoming peer  : 192.168.113.4:37814                                  |
|     | Incoming local : 10.0.0.194:11011                                     |
|     | Outgoing local : 127.0.0.1:6837                                       |
|     | Outgoing peer  : 127.0.0.1:23                                         |
|     | Handshake(s)   : 1                                                    |
|     | First Handshake: 30Jul12-11:25:09                                     |
|     | Last Handshake : 30Jul12-11:25:09                                     |
+-----+-----------------------------------------------------------------------+
| 6838| Incoming peer  : 192.168.113.4:37815                                  |
|     | Incoming local : 10.0.0.194:11011                                     |
|     | Outgoing local : 127.0.0.1:6838                                       |
|     | Outgoing peer  : 127.0.0.1:23                                         |
|     | Handshake(s)   : 1                                                    |
|     | First Handshake: 30Jul12-11:25:12                                     |
|     | Last Handshake : 30Jul12-11:25:12                                     |
+-----+-----------------------------------------------------------------------+
| 6839| Incoming peer  : 192.168.113.4:37817                                  |
|     | Incoming local : 10.0.0.194:11011                                     |
|     | Outgoing local : 127.0.0.1:6839                                       |
|     | Outgoing peer  : 127.0.0.1:23                                         |
|     | Handshake(s)   : 1                                                    |
|     | First Handshake: 30Jul12-11:25:14                                     |
|     | Last Handshake : 30Jul12-11:25:14                                     |
+-----+-----------------------------------------------------------------------+
+--------------------------------- END ---------------------------------------+
%
```

Note: The EXTENDED format can be viewed with the common width of 80 characters on the terminal emulator

CONNECTIONINFOFORMATDETAILED ORIGINAL (default before HP NonStop SSL AAE):

Note: The output of the command is best viewed with a terminal emulator displaying 132 characters per line.

```
% connections, detail
connections, detail
| Port|--------remote connection----------|----------local connection---------|#HS|First-Handshake-|Last--Handshake-|
| 6843|10.0.0.194:11011<--192.168.113.4:38002|127.0.0.1:6843-->127.0.0.1:23      |  1|30Jul12-11:39:39|30Jul12-11:39:39|
| 6844|10.0.0.194:11011<--192.168.113.4:38003|127.0.0.1:6844-->127.0.0.1:23      |  1|30Jul12-11:39:41|30Jul12-11:39:41|
| 6845|10.0.0.194:11011<--192.168.113.4:38004|127.0.0.1:6845-->127.0.0.1:23      |  1|30Jul12-11:39:43|30Jul12-11:39:43|
%
```

CONNECTIONINFOFORMATDETAILED CSV:

```
% connections, detail
connections, detail
Port,Local Conn. Local IP, Local Conn. Local Port,Direction,Local Conn. Remote IP, Local Conn. Remote Port, Direction, Remote Conn. Local
IP, Remote Conn. Local Port, Remote Conn. Remote IP, Remote Conn. Remote Port,Handshake(s),First Handshake,Last Handshake
6840,10.0.0.194,11011,<--,192.168.113.4,37950,127.0.0.1,6840,-->,127.0.0.1,23,1,30Jul12-11:36:15,30Jul12-11:36:15
6841,10.0.0.194,11011,<--,192.168.113.4,37951,127.0.0.1,6841,-->,127.0.0.1,23,1,30Jul12-11:36:17,30Jul12-11:36:17
6842,10.0.0.194,11011,<--,192.168.113.4,37952,127.0.0.1,6842,-->,127.0.0.1,23,1,30Jul12-11:36:19,30Jul12-11:36:19
%
```

Note: The content at the right end of the display is the abbreviated content of the section "SSL handshake information" in the result of the INFO CONNECTION command covered in the next paragraph.

INFO CONNECTION

The INFO CONNECTION command displays detailed information about a single session as in the following example:

```
% info connection 3625
info connection 3625
accepting socket:
=================
   <Sec rem acc PROXY>[TLS_SERVER](0/1): 10.0.0.198:8989<--10.0.1.24:2000
connecting socket:
==================
   <Pln loc conn PROXY>: 127.0.0.1:3625-->127.0.0.1:23
peer certificate information:
=============================
   issuer=/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98/CN=VeriSign C

lass 1 CA Individual Subscriber-Persona Not Validated
subject=/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=www.verisign.com/repository/RPA Incorp. by Ref.,LIAB.LTD(c)98/OU=Persona Not

Validated/OU=Digital ID Class 1 - Microsoft Full Service/CN=Thomas R. Burg/emailAddress=thomasburg@web.de
not_valid_before=Feb 20 00:00:00 2004 GMT
not_valid_after=Feb 19 23:59:59 2005 GMT
md5=C7D442A51F7790721E3F36C383E58DF5
SSL handshake information:
==========================
   1 SSL handshakes; First at 05Aug04,21:26:23, Last at 05Aug04,21:26:23
%
```

The command displays details about:

- Accepting socket: the socket of the application which connects to HP NonStop SSL. For instance in TELNETS mode, that is the connection to the remote client using SSL
- Connecting socket: the socket on which HP NonStop SSL connects to the target application. In TELNETS mode, that is the connection to TELSERV
- Peer certificate information: if the accepting socket in TELNETS or PROXYS mode has sent a client certificate, the contents are displayed here. See section "Requesting the SSL Client to Present a Client Certificate" for details on enforcing client authentication.
- SSL handshake information: displays the number of SSL handshakes on the accepting socket and the timestamp of the first and last handshake.

RENEGOTIATE CONNECTION

The SSL protocol allows both parties to initiate a new SSL handshake to refresh the session keys. The RENEGOTIATE CONNECTION command lets HP NonStop SSL do that from the server side. The following two log messages show that a renegotiation has been successful.

```
22:34:08.19|50|T3|session 10.0.0.198:8989<--10.0.1.24:2002: SSL renegotiation          starting
22:34:10.35|50|T3|session 10.0.0.198:8989<--10.0.1.24:2002: cipher suite TLSv1/RC4-MD5 negotiated
```

The output of the INFO CONNECTION command will display the fact that a new handshake has happened as well:

```
%info connection 3625
info connection 3625
accepting socket:
=================
   <Sec rem acc PROXY>[TLS_SERVER](0/1): 10.0.0.198:8989<--10.0.1.24:2000
connecting socket:
==================
   <Pln loc conn PROXY>: 127.0.0.1:3625-->127.0.0.1:23
peer certificate information:
=============================
   issuer=/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98/CN=VeriSign C

lass 1 CA Individual Subscriber-Persona Not Validated
subject=/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=www.verisign.com/repository/RPA Incorp. by Ref.,LIAB.LTD(c)98/OU=Persona Not

Validated/OU=Digital ID Class 1 - Microsoft Full Service/CN=Thomas R. Burg/emailAddress=thomasburg@web.de
not_valid_before=Feb 20 00:00:00 2004 GMT
not_valid_after=Feb 19 23:59:59 2005 GMT
md5=C7D442A51F7790721E3F36C383E58DF5
SSL handshake information:
==========================
   2 SSL handshakes; First at 05Aug04,21:26:23, Last at 05Aug04,22:38:07
%
```

SSLINFO Command

The SSLCOM command SSLINFO will display the local certificate chain configured through the parameters SERVCERT and CACERTS when HP NonStop SSL is running as an SSL server.

RELOAD CERTIFICATES Command

The SSLCOM command RELOAD CERTIFICATES allows the changing of the server certificate chain without having to restart HP NonStop SSL. The command has two possible syntaxes:

1. If used without an additional parameter, the command assumes the configuration parameters for the new certificate chain (SERVCERT, SERVKEY, SERVKEYPASS, CACERTS) are present in the currently configured CONFIG2 file. If no CONFIG2 file has been configured for startup, the command will fail.
2. If used with an additional parameter containing the filename of a configuration file in double quotes, the new values will be loaded from that file.

Some considerations for the command:

- The success or failure of the command will be returned to SSLCOM. If the command fails, the prior certificate chain will remain active.
- HP NonStop SSL does some limited tests on the new certificate chain. However, some errors in the certificate chain cannot be detected by merely loading the certificates. It is thus recommended to immediately check the new certificate chain with the SSLINFO command as well as with creating a new client connection.
- If the syntax 2 of the command is used and if the name of the config file in the command differs from the startup CONFIG2 file, the changes will *not* be permanent unless the startup configuration of HP NonStop SSL is updated with the changes. It is highly recommended to always keep the certificate chain information in a CONFIG2 and to use syntax a) as in that case the changes *will* be permanent without further action.

SSL Reference

Secure Sockets Layer

The SSL (secure sockets layer) protocol is an open, non-proprietary protocol originally designed by Netscape. It has been standardized by the IETF as Transport Layer Security (TLS) protocol. SSL has been universally accepted on the Internet for authenticated and encrypted communication between clients and servers and is used in millions of browsers around the world.

HP NonStop SSL implements SSL using OpenSSL (© acknowledged).

SSL Features

The SSL protocol has the following basic properties:

- Privacy
  After an initial handshake, client and server agree on a session key which is used for a symmetric cipher algorithm to encrypt the session's payload. Example ciphers are RC4, 3-DES or AES.
- Mutual Authenticity
  Using a public-key cryptography and digital signatures, the SSL protocol allows to authenticate the server or client before exchanging confidential data.
- Session Integrity
  SSL ensures the integrity of the messages exchanged allowing client and server to verify if it has been modified by an attacker, using a Message Authentication Code (MAC). Example MAC algorithms are MD5 or SHA.

Further References

For more information on SSL we recommend the following reading:

- Eric Rescorla, "SSL and TLS: Designing and Building Secure Systems", Addison-Wesley Professional 2000
- Stephen Thomas, "SSL and TLS essentials", Wiley Publishing 2000
- http://en.wikipedia.org/wiki/Transport_Layer_Security

For information on public key cryptography and digital signatures on relation to SSL please read:

- Russ Housley: "Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure", Wiley Publishing 2001
- SSL Certificates and PKI in the NonStop World - and Other Worlds, The Connection May/June 2004.
- http://en.wikipedia.org/wiki/Pki

Implementation Overview

Cipher Suites

HP NonStop SSL uses the SSL protocol - as used in standard browsers and servers - for session security. It supports SSL 2.0, SSL 3.0 and the latest version SSL 3.1, which has been standardized by the IETF as Transport Layer Security (TLS) protocol. This protocol allows for negotiating cipher suites for secure exchange of data as well as exchanging the necessary secrets at the beginning of each session in a way which is particularly strengthened against replay, insertion and man-in-the-middle attacks.

Note: Usage of SSL 2.0 is not recommended as it has some serious design flaws.

The selection of cipher suites is configurable, in order to make our solution tolerable to the needs of individual security requirements:

- RSA certificate-based key-exchange, where the server certificate is validated in the SSL client.
- Either of 3-DES, RC4 or AES as bulk-ciphers.
- Either of HMAC-SHA or HMAC-MD5 as message authentication codes.
- The actual choice of the cipher suite is at the discretion of the server and configurable.

The key lengths for symmetric encryption are:

- (Triple-DES) 3x56 = 168 bits.
- RC4 = 128 bits
- AES = 128 or 256 bits

The key lengths for message authentication are:

- (HMAC-MD5)= 128 bit
- (HMAC-SHA)= 160 bit

  The cipher block chaining mode (CBC) in 3-DES guarantees the utmost security against replay/insertion as well as brute force attacks. At the current state of computer technology triple encryption is no longer a (speed) obstacle.

  The authenticity of messages is granted by the 160 bit SHA hash algorithm. (HMAC-SHA) or by the 128 bit MD5 hash algorithm (HMAC-MD5).

  Modulus lengths of up to 8192 bits are supported for public key values.

Auditing

An indispensable part of every security strategy is Security Auditing. The TLS protocol defines 23 Alert Messages, which may be sent or received. All these alerts are handled by HP NonStop SSL; most of them are fatal for the connection. HP NonStop SSL logs these alerts to the configured log targets e.g. on the console.

X.509 Certificates

Certificates are a form of digital id issued by a certificate authority. A certificate authority signs a certificate with its private key, vouching for the correctness of the certificate contents. Certificates used with SSL are standardized by the X.509 specification. It is possible to build hierarchies of certification authorities, where the top level authority is called the root CA. The root CA's certificate is issued by the root CA itself; it is a so called self-signed certificate.

For SSL, the certificates are used to provide mutual authenticity. Before establishing a session, clients can authenticate a server to ensure it is connecting to a trusted site (SSL server authentication). In this case the server presents its "server certificate" along with the "certificate chain" to the client. The certificate chain is a series of certificates issued by successive CAs that reflect the certificate hierarchy up to the root certificate

Vice versa, the server can optionally request the client to present a certificate for authentication (SSL client authentication, this is currently not supported by HP NonStop SSL).

HP NonStop SSL supports X.509 certificates for server authentication as follows:

- If HP NonStop SSL is running as SSL server (run modes FTPS, TELNETS, PROXYS, EXPANDS) HP NonStop SSL will send the configured server certificates to the client. It is up to the client to check for the proper server certificates. The certificates are configured using the parameters SERVKEY, SERVKEYPASS, SERVCERT and CACERTS; please see in the parameter reference for usage of those parameters. Please see the next section on how to generate your own certificates.
- If HP NonStop SSL is running as a SSL client (run modes FTPC, PROXYC), the TRUST parameter is used to configure a list of trusted root certificates. It is up to the SSL server to send the certificates; HP NonStop SSL will validate the integrity of the certificate chain and check if the root certificate's fingerprint is configured in the TRUST parameter. Note that the default value * for the TRUST parameter is interpreted as "do not validate the remote certificate".

Configuring SSL for Production as SSL Server

The default installation of HP NonStop SSL is streamlined to enable an easy setup and immediate testing. HP NonStop SSL is delivered with a set of certificate and key files which can be used out-of-the-box for testing and evaluation purposes.

For a secure production installation, it is mandatory to configure HP NonStop SSL to use your own certificate and key files. Using the default files and settings for a production installation may compromise the security of the system. OpenSSL toolkits, available as shareware, can help you generate your own SSL certificate.

This section will describe how to generate your own certificates. It also explains how HP NonStop SSL is configured to use these certificates for a production installation. For a more detailed explanation about the concept of certificates, see the section "X.509 Certificates" of this chapter.

Using Your Own Server Key and Certificate Files

You will need at least the following components to configure SSL Server Authentication with your own production certificates:

1. A private key (protected by a pass phrase).
2. A server certificate incorporating the public key matching the private key.
3. The certificate of the root CA that issued (i.e. signed) the server certificate.

To obtain the certificates required for SSL server authentication you may choose one of the following options:

- Purchase a server certificate from a commercial CA
- Obtain a server certificate from an existing internal Certificate Authority of your organization.
- Be your own (root) Certificate Authority to issue a server certificate.

Which option you choose for your production system depends on the nature of your application, the type of users accessing it and on the existing security infrastructure.

If your organization already maintains an internal public key infrastructure (PKI), you would want to obtain a server certificate from an internal CA.

If your server is accessed by external internet users (e.g. customers) that do not know your organization yet, you would probably purchase a server certificate. Remember certificates are used to establish trust. The users trust the CA you purchased your server certificate from, while the CA vouches for your certificate's correctness.

If you want to secure access to an application for internal users only, you would probably prefer using your own root CA to issue the server certificate. As your users know your organization already, they can choose to trust your root CA that issued the server certificate.

The Public/Private Key Pair

Regardless of how you choose to obtain a certificate, you will need to generate a private/public key pair. The private key is stored in encrypted format protected by a pass phrase in a file complying to the PKCS#8 standard. This file is later passed to a secure HP NonStop SSL process with the SERVKEY/CLIENTKEY parameter. For HP NonStop SSL to be able to decrypt the private key, the password must be specified by the SERVKEYPASS/CLIENTKEYPASS parameter.

Warning: Do not give other users access to your private key! In general, private keys should be encrypted for security. The longer your pass-phrase is, the better the protection of your keys.

The public key matching the private key is incorporated into the certificate along with your identification data (the server's X.509 "distinguished name").

The Certificate Signing Request

To obtain a certificate you submit your public key along with some identification data to a Certificate Authority. This so called Certificate Signing Request (CSR) is used by the CA to generate your certificate and sign it with the CA's own private key. CA's expect the CSR to adhere to a certain format. The most widely used format is specified by the PKCS#10 standard.

Obtaining a Certificate from a Third Party CA

In case you choose to obtain a certificate from an internal or external (commercial) CA, you would generate a private key and a PKCS#10 CSR. You will then submit the CSR to the CA, typically by pasting it in BASE64-encoded format to the CA's web site, or sending it via email. The CA will then return the signed certificate to you, typically also in BASE64 encoded format attached to an email. The BASE64-encoded certificate can then be converted to binary certificate file, which is passed to HP NonStop SSL with the SERVCERT/CLIENTCERT parameter.

HP NonStop SSL needs to send the root CA certificate along with the server/client certificate to SSL clients/server for validation. Typically, the third party CA will provide their public root certificate that was used to sign the certificate. To be able to pass the root CA certificate to HP NonStop SSL with the CACERTS parameter, the root CA certificate file need to be uploaded to the system you have HP NonStop SSL installed on. If you received the root CA certificate in BASE64-encoded format, you may convert for HP NonStop SSL usage just like the BASE64-encoded server certificate.

Acting As Your Own CA

If you choose to issue a certificate as your own CA, you would need to generate a root CA certificate and private key. The root CA certificate is a "self-signed" certificate as it is signed with the root CA's own private key.

Warning: Do not give other users access to your root CA private key! If this key is compromised, malicious users can create certificates that will appear to be signed by your CA certificate. In general, private keys should be encrypted for security. The longer your pass-phrase is, the better the protection of your keys. The root CA's private key should also be stored at a secure place. For example, you could store it on a removable disk that you can lock away.

Using the root CA private key and certificate you would then generate a certificate from a previously created CSR. In other words, you would perform the same task as a third party CA.

Example: How to Generate SSL Certificates Using OpenSSL

This example shows how to create a self-signed CA certificate and a server certificate signed by the CA certificate, and how to convert the certificates into the format used by HP NonStop SSL, as well as setting the appropriate configuration parameters. The example also shows how to create a signing request to be submitted to a Certification Authority (CA).

Many customers require server certificates to be signed by a trusted CA (e.g. Verisign, Thawte, etc.). Some customers also purchase an intermediate CA certificate from a trusted CA so they can sign/issue their own corporate server certificates with the intermediate CA certificate purchased from a trusted CA. Note that generating self-signed certificates or certificates signed by an OpenSSL self-generated CA (as in the example) may not be appropriate for a customer's security environment. Customers should only use this procedure if their security environment allows generating their own CA certificate and server certificates, otherwise they should consult their security department for recommended actions to obtain SSL certificates.

When submitting a signing request to a trusted CA authority, the steps for generating a CA certificate are skipped, and all steps from "To Generate the Server Certificate" are executed except steps (4) and (5). The signing request (CSR) is sent to the CA, and the signed server certificate and the root certificate are returned. If the certificates are returned in PEM format, they must be converted to DER format.

To Prepare for Certificate Generation

1. Obtain a copy of OpenSSL. For Unix/Linux systems, build it from source code or obtain it via your package manager. For Windows-based systems, it is best to install the prebuilt OpenSSL for Windows binaries, see http://www.openssl.org/related/binaries.html. In this example, OpenSSL version 1.0.0e (6 Sep 2011) for Windows was used. The commands shown apply for Unix/Linux systems as well.

2. Create a directory OpenSSL_certificates, and within it directories "ca", "server", and "newcerts".

3. All OpenSSL commands shown in this example except the genrsa command require a configuration file. By default, a file named openssl.cfg is expected in the directory where the OpenSSL binary resides. A different filename and location can be specified with the -config option. Windows OpenSSL implementations usually include a configuration file, but if not, an example can be easily obtained by searching the Internet for "openssl.cfg example". Note that for OpenSSL versions 0.9.8g and lower the default config file name is "openssl.cnf". Save the config file example and edit a copy of it as desired. In this certificate generation example, the "dir" statement in the config file was set to "./" (current directory "OpenSSL_certificates"). For convenience, the countryName_default was set to "US", stateOrProvinceName_default to "California", 0.organizationName_default to "comForte Inc", and organizationalUnitName_default to "Development" in section [req_distinguished_name].

To Generate the CA Certificate

(Skip this if an external CA is used)

1. Generate the private key for a root CA. Give a pass phrase when prompted. The key is used later during the signing process.

   SSL encryption is based on public key cryptography and always uses a pair of keys: the private key (generated in this step) and the public key encapsulated in the certificate (see next step). The following command will generate a 4096 bit RSA key which will be encrypted with the AES256 algorithm. The file generated is "cakey.pem" where the extension "pem" indicates PEM formatting.

   ```
   openssl genrsa -aes256 -out ca\cakey.pem 4096
   ```

2. Create the internal root CA. Enter the pass phrase from (1) and supply Country Name, State or Province Name, Locality Name, Organization Name, Organizational Unit Name, Common Name, and Email Address. Defaults can be specified in the config file (as in the example). The following command generates the root certicate "cacert.pem" in PEM format, and will contain the public key. The certificate is valid for 365 days.

   ```
   openssl req -out ca\cacert.pem -new -key ca\cakey.pem -x509 -days 365
   ```

3. Convert the PEM format certificate to DER format. The file "cacert.der" will contain the CA certificate in DER format.

   ```
   openssl x509 -inform PEM -outform DER -in ca\cacert.pem -out ca\cacert.der
   ```

   The "ca" directory now should contain three files: cacert.pem, cakey.pem, and the root certificate cacert.der.

To Generate the Server Certificate

1. First, generate the private key for the server certificate and assign a pass phrase to be used later as value of the SERVKEYPASS parameter. SSL encryption is based on public key cryptography and always uses a pair of keys: the private key (generated in this step) and the public key encapsulated in the certificate. The following command will generate a 4096 bit RSA private key which will be encrypted with the AES256 algorithm. The file generated is "servkey.pem" where the extension "pem" indicates PEM formatting.

   ```
   openssl genrsa -aes256 -out server\servkey.pem 4096
   ```

2. Create the certificate signing request (CSR). Enter the pass phrase from (1) and supply Country Name, State or Province Name, Locality Name, Organization Name, Organizational Unit Name, Common Name, and Email Address. Defaults can be specified in the config file (as in the example). Be sure to list the registered fully qualified domain name of your server (or your IP address if you don't have one) in the Common Name (CN) field. This field must be different from the CN field in the CA certificate or else a naming collision will occur and you'll get errors later on. You can skip the Challenge Password and Optional Company Name prompts.

   ```
   openssl req -out server\csr.pem -new -key server\servkey.pem
   ```

3. Convert the key to PKCS#8 DER format. Specify the pass phrase given in (1) and enter an Encryption Password. You can make the encryption password the same as the pass phrase. The following command will produce the server private key file "servkey.der" in DER format.

   ```
   openssl pkcs8 -topk8 -outform DER -in server\servkey.pem -out server\servkey.der
   ```

   If the signing request will be submitted to a CA authority, skip to step (6) after having received the signed server certificate from the CA and placed it in directory "server" as "servcert.pem" (assuming it was returned in PEM format). The CA root certificate received should be placed in directory "ca" as cacert.pem, and then converted to DER format using step (3) from "To Generate the CA Certificate".

4. In the base directory "OpenSSL_certificates", create an empty text file with the name given in the config file directive "database". Also create a file with the name given in the directive "serial", and add a single line containing a 2-digit hexadecimal serial number (i.e. "01").

   The serial number is used in the signed server certificate. Each time you generate a new certificate, especially before a previously-signed certificate expires, you willl need to increase the serial number by one. Note that it must be a hexadecimal number.

5. Sign the new CSR with the CA key. Specify the pass phrase given for the root certificate. A signed server certificate "servcert.pem" with a 365 day validity will be generated.

   ```
   openssl ca -days 365 -policy policy_anything -keyfile ca\cakey.pem -cert ca\cacert.pem
   -in server\csr.pem -out server\servcert.pem
   ```

6. Convert the signed certificate to DER format. A file named "servcert.der" will be generated by the following command.

   ```
   openssl x509 -inform PEM -outform DER -in server\servcert.pem -out server\servcert.der
   ```

   The directory "server" should now contain files csr.pem, servkey.pem, servcert.pem, servcert.der, and servkey.der. There will be a file 01.pem in the "newcerts" directory if "new_certs_dir = newcerts" was given in the config file. It is identical to servcert.pem in the "server" directory.

7. Transfer the files CACERT, SERVCERT and SERVKEY to the NonStop server. Be sure to select binary transfer mode. Note that the private key file SERVKEY must NOT be transferred over a plain connection such as FTP. If NonStop SSL is not already installed, it is recommended to use SFTP for certificate/key transfers.

NonStop SSH is installed by default on NonStop servers for maintenance LANs, and can be accessed from the system console via an SFTP client such as available in Win6530 or OpenSSH in a DOS window.

```
sftp ..
cd $SYSTEM.SSLCERTS
put server\servcert.der SERVCERT
put server\servkey.der SERVKEY
put ca\cacert.der CACERT
```

Note: the NonStop SSL installation subvolume $system.znsssl contains a set of test certificates that should not be used in production systems. Unless the configuration file contains parameter settings pointing to a different set of certificates, the default certificates in znsssl will be used.

8. Add these parameter settings to the configuration of your HP NonStop SSL server process:

```
SERVCERT $SYSTEM.SSLCERTS.SERVCERT
CACERTS $SYSTEM.SSLCERTS.CACERT
SERVKEY $SYSTEM.SSLCERTS.SERVKEY
SERVKEYPASS <pass phrase>
```

Captured output from certificate generation

```
C:\Comforte\OpenSSL_certificates>openssl genrsa -aes256 -out ca\cakey.pem 4096

Loading 'screen' into random state - done
Generating RSA private key, 4096 bit long modulus

Enter pass phrase for ca\cakey.pem:
Verifying - Enter pass phrase for ca\cakey.pem:

C:\Comforte\OpenSSL_certificates>openssl req -out ca\cacert.pem -new -key ca\cakey.pem
-x509 -days 365
Enter pass phrase for ca\cakey.pem:
Loading 'screen' into random state - done
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:
State or Province Name (full name) [California]:
Locality Name (eg, city) []:
Organization Name (eg, company) [comForte Inc]:
Organizational Unit Name (eg, section) [Development]:
Common Name (eg, YOUR name) []:CRS
Email Address []:

C:\Comforte\OpenSSL_certificates>openssl x509 -inform PEM -outform DER -in
ca\cacert.pem -out ca\cacert.der

C:\Comforte\OpenSSL_certificates>openssl genrsa -aes256 -out server\servkey.pem 4096

Loading 'screen' into random state - done
Generating RSA private key, 4096 bit long modulus

Enter pass phrase for server\servkey.pem:
Verifying - Enter pass phrase for server\servkey.pem:

C:\Comforte\OpenSSL_certificates>openssl req -out server\csr.pem -new -key
server\servkey.pem
Enter pass phrase for server\servkey.pem:

Loading 'screen' into random state - done
You are about to be asked to enter information that will be incorporated
```

```
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:
State or Province Name (full name) [California]:
Locality Name (eg, city) []:
Organization Name (eg, company) [comForte Inc]:
Organizational Unit Name (eg, section) [Development]:
Common Name (eg, YOUR name) []:www.crs.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

C:\Comforte\OpenSSL_certificates>openssl pkcs8 -topk8 -outform DER -in
server\servkey.pem -out server\servkey.der
Enter pass phrase for server\servkey.pem:
Enter Encryption Password:
Verifying - Enter Encryption Password:
Loading 'screen' into random state - done

C:\Comforte\OpenSSL_certificates>openssl ca -days 365 -policy policy_anything -keyfile
ca\cakey.pem
-cert ca\cacert.pem -in server\csr.pem -out server\servcert.pem

Using configuration from C:\OpenSSL-Win32\bin\openssl.cfg
Loading 'screen' into random state - done
Enter pass phrase for ca\cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Oct 13 20:03:52 2011 GMT
            Not After : Oct 12 20:03:52 2012 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = California
            organizationName          = comForte Inc
            organizationalUnitName    = Development
            commonName                = www.crs.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                F3:4A:8C:9A:B6:DA:D7:D6:03:3B:4A:24:D8:25:C9:5C:B8:7A:85:77
            X509v3 Authority Key Identifier:
                keyid:CE:DA:69:91:47:58:0E:FC:6B:57:A2:56:37:36:42:F5:DD:63:4C:8E

Certificate is to be certified until Oct 12 20:03:52 2012 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

C:\Comforte\OpenSSL_certificates>openssl x509 -inform PEM -outform DER -in
server\servcert.pem -out server\servcert.der

C:\Comforte\OpenSSL_certificates>
```

Requesting the SSL Client to Present a Client Certificate

HP NonStop SSL supports client authentication when running in SSL server mode (PROXYS, FTPS, TELNETS, ODBCMXS, EXPANDS). The behavior is controlled by the TRUST parameter (please note: the parameter has different meanings for HP NonStop SSL running in server or client mode).

TRUST set to "*" (default) will disable the checking, thus no client cert will be required.

When TRUST contains a certificate filename this certificate will be sent to the client. The client will send back a certificate signed by the one sent to it. If the client sends no certificate or an invalid one, the connection will be rejected.

Configuring SSL for Production as SSL Client

In run modes PROXYC and FTPC, HP NonStop SSL will be an SSL client. This section only is relevant for those run modes.

The default installation of HP NonStop SSL is streamlined to enable an easy setup and immediate testing and will not verify the remote certificate for authenticity, nor will it send an SSL client certificate to the server for client authentication.

For a secure production installation, it is recommended to configure HP NonStop SSL to verify the remote certificates using the TRUST parameter. Not doing so may compromise the security of the system.

To Configure HP NonStop SSL to verify the remote certificate

1. Obtain the root CA certificate that signed the server certificate of the target SSL server.
2. If required, convert the root CA certificate into the DER format (e.g. with OpenSSL)
3. Upload the DER-encoded root CA certificate file to your NonStop server in binary mode.
4. Configure the TRUST parameter to point to DER-encoded root CA certificate file, e.g.

   ```
   TRUST ROOTCA
   ```
5. If desired, configure the PEERCERTCOMMONNAME parameter to validate the common name of the server certificate.
6. Restart your HP NonStop SSL client proxy process and check these initialization log messages for any errors.

Note: If you have multiple target systems with different root CA certificates, you can enter a list of comma-separated file names for the TRUST parameter.

Presenting a Client Certificate to the SSL server

If Client Authentication is required, HP NonStop SSL can send a client certificate or a client certificate chain to the server.

HP NonStop SSL can be configured to cover 3 scenarios:

1. If CACERTS and CLIENTCERT are set to ‘*’, HP NonStop SSL will sent NO certificate to the server (this is the default setting).

   ```
   CACERTS *
   CLIENTCERT *
   ```
2. To send a self-signed certificate to the server, CACERTS must be set to ‘*’ and CLIENTCERT/CLIENTKEY/CLIENTKEYPASS must point to a valid self-signed certificate.

   ```
   CACERTS *
   CLIENTKEY $SYSTEM.MYCERT.CLNTKEY
   CLIENTKEYPASS mysecret
   CLIENTCERT $SYSTEM.MYCERT.CLNTCERT
   ```

3. If CACERTS contain the signing certificate(s), HP NonStop SSL will sent the whole certificate chain to the server.

```
CACERTS $SYSTEM.MYCERT.CACERT
CLIENTKEY $SYSTEM.MYCERT.CLNTKEY
CLIENTKEYPASS mysecret
CLIENTCERT $SYSTEM.MYCERT.CLNTCERT
```

Security Considerations

While SSL is a very powerful and flexible protocol to encrypt TCP/IP traffic, it has to be used properly to be protected against some common attacks. The two most important factors in making an SSL installation fully secure are:

- protecting against the man-in-the middle attack through proper usage of certificates
- protecting the private key file

Note: Ignoring those two factors will result in a system open to well-known attacks. Please read this section and follow the recommendations to make sure you are deploying SSL properly.

Protecting Against the Man-in-the-Middle Attack

The man-in-the-middle attack is based on a weakness of the TCP/IP protocol which allows adding an "intermediary" between two systems communicating via TCP/IP.

To protect against that kind of attack, SSL uses certificates. See the following sections of the chapter "SSL Reference", for more information:

- "X.509 Certificates".
- "Configuring SSL for Production as SSL Server".
- "Configuring SSL for Production as SSL Client".

Make sure to generate your own certificates for production and to configure all your SSL clients to verify the certificates used by the SSL server.

Protecting the Private Key File

If an attacker gets access to the private key file, he can attack the SSL protocol in various ways. Therefore it is important that you protect the private key file residing on your NonStop system.

The private key file is created during the generation of your certificates and is a file in your Guardian file system. The location of the file is configured using the parameter SERVKEY. Standard procedures (such as SAFEGUARD ACL's) should be employed so that only the HP NonStop SSL process can open that file.

The private key file is encrypted using a so-called pass phrase. An attacker needs both the private key file and the pass phrase for a successful attack. The pass phrase is configured through the SERVKEYPASS parameter, that parameter is probably present in some startup file or macro. This startup file needs again to be protected properly.

Note: Never send the private key file and/or the pass phrase to anybody via e-mail. Make sure the file resides only on your NonStop system and is properly protected via SAFEGUARD.

If the Private Key is Compromised

If you have reason to believe that your server private key file has been compromised, you should immediately install a new server certificate along with a private key file encrypted with a different pass phrase.

Note: If you authenticate the HP NonStop SSL server in your clients, you should consider basing trust on the Root CA certificate (e.g. check the Root CA fingerprint). In case the server certificate is compromised you can simply replace it without having to update your client configuration.

TLS Alerts

If a TLS Alert happens on an SSL-encrypted session, the TLS alert number will be logged. The following message is an example for a log message of this type: a plain Telnet client tried to connect on the encrypted socket, resulting in a TLS alert "50" (DecodeError).

```
13:37:18.53|30|TLS Alert: 50
```

The following table contains the TLS alert numbers for TLS 1.0. For more information about the individual alerts, please refer to the TLS specification RFC 2246 (available under http://www.ietf.org).

TLS Alert Number	TLS Alert name
0	close_notify
10	unexpected_message
20	bad_record_mac
21	decryption_failed
22	record_overflow
30	decompression_failure
40	handshake_failure
42	bad_certificate
43	unsupported_certificate
44	certificate_revoked
45	certificate_expired
46	certificate_unknown
47	illegal_parameter
48	unknown_ca
49	access_denied
50	decode_error
51	decrypt_error,
60	export_restriction
70	protocol_version
71	insufficient_security
80	internal_error
90	user_canceled
100	no_renegotiation

Remote SSL Proxy

The RemoteProxy Component

The RemoteProxy component included with HP NonStop SSL is used to enable SSL encryption for HP client components running on Microsoft Windows systems. Usage of the RemoteProxy component is supported for selected HP NonStop products only, including HP NonStop Remote Server Call (RSC/MP) and HP NonStop ODBC/MX. Additionally, the RemoteProxy can act as an SSL enabling LPD server proxy in order to secure LPD printing off the HP NonStop platform. Usage of the LPDS server mode is supported in combination with the Microsoft Windows platform only. Further note that the HP NonStop SSL RemoteProxy does not support being installed as a Windows service.

RemoteProxy Installation

HP NonStop SSL is shipped with a RemoteProxy InstallShield installation package (PROXYEXE) that can be downloaded and executed on the target Windows workstation.

The HP NonStop SSL RemoteProxy setup program does the following:

- It moves the files to the target PC.
- It installs shortcuts in the Windows Start menu.
- It adds RemoteProxy to the Autostart folder.

To install RemoteProxy on a Client Workstation

1. Download $SYSTEM.ZNSSSL.PROXYEXE in binary format to your Windows workstation, renaming it to PROXY.EXE.
2. On the workstation, run PROXY.EXE to start the RemoteProxy installation program and follow the installation instructions.

   After completing the installation, you will see the HP NonStop SSL RemoteProxy icon in your system tray.

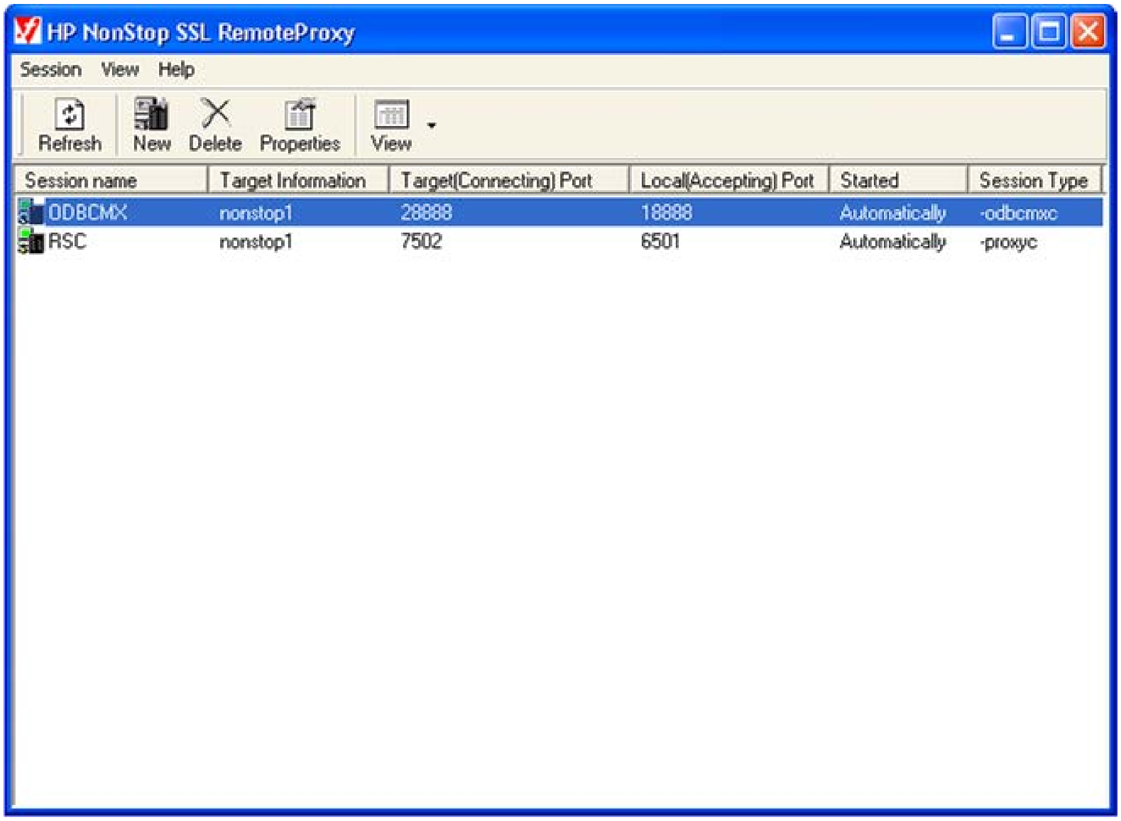
RemoteProxy Configuration

General Configuration Considerations

You will need at least one session entry for one of the available protocols (Generic TCP/IP which allows TELNET, LPD Server and ODBC/MX Client) as shown in the illustrations below. You can name the sessions as desired.

The Main Configuration Screen

After you have installed RemoteProxy, you will have the icon of the Proxy manager in your system tray. Double-clicking the icon will bring up the main configuration screen:



Session name	Target Information	Target(Connecting) Port	Local(Accepting) Port	Started	Session Type
ODBCMX	nonstop1	28888	18888	Automatically	-odbcmxc
RSC	nonstop1	7502	6501	Automatically	-proxyc

The "HP NonStop SSL RemoteProxy" window will list all configured "proxy" sessions. After the installation the list will be empty. The example pictured above shows 2 configured "proxy" sessions (for ODBC/MX and RSC), with the following information:

- "Target Information" shows either the host name or the IP address of the host to which the RemoteProxy connects for the depicted session. This could be a remote host (see last session) or the local host (see first session).
- "Target (Connecting) Port" refers to the port number under which the target host can be reached for the referenced session.
- "Local (Accepting) Port" denotes the port number under which the RemoteProxy listens for incoming connection requests to be forwarded to the target host/target port of the referenced session.

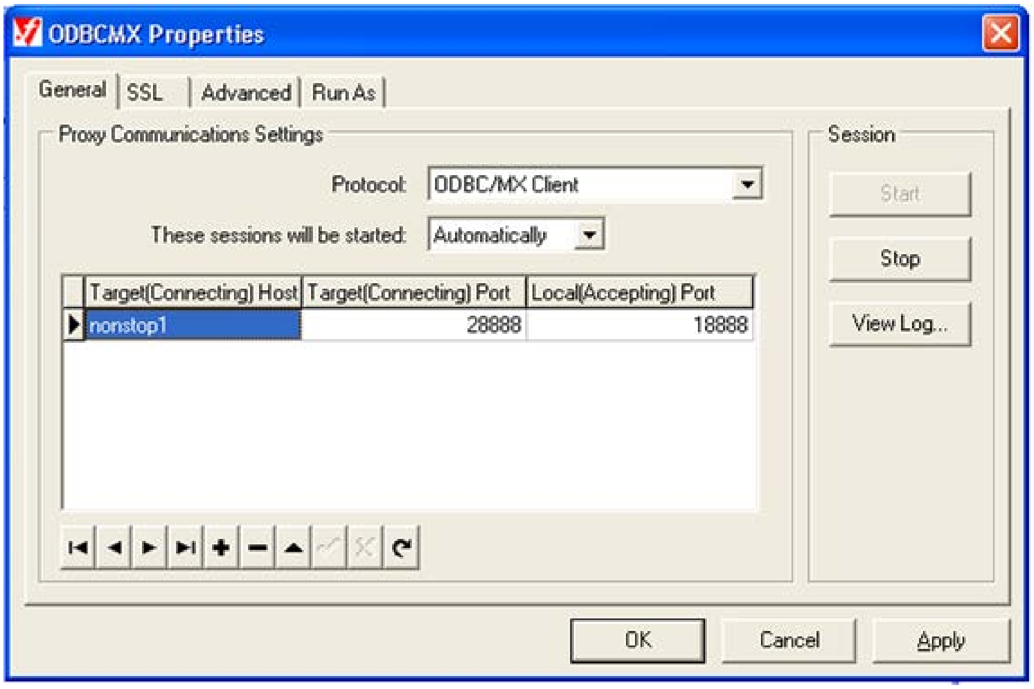
The "HP NonStop SSL RemoteProxy" window also allows you to click on

- "New" to add a sessions
- "Delete" to delete a session
- "Properties" to configure a session

Right-clicking on a session will bring up a context menu, which also allows you to start or stop a session.

The Session Properties Window

Using the "Properties" button or double-clicking on a session entry in will bring up the session properties dialog for the selected session:



ODBCMX Properties

Proxy Communications Settings

Protocol: ODBC/MX Client

These sessions will be started: Automatically

Target(Connecting) Host	Target(Connecting) Port	Local(Accepting) Port
nonstop1	28888	18888

The table in the middle of the "General" tab of the session properties configuration allows configuring multiple proxy instances of the same protocol with the same or with different Target Hosts. Please note, that only the first session is being displayed in the main configuration screen (see previous chapter). Editing, navigating, adding or removing sessions is done using the buttons below the table.

Please note, that the instances configured in the table above cannot be started and stopped separately. If this is required, they need to be configured as separate sessions (i.e. using the NEW button in the configuration main window for each separate session).

Note: RemoteProxy uses a Java Virtual Machine (JVM) to execute proxy sessions. Each proxy session is executed in a separate JVM instance whereas multiple proxy session instances configured for a single proxy session are all executed within the same JVM. Configuring multiple proxy instances rather than separate proxy sessions may lead to a performance advantage.

The "SSL" tab allows you to configure SSL-related parameters. If the RemoteProxy acts as a client for the configured session, then in this tab the checking of the Root Certificate Authority can be enabled. If the "Enable Root Certificate Authority Checking" is enabled, the expected fingerprint of the Root Certificate (i.e. the certificate received from the server to which the RemoteProxy connects for this session) must be entered into the field "Certificate Fingerprint". The fingerprint must be derived from the Root Certificate separately and calculated using the MD5 algorithm.

Authentication on SSL protocol level is performed with the help of SSL certificates. When configuring a RemoteProxy session for LPDS Server mode, configuration of corresponding server certificates per session is always required. In case of client run modes, certificates only have to be configured when client authentication is to be performed.

According to this, the pass phrase of the private key file (as opposed to its public RSA key) must be configured in the "Pass-Phrase" field in the following cases:

- Session is configured as LPDS server (SSL server proxy for LPD)
- Session is configured for running as a client and SSL Client Authentication is to be used.

Note: We strongly recommend not to rely on the default test certificates which are delivered with the HP NonStop SSL for testing purposes only!

The "Advanced Tab" allows you to configure some advanced options. Please refer to the table in the "Session Parameter List" section for details.

The "Run As" Tab allows the configuration of a different user under which the processes spawned by RemoteProxy will run. This will be necessary under rare circumstances only.

Session Parameter List

The following table lists all session parameters, their meanings, and default values:

Field	**Tab**	**Meaning**	**Default Value**
Protocol	General	The type of protocol that is used for the communication. Usually you check "Generic TCP/IP", as with most protocols all data is transferred via a single TCP/IP session. Some protocols, such as ODBC/MX, use multiple sessions and negotiate ports to be used.	Generic TCP/IP
Target Host	General	The address of the target computer to which RemoteProxy should connect for the session referenced by the current table entry.	None
Target Port	General	The port under which the target application is listening on the target host.	none
Local Port	General	The local port on the PC your terminal emulator will connect to. Please note that this port number MUST NOT be used as a Local Port for any other session configured with the RemoteProxy.	none
This session will be started	General	If you choose "automatically" the session will be started on startup of the proxy. If you choose "manually", the session needs to be started manually after startup of the proxy.	Automatically
Enable Root Certificate Authority checking	SSL	If you are using a server certificate to authenticate your NonStop system to the PC, you should activate this check box.	not checked
Fingerprint	SSL	If you are using a server certificate to authenticate your NonStop system to the PC, you need to fill in the MD5 fingerprint of the server certificate here. If "Root Certificate Authority checking" is not checked, you cannot enter a value here.	empty and inactivated
Private Key Pass Phrase	SSL	Authentication on SSL protocol level is performed with the help of SSL certificates. When configuring a RemoteProxy session for LPDS Server mode, configuration of corresponding server certificates per session is always required. In case of client run modes,	empty

Field	Tab	Meaning	Default Value
		certificates only have to be configured when client authentication is to be performed. According to this the pass phrase of the private key file (as opposed to its public RSA key) must be configured in the "Pass-Phrase" field in the following cases: a) Session is configured as LPDS server (SSL server proxy for LPD) b) Session is configured for running as a client and SSL Client Authentication is to be used.	
Turn On Debugging Message Output	Advanced	Should only be checked when you are tracking a problem after contacting HP or comForte Support.	not checked
Allow connection from localhost only	Advanced	If you check this option, the SSL proxy will only accept connections from your local computer. Checking this option if the local computer is the client will prevent misuse of the proxy by external attackers. If the local computer is the server, then you need to disable this option to allow connections from your NonStop server acting as a client.	checked
Turn off SSL session resumption	Advanced	SSL session resumption will speed up the setup of multiple sessions to the same server by reusing session secrets from previous sessions. Turn SSL session resumption off only for testing purposes.	not checked
Initialize secure random seeding on startup	Advanced	If you check this option, the SSL proxy will generate random data at session startup time. Otherwise, it will generate the random data when required during the first connection request. The random data generation process will take a few seconds and will heavily utilize the CPU. This option allows to control when this process is performed.	not checked
Use Microsoft Certificate Store	Advanced	For SSL client authentication the SSL proxy can optionally access the Microsoft Certificate Store to read the certificates and private keys.	not checked
Additional Run Options	Advanced	This field allows you to specify additional startup parameters for the proxy. You should only enter values when advised so by HP or comForte Support.	none

Copying a Configuration to Other Workstations

The RemoteProxy configuration is stored in the registry. Identical configuration of multiple targets PC's can be achieved as follows:

1. Manually configure a single target PC.
2. Export the registry part which contains that configuration (HKEY_CURRENT_USER\Software\Hewlett-Packard\HP NonStop SSL RemoteProxy)
3. Copy the resulting ".reg" file to the target workstation.
4. On the new target PC, import the ".reg" file into the registry by double clicking it.

Appendix

Log Messages and Warnings

This section lists log and warning messages issued by HP NonStop SSL.

Startup messages

This section contains messages which are displayed during startup and which are of an informational nature only.

HP NonStop SSL version <HP NonStop SSL version number> version_info

Appears right after startup and notifies about the version umber of the HP NonStop SSL

using openssl version 0.9.7 - see http://www.openssl.org);

Notifies about the OpenSSL version bound to the HP NonStop SSL

config file: '<filename>

Displays the name of the configuration file the HP NonStop SSL has been started with

runtime args: '<list of runtime args>

If HP NonStop SSL has been started with runtime arguments instead or in addition to a configuration file or TACL PARAMs, those arguments are being displayed.

log level is <log level>

Informs about the log level the HP NonStop SSL has been started with

your system number is <system number>

The system number of the NonStop system on which the HP NonStop SSL has been started. This number must be the same as the system number given in the license file.

starting collecting of random data

Notifies about the process of collecting random data.

collection of <number> bytes random data finished

Informs about completion of collecting random data. <number> is the number of collected bytes.

DEFINE =TCPIP^PROCESS^NAME has value '%s'"

Notification about which TCP/IP process name is being used

parameter SUBNET will be ignored

Notification that the value of the SUBNET parameter will be overridden by the DEFINE =TCPIP^PROCESS^^NAME

TCP/IP process is <process name>

Notification about the TCP/IP process used for the communication in the current context of this message.

parameter SUBNET was evaluated

Notification that a SUBNET param has been found and will be used to determine the TCP/IP process

secure-to-plain proxy started on target host <hostname or ip address>, target port <port number>, source port <port number>

Notification that the HP NonStop SSL has been started in a mode accepting connections for secure data and connecting to the target host on target port for plain data.

plain-to-secure proxy started on target host <hostname or ip address>, target port <port number>, source port <port number>

Notification that the HP NonStop SSL has been started in a mode accepting connections for plain data and connecting to the target host on target port for secure data.

FTP server proxy started on target host <hostname or ip address>, target port <port number>, source port <port number>

Notification about the HP NonStop SSL being started in FTPS mode and connecting to target host, on target port while accepting connections on source port.

FTP client proxy started on source port <port number>

Notification about the HP NonStop SSL being started in FTPC mode and accepting connections on source port.

dumping configuration: <config setting>

Displays the settings of the configuration params

loading Server Certificate from file <filename>

Notification that the server certificate is being loaded from the file given by filename

loading next Certificate Chain file from file <filename>

Notification that the next of a sequence (chain) of signing certificates are being loaded by the HP NonStop SSL.

Fingerprint of Root CA is <MD5 fingerprint>

Notification about the MD5 calculated fingerprint of the root certificate. If fingerprint checking is activated on the client side, this fingerprint must be preconfigured there.

loading private key from file <filename>

Notification about the server private key (see param SERVKEY) being loaded from the file identified by filename.

adding CA Certificate Chain Level <curr number>/<max number> <filename>

Notification about how many certificates the CA certificate chain contains, which of these certificates are currently processed and what the filename of this is.

Connection closed by remote client

This log message will be issued any time a remote client disconnects unexpectedly. In most cases (especially when running in TELNETS mode), this log message can be safely ignored.

ProcessInfo follows:
<process_info>
ProcessInfo --END-

This message will be issued at startup and is an informational message informing about the current process state, including current stack and heap usage.

Warning messages

The following messages are displayed under conditions where HP NonStop SSL can recover from an error and will continue to run.

Firewall: connection rejected from: <ip address>

Warning about a connection from a remote host identified by ip address being rejected. This message may appear in conjunction with the ip filtering function of the HP NonStop SSL, see parameters ALLOWIP and DENYIP for details

certificate not yet valid

Warning that the certificate being currently processed by the HP NonStop SSL is not yet valid, i.e. has a "from date" starting in the future.

certificate expired

Warning that the certificate being currently processed by the HP NonStop SSL has expired i.e. is not valid any longer.

F|#<session>-<ip address> login without SSL rejected

[FTPS mode only] The remote client identified by ip address attempted an unsecured login on a secured port. The login was rejected.

TLS Alert: <TLS alert number>

Warning about a TLS alert received within current session

TLS Exception

Warning about a TLS exception received in current TLS session. Watch for message which come along with this and which give additional information.

remote fingerprint <fingerprint> rejected

Warning that a certificate received from the remote SSL server was rejected because the MD5 fingerprint did not match the one configured with HP NonStop SSL TRUST parameter. The calculated fingerprint is displayed.

OnAccept1Complete: error <error number>

Internal error occurred during acceptance of a TCP/IP connection. Watch for other message which come along with this one in order to decide whether and what action has to be taken.

OnAccept2Complete: error <error number>

Internal error occurred during acceptance of a TCP/IP connection. Watch for other message which come along with this one in order to decide whether and what action has to be taken.

F<session>|<-- unexpected reply to PASV command from FTP server: <reply>

Warning about receipt of an unexpected reply from the remote FTP server upon requesting Passive Mode FTP. Check whether the remote FTP server supports passive mode FTP.

F<session>|<-- reply to STOR/RETR/LIST command from FTP server has error: <reply>

Warning that the reply from the remote FTP server upon one of the mentioned requests is erroneous. If this happens frequently contact your support representative.

F<session>|<-- reply to PORT command from FTP server has error: <reply>

Warning about the receipt of an erroneous reply from a FTP server upon requesting active mode FTP. Check whether the remote FTP server supports active mode FTP.

CSocket::Create: could not resolve address <hostaddress> with given IPMode=<ipmode> (<detailed reason>)

This error occurs when a hostaddress was passed which cannot be resolved, at least not in the current IPMODE. Check the DNS entry for the given host address to verify that the <hostaddress> has an IP address assigned to it. Alternatively pass the IP address directly.

Received command "<ipv4_only_command>" although using IPv6 connection, sending error 503 to client.

This command indicates that the FTP client used does not act according to the FTP IPv6 specification. Please contact HP support.

Received PORT command but outgoing connection is IPv6 connection, transition not supported, sending error 500 to client.

This message occurs when an FTP client connected IPv4 and HP NonStop SSL was configured in FTPS mode with IPMODE DUAL and an IPv6 only host as TARGETHOST. In this case the data connection cannot be established since FTP data command protocol is not supported. To resolve the error condition either specify a mapped IPv4 address in the TARGETHOST (e.g. ffff::127.0.0.1) or set up one process for each IPMODE IPv4 and IPMODE IPv6.

AUTH command "<AUTH_command>" from client not understood

This message indicates that a client sent an unsupported authentication command. Please contact HP support.

Protection level indicated by command "<PROT_command>" not supported.

This message indicates that a request for a certain protection level of the FTP connection was requested which is not supported. This might indicates a bug in either the client or the server side. Please contact HP support.

Received command "<command>" although client issued EPSV ALL before, sending error 503 to client.

When FTP data channels are opened up according to RFC 2428 with EPSV/EPRT, the client can indicate (by sending "EPSV ALL") that it will subsequently always open data connections by using passive FTP with the EPSV command. In this way potential NAT gateways analyzing the FTP control connection can be informed to put the connection to (faster) pass-through mode, since no IP address replacing has to be done anymore. If, however, the client sends the "EPSV ALL" command and later tries to open up an FTP data connection with a command other than EPSV, this error message will be generated. If you see this error, please contact HP support of the FTP client vendor.

Warning, could not normalize at least one of the IPAddresses "<ip1>" and "<ip2>" comparing them as they are

This warning might indicate a bug, please contact HP support.

parameter "PREVENT_TLS_1_0_CBC_VULNERABILITY" was explicitly set to false (NOT recommended, default=true), will *NOT* clear SSL_OP_DONT_INSERT_EMPTY_FRAGMENTS option. You will be vulnerable to attacks against CBC algorithms !!

This warning indicates that the parameter PREVENT_TLS_1_0_CBC_VULNERABILITY was set to FALSE by the user, which results in not preventing a certain vulnerability on CBC algorithms, known as the BEAST attack. Changing the default value TRUE of the parameter should only be done if advised to by support.

Invalid value ("<value>") given for parameter TARGETHOST. Override will not be enabled (<detailed reason>)

This warning indicates that an invalid value for TARGETHOST was specified in FTPC mode. Accordingly the targethost override (in case no targethost was given) in the FTPC user command will not work. Please see <detailed reason> for error details and parameter description of TARGETHOST and TARGETHOSTFORCE.

TARGETPORT out of valid range (1-65536). Override will not be enabled

This message indicates that a TARGETPORT was specified in FTPC mode that is out of the valid port range of 1-65536. Please see parameter description of TARGETPORT and TARGETPORTFORCE.

could not connect to FTP server data port

This message indicates that the connection to the data port which the FTP server opened up could not be established. Watch for other message which come along with this one in order to determine the reason for the connect failure. If you see this warning, please contact HP support.

Could not build proxy for passive data connection from client

This warning indicates than the request of an FTP client for a passive data connection could not be successfully fulfilled due to problems setting up the proxy for the passive data connection. Watch for other message which come along with this one in order to determine the reason of the error condition.

unexpected AUTH SSL reply: '<reply>'

This warning indicates that the FTP server sent an unexpected reply to the AUTH SSL command. Please contact HP support.

reply to EPRT command from FTP server has error: '<detailed reason>'

This warning indicates that the FTP server sent an unexpected reply to the ERPT command. Please contact HP support.

Failed to change IP address according to given value <hostaddress> of parameter FTPNATADDRESS: <detailed error reason>

This warning indicates that an invalid value for the parameter FTPNATADDRESS was specified. See <detailed error reason> for why the error occurred and to resolve the error condition.

could not listen on socket on IP <ip>, port <port>, error: '<errorstring>'(errno=<error_number>)

This warning is issued when the attempt to set up a listen on IP <ip> and port <port> failed. Please see the error number and the corresponding error description indicating the reason for the error condition.

could not create listening socket on port <port>, error: '<errorstring>'(errno=<error_number>

This warning is issued when the attempt to create a socket listening on port <port> failed. Please see the error number and the corresponding error description indicating the reason for the error condition.

illegal USER command sent from remote

This warning is issued when an incorrect USER command was received in FTPS mode. Please contact HP support.

Incorrect user command while trying to change to FTPC format: <detailed error>

This warning is issued when PASSREMOTEIP is set to TRUE and an attempt to change the USER message to FTPC format failed. Please contact HP support.

receiving message on $RECEIVE without prior open message - ignoring and replying with length 0

This warning indicates an incorrect sequence in the communication with the HP NonStop SSL process. It e.g happens when SSLCOM is run with a certain HP NonStop SSL process and the HP NonStop SSL process is then restarted with the same process name without restarting SSLCOM. In this case, any command in SSLCOM will result in an empty response. To resolve the error condition, restart SSLCOM.

AWAITIOX file <filename> (filenum <file_descriptor>) completed with error <error number> (<errorstring>)

This message indicates an error on the file <filename>. The <error number> and <errorstring> will give the error reason. Watch for the messages that come along with this one for the context of the error.

CEventDispatcher::Dispatch - unexpected event

This message probably indicates a programming error. Please contact HP support.

received invalid expand tunnel message header

This message indicates a problem on Expand level. Please contact HP support if you see this message.

could not send complete Expand UDP packet

This warning occurs when the HP NonStop SSL process failed to send out an Expand UDP packet. Please watch for messages that come along with this one and check your Expand line status.

ExpandProxy failed to set up SSL Tunnel listen on <srcipaddr>:<port>, errno=<error number> <errorstring>

This message indicates that an error occurred when trying to set up the SSL Tunnel for EXPANDS mode. See the <errorstring> for a detailed reason. This can include an error due to passing an invalid SUBNET or SRCIPADDR, or errors on the TCPIP stack. Please watch for messages that come along with this one to resolve the error condition. As a result of this message the HP NonStop process will sleep for the value specified in LISTENRETRYINTERVAL (default 10 sec).

accepting tunnel connection failed, reinitializing

This message indicates that the HP NonStop SSL process is currently in the handshake phase. When the Expand lines and the HP NonStop SSL processes are set up correctly, this message will occur a few times until the handshake completed successfully. Otherwise please check your Expand and HP NonStop SSL configuration.

Failed to Connect to client side of proxy: host=<host>, port=<port>

This message indicates a failure when establishing the connection to <host>:<port>. Please watch for messages that come along with this one for the context of the message.

Could not refresh IP address of host <host> (<error_reason>), will keep existing IP address <ip>

This message occurs when a DNS host entry was specified, e.g. for TARGETHOST but the DNS resolution fails. HP NonStop SSL will resolve the hostname every time it connects and - to make sure that the DNS entry is valid initially - during startup. If the DNS resolution fails after startup the last successfully resolved IP address <ip> will be used.

Could not refresh ipAddress SocketAddress::ipAddress(): <error reason>

This message might indicate a bug, please contact HP support.

random data was not seeded properly

This message might indicate a bug, please contact HP support.

Wildcard is included in trusted fingerprint strings, this is insecure since every peer is trusted! Please consider using fingerprints (<secure_fingerprint_algorithms>)!;

This message indicates that no fingerprints were set in the TRUST parameter, thus every peer is trusted. Please see parameter TRUST for details.

<hash_algorithm> fingerprint <fingerprint> trusted. <hash_algorithm> is a cryptographically broken hash algorithm! DO NOT USE <hash_algorithm> FINGERPRINTS ANYMORE! It is strongly recommended to specify fingerprints calculated with a secure hash algorithm (<secure_fingerprint_algorithms>)!

This is a warning message indicating that the <hash_algorithm> has been cryptographically broken. It is recommended to use a more secure hash algorithm instead, please see section HASHALGORITHMS for details.

ADH ciphers enabled, these include *NO AUTHENTICATION*!. MITM attack easy! Not recommended, use at your own risk !!!

This is a Warning message to indicate that one of the ADH ciphers is used. These have severe security limitations, so should be used only if you are fully aware of what you are doing.

Apparently out of memory - could not allocate new SSL_CTX for determining all supported ciphers.

The process ran out of heap space and could not allocate further memory.

Could not set cipher string \"ALL\" on dummySSLCtx in internal_set_cipher_list

This message might indicate a bug, please contact HP support.

Could not set cipherstring "HIGH:!ADH:!PSK:RC4" to main openssl_high context, using default ciphers <default_ciphers>

This message might indicate a bug, please contact HP support.

adding client CA file '<filename>' failed

This message is displayed when a client CA file configured via the CLIENTAUTH parameter could not be processed. Please view the prior log messages for detailed error information.

server fingerprint invalid, session rejected: PEERCERTFINGERPRINT='<fingerprint>', MD5='<actual_md5_fingerprint>', SHA1='<actual_sha1_fingerprint>'

This message is displayed only when the parameter PEERCERTFINGERPRINT is used and when the actual fingerprint of the remote certificate does not match the configured fingerprint. Please see parameter PEERCERTFINGERPRINT for details.

server common name invalid, session rejected: PEERCERTCOMMONNAME='<peercert_cn>', CN='<actual_peercert_cn>'

This message is displayed only when the parameter PEERCERTCOMMONNAME is used and when the actual common name of the remote certificate does not match the configured common name. Please see parameter PEERCERTCOMMONNAME for details.

Informational messages

The following messages display information about actual events. No corrective action is necessary.

issuer= <an certificate issuer>

Notification about the issuer of the currently processed certificate during SSL session establishment

F|#<session>-<ip address> close

[FTPS mode only] Notification about a session with host identified by ip address being closed.

F|#<session> - <ip address> new connection

[FTPS mode only] Notification about a new connection being established with remote client identified by ip address

Fatal Errors

The following messages are displayed in situations where a fatal error occurred. HP NonStop SSL will abend because it cannot recover from that error.

Fatal Error: could not listen on port <port number>, error <error number >

Error condition which is caused either by another application listening on same port or by configuring the HP NonStop SSL with a PORT param less than 1024 while not starting the HP NonStop SSL under the SUPER user logon. The HP NonStop SSL terminates.

Fatal Error: AWAITIOX file <filename> (filenum <filenumber>) completed with error <error number>

A nowaited operation on file identified by filename completed with a filesystem error. Watch for other message which come along with this one in order to decide whether and what action has to be taken.

Fatal Error: fatal error in proxy server, OnAccept: Accept failed

Internal error condition. Receiving a connection failed. Watch for other message which come along with this one in order to decide whether and what action has to be taken.

Fatal SSL error <error text>, exiting

A fatal error specified by error text has occurred. SSLOBJ is being terminated. Please consult your support representative for further action.

Can't open input file <filename>

The file given by filename cannot be opened. The severity of this message depends on the context in which it appears. If for example the server key file cannot be opened the HP NonStop SSL terminates. Watch for other messages surrounding this one.

Fatal SSL error <error number>, exiting

A fatal error during SSL processing has occurred which causes SSLOBJ to terminate. Watch for earlier message which give additional information.

illegal parameter <param value> for MINVERSION needs to be one of 2.0/3.0/3.1

Warning about an invalid setting of param MINVERSION. The allowed settings are being displayed with the message. This message appears during startup of the HP NonStop SSL. The HP NonStop SSL will not start.

illegal parameter '%s' for MAXVERSION needs to be one of 2.0/3.0/3.1

Error message about an invalid setting of param MAXVERSION. The allowed settings are being displayed with the message. This message appears during startup of the HP NonStop SSL. The HP NonStop SSL will not start.

MAXVERSION cannot be smaller than MINVERSION

Warning that an invalid param setting have been detected where MINVERSION is smaller than MAXVERSION. This message appears during startup of the HP NonStop SSL. The HP NonStop SSL will not start.

Failed to perform action <action>, no retry configured.

This message indicates that the <action>, which is usually a certain socket action for setting up a listen socket or connect to a peer, failed. For the main listening socket of the process, the <action> is retried after 10 seconds by default (in this case this error will not occur), however if this retry was disabled by setting the USESOCKETRETRY parameter to FALSE, the process will abend with this error. Watch for other message which come along with this one in order to determine why the <action> could not be performed successfully.

FTPClientProxy failed to listen on source port <port>, subnet "<subnet>", interface "<interface>

This error indicates a failure to set up a FTPC proxy with the shown parameter values. Check that these values are correct, and that no other process does already listen on port <port>. Watch for other message which come along with this one for additional hints on why the FTPC setup failed.

Error when processing cipher suite list, empty list. At least one cipher suite has to be given.

This error indicates that an empty value has been given for the CIPHERSUITES parameter. Make sure your command line does not (e.g. due to a wrong comma) include such an empty CIPHERSUITES entry.

Error when importing cipher suite list, given string <givenCipherList> was invalid (No existing cipher suite found for token <givenCipherListEntry>)

This error indicates that an invalid cipher entry <givenCipherListEntry> was given in the value of the CIPHERSUITES parameter. Make sure your CIPHERSUITES entries are according to the manual and are separated by a comma each.

Cannot init EXPANDSecureProxy, SRCIPADDR/DESTIPADDR in invalid format: <detailed reason>

This error indicates that the SRCIPADDR/DESTIPADDR parameters were given in an invalid format. This can e.g. happen when running in IPMODE IPv4 but specifying IPv6 an address. Please see the <detailed reason> for the detailed error reason.

ACTIVE FTP mode not supported in run mode FTPCPLAIN, set parameter PASSIVE to TRUE

This error indicates an unsupported runmode/parameter combination which occurs when an FTP proxy is started in runmode FTPCPLAIN and the parameter PASSIVE was set to FALSE. Either remove set the parameter PASSIVE to true, or in case you don't need to run plain connections use the regular FTPC run mode.

Multi-tunnel configuration not valid for EXPANDS mode

This error occurs when you try to specify a multi tunnel configuration by passing multiple entries in at least one of PORT/TARGETPORT/TARGETHOST (unused parameters for EXPANDS anyway). Remove any of PORT/TARGETPORT/TARGETHOST to resolve the error condition.

Invalid (Multi-) Proxy configuration given: <detailed reason>

This error indicates that one or more values given in parameters PORT/TARGETPORT/TARGETHOST are invalid. The detailed reason at the end of the message should give you enough information about how the error condition can be resolved.

Invalid or inconsistent IP Address given for INTERFACE or TARGETINTERFACE: <detailed reason>

This error indicates that at least one of the values for INTERFACE respectively TARGETINTERFACE was invalid. The detailed reason at the end of the message should give you enough information about how the error condition can be resolved.

IP address <IP> given for parameter <param> invalid in the corresponding IPMODE(=<ipmode>)

This error occurs when an IP address was given for param <param> that does not correspond to the IPMODE. This can occur if an IPv6 address was specified in IPMODE IPv4. To resolve the error condition check which runmode is intended and specify the IP address in parameter <param> accordingly.

Host name resolution for <hostname> "<interface param>" failed: <detailed reason>

This error occurs when a hostname was given for at least one of the values of INTERFACE respectively TARGETINTERFACE, and the hostname could not be resolved to an IP address. Please check the hostname and your DNS settings. Alternatively specify the IP address instead of the hostname.

Invalid IPMODE specified, ODBC/MX does only support IPv4

This error occurs when an IPMODE other than IPv4 was specified in run mode ODBC/MX. Although ODBC/MX supports IPv6 starting with release H06.26/J06.15, NonStop SSL currently only supports ODBCMXS mode for IPv4.

Invalid IPMODE specified, run mode EXPANDS does not support IPMODE DUAL

This error occurs when IPMODE DUAL was specified in run mode EXPANDS, however this is not supported in runmode EXPANDS by design. Please set up one EXPANDS process for both IPMODE IPv4 and IPMODE IPv6 to resolve the error condition.

Setting <interface_param> in IPMode DUAL is invalid since it defies the DUAL mode purpose of allowing both IPv4 and IPv6 connections

This error indicates that at least one of INTERFACE or TARGETINTERFACE was specified in IPMODE DUAL. To resolve the error condition, please either do not specify INTERFACE or TARGETINTERFACE or do run a proxy process with the interface parameters set for IPMODE IPv4 and IPMODE IPv6.

Failed to restrict listen socket use to IPv6 only according to given IPMode IPv6

When opening up an IPv6 socket, by default it listens for incoming connections on both IPv4 and IPv6. When running in IPMODE IPv6 a special socket option has to be set on the socket. This error message will occur if setting this restriction flag fails. If you experience this error, please contact HP support.

Invalid <ALLOWIP/DENYIP> value given: <detailed reason>

This error indicates that the ALLOWIP and/or DENYIP parameter was set incorrectly. See end of the message for the detailed reason which should give enough information to resolve the error condition. Also see parameter description for ALLOWIP/DENYIP for further information.

Troubleshooting of Typical Errors

Address already in use

If the message "Fatal error: Could not listen on socket: Address already in use" appears, please check whether the Source Port, which you assigned as PORT parameter is not in use by any other process.

Could not open xxx file

If the message "Could not open xxx file" appears, please check whether the file with the specified name (e.g. key file, certificate file, log file) is in use by any other process.

Decode Error

If a message with a "Decode Error" occurs in the HP NonStop SSL log, a client may have tried to create a non-secure connection to a secure HP NonStop SSL server (FTPS, TELNETS, etc.).

Handshake Error

If a message with a "handshake error" occurs in the HP NonStop SSL log, please check the following:

- does the client support the configured SSL protocol versions (MINVERSION, MAXVERSION)?
- does the client support the configured CIPHERSUITES?
- Is NonStop SSL run in client mode and the server side rejects the Client Hello? Please add the parameter TLSEXTENSIONSESSIONTICKET with value FALSE to your NonStop SSL startup configuration. This will disable one specific TLS Extension which is used by default in the Client Handshake. In general no TLS implementation should fail due to this extension being sent since following the respective RFC, TLS extensions shall be ignored if not understood. However there are some buggy implementations around which fail and reject the Client Hello for that reason.

Invalid address

If the message "Invalid address..." appears, please check whether PARAMS TARGETHOST and TARGETPORT describe a valid host::port address in your network.

Security violation (error 4013)

If HP NonStop SSL fails with a security violation, you may have attempted to start HP NonStop SSL to listen on a PORT smaller than 1024 without having a SUPER group user id.

Excerpt from the "Tandem TCP/IP programming manual":

> EACCES (4013)
>
> Cause. A call to bind or bind_nw specified an address or port number that cannot be assigned to a non-privileged user. Only applications whose process access ID is in the SUPER group (user ID 255,n) can bind a socket to a well-known port.
>
> Effect. The bind or bind_nw call failed.
>
> Recovery. Specify another port number or address, or rerun the application with a process access ID in the SUPER group (user ID 255,n).