DEFINE TLSD

The DEFINE TLSD command initiates an SSL/TLS Daemon to handle encryption and decryption.

Syntax:

```
DEFine TLSd ID=id ,CERTLibrary=name8 ,CERTMember=name8
            ,CERTSublibrary=name8 [,CIPher=09] [,MINVers={300|301}]
            [,PORT=443] ,[PASSport=80] [,TYPE={1|2}]
```

Arguments:

ID= - A unique name that will identify this Daemon.

CERTLibrary= - Identifies the library name that contains the private key and certificates to be used by this Daemon.

CERTMember - Identifies the member name that contains the private key and certificates to be used by this Daemon.

CERTSublibrary Identifies the sub library that contains the private key and certificates to be used by this Daemon.

CIPher= - A string of hexadecimal values that indicate the acceptable cipher suites. When a connection is being negotiated, the client will be required to select a cipher suite from this list.

- 01 - RSA_NULL_MD5
- 02 - RSA_NULL_SHA
- 08 - RSA_DES40CBC_SHA
- 09 - RSA_DESCBC_SHA
- 0A - RSA_3DESCBC_SHA
- 62 - RSA_EXPORT_DESCBC_SHA
- 2F - RSA_AES128CBC_SHA
- 35 - RSA_AES256CBC_SHA

MINVers= This value specified the minimum acceptable version of SSL/TLS.

- 0300 - This is the “old” standard. Most clients will be able to provide this level of support.
- 0301 - This value will require that clients adhere to the newer TLS standard as set forth in IETF RFC2246. This version is more secure, but not all clients will support it.

PASSport= This is the unique port number that the SSL-enabled application will connect with. If this value is identical to the PORT= value, then this indicates the application has directly implemented the SSL/TLS API.

PORT= This is the unique port number that clients will connect with. Any port number (1 through 65,535) may be specified.

TYPE= - One of two numeric values may be specified to indicate whether or not the client must provide authentication when connecting.

- 1 - No client authentication is performed. Default.
- 2 - Client authentication enforced.

Example:

```
IPN237I define tlsd,id=tls01,port=992,passport=992,cipher=08090a2f35, -
IPN237I   certlib=prodlib,certsublib=phase,certmember=sample01

IPN237I query tlsd
IPN253I << TCP/IP TLS Daemons >>
IPN617I  ID: TLS01 Cipher: 08090A2F35
IPN618I    Port: 992  Passport: 992 Type: Server
IPN619I    Driver: SSLD  Minimum version: 0300

IPN237I define telnetd,id=teln01,tcpappl=telnlu01,menu=menu01,pool=yes,port=992
TEL900I Daemon Startup Telnet Termname: TELNLU01 Port: 992

IPN237I query telnet
IPN253I << TCP/IP Telnet Daemons >>
TEL920I  ID: TELN01 (Inactive)
TEL921I    Terminal: TELNLU01   Menu: MENU01
TEL922I    Port: 992 Match IP: 0.0.0.0
```

Notes:

- SSL/TLS servers must always provide a certificate to the client during negotiation. The client then uses the certificate to authenticate the server.
- Given the library, sub library, and member name specified, three members with the extensions of ".PRVK", ".ROOT", and ".CERT" must exist and contain valid information.
- Consult the *TCP/IP Optional Features Guide* for more information on configuring *TCP/IP for VSE*'s SSL and TLS.
- Cipher suites 0A and 2F are sufficient for most applications.
- Cipher suite 35 provides the strongest encryption for more sensitive applications.
- Cipher suites 01 and 02 provide no encryption and generally should not be included in the list.
- Coding CIPHER=08090A622F35 provides the greatest flexibility for establishing an encrypted connection.
- Please note that using an encrypted connection protects the data ONLY during transmission. In most instances where data is stolen or forged, the act is performed at the endpoints, before the data is encrypted or after it is decrypted.

Related Commands:

DEFINE FTPD	-	Create a File Transfer Protocol Daemon.
DEFINE HTTPD	-	Create a Hypertext Transfer Protocol (web server) Daemon.
DEFINE TELNETD	-	Create a TN3270 or TN3270E Daemon
DELETE TLSD	-	Terminate an SSL/TLS Daemon.
QUERY TLSD	-	Displays currently-active TLS Daemons.