HOST LINKS

G&R GLINE™

Line Handlers and Configuration

http://www.gar.no/hostlinks/

G&R GALLAGHER ROBERTSON

Version 6.6

GALLAGHER & ROBERTSON AS, Kongens gate 23, N- 0153 Oslo, Norway
Tel: +47 23357800
www: `http://www.gar.no/`
e-mail: support@gar.no

Contents

Host Links Gline

Installation

The G&R emulations and gateways are independent programs, but part of the `G&R Host Links` product set available on all major UNIX/Linux platforms. Many of the products are also available for Windows servers. For details on platforms supported, software delivery and installation refer to the *Host Links Installation and Configuration* manual.

Host Links Product Overview

Terminal environment

Host links products that run on UNIX or Linux servers with a terminal driven user interface include emulators and concentrators, as well as various utilities.

- G3270 provides synchronous IBM3270 functionality. G3270 emulates IBM LU type 2, including base and extended colour together with extended highlighting.

- Qsim provides synchronous Questar terminal functionality. Qsim simulates all Questar models, including the DKU7007, DKU7107, DKU7105 and DKU7211 (Mono, four colour A/B and seven colour modes are supported). It also simulates the VIP7760 and the VIP7700.

- V78sim provides Bull VIP78xx (BDS) functionality. V78sim emulates all models of the VIP7800 family; the actual reference is the BDS7. All visual attributes including colour are supported.

- Pthru provides transparent VIP7800 visibility to Bull mainframes for users with asynchronous VIP7800 terminals or emulators. The terminals are used in text or forms mode.

Server environment

Host Links products that run on UNIX, Linux or Windows servers.

- Ggate is a transparent gateway to the Bull native DSA network. It avoids all need for Front-ends (MainWay) or other gateways. It can be used to connect G&R/Glink (for Windows or Java) emulators or any of the emulators, concentrators, network printer emulators and file transfer clients/servers in the Host Links product set. It also supports third party clients using the TNVIP, TN3270, TN3270E and standard asynchronous Telnet protocols.

- Gweb provides a web browser interface to any host application that is otherwise accessible using the *`Host Links Qsim`*, *`V78sim`*, or *`G3270`* emulations.

- Gspool is designed to run as an unattended process and accept transparent print output from any type of host application (GCOS8, GCOS7, GCOS6, IBM) that normally sends print data to network printers (ROPs), or to a remote spooling system (DPF8-DS). On the Gspool system the print may be directed to a physical printer or to the local spooling system. Gspool operates in different modes, Connect mode, Terminal Writer mode, DPF8 mode, SNM mode, IBM mode, TN3270 mode and TN3270E mode.

- GUFT is a G&R implementation of the Bull UFT file transfer protocols. It enables transfer of data files between Host Links and GCOS systems over a DSA network.

- Gproxy is a network management program used for supervision, management, load balancing and license sharing of G&R *`Host Links`* applications. `Gproxy` can be set up as a freestanding monitor program and/or report generator in a small network, or play a bigger role in a larger network.

- Gsftp is a transparent gateway between two different File Transfer protocols: FTP (RFC 959) and SFTP (the SSH File Transfer Protocol). The purpose is to present a seamless integration between the two protocols, with automatic conversion.

Scope of the product

G&R/Gline comprises a set of communications line handlers with a standard interface used by the *G&R Host Links* product set. The Gline line handlers standardize handling of any kind of communications line. This reduces application complexity, and makes applications portable across network environments. Applications are written using one of the communications libraries supporting the standard communications interface, *GlAPI* (Gline Application Programming Interface). The interface works with any Gline line handler, and can be used to program applications for communication over any network connection for which there is a Gline line handler, including:

Asynchronous	direct or modem connected
X.25	raw or PAD
TCP/IP	dial-up, ISDN, leased line, raw, Telnet, TNVIP TN5250 and TN3270
DSA/OSI	native Bull DSA and DSA/ISO Work Station over OSI-transport
DSA/RFC1006	native Bull DSA session over TCP/IP transport

GlAPI is available on all the supported platforms: UNIX/Linux and Windows servers and workstations.

The *G&R Host Links* product set uses the native mode *Gline API* for all communications.

Third party software houses and customers also use GlAPI when developing communications applications. GlAPI includes CPI-C for use in applications following the X/Open standard for programmatic access to communications. For details of the various APIs please refer to the *GlAPI Application Programming Interfaces* manual.

Overview

G&R product architecture

The G&R products are built in a modular way, and designed to take advantage of the multitasking capability of the Windows and UNIX/Linux platforms. Thus a single instance of a product will in general consist of two or three quite separate processes:

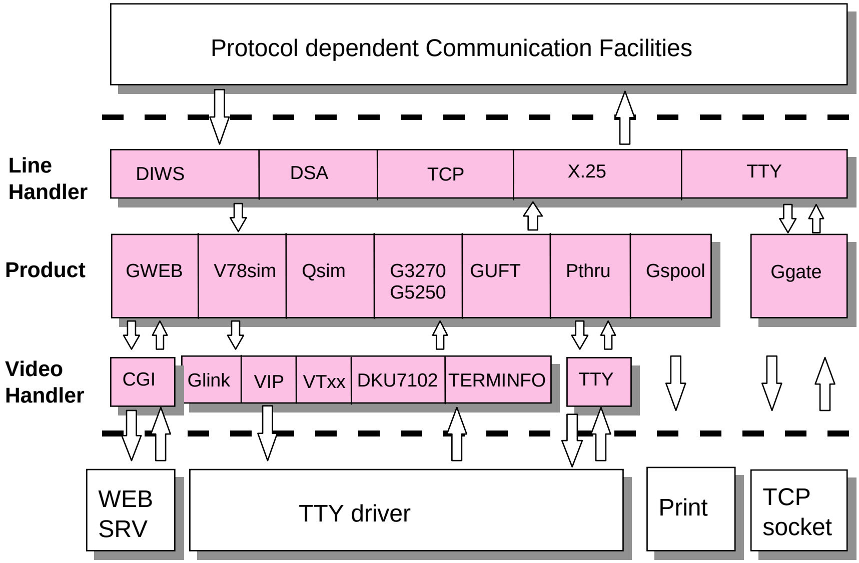
- A video handler: which maps all updates of the internal screen image into the control sequences required to update the specific screen being used. The video handlers are used by UNIX/Linux products, which must communicate with various different terminal types. The Windows products use the standard Windows graphical interface.
- A product: for example an emulator that updates the internal screen image to duplicate the terminal being emulated. The product will also in general send and receive data to the host system using the global interface, which applies to all communications protocols.
- A line handler: that maps the global communications interface into the specific line protocol being used.

This structure has proven itself extremely efficient and very robust. We are able to develop a video handler for a new screen type, and we know that once it works with one G&R product it will work with all. We are able to develop a new line handler and know that once it works with one product it will work with all. We are able to develop a new product using a given screen and communications protocol, and know that the product will work with all screen types we support, and with all the communications protocols we support.

The server programs (e.g. `Ggate, Gspool, GUFT`) differ slightly from the above model, as they don't need a video handler, only a line handler. `Pthru` doesn't need a video handler either; it needs one line handler to connect to the host and one line handler (TTY) to pass all presentation data through to the terminal.

Architecture diagram

The diagram shows clients for terminal users (*`V78sim`*, *`Qsim`*, *`G3270,`* `G5250` and *`Pthru`*) on UNIX and Linux only.



Line handlers

A *`Host Links`* product will generally start up the line handler automatically, and will in general itself accept parameters for tailoring the line handler, and pass these to the handler after contact is established.

We currently have the following line handlers: `gl_dsa`, `gl_tcp`, `gl_x25` and `gl_tty`, handling DSA, TCP/IP, X.25 and Asynchronous communication respectively (`gl_x25` and `gl_tty` are currently only available for UNIX/Linux). Most of the *`Host Links`* products use one line handler at a time, but *`Pthru`* is an exception as it uses two line handlers, normally `gl_dsa` on the mainframe side and `gl_tty` on the terminal side.

Handler/product relationships

Any product using the *Gline API* to communicate with the line handler can be run using any line handler. This is one of the strengths of the architecture.

There are restrictions enforced by the external environment. For example the Questar DKU emulation *Qsim* can normally only be used over a link that the Bull primary network regards as synchronous. In practice this means it can only be run over a DSA or TNVIP link.

The following is a list of the all the *Host Links* products and the line handlers they normally use on the host side:

Qsim	gl_dsa, gl_tcp
V78sim	gl_dsa, gl_tcp, gl_tty
G3270	gl_dsa, gl_tcp, gl_x25
G5250	gl_tcp
Pthru	gl_dsa, gl_tcp, gl_tty
Ggate	gl_dsa
Gspool	gl_dsa, gl_tcp
GUFT	gl_dsa
GlAPI	gl_dsa, gl_tcp, gl_x25, gl_tty

DSA connections are made over a TCP/IP network, using the Internet standard RFC1006 transport protocol. Both GCOS8 and GCOS7 support RFC1006 DSA connections.

The *G&R/Ggate* product off-loads the DSA session protocol into gateways. By running Ggate on the system(s) with the mainframe connections the emulator platforms need only the very small and efficient Ggate protocol layer to connect over TCP/IP to a Ggate gateway with full native Bull network functionality. Ggate makes the mainframe connection using RFC1006.

The ***Host Links Installation and Configuration*** manuals cover the G&R DSA stack installation and setup for the supported UNIX, Linux and Windows server systems.

Bull systems can be accessed using Telnet, TNVIP, TN3270 or TN3270E to connect to a Bull front-end. However, Ggate with RFC1006 increases throughput as compared to these gateways. Glink when connecting to Ggate gives a real, fully functional DSA session over the TCP/IP network between Glink and Ggate as compared to the limited terminal session offered by TNVIP, TN3270 or Telnet. Third-party clients with only Telnet/TN3270/TNVIP protocol can connect via Ggate, which supports these protocols on the client side.

IBM systems can be accessed using DSA into the Bull front-end and the OSF/SNA gateway (supported by MainWay). IBM systems can also be accessed using Telnet 3270 (TN3270 or TN3270E) to connect to any TN3270⇔SNA gateway or front-end. The MainWay gateway, the TN3270 server on the Bull DPX/20 UNIX systems, the IBM TN3270 front-end and the TN3270 server for Windows are all qualified. Additionally, IBM AS400 systems, or other host systems that supports the 5250 family of terminals, can be accessed using Telnet 5250 (TN5250) connecting through a Telnet 5250 server.

Enclosures and ‘turn’

The interface between a *Host Links* product and the line handler is global throughout the G&R product range, and for all user applications using one of the *GlAPI* programmatic interfaces for communication.

The functions for receiving and sending data include an indication of the ‘enclosure’ level for the data received, or to be sent. Enclosures are very important in the more advanced line handlers, particularly the Bull DSA handler. Enclosures are supported for all line handlers to make the API global and to make applications using the API independent of the type of communications line. If the enclosure level signaling is unreliable the product using the line handler will need special parameters to help it 'guess' when mainframe output is finished and it is able to process the data received.

The following enclosures are defined:

none
the enclosure is used for convenience of delivery to or from a lower layer of protocol and has no significance for the product using the interface.

end of segment
This is a complete session protocol unit, but is treated as ‘none’ above.

end of message
Defines a complete logical block of data transmitted or received, but indicates that more logical blocks will follow. This is the Bull DSA 'End of Quarantine Unit'.

end of group
Used by interactive products to indicate that the sender will now wait for input before continuing. This is known as the ‘turn’ concept.

Using Bull DSA terminology; when transmitting or receiving data, an application record, called a ‘message’, can be split (partitioned) into several DSA session protocol units called ‘letters’. The letter can be segmented due to size constraints at the session level (the maximum letter size). The session protocol units might have to be split again into several transport protocol units called ‘fragments’ for transmission due to size constraints at the transport level. Finally the transport units might be split again into ‘packets’ because of constraints at the network level.

Whereas a simple handler might have no session protocol unit size restriction it might have a transport size restriction, or a network packet size restriction (e.g. TCP/IP packet size) that would cause fragmentation of the application record for transmission purposes. Depending on the interface between the handler and the lower level protocol this might cause the receiver to see a series of data blocks with enclosure ‘none’.

In two-way-alternate sessions ‘end of group’ is used to signal that a reply is now required (turn). The DSA record header indicates enclosure and turn, and allows the holder to transmit. There is really no turn concept in a two-way-simultaneous dialog. However, the DSA records of the DSA handler can still use ‘end of group’ enclosures and indicate turn to the Bull mainframe applications, which although they might support two-way-simultaneous dialog, can still require a DSA ‘end of group’ before they will handle an incoming message.

Note that most *Host Links* products and host applications are not in fact two-way-simultaneous in the sense that they send and receive simultaneously at the application level. Two-way-simultaneous means in practice that the application or communications product can break the ‘send, receive, send, receive’ pattern of interactive dialog. Thus an emulator or concentrator connected to a TP8 application configured for two-way simultaneous dialog will be able to send a second transaction before it receives an answer to the first. In general this allows applications to not answer certain transaction types (data entry) for efficiency.

`Host Links` products by default always offer to establish either two-way simultaneous or two-way alternate sessions, and it is the mainframe application that chooses the mode.

Host Links Trace

If you experience any kind of problem when using a Host Links application, the application trace file and/or the line handler trace file will provide useful documentation of the problem.

Trace activation

The Host Links products automatically create sub-directories in the debug directory when debug is activated: at product level using the `-dbg` parameter, or at line level using the `-d_` or `-s_` parameters to the line module.

**Windows
server**
	`gspool -id gs1 –dbg –ps \\SERVER\LEXMARK`
	`-li dsa -da tptst -d_ on`

**UNIX
Linux**
	`gspool -id gs1 -dbg-pc lp -li dsa`
	`-da tptst -d_ on`

Most G&R products include a facility for setting product or line parameters dynamically. It is therefore generally possible to turn on debug or trace without modifying the command line or configuration of a production system.

Trace types

All Host Links products accept a parameter `–dbg,` which starts an application level trace of internal events. This is useful when investigating malfunctions or looking closely at product behavior.

All Gline line handlers accept a parameter `-d_` to turn on a data trace. It records data and enclosure level being exchanged with the line handler. It is useful when documenting product malfunction e.g. an emulation error, because it records exactly what the host sends and what the G&R application replies. It can be used to simulate a customer situation, reproduce a problem and to verify that a correction fixes the documented problem.

All Gline line handlers accept a parameter `-s_` to turn on a session trace. It records the raw data being exchanged between the line module and the underlying transport layer (e.g. TCP socket), as well as internal events and protocol states. It is useful when investigating protocol failures such as unsuccessful connect attempts or abnormal disconnections.

Structure

The Host Links file structure includes a debug directory to collect the trace and debug files in one location where the permissions can be adjusted as required for security. By default only the Host Links administrator can access the directory. The debug directory is created by the initialization procedure and located (by default) in:

Windows server	`\gar\debug`
UNIX Linux	`/usr/gar/debug`

If the application is a client type of application, a debug sub-directory with the same name as the user (UNIX username or PC login name) is created and all debug files are located there. This includes the line level trace except in the special case where the client application connects via Ggate and the line level trace is written on the Ggate system using the Ggate DSA node name as a debug sub-directory.

If the application is a server type of application, then a sub-directory will be created using the DSA node name on behalf of which the server application is executing. If the server does not use DSA the default local session control name is still used if there is a `dsa.cfg` file. If there is no `dsa.cfg` file then the system’s UNIX or Windows communications node name is used. You can find this name using the command `uname –n` on UNIX systems, or the Network section of the control panel on Windows systems. This covers situations where several instances of a server are executing on the same system and accepting incoming calls to different DSA node names, or where several Host Links systems using the same server product share a file system.

Tracing Ggate

When Glink, a Host Links client or a customer application based on GlAPI connects through Ggate to the application, the line handler trace is generated on the Ggate system, with the name and location shown in the table:

Windows server	`\gar\debug\NODE\gga`**`NN-PPPP`**`.dbg`
UNIX Linux	`/usr/gar/debug/NODE/gga`**`NN-PPPP`**`.dbg`

NODE is the local DSA node name used by the Ggate system.

The trace file name consists of the prefix `ggaNN-` followed by the IP-address of the client, suffixed by `.dbg` for a terminal session or `–dbp` for a printer session. The following is a trace file name for Ggate session sequence number 5 executing on Host Links system `GRDL` initiated from a Glink client on IP-address `jim.gar.no`:

```
gga05-jim.gar.no.dbg
```

This file, and possibly also a Glink debug file and a Glink communication trace file activated by the `/J` command line parameter will be needed by the support engineer investigating any problem.

To enable a line handler trace through Ggate the product’s start-up command or configuration file would look like this:

```
-LI YYY:ZZZZ -S_ -D_
```

(YYY =line handler identification, i.e. DSA)
(ZZZZ =IP-address of the system running Ggate)

Examples - G&R products

Examples of directory and file names in the debug structure are:

`/usr/gar/debug/jim`	**Debug directory for user ‘jim’**	
`qsm.dbg`	Qsim emulator debug file	`-dbg`

`qsm-gli.dbg`	Qsim host line trace	`-li dsa -s_`
`pth-glit.dbg`	Pthru terminal line trace	`-term -s_`
`pth-glih.dbg`	Pthru -host line trace	`-li dsa -s_`
`g32.dbg`	G3270 emulator debug file	`-dbg`
`g32-gli.dbg`	G3270 host line trace	`-s_`
`/usr/gar/debug/mike`	**Debug directory for user ‘mike’**	
`v78.dbg`	V78sim emulator debug file	`-dbg`
`v78-gli.dbg`	V78sim host line trace	`-li dsa -s_`
`guf.dbg`	GUFT client debug file	`-dbg`
`guf-gli.dbg`	GUFT client host line trace	`-li dsa -s_`
`/usr/gar/debug/en01`	**Debug directory for node ‘en01’**	
`guf.def`	GUFT server debug file	`-dbg`
`guf-gli.def`	GUFT server host line trace	`-li dsa -s_`
`gli-gli.dsa`	DSA listener host line trace	`-s_`
`gsp.def`	Gspool (default `-id`) debug file	`-dbg`
`gsp-gli.def`	Gspool (default `-id`) host trace	`-li dsa -s_`
`gga01-mike.gar.no.dbg`	Ggate line trace, first Glink	`-s_`
`gga02-mike.gar.no.dbg`	Ggate line trace second Glink	`-s_`
`/usr/gar/debug/en02`	**Debug directory for node ‘en02’**	
`gsp.abc`	Gspool (`-id abc`) debug file	`-dbg`
`gsp-gli.abc`	Gspool (`-id abc`) host trace	`-li dsa -s_`
`gspc-gli.def`	Gspool DPF8 command trace	`-li tcp -s_`
`gspd-gli.def`	Gspool DPS8 data trace	`-li tcp -s_`
`gsp._00`	Gspool started on demand	`-dbg`

	debug	
`gsp-gli._00`	Gspool started on demand trace	`-li dsa -s_`

CPI-C and Gweb trace files

Gweb uses the CPI-C libraries so the Gweb debug structure is exactly the same as for CPI-C, except that Gweb inserts its own product identifier into the file name structure. CPI-C applications use the ‘client’ style of debug and create a debug directory with the UNIX username or PC login name used by the process that started them.

The application level debug (`-dbg`) and line trace (`-s_` and `-d_`) are set in the `cpic.cfg` file. The line trace goes to the debug directory, with the name built up as follows:

```
<product_id><session_id>-<process_id>.<debug_type>
```

`product_id`	*Value*	*Comment*
	`cpi`	CPI-C API
	`cp3`	CPI-C 3270
	`cp7`	CPI-C 7800
	`cpd`	CPI-C DKU
	`gw3`	Gweb3270
	`gw7`	Gweb7800
	`gwd`	Gwebdku
`session_id`	`(nn)`	If multi-session application, 1-63
`process_id`	`n (n n n...)`	Varies by platform
`debug_type`	`dgb`	Application level debug
	`gli`	Line trace

Example:

\gar\debug\system	debug directory for user "system"	
cpi-16.dbg	CPI-C single session debug	-dbg
cpi-16.gli	CPI-C single session line trace	-li dsa -s_
cpi2-123.dbg	CPI-C session 2 application debug	-dbg
gw7-20172.gli	Gweb7800 host line trace	-li dsa -s_

The DSA line handler

You start this handler when you specify `-li dsa` as the line handler for a *`Host Links`* product. If you connect via *`Ggate`* it creates a new line handler process or thread for you, and the new process/thread runs `gl_dsa` on your behalf.

The handler is used to connect to Bull systems directly, both GCOS7 and GCOS8.

The DSA line handlers can (like all others) be used in communication between two Host Links applications; for example for test purposes or when running *`G&R/GUFT`* file transmission between 2 systems.

- The DSA line handler in DSA protocol mode is the preferred way to connect to Bull systems. It implements DSA presentation over DSA session, interfacing to the transport stack (RFC1006).

DSA on Windows servers

Due to limitations in the multitasking capabilities of Windows servers the DSA line handler has been implemented slightly differently than the UNIX/Linux counterpart. On UNIX/Linux, when you specify `-li dsa` for a *`Host Links`* application, a new line handler process `gl_dsa` is launched. Windows servers cannot multitask multiple processes efficiently, and so need to handle multitasking by using threads within a single process. The Windows server version of *`Host Links`* uses a line module server, to deal with this. The line module server is a program with the same name as the listener products on UNIX/Linux, i.e. `nl_dsa`. When you specify `-li dsa` for a *`Host Links`* product running on a Windows server, it contacts the line module server. The line module server creates a new thread to handle the session. Note that if you are using SSL (Secure Socket Layer) between clients and Ggate then you need `nl_dsas`, which is compiled with the necessary SSL routines.

As a result of this, the `nl_dsa` line module server **must** be started before any *`Host Links`* product that uses the DSA line handler. Like all the other *`Host Links`* servers on Windows, the line module server must be launched as a Windows service. Do this by adding the command line to the beginning of the Gservice configuration file:

```
nl_dsa.exe
nl_dsas.exe
```

Transport stacks

The DSA handler interfaces to a transport stack. The transport stack traditionally used by Bull front-ends and mainframes was the OSI-transport stack, and the Host Links products needed a corresponding OSI-transport stack in order to connect. This is no longer the case. Bull supports the Internet standard RFC1006 protocol layer, allowing native DSA sessions to be established over TCP/IP networks.

The DSA line module includes the TCP/IP based RFC1006 interface. We deliver a DSA line module that includes only the RFC1006 interface and this module is optionally installed during the Host Links installation process:

```
Your choices are:
1. DSA over OSI and RFC1006 (requires OSI stack)
2. DSA over RFC1006 only
Your choice is [2]?
```

Please choose 2 if you will be using only RFC1006.

RFC1006

For customers with a GCOS8 Helios system or a GCOS7 Diane system, RFC1006 support is standard. For customers with RFC1006, the G&R products can make genuine, fully functional DSA connections over a TCP/IP network. We supply the RFC1006 protocol stack as a part of the standard delivery of Host Links. No communications software other than the standard TCP/IP stack is needed on the Host Links platform. See the `-who gar` parameter to the `tp` (transport provider) directive in the `dsa.cfg` file.

If you intend to accept incoming connections over RFC1006, you **must** start the `nl_dsa` listener, and it **must** be started from `root` because only `root` can reserve the RFC1006 reserved port 102.

Accepting connections

Products and applications receiving connections must configure a local node name using the `-ln` option, causing the line handler to accept incoming calls. The local mailbox name should then also be specified using the `-mn` option. Because a line handler can deal with only one incoming connection, a system accepting multiple simultaneous connections to a single node name **must** use a listener, described below. The `-ln` option then registers the application with the listener for the given node name as listening on the given mailbox. It is then the listener for the given node name which accepts connections and forwards them to the applications which have registered that they are listening to mailboxes on that node name.

The listener

The `nl_dsa` program listens for incoming connections on a given `SC` (Session Control name, also referred to as DSA node name), and dispatches incoming connections to waiting programs. This allows you to have many programs listening for incoming connections on the same `SC`, but for different mailboxes.

If you are using SSL (Secure Socket Layer) between the clients and Ggate running on a Windows system you must use nl_dsas, which has been compiled with the SSL routines. This is because the listener supplies a DSA line handler thread for each outgoing session on Windows, and the threads used by Ggate must have SSL routines for encryption between the client and the line handler.

Listener parameter overview

Option	**value**	**Explanation**
`-prot`	DSA	Session protocol to be used. Default DSA
`-amp`	on/OFF	Allow multiple clients (Glink) with same `–PCO` (printer mailbox name)
`-dbg`		Enable data and event tracing. Writes to the debug file `gli-gli.XXX` in the default debug directory
`-e_`		**Windows** only: enable Windows Event Log Trace

`-icolim`	n 50 200	Number of simultaneous incoming sessions accepted by the **Windows** listener before it begins to reject them in order to avoid unpredictable errors due to Windows service resource limitations. The default is 50. You can increase the default to 200 by running the Host Links Windows services in the foreground. Use the ‘foreground’ parameter to Gservice (see the Gservice documentation in the Host Links for Windows installation manual).
`-icorej`	xxxx	When the **Windows** listener begins to reject incoming sessions because it has reached the limit `-icolim` it will by default use DSA reason code 0106 (mailbox saturated). You can change it to e.g. 0105 (mailbox inoperable).
`-ln`	SCID	Set if multiple SC records are defined and you want to listen for incoming sessions on other than the default SC. Also set if you want to run several listeners
`-lp`	Port number	Local port number (applicable to RFC1006). Default 102
`-rej`	Reason code	Overrides the DSA reason code used when rejecting an incoming session connect from the mainframe. Defaults to 0104 for unknown mailbox and 0106 for busy mailbox

Using the Windows listener

For Windows servers the `nl_dsa` line module server (listener) **must** be started before any *`Host Links`* product can use the DSA line handler, see the section describing use of the DSA handler on Windows servers. The same line module server also carries out the listen function, and so any product using the `-ln` option registers the mailbox for which it will accept connections with the line module server (listener) that is handling connections for the node name specified in the `-ln` parameter.

Using UNIX/Linux listeners

If you need more than one listening application on a UNIX/Linux system, either using the `-ln` option directly or implicitly through the `-co` or `-pco` options explained below, then you must start the listener process, `nl_dsa`. It can be started manually, but typically you should set it up such that the command is run each time the UNIX/Linux platform boots. This process has to be started before any other *`Host Links`* product that receives incoming connections. If the DSA listener is accepting incoming RFC1006 connections, it must run as `root`, because access to the RFC1006 port is restricted to `root`.

On systems running AIX or HP-UX, you can do that by starting it directly from `/etc/inittab`, by placing a line like this at the end of the file:

```
nl_dsa:2:once:/usr/gar/bin/nl_dsa
```

On other UNIX or Linux systems and you can create a shell script, e.g. `S99nl`, with the content below, and place it in the `/etc/rc2.d` directory:

```
/usr/gar/bin/nl_dsa
```

The `nl_dsa` program accepts a node name parameter, `-ln nodename`. This option needs only be specified if you have defined multiple `SC` records in your `dsa.cfg`, and want to listen for incoming connects on other than the default `SC`. This would be the case if you were accepting incoming calls to multiple node names and needed multiple listeners.

The listeners also accept the line session trace parameter `-s_`, which causes them to produce a line trace in the debug directory where they record the connections established through the listener.

Commands accepted from Gmanager

Standard commands

The commands that are accepted by all servers are:

- DOWN - terminates the server
- STATUS - reports server-specific status information to the log file
- PARAM - brings up a dialog box that allows the operator to give a command line parameter to the server. Note that some parameters do not work when given interactively i.e. they can only be handled at server startup time
- DEBUG ON/OFF - toggles on and off tracing interactively

Listener-specific commands

Windows

RELO reloads the on-demand startup file:

```
config.dsa / config.diw.
```

LSOD lists all active on-demand processes

UNIX/Linux

The listener supports only the default commands.

Starting applications on demand

You can start applications on demand when DSA connections arrive. All the G&R server-type applications that accept incoming connections (e.g. Gspool, GUFT and Gmailer) can be started in this way, as well as user-written applications using GlAPI.

Start on demand is especially useful for sites that need to emulate many printers. Pre-starting one Gspool process per printer is an administrative problem for sites that have hundreds of them.

You can also accept multiple simultaneous incoming connects to GUFT servers with the same Host Links node name and mailbox name.

The `nl_dsa` listener handles the implementation; so this must be started. A configuration file controls the mapping from the DSA mailbox name (`-mn`) and, optionally, extension (`-mx`) to the command line for the application to be started. There is one file for each DSA node name (`SCID`) for which connections are being accepted. It is placed in the directory:

```
/usr/gar/servers/<scid>.gli
```

The file is `config.dsa`.

Example file `/usr/gar/servers/grdl.gli/config.dsa`:

```
* Three printer mailboxes
listen -mn printer1 -cmd gspool -pc “lp -dprt1”
listen -mn printer2 -cmd gspool -pc “lp -dprt2”
listen -mn printer3 -cmd gspool -pc “lp -dprt3”
* UFT server
listen -mn filetran -cmd guftsrv
* User written application
listen -mn userapp -lim 1 -ext -cmd userapp arg1 arg2
```

Explanation:

All blank lines and lines starting with # or * are ignored. The only directive handled is `listen`. It takes the following options:

`-mn`	Local mailbox name
`-mx`	Local mailbox extension
`-lim`	Used only if you want to enforce an upper limit on the number of simultaneous connections to this mailbox
`-ext`	Used to indicate that this is an external command, i.e. not a G&R product, but a user application written using GlAPI
`-cmd`	The command to be executed. It must be the last option on the line

The file can be updated at any time without terminating nl_dsa. The listener checks the timestamp on config.dsa every 5 minutes. If it has been updated nl_dsa reloads it. On Windows systems you can force an immediate reload using *`Gmanager`* to issue the command `relo` to nl_dsa.

The file above allows an unlimited number of simultaneous connections to mailboxes `printer1`, `printer2` and `printer3`. For each connection to one of the mailboxes a new instance of the corresponding Gspool is started, regardless of how many were already executing. Connections to mailbox `filtran` each start a new instance of *`GUFTSRV`*. Only one instance of `userapp` is allowed to start, new connections to `userapp` are refused until the first instance terminates.

If an application connects using both a mailbox name and extension then there must be an entry that matches both. The entry will be used to start the application unless the limit (`-lim`) is reached. Two entries with the same mailbox name must have different extensions.

If an application connects using only the mailbox name then the first entry in the file with a mailbox name that matches will be used to start the application until the limit (`-lim`) is reached. If the limit is reached then the next entry with a matching mailbox name is used.

User written applications using this feature should be written like any other application using GlAPI, but be aware that:

- The session connection is passed on the standard input file descriptor, so this descriptor must not be changed.

- Additional information is passed using environment variables, so these must not be cleared.

- The application should terminate as soon as the session terminates. It should not itself go on to listen for further incoming connections. In other words, it should handle one session and then die. A new instance of the application will be started to handle the next incoming connect.

Issuing connections

When the handler issues a connect it asks by default for either a two-way simultaneous or alternate session, leaving it up to the acceptor to decide. TSS on GCOS8 would always choose alternate, but the DACQ gateway to DMIV TP or the CXI interface to TP8 can choose either, depending TP configuration. Two *`Host Links`* products communicating with each other would by default choose a two-way-simultaneous session.

Two-way-simultaneous sessions do not have a 'turn' concept, but the DSA session records indicate the enclosure level specified by the sender. Thus data blocks may be delivered with enclosure 'end of group' indicating 'turn'. GCOS8 DACQ applications using interactive dialog will generally indicate 'turn' at the end of their output even if they are configured as two-way simultaneous.

The handler by default asks for an ASCII session when connecting to GCOS8, and an EBCDIC session for GCOS7 and IBM hosts. When an ASCII product like Qsim uses an EBCDIC session to GCOS7 the code set is transliterated in the line handler. The transliteration can be modified to map ÆØÅ æøå 'correctly' for Scandinavian users (`-SX`). When *`GUFT`*, *`G3270`*, *`G5250`*, the *`3270 CPI-C API`* or the *`5250 CPI-C API`* use an EBCDIC session they do their own transliteration.

Service messages

The line handler accepts service messages for connect `$*$CN`, disconnect `$*$DIS`, break `$*$BRK` (attention, and possibly purge undelivered data, depending on configuration of the mainframe, and/or application), the two attention signals `$*$A1` (attention type 1), `$*$A2` (attention type 2), identification query `$*$ID` and lower case support `$*$LC ON/off` from the communications product using the handler. You can type them and transmit when using a Host Links emulator or Glink/Ggate. Unknown service messages (messages beginning with `$*$` but where the 2-3 following characters are not recognized) will cause an error message.

Interpretation of service messages within the message text can be disabled (`-DSD OFF` as an emulation parameter or `CFIX 2875` in Glink) so that they are transmitted as data to the mainframe. The emulators all have command keys for issuing service messages that may be used instead of typing `$*$` if interpretation is disabled.

User applications using one of our communications APIs from the *`GlAPI`* library must send service messages themselves when the *`Gline API`* is used, but they will be sent automatically by `CMALLC, CMDEAL` and `CMSERR` when one of our *`CPI-C API`*s is used.

The connect command

3 different `$*$CN` formats can be used:

Using a predefined CONAME from `dsa.cfg` to connect:

```
$*$CN coname
```

If the name given is not a predefined CONAME, it is assumed to be the application name and the connect will be carried out using the parameters set for the DSA session (i.e. explicitly by parameter setting from command line and/or the configuration file)

Using the traditional G&R positional `$*$CN` format:

GCOS8 connect to 'appl' (application)

```
$*$CN appl,node[,str,password,person,project,billing]
```

GCOS7 connect to 'appl' (application)

```
$*$CN userid,project,billing,appl,password,node[,,str]
```

Using the Terminal-Manager keyword connect format:

```
$*$CN -scid NODE -dmb APPL –ext . . .
```

The following key words can be used:

```
-dmb, -ext, -scid, -lmb, -str, -usr, -ba, -pj, -pw, -mdmp
```

A GCOS8 connect normally requires only application and node, but you may need to specify a GRTS ID or LID depending on configuration. This is supplied as `-str`. You will need `password`, `person`, `project` if the target site has the ‘Secure Workstation Environment’.

A GCOS7 connect normally requires you to set all of the documented parameters. The string (`-str`) is used by GCOS7 to set application options. These are:

IOF:

```
![NS] [NM] [NEW]     e.g. –str “!NM NEW”
$station-name        e.g. –str “!NM NEW $STA1”
*environment         e.g. –str “!NM NEW $STA1 *ENV1”
```

Note that when using multiple space-separated options they must be enclosed in quotes.

NS	No Startup, user specific information suppressed
NM	No Mail, user messages suppressed
NEW	New session, previous interrupted session lost
station name	station for JOR and SYSOUT
environment	IOF user context for session

TDS

```
?<option>[%string OR <length><system-header]

e.g.    –str ?B40D25
        -str ?A%JIM
```

option	A: system messages and edited mode B: system messages and unedited mode C: no system messages and edited mode D: no system messages and unedited mode.
string	8 alphanumeric characters to be input to the first TPR of the LOGON transaction
header	service message header in the form of an alphanumeric prefix or a control code

Note that you can set the default values for the connect commands with parameters in the configuration file, but that if no default is configured then the last value used for this particular field becomes the default. So if you make an error in one of the parameters you need not specify the others in your next connect.

If you set all connect parameters in the configuration file you can connect from most of the *`Host Links`* products using a single key connect command, or by transmitting `$*$CN`.

Connecting to GCOS8 CXI applications

You must use `-HM CXI` if you log in to GCOS8 via CXI. When you log in the endpoint (at least for TP8) is:

```
node/mailbox/mailbox extension
```

When you define a `LID` in TP8 one of the parameters is mailbox extension, usually set to be equal to the `LID`. In the connect (`$*$CN`) the first parameter (mailbox name) can be specified as follows:

```
$*$CN XXXXXX YYYY,node
```

`XXXXX`	the TP8 mailbox
`YYYY`	the extension for your LID

When the remote mailbox string contains 8 characters you must follow the mailbox string with the mailbox extension string without a space separator.

Alternatively the remote mailbox is configurable as default application (`-DA`) and in the same way the extension is configurable as default extension (`-DX`).

The disconnect command

```
$*$DIS
```

force disconnection (normally done by application).

The break commands

```
$*$BRK
```

sends attention type 1, against turn if necessary. The host, and/or application session may be configured to purge undelivered data when a break is sent.

The default configuration for break is:

GCOS8	Attention and purge
GCOS7	Attention only
DNS	Attention only

For DSA300 applications (CXI applications such as TP8) an option 'data attention' may be negotiated. For these applications a data attention is used instead of attention when transmitting a break.

```
$*$A1
```

sends attention type 1 as for break, but never purges undelivered data.

```
$*$A2
```

sends attention type 2, no purge.

The identification inquiry

```
$*$ID
```

returns the local DSA session control name, the local mailbox name and the terminal type used e.g.

```
$$DSA: Your ID is: SC:EN3D MB:D24701   TM:DKU7107
```

Connection errors

When a connection request from any of the *`Host Links`* applications fails to reach the host application, the handler will return an error message that will help you find the problem. The following sections list the three different classes of errors you may experience, and suggest where you should continue your investigations.

Transport disconnect

'Transport disconnect received' errors after a connection attempt typically mean that you have a physical problem with the link or a network configuration problem. Check with trace tools on the UNIX side if any packets leave the UNIX system. The problem may possibly be on the other side if transport does not manage to establish connection with the session layer there. In this case some of the stacks will deliver the DSA reason code, in which case see the section entitled ***Error codes***.

Session refused

Session refused or *session aborted* errors after a connection attempt typically mean that you have mainframe configuration errors, or a mismatch between the Host Links and mainframe configuration.

Look up the error code displayed in the DSA network system messages and return codes manual, see the section entitled ***Error codes*** or use the *`Gerror`* utility, see section entitled ***DSA test utilities***.

Parameters for the DSA handler

The following parameters apply to this handler (defaults in uppercase):

Parameter	Description
`-LI DSA`	Start gl_dsa in DSA protocol mode

`-LI DSA:GATEWAY`
`:PORT`
`/GATEWAY[:PORT]`
	Start gl_dsa through Ggate running on GATEWAY
	If the port is not standard
	If there is an alternative Ggate

`-AM XXX`	Application mode, modes follow below
`AST`	Datanet AST
`CON`	GCOS8 Console application
`GEN`	General
`SNM`	GCOS6 SNM
`VMF`	GCOS8 Virtual Machine Facility
`-AW on/OFF`	Delay initial turn from host (in the session accept) until real data is received on the session.
`-BRK AT/ATPGLT`	Can be used to override the defaults for purge handling on break. AT gives no purge, ATPGLT gives purge. See section entitled *DSA service messages*.
`-BT n`	Break type 1 (Attention 1 default for DSA200), 2 (Attention 2 special cases) or 3 (Data attention default for DSA300)
`-CM xxxx`	Connection mode to be used when connection is initiated. Defaults to offer all modes.
`TWSI`	Two Way Simultaneous, Initiators turn
`TWSA`	Two Way Simultaneous, Acceptors turn
`TWAI`	Two Way Alternate, Initiators turn
`TWAA`	Two Way Alternate, Acceptors turn
`-CO xxxxxx`	Connection name. CONAME in `dsa.cfg`

Parameter	Description
`-CP DSA300`	Connection protocol DSA300
`DSA200`	Connection protocol DSA200
`-CS ASCII/EBCDIC`	Explicit setting of code set (ASCII or EBCDIC)
`-CW nn/30`	Connect wait time in seconds before failing
`-D_ on/OFF`	Trace data exchange with transport layer

`-D? xxxxxx`
`-PW`
`-PSSW`
	Default password

`-DA xxxxxxxx`
`-DMB`
	Default application mailbox for connect. `GCOS7DEF` can be used to deliver a blank mailbox to GCOS7 when the mailbox is set by the project

`-DB xxxxxx`
`-BA`
`-BILL`
	Default billing

`-DDT`	Delayed turn in tenths of a second. Often needed by GlAPI programs, GlinkWeb and Gweb. In case the mainframe gives the turn, then demands it back.

`-DN xxxxxx`
`-SCID`
	Default node to which you will connect

`-DP xxxxxx`
`-PJ`
`-PROJ`
	Default project

`-DU xxxxxx`
`-US`
`-USR`
`-USERID`
	Default userid

`-DX xxxx`
`-EXT`
	Default application mailbox extension
	4 characters.

`-DEE`	Delayed End to end ACK
`-E8 ON/off`	8 bit bytes on communication line
`-EE on/OFF`	End to end ACK
`-FA on/OFF`	Do not allow change of application

Parameter	Description
`-FM on/OFF`	Do not allow change of mailbox
`-FN on/OFF`	Do not allow change of node
`-HM DPS8`	Host type GCOS8
`    CXI`	Host type GCOS8 with CXI
`    DPS7`	Host type GCOS7
`    DPS6`	Host type GCOS6
`    ROUTER`	Host type GCOS6 NTM
`    DN`	Host type Datanet application
`    HostLink`	A Host Links system (UNIX/Linux/Windows)
`    IBM`	Host type IBM via Datanet SNA gateway
`-IE ON/off/0/1/2`	Insert ETX/CR at end of message. 0 is OFF, 1, is ON, and 2 is ON for messages from the mainframe
`-II nn`	Inactivity timer in minutes before automatic disconnect
`-IOC nn`	Periodically (every nn seconds) update IO counts for the session. Reported by Ggate to Gproxy
`-LM T32782`	Log mode, for host type IBM only
`-LN xxxx`	Local node name used when accepting incoming connection (if other than default local SCID)
`-LO ON/off`	Automatic log out of Ggate if the mainframe session is disconnected
`-LT xxxx`	Local node name used for outgoing connections (if other than local SCID)

`-MN xxxxxxxx`
`-LMB`
	Local mailbox name

`-MX xxxx`	Local mailbox extension.
`-NA xxx yyy zzz`	Send statistics to Gproxy network administration stations (up to 4 IP addresses may be given)
`-NB on/OFF`	Enable broadcasting for any/all Gproxy stations

Parameter	Description
`-PCO xxxxxx`	Associated printer session connection name
`-PCN on/OFF`	Associated printer session auto-connect
`-PTS ON/off`	Associated printer session turn simulation
`-PL nnn`	Page/screen length of terminal in lines (max. 255)
`-SX on/OFF`	Scandinavian transliteration
`-S_ on/OFF`	Trace session events and transport data
`-TELBS ON/off`	Controls interpretation of backspace in the data stream to cancel the preceding byte (Ggate/Telnet)
`-TELECHO ON/off`	Controls default for echo (Ggate/Telnet)
`-TI xxxx`	Terminal identity sent to host

`-TM xxxxxx`
`-MD`
`-MDMP`
	Terminal type (Sender Identifier) sent to host. The host replies with its own Sender Identifier. The list of recognized types follows:

`TTY33    TN300`	Async, echo/character mode, treated as TTY
`TTU8221  TTU8223`	Async, echo/character mode, treated as TTY
`NVT`	Async, Network Virtual Terminal, treated as TTY
`VIP7801  VIP7802`	Async, echo/character mode, treated as TTY
`TXT7801  TXT7802`	Async, but in text/forms mode
`VIP7804  VIP7805`	Sync, text/forms mode
`VIP7804V VIP7805V`	Sync 72-line text/forms mode
`VIP7814`	Sync, text /forms. Unlimited attributes
`VIP8800`	Sync, 8-bit ASCII terminal. Forces `-CS ASCII`
`VIP9800`	Sync, 8-bit PLW for GCOS7 Diane. `-CS EBCDIC`
`HDS5    TWS2255`	Async HDS3/5/7, echo/char mode, treated as TTY
`HDS5T   TXT2255`	Async HDS3/5/7 in text/forms mode
`HDS7    TWS2255V`	Sync, 7800 8-bit 72-line `-CS ASCII`
`VIP7700 VIP7760`	Early sync text/forms terminals. Replaced by VIP7800 and DKU

Parameter	Description
DKU7001 DKU7002	Async, treated as TTY
DKU7102	Async, but with forms and attributes
DKU7007 DKU7107	Sync, text/forms
DKU7007D DKU7107D	Sync, text/forms, supporting local forms
DKU7211 DKU7211D	Colour model, without/with local forms
DKU9107	8-bit PLW DKU for GCOS7 Diane. -CS EBCDIC
PRT722X A2	Synonyms. Printers for DKU
TTU8124 TTU8126	TTU is an async printer
PRT9220	8-bit printer for GCOS7 Diane. -CS EBCDIC
IBM3270 IBM3270A	EBCDIC/ASCII 3270 models
IBM3278 IBM3278A	EBCDIC/ASCII 3270 models
IBM3279 IBM3279A	EBCDIC/ASCII 3270 models
IBM3287	Printer
DSAIP	DSA information processor. Used by TS8 (1001)
G8TP	GCOS8 TP8. Used also by SAFE (100A)
HL64	All GCOS7 applications (1004)
HL66	GCOS8 DAC applications, TSS (1005)
HXnnmm	nnmm are four hex digits
nn	list, 10 for information processor, 20 for terminal
mm	DSA code of processor or terminal
HX0000	Some CPI-C application/application connections
BROWSER	Terminal for TDS-Web and Web8
-UD xxxxxx -UR -STR -STRG	Default ‘GRTSID/LID/user string’ for GCOS8 Application options for GCOS7
-WI nnn	Page/screen width in characters (max. 255)

DSA parameters in detail

-LI DSA and -LI DSA:GATEWAY[:PORT]/GATEWAY

This parameter starts the DSA line handler. If the colon and a gateway name are included the handler is started on the Ggate gateway with the symbolic name or IP address `GATEWAY`. If the Ggate port is non-standard, then it is specified with `:port`. If there is an alternate Ggate then this is specified with `/GATEWAY` and `:port` if non-standard.

It also signals that the parameters following are for the line handler. The first group of line handler parameters **MUST** be preceded by a `-LI` parameter.

Parameters to *`Host Links`* applications are positional in the sense that they must be delivered to either the application or to the line handler. Parameter `-USER` can be used to switch back to parameters for the application and `-HOST` to switch back again to the line handler. They must be repeated if necessary before each group of parameters.

-AM AST/CON/GEN/SNM/VTM

```
v78sim -li dsa -hm DPS6 -am gen
```

This parameter sets the communication parameters needed for connections to applications that do not follow the normal pattern. The default is `GEN` (generic).

You can use it to set `CON` to connect to the GCOS8 console application, or `VMF` to connect to the GCOS8 Virtual Machine Facility.

-AW on/OFF

Macro driven connects or applications using GlAPI sometimes start before the session is fully established because some host applications give Host Links products the turn, and then take it back, causing the macro to fail. This parameter causes the line handler to wait until the first message is received from the host before telling the application that the session is established.

-BS ON/off

This parameter is only applicable to Ggate connections, where Ggate is acting as a Telnet server for asynchronous terminals. It causes the Ggate/Telnet terminal manager to interpret Backspace (0x08) as an editing command that cancels the preceding byte.

-BT 1/2/3

This parameter changes the default for the break signal. The default for DSA200 is ‘Attention 1’ (`-BT 1`), with ‘Attention 2’ (`-BT 2`) used in special circumstances. The default for DSA300 is ‘data attention’ (`-BT 3`). Data attention is normally negotiated when needed by the mainframe, and this parameter is not necessary.

-CM xxxx

```
qsim -li dsa -cm TWAI
```

This parameter sets a connection mode to be suggested to the other party when the connection is initiated. Four connection modes are supported:

TWSI	Two Way Simultaneous Initiators turn. The applications can send without having 'turn'. The application establishing the session starts.
TWSA	Two Way Simultaneous Acceptors turn. The applications can send without having 'turn'. The application accepting the session starts.
TWAI	Two Way Alternate Initiators turn. The applications must have 'turn' to send. The application establishing the session starts.
TWAA	Two Way Alternate Acceptors turn. The applications must have 'turn' to send. The application accepting the session starts.

By default the handler offers to do both TWS and TWA and leaves the initial choice of turn to the other end. When Gline receives an incoming connect it accepts whatever the other end suggests unless you set the `-CM` parameter.

-CO xxxxxx

```
qsim -li dsa -co b8dcacce
```

This parameter sets a symbolic connection name. The line handler parameters are set in `dsa.cfg` for the `CONAME`. The line handler processes the `-CO` parameter by **first** resetting **all** line handler parameters to their default value, canceling any parameters set before `-CO`, and then expanding the `CONAME`. Any parameters **after** the `-CO` override the parameters set by the expansion, if allowed in `dsa.cfg`. Those of you familiar with DNS will notice many similarities between a connection name and the DNS correspondent object.

-CP xxxxxx

```
qsim -li dsa -cp DSA300
```

This parameter forces the connection protocol to `DSA200` or `DSA300`. For host modes: `DPS7`, `DPS8` and `UNIX` the default connection protocol is chosen by the host (historically it was `DSA200`). For host mode `DPS6,` `CXI`, `IBM`, `ROUTER` and `DN` the default is `DSA300`.

-CS xxxxxx

```
qsim -li dsa -hm dps7 -cs ASCII
```

This parameter sets the character set for the session. The default value is `EBCDIC` for IBM, and for GCOS7 except when using an ASCII terminal mode. It is `ASCII` for all other connections.

-CW nn

```
qsim -li dsa -cw 10
```

This parameter defines the time in seconds that Gline waits for a connection to complete before interrupting it and declaring it to be unsuccessful (default 30 seconds).

-D_ on/OFF

Windows Server	`gspool -id gs1 -ps \\SERVER\LEXMARK`
	`-li dsa -da tptst -d_ on`

UNIX Linux	`gspool -id gs1 -pc lp -li dsa -da tptst`
	`-d_ on`

This parameter enables the line handler’s data trace. Data sent and received to the client application is written to a file in the Host Links debug directory. If the `-PCO` parameter is included an additional trace file is written for the printer session. The trace file generated when this parameter is set, can be used as input to various debug tools used by G&R support in order to reproduce e.g. emulations problems. See section entitled ***Host Links trace***.

-D? -PW, -PSSW xxxxxx

```
qsim -li dsa -d? garpassw
```

This parameter defines the default password for the mainframe.

-DA, -DMB xxxxxx

```
qsim -li dsa -da tss
```

This parameter defines the default application mailbox for the mainframe application. It may be up to 12 characters in length. Characters 9-12 become the extension and can be set separately with `-DX`. Normally a connect requires an application mailbox, but in the special case of GCOS7 configurations where the mailbox is configured under the project you can use `GCOS7DEF` to deliver a blank mailbox and let the project decide it.

-DB, -BA -BILL xxxxxx

```
qsim -li dsa -db expensive
```

This parameter defines the default billing for the mainframe.

-DDT n

```
qsim -li dsa –ddt 5
```

This parameter delays delivery of the turn to the application by `n` tenths of a second. Sometimes a mainframe sends data and the turn, but then demands the turn back again in order to deliver more data. If the application is waiting for the turn as a signal to take action, it might take action prematurely if the first turn is delivered. By waiting a few tenths of a second at the line level, delivery of the 'false' turn can be cancelled if the mainframe demands the turn back.

-DN, -SCID xxxx

```
qsim -li dsa -dn b8dt
```

This parameter defines the default node for the mainframe. It must be an RSC record in the `dsa.cfg` configuration file.

-DP, -PJ -PROJ xxxxxx

```
qsim -li dsa -dp unixcn
```

This parameter defines the default project for the mainframe.

-DU, -USR, -USERID xxxxxx

```
qsim -li dsa -du mmluser
```

This parameter defines the default userid for the mainframe.

-DX, -EXT xxxx

```
qsim -li dsa -hm cxi -dx gr01
```

This parameter defines the default mailbox extension for the mainframe. The mailbox extension is used when connecting to CXI (GCOS8) subsystems such as TP8 (CXI requires that `-HM CXI` be set), and will normally contain the LID (Logical ID). The mailbox extension can be up to 4 characters long.

-E8 ON/off

This parameter can be used to override the Gline default, which is to allow 8 bit bytes on the communication line. A number of mainframe applications were developed in a period when communications were restricted to 7 bit characters (the eighth bit being used for parity, and stripped by the terminal). These applications sometimes set the eighth bit, but work because the terminals stripped it. Setting `–E8 off` will give the same effect for Gline connections.

-EE on/OFF (DSA200 to GCOS8 only)

```
qsim -li dsa -ee on
```

This enables end-to-end-ACK. The line handler attempts to negotiate EEACK with the DAC or DAC-Q gateway. If successful Gline acknowledges DSA records that request it by sending the required ACK as an expedited message. EEACK is designed to delay further mainframe output until the current block has been processed. It is often used by applications that send print reports to slow ROP printers connected to the front-end, to stop accumulation of unprocessed print blocks. Normally G&R products can process blocks faster than the line speed so this is never necessary.

-DEE on/OFF (DSA200 to GCOS8 only)

```
gspool -ts -li dsa -dee on
```

The line handler attempts to negotiate EEACK with the DAC or DAC-Q gateway, but it converts it to a ‘turn’ based mechanism on the client side. If successful Gline will convert any received DSA record that requests an ACK to be an EGI (End of Group) record, giving the ‘turn’ to the G&R application. When the application replies Gline will ACK the previous record. This will stop accumulation of unprocessed host output in the Gline buffers.

-FA on/OFF

```
qsim -li dsa -da tss -fa on
```

This restricts connections to the default application name (`-DA appl`). It must be entered after the `-DA` parameter.

-FM on/OFF

```
qsim -li dsa -mn jim -fm on
```

This locks the mailbox name (`-MN mbxname`). It must be entered after the `-MN` parameter. The local mailbox name is used for unique identification of the terminal. It can be used to generate the GRID or LID in the Datanet.

-FN on/OFF

```
qsim -li dsa -dn b8dt -fn on
```

This locks the configured default mainframe node (`-DN node`). It must be entered after the `-DN` parameter.

-HM DPS8/xxxx

```
v78sim -li dsa -hm dps6
```

This sets the target mainframe type. You select a set of internal parameters that control the connection and dialog with the mainframe. The default is DPS8.

mode	Explanation
`DPS8`	Host type GCOS8 with DAC/DACQ Gateway. This mode is used when communicating with TSS and DMIV/TP.
`CXI`	Host type GCOS8 with CXI. This mode is used when communicating with TP8 and other CXI applications.
`DPS7`	Host type GCOS7. Used when communicating with GCOS7.
`DPS6`	Used when communicating with GCOS6 systems through SNM.
`DN`	Host type DNS. Used when connecting to DNS applications.
`ROUTER`	Host type GCOS6 NTM.
`HostLink`	Used when communicating with another Host Links system. Gmailer and GUFT can make such connections.
`IBM`	Used when communicating with IBM hosts through the Janus/OSF gateway in the Datanet.

-IE ON/off/0/1/2

```
v78sim -li dsa -ie OFF
```

This controls insertion of ETX/CR at the end of messages. If the terminal is in the TTY class (e.g. `VIP7801`) CR is appended, otherwise ETX is appended. The default is `ON` for data sent to DPS6 and DPS8 mainframe applications. The values 0 and 1 are OFF and ON respectively. The value 2 is a special case where ETX is not added when sending to the mainframe, but is added to messages from the mainframe.

-II nn/15

```
v78sim -li dsa -ii 30
```

This sets an inactivity timer, which defines in minutes how long the session can be inactive before being automatically disconnected. By default sessions are never disconnected. If `-ii` is used but `nn` is not set it defaults to 15 minutes. It can be used at the DSA node level (`RSC` directive) in the `dsa.cfg` file to set the timer for all sessions to a specific remote node.

-LL nn

```
v78sim -li dsa -ll 2048
```

This defines the ‘line length’, which is in fact the size of the largest block of data that can be sent or received by an application. The default is 6 k (6144).

-LM T32782

```
g3270 -li dsa -lm d4c32782
```

This defines a log mode, and is used for `IBM` mainframes only.

-LN XXXX

```
gspool -li dsa -ln
```

This enables accept of incoming connections. The `XXXX` is optional and specifies the local node name; otherwise the default node name is taken from `dsa.cfg`. `XXXX` must match an `SC` record in `dsa.cfg`.

-LO ON/off

This line parameter can be used by Ggate clients to stop the default automatic log out of Ggate if the mainframe session is disconnected. The client then remains connected to Ggate with a $*$ prompt, and can issue local dialog commands. The printer session, if any, remains connected or listening if it is accepting connects from the mainframe.

-LT XXXX

```
gspool -li dsa -lt grdw
```

This defines a non-standard local node name for outgoing connections. `XXXX` must match an `SC` record in `dsa.cfg`.

-MN, -LMB xxxxxx

```
qsim -li dsa -mn mike
```

This sets the local mailbox name. It can be up to 12 characters long. Characters 9-12 become the extension and can be set separately with `-MX`. By default the handler assigns unique mailbox names based on the program's process id (`pid`) in the form `Dnnnnn.`

The mailbox name can be used as additional user identification for security. On GCOS8 systems characters 3 and 4 may be used to generate a GRID (see -GRID and -NGRID parameters in the DNS configuration), or the first three (BCD TDS) or four (ASCII TP) characters may be used to generate the LID (see -LID and -NLID parameters in the DNS configuration). For IBM (OSF/Janus) connections it can be used to select a specific LU or LU pool.

-MX xxxx

```
qsim -li dsa -mn arild -mx gr01
qsim -li dsa -mn “arild   gr01”
```

This defines a local mailbox extension. The mailbox extension is a 4-character string that is appended to the 8 first characters of the mailbox name. The mailbox extension can be included in the local mailbox name (`-MN`) as shown, but quotes are necessary if spaces are needed to make the mailbox name 8 characters long.

-NA wwwwww xxxxxx yyyyyy zzzzzz

```
qsim -li dsa -na 192.150.211.11
```

This directs network administration information from the line handler to systems running *`Gproxy`*. Up to four IP addresses (space separated) can be included. See also the `-NB` parameter.

-NB on/OFF

```
gg_tcp -nb on (-li not necessary for servers)
```

This enables broadcasting of network administration information to systems running *`Gproxy`*. Broadcasts allow multiple, but unspecified, systems running *`Gproxy`* to monitor G&R products. This is especially important for servers such as *`Ggate`*, *`GwebS`* and *`GlinkJS`* for which *`Gproxy`* can load-balance and license-share.

-PCO xxxxxx

```
qsim -li dsa -co tp8_app -pco printsess
```

This associates a CONAME for a printer with the terminal session. The Bull mainframes often send print over an independent print session, which has to be associated with a terminal session so that the print can reach the terminal's printer. The printer's CONAME must be defined in dsa.cfg and must expand into the set of parameters needed to establish the printer session.

Glink, the *Host Links* emulators and *Pthru* can all accept print output directed by the GCOS mainframe to an independent print session. For example, print directed to a GCOS7 Twriter queue or TDS/TP8 LID defined as a ROP printer can be printed on a printer attached to a PC with *Glink* or to the configured print path of *V78sim* or *Qsim*.

-PCN on/OFF

This causes the associated printer (-PCO) to initiate the connection. The default is to wait for the mainframe to connect to the printer.

-PTS ON/off

This can be turned OFF if the host does not want to get the 'turn' back each time it sends 'turn' to the printer. Leave it to the default value in most cases.

-PL nnn

```
gspool -li dsa -pl 48
```

This specifies the screen or page length of the terminal in lines (max. 255). The default value for a screen is 24, and for a printer is 0. It is required by some mainframe applications that need to know the length of the screen (24 or 72). Some mainframe print spooling systems have to know the page length of your printer. You must then add it to the parameters associated with CONAME (-PCO coname) if you use an attached printer, or to the parameters for Gspool.

-SX on/OFF

```
qsim -li dsa -sx on
```

This selects Scandinavian transliteration tables for conversion between EBCDIC and ASCII. The -SX parameter transliterates EBCDIC #@$ to ASCII [\]. It is seldom used, even in Scandinavia.

-S_ on/OFF

Windows Server	`gspool -id gs1 -ps \\SERVER\LEXMARK`
	`-li dsa -da tptst -s_ on`

UNIX Linux	`gspool -id gs1 -pc lp -li dsa -da tptst`
	`-s_ on`

This enables the session trace in the line handler. It documents session, transport and internal events along with inbound and outbound data for the transport interface. If the `-PCO` parameter is used then an additional trace file will be written for the printer session. See section entitled ***Host Links trace***.

-TI xxxx

```
qsim -li dsa -ti T1234
```

This defines the terminal id, used by the line handler when connecting to the remote system. It can be used by the remote system to generate a LID. The default value is `T000`.

-TM, -MD, -MDMP xxxxxx

```
pthru -li dsa -tm vip7804
qsim -li dsa -tm hx2033
```

This selects the terminal mode that will be delivered to the mainframe. It tells the application which terminal presentation protocol you will use. See section ***Parameters for the DSA handler*** for possible values.

If you need to use some other mode, and know its hexadecimal identification (see the DNS V4 - *Terminal Management Manual*), then you can configure it using `-TM HXxxxx`, where `xxxx` is 2 bytes in hexadecimal.

-UD, -UR, -STR, -STRG xxxxxx

```
qsim -li dsa -hm dps8 -ur glid
qsim -li dsa -hm dps7 -ur !NM NEW $MYSTA *MYENV
```

This defines the default user record for the mainframe to which you are connecting. It can be 2-32 characters long, and is normally used to deliver the GRTS id (GRID) for connection to DAC programs on GCOS8, or the Logical id (LID) when connecting to TDS or DMIV-TP by DACQ.

When connecting to GCOS7 it defines the application options; if you use multiple space-separated IOF options they must be enclosed in quotes. There are no spaces in the TDS string:

```
IOF: ![NM] [NS] [NEW] [$Station] [*environment]
TDS: ?<option>[%string OR <length><syshead>]
```

-WI nnn

```
gspool -li dsa -wi 255
```

This parameter specifies the screen or page width of the terminal in characters (max. 255). The default value for a screen is `80`, and for a printer is `132.` It is required by some host print spooling systems that have to know the page width used in your printer, and must then be added to the parameters associated with `CONAME` (`-PCO coname`) if you use an attached printer, or to the parameters for Gspool.

Other line handlers

The TCP/IP handler

You start this handler (`gl_tcp`) when you specify `-li tcp` as the line handler for a *`Host Links`* product. When you connect from one G&R product (or GlAPI application) to another the TCP/IP handler uses a 'raw' mode; the application data is simply packed in TCP packets and transmitted with no higher-level protocol involved. Terminal sessions use the Telnet protocol to reach standard Telnet servers, TN3270/TN3270E to reach TN3270 servers, TN5250 to reach TN5250 servers and TNVIP to reach TNVIP servers. Terminal sessions use Rlogin protocol when user identification is to be passed between UNIX/Linux client and server systems.

Making connections using TCP

Accepting a connect

If you configure a local port number (`-lp`), products and applications can use the TCP handler to receive connections.

Note that you must give the local port parameter at startup in the command line or configuration file for it to take effect. You can start two products on the same or two different systems and then connect from either to the other to establish interactive communication between two terminals or applications.

Issuing a connect

TCP connections in 'raw mode' are always two-way simultaneous. Any concept of 'turn' to determine initiative must be implemented in the applications. The 'end of data' marker is set following criteria depending on application mode, and is mapped to 'end of group' for the application using the handler. It can be interpreted as 'turn'.

For TNVIP, TN5250 and TN3270 connections the concept of turn is included in the protocol, and mapped to 'end of group' for the applications using Gline.

Service messages

The TCP handler accepts service messages for connect (`$*$CN`), disconnect (`$*$DIS`) and break (`$*$BRK`) from the communications product using the handler. You can type them and transmit when using an emulator or concentrator.

User applications using one of our communications APIs from the *`GlAPI`* library must send the `$*$CN`, `$*$DIS` and `$*$BRK` messages themselves when the *`Gline API`* is used, but they will be sent automatically by `CMALLC`, `CMDEAL` and `CMSERR` when one of our *`CPI-C API`*s is used.

The connect command

The only information required is the IP address in symbolic or numeric format and the port number if it is not the standard Telnet port e.g.

```
$*$CN hostname:portnumber
```

Make connection to ‘hostname’.

Telnet, TN3270, TN5250 and TNVIP servers are all normally configured to use the standard Telnet port, and the TN3270, TN5250 or TNVIP dialects are arrived at by Telnet negotiation.

Note that you can set the default values for the connect commands with parameters in the configuration file, but that if no default is configured then the last value used for this particular field becomes the default. So if you make an error in one of the parameters you need not specify the others in your next connect.

If you set all connect parameters in the configuration file you can connect from most of the *`Host Links`* products using a single key connect command, or by transmitting `$*$CN`.

The disconnect and break commands

All systems accept:

```
$*$DIS
```

force disconnection (normally done by application).

```
$*$BRK
```

interrupt application, against turn if necessary.

Parameters for the TCP/IP handler

The following parameters apply to this handler (defaults in uppercase):

Parameter	**Description**
`-LI TCP`	Start `gl_tcp`
`-AM XXXXXX`	Application mode
`RAW`	Default for other ports than Telnet
`CRTERM`	
`LFTERM`	
`TELNET`	Default for the Telnet port
`TN3270`	
`TN3270E`	
`TN5250`	
`TNVIP`	
`RLOGIN`	
`-AP on/OFF`	Select the printer LU name associated with the LU name used by the screen (`-LU`)
`-DN hostname:port`	Numeric/symbolic IP-address, optional port
`-D_ on/OFF`	Data trace between application and host
`-HO hostname:port`	Numeric/symbolic IP-address, optional port

Parameter	Description
`-HW n`	Enable ‘delayed enclosure’. Number of sec/10 to wait for more downstream data before ‘end of group enclosure’ is posted
`-LP local_port`	Numeric/symbolic: Incoming connections
`-LU luname`	TN3270 LU name
`-DEV devicename`	TN5250 device name
`-NU on/OFF`	Enable Telnet `CRNUL` mode
`-RES mailboxname`	Resource name for TNVIP
`-RP remote_port`	Numeric/symbolic: Outgoing connections
`-SU username`	User name for Rlogin
`-S_ on/OFF`	Session trace, internal events and data
`-TM terminal_type`	Telnet terminal type negotiation

TCP parameters in detail

-LI TCP

This starts `gl_tcp`. It also signals that the parameters following are for the TCP line handler. The first group of line handler parameters **MUST** be preceded by a `-LI` parameter.

Parameters to *`Host Links`* applications are positional in the sense that they must be delivered to either the application or to the line handler. Parameter `-USER` can be used to switch back to parameters for the application and `-HOST` to switch back again to the line handler. They must be repeated if necessary before each group of parameters.

-AM XXXXXX

This is used to select a mode of operation. The following modes are available:

-AM mode	Description
`RAW`	In this mode all data is passed to the application with an end of data marker.
`CRTERM`	In this mode all data buffers are marked with a ‘more data’ marker until a Carriage Return (CR) is found. When CR is received the data passed to the application is marked with an end of data marker.
`LFTERM`	In this mode all data buffers are marked with a ‘more data’ marker until a Line Feed (LF) is found. When LF is received the data passed to the application is marked with an end of data marker.
`TELNET`	In this mode the Telnet protocol is used for connection, negotiation/setting of Telnet options and data traffic.
`TN3270`	In this mode the TN3270 protocol (RFC1576) is used for connection, negotiation/setting of Telnet options and data traffic for a 3270 screen-session. LU name selection and print are supported using the TN3270 extensions in RFC1646.
`TN3270E`	In this mode the TN3270 extensions from RFC1647 are used to improve handling of screen and print sessions.
`TN5250`	In this mode the TN5250 (RFC1205) protocol is used.
`TNVIP`	In this mode the TNVIP protocol is used (RFC1921). The gl_tcp handler acts as a TNVIP client and communicates with a TNVIP server.
`RLOGIN`	In this mode the RLOGIN protocol is used for connection.

Telnet

```
v78sim -li tcp -rp 9998 -am telnet
```

The default application mode is `TELNET` for outgoing connections, unless connecting to a port number other than the `TELNET` default (`23`). If another port is chosen the default mode is `RAW`, which is also the default for incoming connections. If Telnet is required on a non-standard port it must be set.

TN3270 and TN3270E

Gweb, G3270 and Gspool use these modes. In TN3270 mode Gweb, G3270 and Gspool can select LU name. Gspool supports TN3270 print as specified in RFC1646 and used in the SNA gateway from Open Connect Systems.

In TN3270E mode G3270 supports the ATTN and SYSREQ keys as well as selection of a specific LU name. Gspool can be configured as a generic printer or associated printer through the use of TCP line handler parameters `-LU` and `-AP`. Microsoft's MS SNA Server (Windows), Bull's SNA/20 (AIX 4.1) and MainWay are examples of gateways that support TN3270E (RFC1647).

TNVIP

```
qsim -li tcp -am tnvip -tm dku7211
v78sim -li tcp -am tnvip -tm vip7814
```

The Bull TNVIP server on AIX uses a non-standard port, normally 7323, and requires parameter `-rp 7323`. The MainWay TNVIP server normally uses the default (23), but may need the `-res` parameter to select a specific configuration. When using TNVIP you must select one of the allowed terminal types, see `-TM.`

We have implemented the Telnet and TNVIP client protocols in the interest of completing our connectivity offer. Please note however that RFC1006 is supported by all GCOS systems, and if used when communicating with G&R products it will increase throughput as compared to using TNVIP. It will also give a real, fully functional DSA session over the TCP/IP network, as compared to the limited terminal session offered by TNVIP.

-AP on/OFF

Windows Server	`gspool -id gs8 -ps \\SERVER\LEXMARK -li tcp`
	`-am tn3270e -ho some.ibm.system`
	`-lu luname -ap –tm ibm-3287-1`

UNIX Linux	`gspool -id gs8 -pc lp –li tcp -am tn3270e`
	`-ho some.ibm.system -lu luname –ap`
	`-tm ibm-3287-1`

This tells the TN3270E server that the `-LU` parameter is the `LU` name of a screen, and it must allocate the printer associated with the screen's `LU` name in the TN3270E configuration.

-DN ipaddress:port

```
v78sim -li tcp -dn 192.150.211.4:9998
```

This selects the server to which you connect. You can use its numeric IP-address or you can use its symbolic name. In the latter case it must be registered in your hosts file or with your name server. You can append a colon and a numeric port number or use a symbolic name. In the latter case this symbolic name must be registered in your services file.

-D_ on/OFF

Windows Server	`gspool -id gs1 -ps \\SERVER\LEXMARK`
	`-li tcp -am tnvip -d_ on`
	`-ho some.gcos.system`

UNIX
Linux
	`gspool -id gs1 -pc lp -li tcp -am tnvip`
	`-d_ on -ho some.gcos.system`

This enables the line handler's data trace. Any data sent/received to the client application is written to a file in the Host Links debug directory. The trace file generated can be used as input to various debug tools used by G&R support in order to reproduce e.g. emulations problems. See section ***Host Links trace***.

-HO ipaddress:port

```
v78sim -li tcp -ho 192.150.211.4:9998
```

This selects the server to which you connect. You can use its numeric IP-address or you can use its symbolic name. In the latter case it must be registered in your hosts file or with your name server. You can append a colon and a numeric port number or use a symbolic name. In the latter case this symbolic name must be registered in your services file.

-HW interval (tenths of seconds)

```
v78sim -li tcp –hw 5
```

This sets a timeout for signaling 'turn'. It is needed when, after output, the server does not pass a turn signal to a Host Links client that needs it; typically a client simulating an asynchronous terminal connected by Telnet (e.g. *Gweb* connected to a HVX system). In the example, the line handler will pass the turn signal (enclosure 3) to the application when 0,5 seconds have passed without any further output from the server.

-LP local port

```
myglineapp -li tcp -lp 9999
```

This parameter selects the local port for incoming connections, and enables accept of incoming connections. You can use the port number or a symbolic name. In the latter case this symbolic name must be registered in your services file. The default is to not accept incoming connections. It defaults to raw mode.

-LU LUname

This is used in TN3270 mode to ask for a specific LU name on the host side.

-NU on/OFF

```
  v78sim -li tcp -nu on
```

This enables Telnet `CRNUL` mode. Normally Telnet in ASCII mode encodes a carriage return (CR) as a Telnet ‘new line’ by appending a line feed (LF). In this mode a NUL character is appended instead.

-RES resource name

This parameter is used in application mode `TNVIP`. It selects a non-default ‘mailbox name’, which acts as a special access point in the TNVIP server. It has a maximum length of 12 characters, and will be appended to the terminal type (given by the `-TM` parameter), separated by a delimiter. It is used to select a specific terminal configuration in the terminal manager.

-RP remote_port

```
  v78sim -li tcp -rp 9998 -am telnet
```

This selects a remote port for outgoing connections. You can use the port number or a symbolic name. In the latter case this symbolic name must be registered in your services file.

The default port number for the Telnet application modes, `TELNET`, `TN3270` and `TNVIP` is the ‘Telnet’ port number, `23`. If you set another remote port the default application mode is `RAW`. If you want to use Telnet you must set it.

-SU username

This sets the user name to be used when logging in to the Rlogin server.

-S_ on/OFF

Windows Server	`gspool -id gs1 -ps \\SERVER\LEXMARK`
	`       -li tcp -am tnvip -s_ on`
	`       -ho some.gcos.system`

UNIX
Linux
	`gspool -id gs1 -pc lp -li tcp -am tnvip`
	`       -s_ on -ho some.gcos.system`

This enables the line handler's session trace. It documents session, transport and internal events along with inbound and outbound data sent and received through the underlying socket interface.

-TM terminal_type

This selects terminal type for Telnet negotiation performed in application-mode `TN3270`, `TN5250` or `TNVIP`. It is very important to select a terminal type acceptable to the server (IBM TCP/IP front end, generic TN3270⇔SNA gateway or the TNVIP server in a Bull MainWay, AIX gateway, GNSP or Diane).

The 3270 terminal types listed in *RFC1340 Assigned Numbers* are

```
IBM-3278-2          IBM-3278-3
IBM-3278-4          IBM-3278-5
IBM-3279-2          IBM-3279-3
IBM-3278-2-E        IBM-3278-3-E
IBM-3278-4-E        IBM-3278-5-E
IBM-3279-2-E        IBM-3279-3-E
```

The default value in application mode `TN3270` is `IBM-3279-2-E`.

```
g3270 -li tcp -tm ibm-3279-2-E
```

The supported 5250 terminal types are (see *RFC1205)*

```
IBM-5555-C01        IBM-5555-B01
IBM-3477-FC         IBM-3477-FG
IBM-3180-2          IBM-3179-2
IBM-3196-A1         IBM-5292-2
IBM-5291-1          IBM-5251-11
IBM-3812-1          IBM-5553-B01
```

The default value in application mode `TN5250` is `IBM-3179-2`.

```
G5250 -li tcp -tm ibm-3179-2
```

For TNVIP the terminal types listed in *RFC-1921* are

```
 VIP7700             VIP7804
 VIP7760             VIP7804V
 DKU7005             VIP7814
 DKU7007D            HDS7
 DKU7105             VIP8800
 DKU7107D
 DKU7211
 DKU7211D
```

DKU terminal types ending in D imply local form support. Local forms are not supported in our UNIX/Linux based DKU emulations (they are supported in Glink).

```
qsim -li tcp -am tnvip -tm dku7211
```

The X.25 handler

You start the X.25 handler (gl_x25) when you specify -li x25 as the line handler for a *Host Links* product. It is only available for the UNIX/Linux version of *Host Links*, and only on some of the platforms.

Making connections using X25

Accepting a connect

The X25 handler by default does not accept connections because the 'outgoing only' parameter (-OO) is by default ON. If it is set OFF products and applications receiving connections can use the handler.

Note that the local sub-address parameter, the local user data parameter or both must be given at startup in the command line or configuration file to take effect and direct incoming calls to the correct product. You can start two products on the same or two different systems and then connect from either to the other to establish interactive communication between two terminals or applications.

Issuing a connect

X25 connections are always two-way simultaneous. Any concept of 'turn' to determine who may send data must be implemented in the applications. The handler recognizes the X.25 'more data' flag, and packets with this flag are delivered as enclosure 'none'. It delivers the terminating packet without the flag with enclosure 'end of group', and this may be interpreted as 'turn'.

Service messages

The X25 handler accepts service messages for connect ($*$CN), disconnect ($*$DIS) and break ($*$BRK) from the communications product using the handler. You can type them and transmit when using an emulator or concentrator.

User applications using one of our communications APIs from the *GlAPI* library must send the $*$CN, $*$DIS and $*$BRK messages themselves when using the *Gline API*, but they will be sent automatically by CMALLC, CMDEAL and CMSERR when one of our *CPI-C API*s is used. The PAD service messages can only be used from the *Gline API*.

The connect command

```
    $*$CN remoteaddress,userdata
```

The only information required is the X.25 number to be called in numeric format, and the user data if needed in hexadecimal. Remote address is given as a numeric X25 subscription. User data is three hexadecimal bytes.

Note that you can set the default values for the connect commands with parameters in the configuration file, but that if no default is configured then the last value used for this particular field becomes the default. So if you make an error in one of the parameters you need not specify the others in your next connect.

If you set all connect parameters in the configuration file you can connect from most of the *`Host Links`* products using a single key connect command, or by transmitting `$*$CN`.

The disconnect and break commands

All systems accept:

```
    $*$DIS
```

force disconnection (normally done by application).

```
    $*$BRK
```

interrupt application, against turn if necessary.

Parameters for the X.25 handler

The following parameters apply to this handler (defaults in uppercase):

Parameter	Description
`-LI X25`	Start `gl_x25`
`-AC on/OFF`	Accept reverse charges
`-CR on/OFF`	Insert CR at end of message
`-D_ on/OFF`	Write Gline data trace to Host Links debug
`-DS xxxxxx/NONE`	Default Remote address.

Parameter	Description
`-DU hxhxhx/NONE`	Default Remote User data
`-EB on/OFF`	Use EBCDIC on line
`-LU hxhxhx/NONE`	Local User data
`-HW n`	Enable ‘delayed enclosure’. Number of sec/10 to wait for more downstream data before an ‘end of group enclosure’ is posted.
`-OO ON/off`	Outgoing call only
`-PD on/OFF`	X29 PAD support
`-RC on/OFF`	Facility code: Make call with reverse charges
`-RF nnn/NONE`	Facility code: Receive throughput class override
`-RS nnn/NONE`	Facility code: Receive packet length override
`-RW nnn/NONE`	Facility code: Receive window size override
`-S_ on/off`	Write Gline session trace to Host Links debug
`-SC nn/NONE`	Local Sub-address
`-SP on/OFF`	Send data with even parity added
`-UG nnn/NONE`	Facility code: Closed User Group
`-XF nnn/NONE`	Facility code: Transmit throughput class override
`-XS nnn/NONE`	Facility code: Transmit packet length override
`-XW nnn/NONE`	Facility code: Transmit window size override

The Asynchronous (TTY) handler

You start the TTY handler (gl_tty) when you specify -li tty as the line handler for a *Host Links* product. It is only available for the UNIX/Linux version of *Host Links*.

Making connections using TTY

Accepting a connect

The TTY handler accepts connects in the sense that it sees the status of the line change if the other side (e.g. a modem) takes up the correct signals, and reports this as an incoming connect to the *Host Links* product.

Issuing a connect

TTY connections are always two-way simultaneous. Any concept of ‘turn’ to determine who may send data must be implemented in the applications. By default, TTY works completely asynchronously i.e. all data delivered to TTY is transmitted immediately on the line, and all data received is delivered immediately to the *Host Links* product with enclosure none. This applies even to single characters, so products running in ‘echo’ mode can use the TTY handler.

There are parameters for setting various control characters as the ‘end of record’ marker. These parameters implement ‘block mode’ operation. Incoming data is scanned for the end of record marker, and when found the accumulated data is delivered as an ‘end of group’ enclosure, which may be interpreted as ‘turn’.

When sending data there are parameters to map enclosure level to control characters used as ‘end of record’ markers. This functionality is used to deliver the concept of enclosure and turn to our *Glink* emulator running on a PC when connected by an asynchronous line. This is necessary for some ‘synchronous’ emulation modes, particularly Questar DKU emulation.

Service messages

The TTY handler accepts service messages for connect ($*$CN), disconnect ($*$DIS) and break ($*$BRK) from the communications product using the handler.

User applications using one of our communications APIs from the *GlAPI* library must send the $*$CN, $*$DIS and $*$BRK messages themselves when the *Gline API* is used, but they will be sent automatically by CMALLC, CMDEAL and CMSERR when one of our *CPI-C API*s is used.

The connect command

There is no information needed when connecting to an asynchronous line. The TTY handler does not support dial-up modems.

```
$*$CN
```

Connect to the line. This will cause the correct signals to be given on the physical interface and the other side will recognize the line as active.

You can connect from most of the *Host Links* products using a single key connect command, or by transmitting $*$CN.

The disconnect and break commands

All systems accept:

```
$*$DIS
```

force disconnection (normally done by application).

```
$*$BRK
```

interrupt the application, against turn if necessary.

Parameters for the TTY handler

The following parameters apply to this handler (defaults in uppercase):

Parameter	Description
-LI TTY	Start gl_tty.
-BM on/OFF	Block mode. ETX/EOT used as end of record mark.
-CR on/OFF	CR mode. CR used as end of record mark.
-D_ on/OFF	Write Gline data trace to Host Links debug directory.

`-DN XXXXX`	Device name. Name of async port.
`-EB on/OFF`	Add ETB to send data when enclosure is `e_message`
`-EX on/OFF`	Add ETX to send data when enclosure > `e_message`
`-HW n`	Enable ‘delayed enclosure’. Number of sec/10 to wait for more downstream data before an ‘end of group enclosure’ is posted.
`-LF on/OFF`	LF mode. LF used as end of record mark.
`-NX on/OFF`	Disable IXON/IOFF.
`-S_ on/OFF`	Write Gline session trace to Host Links debug
`-ST XXXXX`	stty options

DSA Configuration

Step-By-Step Guide

Configuration check list

Before you start to configure the mainframe and Host Links system you must obtain the configuration parameters you will need. If you already have a functional network, you do not need to change the configuration of the mainframe. Obtain the parameters from the network administrator.

The mainframe system

You will need:

- The DSA node name (SCID, session control ID) of the system

  If DSA session is to be used for DSA200 connections:

  - The DSA area address of the host (the TSN, e.g. 01:24)

- The network address of your mainframe system:

  - The IP address of the RFC1006 port on the FEP or mainframe

The Host Links system

- The DSA node name (SCID, session control ID) for the Host Links system

  If DSA session is to be used for DSA200 connections:

  - The local DSA area address (the TSN, e.g. 05:36)

- The network address of the Host Links system:

  If connecting over a TCP/IP network (RFC1006)

  - The IP address

Configuring the mainframe

GCOS8 configuration

Set up the `routes.wdf` and `dsa_config` files.

This is done using the G8CM utility.

GCOS7 configuration

The network generation source file must be modified to include the description of the Host Links node. This is normally done with the `SYS` or `SYSTEM` macro directive, or with an `RSYS/RSC` pair of directives.

This can be done as follows for a DSA Host Links system:

```
RSYS NAME = GRDL, PF = ‘STID/ISO/SID3’, RSC = GRDL;
RSC NAME = GRDL, ADDR = DSA, SCID = 54:060, SVR=NCC1;
```

Configuring the Host Links system

Configuring the transport stack

DSA is a session level protocol that runs over a transport stack. Today networks are built of commodity products using the TCP/IP communications stack. To facilitate use of existing OSI-based applications over TCP/IP networks a ‘Request for comment’ (RFC1006) was submitted to the Network Working Group that manages the introduction of Internet standards. The RFC is entitled ‘ISO transport services on top of the TCP’ and has become an Internet standard. DSA runs over RFC1006 and TCP/IP networks.

RFC1006 transport

GCOS8 Helios systems support RFC1006. GCOS7 Diane systems also support RFC1006.

When you use the RFC1006 protocol you don’t need any further communications software or configuration of the transport stack. We provide our own RFC1006 transport stack, which is controlled completely from `dsa.cfg`. RFC1006 is an Internet standard protocol that encapsulates an OSI-transport (TP0) protocol in TCP/IP frames. We assume that TCP/IP is already installed on the Host Links platform.

Configuring Host Links

It is assumed here that the Host Links software is already installed on the target platform. For a detailed description of the installation process, see the ***Host Links Installation and Configuration*** manuals for UNIX/Linux and Windows.

The dsa.cfg file

You must configure the local Host Links system as well as the remote host systems you will access. For the local system you need an `SC` directive that contains the DSA session control name and the DSA address if DSA200 is to be used. The following assumes session control ID `is2c` and DSA address `05:44.`

```
sc is2c -addr 05:44
```

For each remote system to which you will connect you need a pair of `rsc`/`ts` directives. The following assumes a host named ph8a

```
rsc ph8a -ts ph8a_rfc
ts ph8a_rfc -class 0 -ns 141.112.xx.yy -tp rfc
```

Finally for each remote system you need a description of the local entity (transport provider) that is used to reach that system. The provider is pointed to by the `-tp` parameter in the `ts` directive. The `tp` directive should have a `-who` parameter to identify the provider.

```
tp rfc -who gar
```

The `dsa.cfg` file may also contain security filters and detailed connection parameters used when making defined host connections (`CONAME`). For a detailed description please refer to the section entitled ***The dsa.cfg configuration file***.

Remember to compile the configuration file using the `glcc` compiler!

The listener

The listener is used for distribution of incoming connections from the remote host(s) to the Host Links applications. The listener must be used if several Host Links applications will receive incoming connects using the same local DSA session control name (TSAP).

The listener should be started automatically whenever the Host Links system is restarted. If the receiving applications are started before the connections arrive then no listener configuration is necessary. Each receiving application registers the mailbox name for which it will accept connects with the listener when the application starts.

If the session listener is to start the Host Links application on demand (when the DSA connection arrives), then you configure it in the listener’s file:

```
/usr/gar/servers/NODE.gli/config.dsa
```

`NODE` DSA session control name of the local system

A sample directive is:

```
listen -mn filetran -cmd guftsrv
```

This starts the GUFT server application whenever a DSA connect to mailbox name `filetran` is received. For details see the section entitled ***The DSA line handler***, sub-section ***Accepting connections***.

Application configuration files

Most of the host links applications have their own configuration files. These are located under the `/usr/gar/config` catalog structure and are used to minimize command line parameters. You do not need to set any of these until the connections are tested out successfully. These files are described in the application manuals.

Test the connections

Make sure the configuration changes described above have been activated. The DSA configuration file on the Host Links system must be compiled. The system may need to be rebooted. If the connections are unsuccessful you need to analyze the communications problems.

Third party tools

There are software tools that may help you analyze most problems. Wireshark is an excellent tool.

Host Links tools

There are several Host Links utilities and debug tracing routines. You don't need to set up any Host Links application to do the initial tests, simply try the *`Gping`* or *`Gtrace`* utilities described in the section: *DSA test utilities.*

When starting any Host Links communication application, you can activate a session trace routine by supplying the `-s_` parameter. For example:

```
qsim ..... -li dsa ...... -s_
```

This will generate a session trace file in the Host Links debug structure. See the *Host Links Trace* section for more detailed information.

The dsa.cfg configuration file

Overview

One or more local session control entities (SC) must be defined. These are looked up using the value of the -ln line module option and contain the DSA session control name and number of the local UNIX/Linux or Windows server system, to be used by applications accepting incoming connects. The SC is also used to identify the calling Host Links node on outgoing connections.

Remote session control entities (RSC) must be defined. These are looked up using the value of the -dn or -scid line module options, the corresponding node name field in *Glink* or $*$CN commands in *Host Links* emulator products. They contain the DSA session control name and number of remote systems needed by applications making connects. They also define the transport station where the remote session control entity is to be found. If there are several paths to the remote session control entity several transport stations can be defined for backup or for load balancing.

Transport stations (TS) may be defined. They describe how to open transport connections to reach remote session control entities (DSA nodes).

Transport providers (TP) may be defined. They describe which transport stack to use on the local UNIX/Linux or Windows server system.

Symbolic ‘connection names’ (CONAME) may be defined. Clients then connect using these symbolic names instead of directly specifying a long list of connection parameters. Security filters (-FILTER) can be used to restrict access to these connection names.

A connection name may be defined so it picks up different connection parameters depending on which client is connecting to it, again by the use of security filters (PARAMS with -FILTER).

Connection names may also include access to pools of connection parameter lists, typically used to aid in generation of unique TP LIDs or GRTS IDs. These pools are also subject to security filtering (POOLS with -FILTER).

Access to undefined connection names may be configured to be either an error or to be ignored (COSTRICT).

Arbitrary options or groups of options may be restricted to clients passing a security filter (`-FILTER`).

The security filters referred to above are lists of IP addresses, IP network numbers, names of local hosts or names of applications. Each element in the list may either be permitted or denied access.

There is no limit on the number of objects that may be defined.

There is no limit on the size of any of these objects.

You configure these features by editing the `dsa.cfg` text file on the server. The file path is:

Windows Server	`\gar\config\dsa.cfg`
UNIX/Linux	`/usr/gar/config/dsa.cfg`

The formats of the directives in this file are described below. After making a change you must run the `glcc` configuration compiler program. This validates that the file is correct and copies/merges it into the file `dsa.dat`. It's the `dsa.dat` file that is actually used by the *Host Links* software at runtime. You must **not** edit this file directly. Always modify the `dsa.cfg` file and run `glcc`.

Note that editing the `dsa.cfg` file and running `glcc` while sessions are active is explicitly supported. The software has been carefully written to allow dynamic changes. You're even allowed to make changes in pool definitions where some entries are currently in use.

The Windows version of `glcc` has a `-s` option (silent mode). This option suppresses all message box displays, allowing for batch file installations.

Configuration file directives

In the syntax used below UPPERCASE is used for the fixed keywords and lowercase for the variable parameters. Brackets [...] are used around optional keywords. The vertical line | is used to separate alternatives. Symbolic names you define may be from 1 to 12 characters long and if they contain spaces the name must be enclosed in quotes. The order of the directives in the file is irrelevant except for the `PARAMS` and `POOL` cases mentioned below. Case is insignificant throughout the file. Space and tab characters can be placed anywhere between words and are ignored. Blank lines or lines starting with '*' or '#' are ignored. Comments can be put at the end of lines. The comment must be started with a # or * character.

SC - Session control

```
SC name [-ADDR x:x] [-SCID xxxx] [-DFLT] [-UNAME xxxx]
```

Defines a local DSA session control entity.

The name need not follow DSA name rules; it's only used as a lookup key for whatever is specified in the `gl_dsa -ln` option. In practice it will normally be the DSA session control entity name (node name). Normally you need only one of these directives.

`-ADDR` defaults to 1:2 and is optional. It's the DSA200 address (AREA:TSN) and need only be included if you will be using GL_DSA to make DSA200 connections. DSA200 is typically used for DPS7 native DSA access and for some DPS8 subsystems. The values are given in decimal.

`-SCID` defaults to be the same as the name when the name is 4 characters or less in length, otherwise `-SCID` is required. `-SCID` may in any case be specified to override the default. The SCID is used as the calling DSA session control name on outgoing connections.

`-DFLT` means this is the `SC` to use if a `gl_dsa -ln` option with no node name is used. Otherwise the first `SC` defined automatically becomes the default.

-UNAME is optional. It's the system's UNIX/Linux or Windows server communications node name. **NB!** it is not the DSA node name. This name can be displayed using the command uname –n on UNIX/Linux systems, and displayed in the Network section of the control panel on Windows server systems. When looking for an SC the software first looks for one with a matching UNAME, but if none is found it looks for one without a UNAME. It's only of use in setups where multiple UNIX/Linux and/or Windows server systems share a single dsa.cfg file on a file server.

```
sc is2c
sc is2c -addr 54:97
sc testsc -scid is2c
```

RSC - Remote Session Control

```
RSC name [-ADDR x:x] [-SCID xxxx] [-DFLT] [-II nn]
[-TS tstation [tstation] [tstation]]
```

This defines a remote session control entity (DSA node).

The name need not follow DSA name rules, it's only used as a lookup key for whatever is specified in the gl_dsa -dn parameter, in the node field in *Glink* or *Host Links* emulators $*$CN commands. In practice it will normally be the DSA session control entity name (node name). You need one of these directives for each remote node to which you connect or from which you will receive connects.

-ADDR defaults to 1:1 and is optional. This is the DSA200 address and need only be included if you will be using GL_DSA to make DSA200 connections to this node. DSA200 is typically used for DPS7 native DSA access and for some DPS8 subsystems. The values are given in decimal.

-SCID defaults to be the same as the name when the name is 4 characters or less in length, otherwise -SCID is required. -SCID may in any case be specified to override the default. This is the DSA session control name of the remote system.

-II nn sets an inactivity timer in minutes. Should the session be inactive for more than the time set here then the session is automatically disconnected. It defaults to 15 minutes if -ii is specified but nn is not given.

-DFLT means this is the RSC to use if no line handler -dn parameter, $*$CN nodename or Glink Host node is used when making a connection.

-TS points to a TS (Transport Station) directive somewhere else in the file. If you omit the option the default TS is used.

```
rsc is2b
rsc en3e -addr 54:62
rsc is2b -ii 15 -ts ph8a_rfc
```

To set up backup routes you just specify multiple -ts routes:

```
rsc ph8a -ts ph8a_rfc ph8a_bak
```

These are then tried in order until a connection succeeds or the end of the list is reached.

Alternatively to configure load balancing you use:

```
rsc -ts route1 route2 route3 -lb 60 20 20
```

In this example route1 gets 60% of the load, route2 20% and route3 20%. If the first priority route has a connect failure the second priority is tried etc. until a connection succeeds or the last priority is reached. A total of 8 routes can be specified.

The software remembers when a transport disconnection or time-out happens while trying to connect, and the node is suspended to avoid repeated connection attempts using a route that is down. The suspension time may be configured with the -si parameter to the ts directive.

If all routes to a given remote session control ID are marked as down then a connection is tried using the first priority route. If the connection succeeds that route is marked as being up again.

TS - Transport Station

```
TS name  -CLASS x -NS 141.112.xx.yy [-TPDU x] [-SI nn] [-DFLT]
[-TP tprovider]
```

This describes how to open a transport connection to reach remote session control entities (DSA nodes). You need one for each remote transport station to which you will be connecting. A transport station is the transport service access point, which resides in GCOS7 or GCOS8.

-CLASS is required and must be set to 0 for RFC1006 connections.

-NS is also required and is the real network address. It must be set to the IP name or address for RFC1006 connections.

-TPDU defaults to whatever the local transport provider does and is optional. It’s the maximum transport PDU buffer size. The actual size will be negotiated.

-SI is optional and specifies how many minutes to suspend after a connect failure; configuring 0 stops it from being suspended. The default is 15 minutes.

-DFLT mans that this is the default transport station. Otherwise the first TS automatically becomes the default.

-TP points to a TP (transport provider) directive somewhere else in the file. If you omit the option the default TP is used.

```
ts ph8a_rfc -class 0 -ns 141.112.xx.yy –tp rfc

tp rfc –who gar
```

TP - Transport Provider

```
TP name  [-WHO GAR] [-DFLT] [-UNAME xxxx]
```

This defines a local transport provider. We advise that it is configured with a minimum of the -WHO parameter to define the provider.

-WHO explicitly defines the provider.

For the G&R/RFC1006 transport (all platforms):

Specify the value ‘rfc’ in order to signal the use of RFC1006 over TCP transport. Please note that in order to be able to accept incoming connections when using RFC1006 you must start the DSA listener, even if only one program is accepting connections, and it must be started from root, since only root can access port 102, which is reserved for RFC1006.

-DFLT means that this is the default transport provider. Otherwise the first TP defined automatically becomes the default.

-UNAME has the same use here as in SC.

CONAME (RESOURCE) - Connection Name

```
CONAME name [-BACKUP coname]
            [-DESC desc]
            [-FILTER filter]
            [-PARAMS params]
            [-POOL pool]
            [-TYPE AUTO | MENU | TM |TERM | PRINT]

[Gline parameter list]
[Gline parameter list]
[-CODE nnnn -TEXT text]
[-CODE nnnn -TEXT text]
```

Defines a connection name e.g.:

```
coname tssprod -desc “TSS production” -filter block_remote
-scid is40 -dmb tss      * Gline parameter list for all users
```

CONAME and RESOURCE are synonymous. The parameters if used must be on the same line.

`-BACKUP` designates another CONAME that will be tried if any problem arises on the connection. Up to eight CONAMES can be chained in this way (7 backups).

`-DESC` adds a description of the `CONAME` (30 characters). It’s currently unused except for documentation purposes.

`-FILTER` blocks access to this `CONAME` unless permitted by the filter.

`-PARAMS` points to `PARAM` directives that add to the Gline parameter list.

`-POOL` points to `POOL` directives that add to the Gline parameter list.

`-TYPE` is an optional identifier:

`TERM` (terminal) or `PRINT` (printer) supply the information to clients that request a `CONAME` list, for display as appropriate.

`AUTO`/`MENU`/`TM` are used to specify TNVIP dialog style for TNVIP clients. Auto gives no local dialog. Menu displays the Gline parameters in a formatted screen. TM gives a ‘Terminal Manager’ style, as specified by `-CODE` and `-TEXT` parameters. `MENU` is default for TNVIP clients.

Gline parameter list for all users:

is one or more lines of Gline parameters that will be used as a start point for building the final connection parameter list.

When compiling dsa.cfg all Gline parameters for a given CONAME are assembled in a single line. That includes the –CODE and –TEXT parameters below. The line is limited to 512 characters.

-CODE NNNN - forces ‘Terminal Manager’ dialog. The following codes may be configured to display information or prompt to ask for user logon parameters:

0000 displays banner information
1000 displays logon information with parameters
1300 generates a -DB parameter (Billing)
1400 generates a -DU parameter (User ID)
1500 generates a -DP parameter (Project)
1600 generates a -PW parameter (Password)
1800 generates a -DN parameter (SCID)
4200 generates a –TM parameter (Model)
4700 generates a -CO parameter (CONAME)
5800 displays a ‘logged in’ message
7900 generates a –DA parameter (remote mailbox)
8100 generates a –DX parameter (remote mailbox extension)
8800 generates a -CO parameter (CONAME)
8900 generates a local dialog command prompt ($*$)

Ggate submits the parameters to *Gline* as they are collected. If an error is detected the prompt is repeated after an error message.

-TEXT is used in conjunction with -CODE. It overrides the standard prompt in the local dialog e.g.:

```
RESOURCE tmsample –TYPE TM
    -DN BE01 –DA IOF
    -CODE 0000
    -CODE 1000 -TEXT "ID:???? SC:???? MD:????"
    -CODE 1400 –TEXT Utilisateur
    -CODE 1600 –TEXT '"Mot de passé"'
    -CODE 1500
```

Would result in the following Terminal manager style dialog:

```
$$ 0000 Banner information (G&R style)
$$ 1000 ID:PHX SC:PH8A MD:DKU7211
$$ 1400 Utilisateur:
$$ 1600 Mot de passe:
$$ 1500 PROJECT:
```

The default texts for the codes are:

```
0000 G&R Host Links ***Ggate*** version
1000 YOUR ID IS ???????? SC: ???? MODEL: ?????? MB: ????
1300 BILLING:
1400 USER ID:
1500 PROJECT:
1600 PASSWORD:
1800 SC:
4200 MODEL:
4700 CONNECT TO:
5800 LOGIN SUCCESSFUL
7900 MB:
8800 CONNECT TO:
8900 $*$
```

The parameters available to the 1000 text are the CONAME, the SCID of the Ggate system, the terminal type and the local mailbox. They can only be used in that order and a group of 4 question marks is needed to indicate the positions where they are to be inserted.

The standard texts can be modified using the `gcptexts` utility. See the ***Host Links Installation*** manual for details.

Example:

```
RESOURCE TNG -desc "USERS type RH" -type TM
-code 0000 –text "*DNS-E V3U4.A00 P1.002 P2.006 P3 etc."
-code 1000 -text "VOTRE ID EST ???? SC:???? MODELE:???? MB:????"
-code 1600 -text "MOT DE PASSE:"
-code 5800 -text "VOUS ETES ENREGISTRE"
-code 4700 –text "CORRESPONDANT: "
-code 8800 –text "CORRESPONDANT: "
```

If the `4700` directive is not the last, and is answered with a null, then the next directive will be used immediately. In this case it is an `8800`. Because the `8800` directive is the last, it will be repeated three times. On the third null response we go to the command prompt (`$*$`).

If a 4700 or 8800 is followed by more directives:

```
  -code 8800 –text "CORRESPONDANT: "
  -code 1800 –text "SCID: "
  -code 7900 –text "MBX: "
```

Then a single null response to the 4700 or 8800 goes to the next directive. These will normally be directives that collect line parameters. Ggate submits the parameters to *Gline* as they are collected. If an error is detected the prompt is repeated after an error message. After three consecutive errors on any directive we go to the command prompt. For directives 1800 (SCID) or 7900 (mailbox) a null response is considered an error. For all other line parameter prompts a null response is accepted and skips to the next directive.

Gline parameters

Gline parameters can be added to the CONAME directly, or via PARAMS or POOL directives. By default Gline parameters can be overridden by the client, but this can be inhibited using the '+' character as a prefix to the parameter in the directives in dsa.cfg. The client can be requested to ask for a parameter interactively using the '*' character for the parameter. If the '*' character is used as a prefix to a value, the value will be suggested as a default when the client asks for the parameter.

Glink uses a Ggate-specific configuration dialog box to display parameters. TNVIP clients using the default MENU style are sent a formatted screen, where parameters that can be changed are in variable fields, but others are protected.

The set of line module parameters used to finally connect to the host is:

- The list from the CONAME directive
- If -PARAMS is specified this adds the options from **all** PARAMS directives with the specified name who's filters, if any, allows the client to access them. The order in which the options are added is that in which they are found in the dsa.cfg file
- If -POOL is specified it selects the **first** POOL directive in the dsa.cfg file with the specified name and a filter, if present, that allows the client to access it. The parameters from a random, as yet unused, line of parameters in the POOL directive are added
- Finally this set of parameters is overridden by any parameters from the user

  For Glink clients: from the Glink/Ggate configuration dialog box

For Host Links clients from the `-LI DSA` command line

For TNVIP clients interactively:
from Terminal Manager style dialog
from the menu

Note that the parameters from the user will not override any which have been inhibited using the ‘+’ prefix. Note also that any `RESTRICT` that applies to the client will be applied to parameters supplied by the client, and may cause rejection of the connect attempt.

PARAMS - Parameters

```
PARAMS name [-FILTER filter]
Gline parameter list
```

This defines connection parameters. There may be multiple `PARAMS` directives with the same name, and all will be examined as candidates for addition to the parameter list for the client. If a filter is specified then only clients that are permitted by it get the parameter list added. Several CONAME directives can point to the same `PARAMS` directives.

Example, two users pick up different parameters from two `PARAMS` directives:

```
coname tssprod -desc “TSS production system” -params tssprod
   -scid is40 -dmb tss      * Gline parameter list for all users

params tssprod -filter john
   -usr john -pw calvin     * parameters for John only
params tssprod -filter sue
   -usr sue -pw hobbes      * parameters for Sue only
```

POOL - Parameter Pool

```
POOL name [-FILTER filter]
Gline parameter list
[Gline parameter list]
```

This defines a pool of connection parameters. There may be multiple `POOL` directives with the same name. They will be examined in the order in which they are found in `dsa.cfg`. If a filter is specified then only clients permitted by it can access the pool. The **first** `POOL` with the correct name and permission is selected. Several `CONAME` directives can point to the same `POOL` and the pool is then shared between clients connecting to the different `CONAME` directives. Each `Gline parameter list` line in the pool is independent of the others. They are allocated in round-robin order and the software is careful never to allocate the same line again as long as it is in use by a client. Typically the parameter lines will define unique LIDs to be shared by a group of clients.

Example: three pools of mailbox extensions for use as TP8 LIDs, one for users in Oslo, the second for users in Bergen and the third for users in Lillehammer:

```
coname tp8prod -desc “TP8 production system” -pool tp8prod
   -scid is40 -dmb prodv2      * Gline parameter list for all users

pool tp8prod -filter oslo
   -dx os01      * will be picked up by one Oslo user
   -dx os02      * will be picked up by one Oslo user
   -dx os03      * will be picked up by one Oslo user
   -dx os04      * will be picked up by one Oslo user

pool tp8prod –filter Bergen
   -dx be01      * will be picked up by one Bergen user
   -dx be02      * will be picked up by one Bergen user
   -dx be03      * will be picked up by one Bergen user
   -dx be04      * will be picked up by one Bergen user

pool tp8prod -filter lillehammer
   -dx li01      * will be picked up by one Lillehammer user


filter oslo
permit tcp 192.150.211.0/24      * Class C network
filter bergen
permit tcp 192.150.212.[00-10]   * macro for group
filter lillehammer
permit tcp 192.150.213.4         * single user
```

Macro facility

A simple macro facility eases configuration of parameters for large numbers of sessions. It expands numeric ranges only. Multiple macros on the same line must have the same range. You may use up to 8 macros on each line. e.g.

```
-mn mbx[01-03] -dx 00[12-14] -ur userrec[9-11]
```

expands to:

```
-mn mbx01 -dx 0012 -ur userrec9
-mn mbx02 -dx 0013 -ur userrec10
-mn mbx03 -dx 0014 -ur userrec11
```

The expansion will use the same number of digits if a numeric is written with leading zeroes i.e. `9-11` could have been written `09-11` to make the first expansion `userrec09`.

```
filter bergen
permit tcp 192.150.212.[00-03]   * macro for group
```

expands to:

```
filter bergen
permit tcp 192.150.212.00
permit tcp 192.150.212.01
permit tcp 192.150.212.02
permit tcp 192.150.212.03
```

COSTRICT - Connection Name Strictness

```
COSTRICT [-FILTER filter]
```

`COSTRICT` defines strict handling of missing connection names. If a client connects and specifies an undefined connection name then the default is to simply ignore this and allow the client to continue and specify other parameters. If you specify `COSTRICT` then the default instead gives an error message. If you specify a filter then an error message is given only to those that are permitted by it.

RESTRICT - Parameter Restrictions

```
RESTRICT [-FILTER filter]
Gline parameter list
[Gline parameter list]
```

RESTRICT defines Gline parameters that should be restricted from general use. If a filter is specified then only those that are permitted by it are subject to the restrictions. While this may seem backwards at first glance it was done this way so that all the directives can be read the same way, namely that the directive is visible only if the filter specified returns the client as permitted. You can specify an unlimited number of RESTRICT parameters and they will all be processed. Each `Gline parameter list` line within a RESTRICT is independent of the others. All Gline parameters in a `Gline parameter list` line need to be matched by options given by the client for the restriction to take effect. An asterisk (*) can be specified as a wild card to match any parameter value. Restrictions are only checked against parameters specified directly by clients, not against parameters specified in other parts of the `dsa.cfg` file.

Examples:

```
* Absolutely nobody is allowed to use his/her own SCID.
restrict
-scid *

* Those permitted through the remote_users filter are not allowed
* to use any pair of -scid or -dmb options, nor allowed to
* use the -ln option.
restrict -filter remote_users
-scid * -dmb *
-ln *

filter remote_users
* note that this filter allows all IP addresses
* so the restrict above applies to all remote users
permit TCP *
```

FILTER - Access Filter

```
FILTER name
PERMIT|DENY TCP|LOCAL|APPL|USER|NODE|SSL *|expression
...
```

FILTER defines a filter for security screening. A filter is a list of one or more lines, each starting with PERMIT or DENY. Each line is checked in sequence until a match is found, then the permit/deny status of that line is returned. There's an implicit line to deny everything at the end of the definition. Local clients running *Qsim*, *V78sim* etc. will match filter lines with LOCAL specified. Local clients are the *Host Links* products on the system that owns the /usr/gar (UNIX/Linux) or \gar (Windows) directory, or on other systems that mount it over the network. Clients coming in through *Ggate* will match filter lines with TCP specified. Any client will match filter lines with APPL specified. Clients using SSL will match filter lines with SSL specified.

TCP expression:

The expression is IPaddress[/bits]. The /bits part specifies how many bits of the IP-address will be used during matching. The default is 32, i.e. use the entire IP-address. If you want to permit/deny the whole of a class C network specify 24 bits. For the whole of a class B network specify 16 bits. For the whole of a class A network specify 8 bits. The asterisk (*) matches anything. Only clients coming into *Ggate* will match TCP filters.

LOCAL expression:

The expression is the system's UNIX/Linux or Windows communications node name. **NB!** it is not the DSA node name. This name can be displayed using the command uname –n on UNIX/Linux systems, and displayed in the Network section of the control panel on Windows server systems. Any application running on a system with this communications node name will match the expression. An asterisk (*) matches any communications node name. A specific expression is only needed when you filter applications running on one of several systems sharing a single dsa.cfg on a file server.

APPL expression:

The expression is one of Glink, Qsim, V78sim, Pthru, G3270, G5250, Padcon, Gmailer, Gspool, GUFT, GUFTSRV or Unknown.

```
filter block_remote
   permit local *              # implicitly denies remote
filter allow_ip
   deny tcp 192.150.211.4      # first deny one terminal
   permit tcp 192.150.211.0/24  # allow rest of CLASS C network
   permit tcp 18.0.0.0/8       # allow a class A network
   permit tcp 128.38.0.0/16    # allow a class B network
filter allow_qsim
   permit appl qsim            # applies to anyone using Qsim
```

USER expression:

The expression is the user name that is supplied by the client in the ‘logon packet’ (in the -us parameter). The user name is normally the user account name in the local client network. The asterisk (*) matches anything. Glink clients need version 6.1.4 or later. For Host Link applications, release 5.3 or later is required.

NODE expression:

The expression is the unique workstation (‘node’) name supplied by the client in the ‘logon packet’ (in the -st parameter). This filter type is suitable in configuration where the TCP expression cannot be used to identify a particular workstation i.e. if DHCP is used or in some firewall configurations (the IP address is generated dynamically). The workstation name is case sensitive. The asterisk (*) matches anything. Glink clients need version 6.1.4 or later. For Host Link applications, release 5.3 or later is required.

SSL expression

The expression is a string, using quotes if there are embedded spaces. The SSL certificate of SSL clients is searched for the string. If the string is found anywhere in the certificate the filter will apply.

Use of filters to restrict access

In this example a coname is defined that can only be used by SSL clients using the G&R Glink demo certificate:

```
CN=Glink demo/C=NO/ST=Oslo/O=Gallagher & Robertson as
/OU=DEMO/emailAddress=glink@gar.no
```

The filter is set up to test for the Common Name in the certificate.

```
coname secure -desc "only for the Glink demo" -filter sslflt
   -dn i163 -da reqfld -du oddvar -s_ -d_

 filter sslflt
    permit ssl "Glink demo"
```

In this example the class C IP networks 192.150.211.0, 193.71.16.0, 193.71.17.0, 193.71.18.0 and 193.71.19.0 should have access to both CONAMEs. All other IP addresses should only have access to the tp8test CONAME. Remember there is an implicit ‘deny’ of everything at the end of a filter.

```
# define a CONAME for a TP8 test system ‘gartest’ with no filters
# the coname is then available to anyone

coname tp8test -desc “TP8 Testapplication gartest” -pool tp8test
   -hm CXI –tm TXT 7801-da gartest -dn ph14

# define a coname for a mailbox ‘2wa-rno’ with a filter allowing only
# the desired IP addresses

coname 2warno -desc “TP8 Testapplication 2wa rec no” -filter external
   -hm CXI -tm TXT7801 -da 2wa-rno -dx jimp -dn ph14

filter external
permit local *                     # allow Qsim, V78sim etc.
permit tcp 192.150.211.0/24        # allow class C network (use 24 bits)
permit tcp 193.71.16.0/22          # allow 16, 17, 18 and 19 (use 22 bits)
```

In this example all IP addresses except the ranges above should be prevented from specifying any node name or application name parameters directly. This is typically used to force access to happen through a `CONAME`.

```
# define a restrict parameter with a filter that allows through all
# TCP addresses which are to be restricted. The TCP addresses that are
# permitted by the filter meet the restriction

restrict -filter internal   # addresses passing the filter cannot choose
   -dn *                # the default node to which they connect
   -da *                # the application mailbox to which they connect
   -ln *                # or to listen for incoming connects

filter internal
deny tcp 192.150.211.0/24   # they are not permitted by the filter
deny tcp 193.71.16.0/22     # and so DO NOT meet the restriction
permit tcp *                # rest are permitted, and meet the restriction
```

Sample dsa.cfg for RFC1006

```
# Your local DSA RFC1006 node name is 'is2c'.
# You want to connect with DSA RFC1006 node name 'ph21'
# that has IP address 1.2.3.4 and 'ka01' that has 5.6.7.8

# is2c needs a dsa200 address when connecting to GCOS7

sc is2c -addr 05:44

# Remote node GCOS8 ph21

rsc ph21  -ts ph21_rfc
ts  ph21_rfc -class 0   -ns 1.2.3.4  -tp rfc

# Remote node GCOS7 ka01 needs a dsa 200 address

rsc ka01 -addr 176:072  -ts ka01_rfc
ts  ka01_rfc -class 0   -ns 5.6.7.8  -tp rfc

# Use Gallagher & Robertson RFC1006 stack

tp  rfc      -who gar

# Coname configuration example

# In Glink you select a coname by specifying it in the Ggate
# configuration dialog box.
# In Host Links a coname is selected with the '-co'
# parameter.
#
# You want to use a coname 'tp8ws2' to access TP8.
# It references a pool of mailbox extensions so it
# can be shared by a group of users.

coname tp8ws2 -desc "TP8 workstation 2" -pool lidpool
-dn ph21 -da tpws2mbx -hm dps8

 pool lidpool
 -dx mg01
 -dx mg02
 -dx mg03
 -dx mg04
```

```
# A convenient shorthand for configuring large pools:

 pool lidpool
 -dx mg[01-04]

# conames are also used for configuration of merged print,
# where data from a separate print session is merged into
#the terminal session, marked as print data.

# Merged print from TWriter on GCOS7.
# Use -pco to get point to printer coname a2_print
# The printer coname must wait for a connect from Twriter to
# the printer mailbox mypctw.
#
coname mytds -desc "TDS on GCOS7"
-dn ka01 -da tdsmbx -hm dps7 -pco a2_print

coname a2_print -desc "Twriter print to MYPCTW"
-ln -mn mypctw -tm a2

# end of dsa.cfg
```

Appendix: Host Links Manuals

Below you find a complete list of all available Host Links manuals:

Installation	
Host Links Servers	Installation and Configuration on UNIX/Linux
Host Links Emulators	Installation and Configuration on UNIX/Linux
Host Links	Installation and Configuration on Windows
Line handling	
Gline	Line Handler and DSA/OSI Configuration
Ggate	Transparent Gateway
Gproxy	Network Manager & SNMP Proxy Agent
G&R SSL	Using SSL for security in G&R products
GlAPI	Application Programming Interfaces
Gsftp	Gateway between FTP and SFTP
Emulations	
Gspool	Network Printer Emulation
GUFT	Unified File Transfer
G3270	Emulating IBM 3270 Terminals
G5250	Emulating IBM 5250 Terminals
Pthru	Gateway to the Bull Primary Network
Qsim	Emulating Questar DKU7107-7211 & VIP7700-7760
V78sim	Emulating VIP7801 & VIP7814
Gweb	Web Browser Front-end for DKU, VIP7700-7760, VIP7800, IBM3270 and IBM5250 Emulations

Appendix: DSA test utilities

The Gline package includes a set of Gline communication utilities. These are used when testing and debugging connection problems. The utilities are delivered as part of the Gline package and can be used without any additional configuration. The nodes to be tested must of course be configured in the `dsa.cfg` file.

Gconame

Lists the parameters generated from a given CONAME. The utility works for both CONAME and RESOURCE e.g.:

```
gconame tnviptm

Checking 'dsa.cfg' for coname 'tnviptm'
Coname: tnviptm, type TM, parameters:
-DA misfld
-S_
-D_
-CODE 0000
-CODE 1000
-CODE 1800
-TEXT Remote SCID?:
-CODE 4700
-TEXT Remote application?:
-CODE 1400
-CODE 1600
-TEXT Password?:
```

Gerror

Shows the text message associated with a DSA reason code. Only the most common codes are supported i.e. the ones related to network, transport and session communication layers. e.g.:

```
gerror 0109
Reporting component: Session control (01) 0109, Dialog
protocol error or negotiation failed (wrong logical record).
```

For a detailed description of all reason codes, please consult the Bull manual ***OSI/DSA Network System Messages and Return codes*** (39A2 26DM).

Glnode

List and verify the communications parameters of the local node e.g.:

```
glnode
Local node name          : IS2C
Local session control id : IS2C
DSA200 address (area:tsm): 5:44 (5:2c)
```

Gping

Connect to a remote system using the Gline parameters set on the command line. If used on a Windows platform the command prompt must be started as an administrator. If successful it returns ‘connected to application’, otherwise it shows the error code returned e.g.:

```
gping -li dsa -dn b7dl -da iof -du jim -pw mydogsname
Gping - $$DSA: Connected to application
```

Grnode

Return the parameters (in `dsa.cfg`) and the state of a remote node e.g.:

```
grnode ph8a
Checking 'dsa.cfg' for node 'ph8a'
  Session control id            : PH8A
  DSA200 address (area:tsm)     : 1:1 (1:1)
  Inactivity interval           : 0
Route 0
  Load balance percentage       : 0
  TP class                      : 0
  TP expedited                  : 0
  TPDU size                     : 0
  Network address               : 141.112.xx.yy
```

Gtrace

Same as `gping` but writes the DSA communication trace on the user’s terminal (applicable to UNIX versions) e.g.:

```
gtrace -li dsa -dn ln40 -da snm151
D6:Application event @ 14:17:17.6003. tokenitem = 00
D6:Application event @ 14:17:17.6082. tokenitem = 00
D6:Connect request called, node = LN40
D6:OurBufferSizes. ApplMaxXmit = 511, ApplMaxRecv = 500
Rec:4000 0002 s:2
Rec:506B 0010 s:16
etc etc
Gtrace - line trace ending.
Gtrace - $$DSA: Connected to application.
```

Gtsupd

Update the state of a transport route. Transport routes can be set automatically in a disabled state if a backup route is configured. When such a state change occurs the route will be set back to the enabled state after a configurable timer has expired. The default is 15 minutes. You can reset the state of such a route with `gtsupd ts-name enbl/used/down/locked` e.g.:

```
gtsupd gars_rfc enbl
TS-entry ‘gars_rfc’, new state = enbl
```

Appendix: Error Codes

OSI/DSA error codes

Below is a list of OSI/DSA error codes and the corresponding description. These are the same descriptions that the `G&R/Gerror` utility will display when given the DSA code as a parameter.

code	Description
00xx	**General Errors**
0001	Open Failure in LC - Reject for unknown reason
0002	Open Failure in LC - Acceptor customer node inoperable
0003	Open Failure in LC - Acceptor customer node saturated.
0004	Open Failure in LC - Acceptor mailbox unknown.
0005	Open Failure in LC - Acceptor mailbox inoperable.
0006	Open Failure in LC - Acceptor mailbox saturated.
0007	Open Failure in LC - Acceptor application program saturated
0008	Connection refused. Transport protocol error or negotiation failed.
0009	Open Failure in LC - Dialog protocol error or negotiation failed
000A	Open Failure in LC - Presentation protocol error or negotiation failed
000B	Open Failure in LC / Connection refused lack of system resources
000C	Open Failure in LC / Connection refused from GCOS7 duplicate user
000D	Open Failure in LC, Duplicate implicit LID / Q class not started
000E	Open Failure in LC, Duplicate GRTS Id / lack of memory resources
000F	Open Failure in LC, No Logical line declared for DACQ / 7 connection refused
0010	Open Failure in LC, GCOS 8 GW Missing translation / Incorrect device length in ILCRL.
0011	Open Failure in LC, DAC connection not initialized / Too many jobs executing
0012	Open Failure in LC, No binary transfer / impossible to start the IOF job
0013	Open Failure in LC, connection is not negotiated in FD mode / impossible to start the IOF job

0014	Disconnection - Timeout resulting from absence of traffic.
0016	Option missing for an RBF mailbox.
0017	Connection refused - Incorrect access right for MB.
0018	Connection refused - Incorrect access rights for the application.
0019	Connection refused - Unknown pre-negotiated message path
001A	Connection refused - Security validation failed.
001B	Connection refused - Unknown acceptor mailbox extension.
001C	Connection refused - Inoperable acceptor mailbox extension.
001D	Connection refused - Invalid Message group number.
001F	Disconnection - no more memory space.
0020	Connection refused - Unknown node.
0021	Connection refused - inaccessible node or Host down.
0022	Connection refused - saturated site.
0023	Connection refused - inoperable mailbox.
0024	(X.25) Packet too long. Problem with packet size. / Connection block already used.
0030	Syntax Error - option not known (received on close VC).
0031	(X.25) No response to call request packet - timer expired.
0033	(X.25) Timer expired for reset or clear indication.
0039	Disconnection - transport protocol error (MUX).
003C	Presentation Control Protocol Error
003E	The application has not the turn
003F	Message group closed
0040	(X.25) Facility code not allowed. / Connection refused - unknown node
0041	Connection refused - path not available.
0042	Connection refused - Duplicate USER ID / Facility parameter not allowed
0044	(X.25) Invalid calling address.
0045	(X.25) Invalid facility length.
0047	(X.25) No logical channel available.
004F	DNSC: (X.25) Invalid call packet length.
0050	Normal disconnection (GCOS3/8)
0051	Error or Event on LC initiated by GW
0052	Error or Event on LC initiated by GW.
0053	Error or Event on LC initiated by GW. TCall
0054	Error or Event on LC initiated by GW. DIA in LOCK State
0055	Error or Event on LC initiated by GW. DIA error
0056	Error or Event on LC initiated by GW. GW has no known explanation.
0057	Error or Event on LC initiated by GW. Reject mailbox permanent

0058	Error or Event on LC initiated by GW. No more input lines in DACQ
0059	Time-out on GCOS 3/8 gateway.
005A	Error or Event on LC initiated by GW. Disconnect from terminal without reason
005B	Error or Event on LC initiated by GW. Wrong letter or wrong record
005C	Error or Event on LC initiated by GW. Forbidden letter received
005D	Error or Event on LC initiated by GW. Forbidden letter received
005E	Error or Event on LC initiated by GW. No buffer for secondary letter
005F	Error or Event on LC initiated by GW. No buffer for fragmented letter
0060	Error or Event on LC initiated by GW. Disconnect on end of phase record
0061	Error or event on LC initiated by GW. No buffer for control letter.
0062	Error or event on LC initiated by GW. Mailbox in closing phase
0064	Error or event on LC initiated by GW. Flow control error.
0065	Error or event on LC initiated by GW. CH locked by operator.
0066	Error or event on LC initiated by GW. Disconnect with a normal TMG F2 exchange.
0067	Error or event on LC initiated by GW. Teletel rerouting error from DACQ
0068	Error or event on LC initiated by GW. Teletel routing error from DACQ
0069	Error or event on LC initiated by GW. Teletel rerouting error from TM
006A	Error or event on LC initiated by GW. Teletel rerouting error from TM
006B	Syntax error - text too long.
006C	Syntax error - illegal object in a GA command.
006D	Syntax error - unknown node Id.
0078	Syntax error - illegal command for this object.
0079	Syntax error - illegal date.
007F	(X.25) No route available for X.25 switching.
0081	No more network routes available for switching.
0082	(X.25) Hop count reached for X.25 switching.
0083	(X.25) Flow control negotiation error.
0085	(X.25) Frame level disconnection.
0086	(X.25) Frame level connection.
0087	(X.25) Frame level reset.
0090	Frame level not set.
0092	(X.25) X.25 Echo service in use.
0093	(X.25) Incorrect password for PAD connection.

0094	(X.25) No more PAD connections allowed.
0096	(X.25) TS SX25 or NU X25 objects locked.
009C	(X.25) Invalid packet header. X.25 protocol error.
009D	(X.25) Incompatible header. X.25 protocol error.
009E	(X.25) Logical Channel Number too high.
009F	(X.25) Incorrect packet type.
00B2	Use of invalid password through PAD
00B6	Unknown mailbox selection for PAD connection using the PAD password.
00C0	(X.25) Normal disconnection.
00D7	(X.25) TS image (of type DSA or DIWS) in LOCK state.
00DE	(X.25) NS RMT or NR SW in LOCK state.
00E1	Connection refused. Mailbox is not in ENBL state.
00E6	QOS not available permanently.
01xx	**Session Control**
0100	Logical connection accepted or normal termination
0101	Rejection for unknown reason or abnormal termination
0102	Acceptor node inoperable.
0103	Acceptor node saturated. When a node has no available resources
0104	Acceptor mailbox unknown.
0105	Acceptor mailbox inoperable.
0106	DNS: Acceptor mailbox saturated.
0107	DNS: Acceptor application program saturated.
0108	Transport protocol error or negotiation failed (DSA 200 only).
0109	Dialog protocol error or negotiation failed. (Wrong logical record).
010A	Time-out on session initiation / unknown LID
010B	Acceptor mailbox extension unknown.
010C	Acceptor mailbox extension inoperable.
010D	Invalid Session Number.
010E	Unknown node.
010F	System error. System generation error or insufficient memory space
0110	Application abnormal termination. Subsequent to an abnormal occurrence in the dialogue
0111	Normal terminate rejected.
0112	Protocol not supported.
0113	Session control service purged by user.
0115	Disconnection Time-out on message group initiation.
0117	Incorrect Access Right for MB
0118	Incorrect Access Right for the Application
0119	Pre-negotiated Message Path Descriptor unknown
011A	Security validation failed
011E	Incorrect object status

011F	Not enough memory space available.
0120	Node unknown.
0121	The channel object (CH) is in LOCK state
0122	Saturation - no plug available
0123	Object status = LOCK
0124	Connection block (TSCNX) already used
0125	Disconnection already running
0126	The connection block (TSCNX) is disconnected (or not connected)
0127	Change Credit value < 0
0128	Ineffective Change Credit (delta = 0)
0129	No more deferred letters
012B	"Reinitialization" Request
012C	"Reinitialization" in progress
012D	"Reinitialization" in progress, letters are dropped
012E	Close virtual circuit. Either no mapping exists between PA/NR or CL and VC/NS
012F	Null connection object index.
0130	Undefined function at Sysgen time.
0131	Letter too large with respect to the negotiated size.
0132	The received letter is longer than the size which was
0133	Disconnection of the session control user
0134	Interface error on EOR (End-Of-Record) processing.
013C	Presentation control protocol error.
013E	You do not have the turn.
013F	Message group closed.
0140	Session is closed.
0151	Request refused, no system buffers available.
0152	Incorrect addressing record.
0153	No presentation record in the ILCAL or ILCRL
0154	Negotiation failed on session mode
0156	Negotiation failed on resynchronization.
0157	Negotiation failed on END to END ACK
0158	No presentation record in the connection letter
0159	Negotiation failed on session mode
015A	Negotiation failed on letter size (in the Logical Connection record).
015B	Negotiation failed on resynchronization (in the Logical Connection record).
015C	Negotiation failed on end-to-end ACK (Logical Connection record).
015D	No support of the "letter" interface because Multirecord is not negotiated.
0160	Incorrect TSPACNX table.
0161	Protocol error on letter reception.

0162	Negotiation failure.
0163	Record header length error.
0164	Protocol error.
0165	Protocol error reception of control letter.
0166	Type or length error on interrupt letter.
0167	Protocol error on reception of data letter.
0168	Dialog protocol error.
0169	Unknown event.
016A	Protocol error on data transfer.
016B	Invalid status for a disconnection request.
016C	Invalid status for a recover
016D	Invalid status for a suspend/resume request.
016E	Negotiation failure.
016F	Unknown command.
0170	Error in presentation protocol
0171	Letter header length error in
0172	ILCAL is not DSA 200 protocol.
0173	Error in session record.
0174	Normal disconnection, without complementary reason code.
0175	Letter is not in ASCII or EBCD.
0176	Connection protocol letter header
0177	Letter header protocol error.
0178	Record header protocol error.
0179	Record header length error.
017A	Mbx record header length error.
017B	Error on buffer transfer.
017C	DSA 200 record header protocol
017D	DSA 300 record header protocol
017E	Unsupported connection options.
017F	Character error in ASCII string.
0180	No segmented record size.
0181	Invalid mailbox object index.
0182	Mapping error for a remote connection.
0190	No more buffers.
0191	Byte count is greater than GP.
0192	Byte count is greater than GP.
0193	Byte count is greater than GP.
0194	Byte count is greater than GP.
0195	Byte count is greater than GP.
0196	Byte count is greater than GP.
0197	Byte count is greater than GP.
0198	No more buffers.

0199	Byte count is greater than GP.
019A	Byte count is greater than GP.
019B	Byte count is greater than GP.
019C	Byte count is greater than GP.
019D	Byte count is greater than GP.
019E	Byte count is greater than GP.
019F	Byte count is greater than GP.
01A0	Invalid transfer state.
01A1	Suspend protocol running.
01A2	Suspend protocol running.
01A3	Recover protocol running.
01A4	Forbidden function in write request. ($WRITE)
01A5	Conflicting parameters for segmented record. (SWBREC)
01A6	Protocol conflict - suspend/recover.
01A7	Protocol not supported - letter/end-to-end ACK. (SWBLET)
01A8	Multi-record letter in progress.
01A9	Interrupt request forbidden.
01AA	Send control record request forbidden. (SCTROL)
01AB	Forbidden for TWA session - turn is here. (SREAD)
01AC	Termination forbidden - suspend or recover in progress. (STERM)
01C0	No space available for downstream connection request. (SMECNX)
01C1	No space available for upstream connection request. (SMUCNX)
01C2	No space available for upstream SCF connection. (SMRCNX)
01C3	No space available for session context. ($SCTX)
01E0	Enclosure or data length error for a write request. ($WRITE)
01E1	Enclosure or data length error for a write segment record request. (SWBREC)
01E2	Enclosure error for 'give turn' request. (SGVTRN)
01E3	Interrupt request is not demand turn, attention/data attention, or purge record.
01E4	Input status for a send control letter is not permitted.
01E8	Write request without turn.
01E9	Write segmented record request without turn.
01EA	Write segmented letter request without turn.
01EB	Send control letter request without turn.
01EC	Disconnection request without turn.
02xx	**Presentation Control**
0201	Protocol level not supported
0202	Application designation protocol error.
0203	Character encoding error. TM cannot support the proposed encoding.
0204	Character set error. TM cannot support the proposed character set.

0205	Character subset error. TM cannot support the proposed character subset.
0206	Incorrect record encoding.
0207	Incorrect parameter encoding.
0230	Data presentation control error. The presentation control proposed for this session cannot be used
0231	Device type is incompatible with the configuration.
0232	TM control protocol is incorrect.
0233	Device-sharing attributes are invalid.
0234	Initiator or acceptor configuration is not correct.
0235	Logical device index error.
0236	Number of logical devices is incompatible with the configuration.
0237	TM protocol record not supported.
03xx	**Terminal Management**
0300	Sysgen error WARNING. There is no mapped object; some objects will be spare.
0301	Operator requested session abort or logged.
0302	Idle time run out after secondary network failure.
0303	Idle time run out for no traffic.
0304	Form not found.
0305	Operator requested suspension.
0306	Destructive attention send on the session.
0307	Unknown TX addressed in this session. TM is unable to a the session.
030A	Protocol error. A record was received which did not comply with current standards
0310	Insufficient resources. The receiver cannot act on the request because of a temporary
031E	Incorrect value for Retry or Wait parameters on UP LL command.
0320	Function not supported.
0321	Parameter error. This can result
0322	Resource not available. The
0323	Intervention required (on principal device).
0324	Request not executable.
0325	EOI required.
0326	Presentation space altered, request executed.
0327	Presentation space altered, request not executed.
0328	Presentation space integrity lost.
0329	Device busy. The device is busy and cannot execute the request.
032A	Device disconnected.
032B	Resource not configured.
032C	Symbol set not loaded.

032D	Read partition state error.
032E	Page overflow.
0330	Subsidiary device temporarily not available.
0331	Intervention required at subsidiary device.
0332	Request not executable because of subsidiary device.
0340	TM cannot accept a new connection.
0341	Object status incorrect.
0342	The TM configuration is not correct.
0343	Unknown TX addressed on this session.
0344	Data presentation protocol error.
0345	Device type is incompatible with the configuration, or is not supported.
0346	TM control protocol incorrect.
0347	Device shareability attributes are invalid.
0348	Initiator or acceptor configuration is not correct.
0349	Logical device index error.
034A	Number of logical devices incompatible with the configuration.
0350	Disconnection of TM after reinitialization of the network.
0360	File not found. (Welcome and Broadcast Messages)
0361	Site not found. (Welcome and Broadcast Messages)
0362	NASF error. (Welcome and Broadcast Messages)
0370	No-session timeout. Device disconnected.
0371	No-input timeout. Device disconnected.
0372	No-output timeout. Device disconnected.
0373	Timeout due to no backup session being initiated.
0374	Timeout due to no backup session being established.
0375	Connection refused because of late activation of back up session.
0376	Disconnection of current session to switch to backup session.
0380	AUTOCN parameter not declared.
0381	Mixed ETB in data sent by VIP screen and cassette
0382	Data header sent by the terminal incorrect.
0383	Desynchronization in the exchange of data.
0384	KDS block count error.
038C	Remote terminal is not connected
0390	Unknown mailbox.
0391	No call packet to return.
0392	No "Possibility" command to return Protocol error
03C0	Slave device disconnection.
17xx	**Network Layer**
1701	PAD connection refused.
1702	Flow control error.

1706	Logical channel number not zero in restart packet.
1707	Illegal packet length or use of D-bit forbidden.
1708	Illegal header.
1709	Illegal Logical Channel Number.
1710	Invalid packet type for the automaton state. Protocol error
1711	Incorrect packet type.
1712	Inconsistent network parameters in the generation file.
1713	No more space.
1714	DSAC network layer object not usable.
1717	USED/ENBL transition. Transport station is locked.
1718	USED/ENBL transition. This is a back-up NR.
1719	USED/ENBL transition. Dynamic close due to load.
171A	USED/ENBL transition. Transfer time-out has elapsed.
171B	USED/ENBL transition. This is a back-up NR.
171C	USED/ENBL transition. Transport station is idle.
171E	USED/ENBL transition. NR object is locked.
171F	ENBL/LOCK transition. NR HDLC has no more memory space.
1721	Remote station is inaccessible via the configured network. Check
1723	Incorrect PAD password.
1724	Virtual circuit already in use. LCN (Logical Channel Number) too high.
1725	Invalid virtual circuit.
1726	Packet too short. Protocol error for the equipment directly connected to the Bull Datanet.
1727	Incompatibility between the generation parameters of two communicating systems on window or packet size.
1729	Packet size in communicating systems not the same.
1731	Timer runs out while waiting for call confirmation.
1732	Timer runs out while waiting for clear confirmation.
1733	Timer has run out while waiting a reset confirm.
1740	Call setup or call clearing problem.
1741	Open failure on virtual circuit. No flow control on this NS.
1742	Incorrect facility. Protocol error for the equipment directly connected to the Bull Datanet.
1744	Unknown subscriber.
1745	End of time-out on reset confirm. Invalid facility length. Protocol error for the equipment directly
1747	No logical channel available.
1749	End of time-out on call confirm.
174F	Incorrect packet length. Protocol error for the equipment directly connected to the Bull Datanet.
1755	Flow control, window, packet size or reset error.

1760	Frame disconnection.
1770	Frame connection.
1771	Frame reset.
1781	No more network routes available for X.25 switching.
1782	Maximum of 15 switches have been used,
1783	Flow control negotiation error.
1785	Frame level disconnection.
1786	Frame level connection.
1787	Frame level reset.
1790	Frame level not established.
1791	No more logical paths available for the PAD.
1792	Echo service busy.
1793	Incorrect PAD password.
1794	All the PAD virtual circuits are used
1795	X.25 initialization not possible.
179B	LCN not null in restart packet
179D	Incompatible header (receive error: all VC of concerned NS
179E	LCN greater than NBVC in NS directive
179F	Incorrect packet type
17A0	Invalid facility.
17B0	Normal disconnection.
17B1	X.25 Echo in use.
17B2	No more logical channels available.
17B3	No more PAD connections allowed.
17B4	TS SX25 or NU X25 object locked.
17B5	Buffer capacity overflow.
17B6	Normal disconnection.
17B8	Unknown calling SNPA (Sub-Network Point of Attachment).
17B9	Internet problem.
17CB	Call collision on VC
17CC	Incompatible generations (NR object without mapping).
17CE	Invalid status NR locked.
17CF	Lack of space.
17D0	Unknown subscriber.
17D4	TSCNX already used for another connection. SCF internal error.
17D7	Transport station locked.
17DD	Proper NS locked.
17DE	Invalid status NR locked.
17DF	Lack of space.
17E0	Forbidden parameter or invalid value.
17E1	Invalid transition.
17E2	Upward-mapped object (TS) not locked.

17E3	No object mapped above.
17E4	NR not locked (MP NR -ADD/-SUB) or virtual circuit already open.
17E5	NR is last in list and the TS is not locked.
17E6	No object mapped above (UP NR -PRIO). NR not mapped on TS.
17E7	Upward mapped object not locked
17E9	Mix of datagram and connection network
17EB	Class inconsistent with NR.
17EE	Incompatible generations. NR object without mapping.
17FF	Wrong parameter in administrative CALL
18xx	**Transport Layer**
1800	Normal disconnection initiated by the correspondent
1801	Local saturation at connection request time.
1802	Failed negotiation at connection time.
1803	Duplicate connection. Two or more requests have been issued for the same connection.
1804	Redundant request.
1805	Retransmission Time-out at transport level.
1806	Survey time-out at transport level.
1807	Transport protocol error.
1808	Session Control specified is not available (inaccessible).
1809	Requested Session Control Id unknown by remote transport.
180A	Termination because of disconnection by administration.
180B	Session Control/Transport interface error.
180C	Connection request on non-sharable VC in case of ISO Transport. ISO: header or parameter length is invalid.
1817	Station in shut-down state.
181F	No memory space at connection time.
1821	Session Control inaccessible by configured session routes. ISO: Session entity not attached to TSAP.
1824	Collision between Close NC and Open TC.
182E	Remote station not configured.
182F	Resource saturation.
1831	ISO: No route for the called NSAP.
1832	ISO: Received NSAP addresses are wrong.
1833	Segmentation violation.
1834	ISO:QOS priority not available temporarily, due to a local condition (for example, lack of resources).
1835	ISO:QOS priority permanently unavailable locally (for example, due to an error in the system generation).
183A	ISO: Remote reason not specified.
183C	ISO: Remote transport entity congestion at connect request time.
1840	Server in terminating state. TC has been re-assigned on another NC.

18A1	An additional NC has been assigned to a TC.
18B0	NC has been re-assigned on another VC.
18EF	Disconnection at Transport level caused by reception of RESTART DSA during the transfer phase.

Windows Sockets error Codes

Below is a list of Windows Sockets return codes and the corresponding description.

Hex code	**Windows Sockets Access Error name**	**Description**
2714	WSAEINTR	The (blocking) call was cancelled via WSACancelBlockingCall()
2719	WSAEBADF	The socket descriptor is not valid.
271E	WSAEFAULT	An invalid argument was supplied to the Windows Sockets API.
2726	WSAEINVAL	An invalid call was made to the Windows Sockets API.
2728	WSAEMFILE	No more file descriptors are available.
2733	WSAEWOULDBLOCK	The socket is marked as non-blocking and no connections are present to be accepted.
2734	WSAEINPROGRESS	A blocking Windows Sockets call is in progress.
2735	WSAEALREADY	The asynchronous routine being cancelled has already completed.
2736	WSAENOTSOCK	The descriptor is not a socket.
2737	WSAEDESTADDRREQ	A destination address is required.
2738	WSAEMSGSIZE	The datagram was too large to fit into the specified buffer and was truncated.
2739	WSAEPROTOTYPE	The specified protocol is the wrong type for this socket.
273A	WSAENOPROTOOPT	The option is unknown or unsupported.
273B	WSAEPROTONOSUPPORT	The specified protocol is not supported.

273C	WSAESOCKTNOSUPPORT	The specified socket type is not supported in this address family.
273D	WSAEOPNOTSUPP	The referenced socket is not a type that supports connection-oriented service.
273E	WSAEPFNOSUPPORT	
273F	WSAEAFNOSUPPORT	The specified address family is not supported by this protocol.
2740	WSAEADDRINUSE	The specified address is already in use.
2741	WSAEADDRNOTAVAIL	The specified address is not available from the local machine.
2742	WSAENETDOWN	The Windows Sockets implementation has detected that the network subsystem has failed.
2743	WSAENETUNREACH	The network address can't be reached from this host. There is probably a problem in the way you have set up TCP/IP routing for your PC (most likely you have not defined a default router).
2744	WSAENETRESET	The connection must be reset because the Windows Sockets implementation dropped it.
2745	WSAECONNABORTED	The connection has been closed.
2746	WSAECONNRESET	
2747	WSAENOBUFS	Not enough buffers available, or too many connections.
2748	WSAEISCONN	The socket is already connected.
2749	WSAENOTCONN	The socket is not connected.
274A	WSAESHUTDOWN	The socket has been shutdown.
274B	WSAETOOMANYREFS	
274C	WSAETIMEDOUT	Attempt to connect timed out without establishing a connection.
274D	WSAECONNREFUSED	The attempt to connect was forcefully rejected. The service on the other side is not available.
274E	WSAELOOP	Too many symbolic links were encountered in translating the path name.
274F	WSAENAMETOOLONG	
2750	WSAEHOSTDOWN	The host machine is out of service.
2751	WSAEHOSTUNREACH	The host machine is unreachable.

2752	WSAENOTEMPTY	
2753	WSAEPROCLIM	
2754	WSAEUSERS	
2755	WSAEDQUOT	
2756	WSAESTALE	
2757	WSAEREMOTE	
276B	WSASYSNOTREADY	Indicates that the underlying network subsystem is not ready for network communication.
276C	WSAVERNOTSUPPORTED	The version of Windows Sockets API support requested is not provided by this particular Windows Sockets implementation.
276D	WSANOTINITIALISED	A successful WSAStartup() must occur before using this API.
2AF9	WSAHOST_NOT_FOUND	Authoritative answer host not found.
2AFA	WSATRY_AGAIN	Non-authoritative answer host not found, or SERVERFAIL.
2AFB	WSANO_RECOVERY	Non-recoverable errors, FORMERR, REFUSED, NOTIMP.
2AFC	WSANO_DATA	Valid name, no data record of requested type.