HOST LINKS

G5250™

Emulating IBM5250 Terminals

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

GALLAGHER
ROBERTSON

Version 6.3

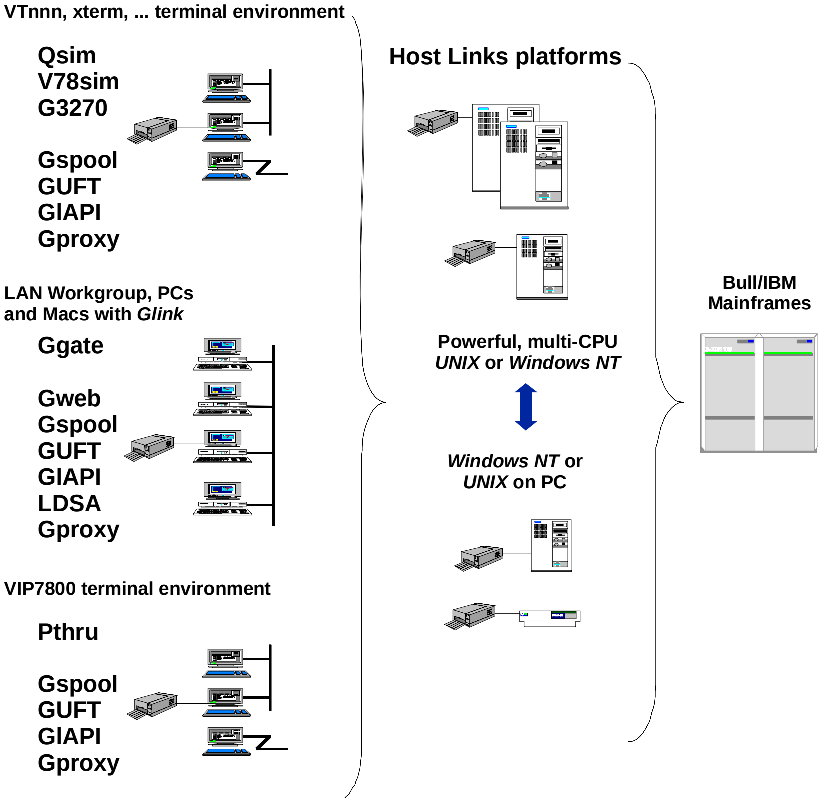
GALLAGHER & ROBERTSON AS, Kongens gate 23, N- 0153 Oslo, Norway
Tel: +47 23357800 • Fax: +47 23357801
www: `http://www.gar.no/`
e-mail: support@gar.no

Contents

Host Links G5250

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.


VTnnn, xterm, ... terminal environment
Qsim
V78sim
G3270
Gspool
GUFT
GIAPI
Gproxy
LAN Workgroup, PCs and Macs with *Glink*
Ggate
Gweb
Gspool
GUFT
GIAPI
LDSA
Gproxy
VIP7800 terminal environment
Pthru
Gspool
GUFT
GIAPI
Gproxy
Host Links platforms
Powerful, multi-CPU *UNIX* or *Windows NT*
Windows NT or *UNIX* on PC
Bull/IBM Mainframes

Scope of the product

Functionality

G&R/G5250 provides synchronous IBM5250 functionality to users with asynchronous terminals connected to a UNIX/Linux system. It allows any UNIX/Linux user to connect to IBM AS/400 servers. G5250 is not available for Windows servers.

Network connections

IBM AS/400 systems can be accessed using Telnet 5250 (TN5250).

Asynchronous terminals

All asynchronous terminals supported by *Host Links* may be used to access G&R emulators, including Digital VTnnn, Bull VIP7801 and DKU7102 and PCs using *G&R/Glink* (but Glink users should normally use Glink's own emulation and connect directly, or via *G&R/Ggate*). The visual attributes shown are limited to the capability of the specific terminal. For VT340s and VT220 clones with colour support use our VT220 video driver with COLOURMODE in the profiles file.

Integration

G&R emulators allow execution of local commands, and the user may easily switch to other *Host Links* products while running the emulator. This includes the Interactive Help System that has all the information available in this manual. System administrators may easily customize the menus provided. The emulators utilize the *Host Links* screen interface. All functionality provided by this interface is therefore available for emulator users.

Run-time licenses

In order to run a G&R emulator, the following license keys must be present in your `/usr/gar/config/licenses` file:

`basic`	For the base G&R run-time system
`emulator`	One of `g3270`, `g5250`, `qsim`, `v78sim`

The licenses file identifies the G&R distributor, the owner of the license and the licensed products. The license key for a product will normally state for how many simultaneous sessions the product is licensed. If a limitation is specified in the license, only the licensed number of sessions can be active at any time.

Connecting to host

You can connect to a host application by typing line feed followed by C (connect) `LF C`. G5250 will then make a connection request using the parameters set in the configuration file or given on the command line. You can instruct G5250 to connect at startup by giving the parameter `-CN ON`. You can also do your own connect by entering a connect command string when G5250 has started (`$*$CN` command).

For TN5250 (Telnet 5250) connections only the IP address of the AS/400 system or gateway is used, either in numeric or symbolic format e.g.

```
$*$CN ouras400.gar.no<XMIT>
```

TN5250 servers are normally configured to use the standard Telnet port, and the TN5250 dialect is arrived at by Telnet negotiation. If the TN5250 server is configured on another port the connect command is:

```
   $*$CN gateway:portnumber
```

G5250 keyboard

G5250 Control keys

Control key table

Function	KEY	Function	KEY
backspace	BACKSP	**G5250 command**	LF
backtab	BACKTAB	**home**	HOME
clear	CLEAR	**insert mode**	INS
delete	DEL	**new line**	RETURN
enter	XMIT	**F1..F12**	F1..F12
erase field	EOF	**F13..F24**	SHIFT/F1..SHIFT/F12
erase input	EOP	**reset**	RESET
field mark	CTL/B	**tab**	TAB

Internal key and function numbering

This table lists all keys known to the `Host Links` emulators, their internal key number for use if quoting the key, and the assignments of functions to keys in G5250. All keys in the table may be redefined as macros, but the original function will still be available by quoting the key. Only control combinations (CTRL/X) marked unassigned are available for your own functions. Others are reserved for present or future use in the video handler.

Mnemonic	Hex	Decimal	G5250 function	**Remark**
CTL/A	$101	257	alfa override	
CTL/B	$102	258	field mark	
CTL/C	$103	259	cursor select	
CTL/D	$104	260	dup	
CTL/E	$105	261	unassigned	
CTL/G	$107	263	unassigned	
BS	$108	264	back space	
HT	$109	265	Forward tab	
LF	$10A	266	command key	Also VT220 F16 (DO)
CTL/L	$10C	268	unassigned	
CR	$10D	269	new line	
CTL/R	$112	274	unassigned	Allocable control key
F1	$130	304	F-key 1	Also VT220 PF1
S1	$131	305	S/F-key 1	
F2	$132	306	F-key 2	Also VT220 PF2
S2	$135	309	S/F-key 2	
F3	$136	310	F-key 3	Also VT220 PF3
S3	$137	311	S/F-key 3	
F4	$138	312	F-key 4	Also VT220 PF4

Mnemonic	Hex	Decimal	G5250 function	**Remark**
S4	$139	313	S/F-key 4	
F5	$13A	314	F-key 5	Also VT220 PF5
S5	$13B	315	S/F-key 5	
F6	$13C	316	F-key 6	F6 to F12; also VT220
S6	$13D	317	S/F-key 6	
F7	$13E	318	F-key 7	
S7	$13F	319	S/F-key 7	
CUP	$141	321	cursor up	Also VT220 cursor keys
CUD	$142	322	cursor down	
CUF	$143	323	cursor forward	
CUB	$144	324	cursor back	
CUH	$148	328	Home	
EOP	$14A	330	erase page	
EOL	$14B	331	erase field	
F8	$150	336	F-key 8	
S8	$151	337	S/F-key 8	
F9	$152	338	F-key 9	
S9	$153	339	S/F-key 9	
F10	$154	340	F-key 10	
S10	$156	342	S/F-key 10	
F11	$15C	348	F-key 11	
S11	$15D	349	S/F-key 11	
F12	$15E	350	F-key 12	
S12	$15F	351	S/F-key 12	
CLR	$160	352	Clear	

Mnemonic	Hex	Decimal	G5250 function	**Remark**
RIS	$163	355	Reset	
RES	$165	357	Reset	
IC	$167	359	insert mode	
IMR	$168	360	unassigned	
XMT	$169	361	Enter	Send
IL	$16A	362	unassigned	
DL	$16F	367	unassigned	
TBS	$170	368	unassigned	
TBI	$175	373	unassigned	
DC	$178	376	Del	Same as delete
CBT	$17B	379	Back tab	
TBC	$17C	380	unassigned	
DEL	$17F	383	Delete	

Control keys and macros

Administration

Macros and redefined control keys are stored in sub-directories of the macro directory, /usr/gar/<emu>_mac where <emu> can be 3270, 5250, qsim or v78. This directory can be redefined with -MP, and would then be shared by all users with the same -MP.

Each sub-directory of the macro directory is a keyboard definition. By default, each user has a personal keyboard definition. The sub-directory name is the user's UNIX identification. This can be changed using -MD, and all users having the same -MD would share the keyboard definition. The first time a user presses a control key or executes a macro, the emulator checks the users sub-directory for a definition of the key. If the key is not defined in the user's sub-directory, it checks a shared sub-directory default. If a definition is found it is loaded for future use, otherwise the standard key definition is used.

By default, all users have permission to define macros and redefine control keys. This can be changed using the Host Links profiles directive NOMACRO. Users with NOMACRO in their profile can only read keyboard definitions made by the administrator. Note that when a group of users have the same -MD then they should not have permission to define macros, since they would all redefine the shared keyboard. Generally, only the administrator of such a group has permission to define macros. The administrator uses a personal keyboard definition and then copies it over to -MD. In the same way the administrator defines the default keyboard definition and copies it to sub-directory default.

```
v78sim -md group_a -li dsa........
qsim -md group_a -li dsa........
g3270 -md group_a -li dsa........
g5250 -md group_a -li dsa........
```

The command lines above would define the users as having their keyboard definitions at: /usr/gar/ under directories v78_mac, qsim_mac, G3270_mac, G5250_mac respectively with subdirectory group_a.

Redefining your keyboard

G&R emulators provide 24 user definable macros that may be executed with the LF F-KEY sequence. Additionally any control key on the terminal can be redefined as a macro. Control keys visible to the emulators are all DKU7102, VIP7800, VT220 or Terminfo function keys defined for the actual terminal. All macros are stored in the macro directory, /usr/gar/<emu>_mac (where <emu> can be 3270, 5250, qsim or v78), in a sub-directory corresponding to the user identification (unless redefined with -MD). The system administrator may define a default keyboard in directory default that will be used for all keys having no definition in the user's own macro directory.

A macro sequence may contain any key other than itself. This includes other macro keys. Macros may be nested to a depth of 8. When a nested macro is executed the current content of the macro is used. This makes it possible to redefine 'subroutine' macros. Macros may contain control characters (LF X). Macros may contain the transmit key, and if so the macro will be suspended until the mainframe replies.

Control keys may be included in a macro simply by pressing them. However, your terminal may not have the control key necessary to reach the emulator function you would like in the macro. To help you in this situation the emulators allow you to 'quote' any control key with the sequence LF NNN " that is interpreted as if you had pressed the key with the internal value of nnn (decimal).

If you use a control key in a macro and if you later redefine the control key (LF K) then your macro will use the new definition. If you want to execute an internal function e.g. Backspace, and also intend to redefine the BACKSPACE key to do something else then you must 'quote' the internal value for the Backspace function (decimal 264) in your macro, rather than use the key.

e.g. use LF 264 " at the point in the macro where you want a Backspace.

Redefining the command key

The command key, LF (CTRL/J and F16, DO on VT200), may be assigned to any visible control key by redefining the key and quoting the internal value of the command function. Thus some key 'funny' could be made into the command key as follows:

`LF K 'FUNNY' Y`	redefines 'funny' and confirms the redefinition
`LF 2 6 6 " LF N`	quotes the command function and terminates the macro

The key 'FUNNY' would thereafter act as the command key. Note that the quote of the command function will NOT act as a command key during the macro definition. This is an exception to the normal rule that all functions used in a macro are executed during macro definition. This means that if you want a special command key you should define it first, and thereafter use the special command key in all macros including a command sequence. If you have no LF key on the terminal you would use CTRL/J while defining a new command key as above.

A key used in LF KEY commands cannot be used as the command key. In addition to F1-F12 and SF1-SF12 (execute macro) the emulators have assigned LF command functions to CR (do nothing), HOME (clear), and DEL (reset), so these may not themselves be used as the command key.

Foreign terminals (VTnnn)

If you are using a 'stranger' terminal, particularly a VTnnn terminal, then you will find few of the standard control keys, but you may find up to 20 function keys, and other control keys. F1 to F12 may be needed in order to signal function key presses to your mainframe application, in which case 24 of the available control keys must be defined as internal functions F1 to F12 and shift/F1 to shift/F12. All of the other keys may be redefined as emulator functions or macros (LF K). Note that the VT200 function key 16 (DO) is used by default as the command key (LF), but this can be redefined. Also note that in general F1 to F5 of the VTnnn are local keys and cannot be used as F1 to F5 for mainframe sequences, unless programmable.

The VT100 video handler

This handler should be used with Digital VT100 terminals and clones and emulations of it. A TERM variable starting with ‘vt1’ selects this handler.

The real VT100 family has only PF1 to PF4 function keys. Since most mainframe applications expect keys F1 to F12 with shift, as well as some other VIP and DKU specific keys, these must be defined in some way on the terminal. The VT100 handler by default uses this mapping:

VT100 key	Result	Result after 0
Keypad 1	F1	Shift F1
Keypad 2	F2	Shift F2
Keypad 3	F3	Shift F3
Keypad 4	F4	Shift F4
Keypad 5	F5	Shift F5
Keypad 6	F6	Shift F6
Keypad 7	F7	Shift F7
Keypad 8	F8	Shift F8
Keypad 9	F9	Shift F9
Keypad ,	F10	Shift F10
Keypad -	F11	Shift F11
Keypad .	F12	Shift F12
Keypad ENTER	XMIT	
PF1	Home	
PF2	Erase to end of line	
PF3	Erase to end of page	
PF4	Clear	

The column ‘Result after 0’ shows what happens if the keypad 0 key is pressed first, then the key in the left column.

The interpretation of the numeric keypad can be toggled from the above to numeric using the CTRL/N key. In numeric mode the keypad may be used for keying numbers. The initial state of this interpretation can be set to numeric by using the KPNUM profile directive.

The VT200 video handler

This handler should be used on Digital VT200, VT300 and VT400 terminals and clones and emulations of it. A TERM variable starting with ‘vt2’, ‘vt3’ or ‘vt4’ selects this handler.

The real VT200 family has 20 function keys but F1 to F5 are allocated to internal functions, and there are no shifted F-keys. Since most mainframe applications expect keys F1 to F12 with shift, as well as some other VIP and DKU specific keys, these must be defined in some way on the terminal. The VT200 handler by default uses this mapping:

VT200 key	Result	Result after 0
KEYPAD 1	F1	SHIFT F1
KEYPAD 2	F2	SHIFT F2
KEYPAD 3	F3	SHIFT F3
KEYPAD 4	F4	SHIFT F4
KEYPAD 5	F5	SHIFT F5
KEYPAD 6	F6	SHIFT F6
KEYPAD 7	F7	SHIFT F7
KEYPAD 8	F8	SHIFT F8
KEYPAD 9	F9	SHIFT F9
KEYPAD ,	F10	SHIFT F10
KEYPAD -	F11	SHIFT F11
KEYPAD .	F12	SHIFT F12
KEYPAD ENTER	XMIT	
PF1	HOME	

VT200 key	Result	Result after 0
PF2	ERASE TO END OF LINE	
PF3	ERASE TO END OF PAGE	
PF4	CLEAR	
TAB	TAB	BACKTAB
FINISH	F4	SHIFT F4
INSERT	INSERT CHARACTER	INSERT LINE
REMOVE	DELETE CHARACTER	DELETE LINE
SELECT	SLC	
PREVIOUS SCREEN	PAGE UP	
NEXT SCREEN	PAGE DOWN	
F6	F6	
F7	F7	
F8	F8	
F9	F9	
F10	F10	
F11	F11	
F12	F12	
F13	LF	
F14	V14	
HELP	F1	
DO	LF	
F17	RESET INITIALIZE	RESET
F18	SET ATTRIBUTE	DELETE ATTRIBUTE
F19	SET TABSTOP	DELETE TABSTOP
F20	ESCAPE	

The column ‘Result after 0’ shows what happens if the keypad `0` key is pressed first, then the key in the left column.

The keyboard interpretation can be changed using profile directives `NOKPAM` and `KPNUM`. The `NOKPAM` directive turns off all interpretation. The `KPNUM` directive interprets the numeric keypad as numeric so that it can be used for keying numbers. There are thus four possible states for the keyboard:

> `NOKPAM` set and `KPNUM` not set. This gives little initial functionality, it does no VT200 keyboard mapping at all. The function keys deliver native VT200 sequences, the numeric keyboard sends the sequences associated with application mode, and these too are delivered as native VT200 sequences.
>
> `NOKPAM` is set and `KPNUM` is set. This is just as the above, except that the numeric keypad is interpreted as numeric and can be used to key numbers.
>
> `NOKPAM` is not set, and `KPNUM` is not set. This is the default as shown in the table above. This gives maximum mapping of the keyboard into keys needed by Host Links products.
>
> `NOKPAM` is not set and `KPNUM` is set. This gives the mapping in the table above, except that the numeric keypad is used to enter numbers.

The `CTRL/N` key can be used to toggle between the modes described above. For each press of `CTRL/N` the keyboard interpretation steps to the next mode. The mode is indicated with a status message.

The profiles option `COLOURMODE` allows the video handler to send ANSI colour sequences that are interpreted by VT340, and also many VT220 clones.

Keyboard mapping suggestions

If your VT200 clone has `F1` to `F5` programmable then you can program them to send numeric pad application mode sequences (`Esc O q` -> `Esc O u`) to give you `F1` to `F5` on the VT200 F-keys. Start with `KPNUM` in your profile, and you need never switch to application mode except for screen reset (`CTRL/N 0 PF4`, then step back to numeric with `CTRL/N`).

Otherwise if you suppress the mapping completely (NOKPAM) then all legal VT200 keys deliver native mode to the emulators. Extensions of the Single shift 3 sequence SS3 (Esc O x, x is A-Z a-z) and CSI (Esc [n ~, n is 0-39) can also be delivered if you have a programmable clone, and may be redefined as a function or macro. The numeric pad in application mode sends native sequences and these are recognized as native VTnnn sequences, and may be freely redefined as emulator functions or macros, independent of your solution for the F-keys.

In general, NOKPAM offers a greater freedom of choice of keys, but will mean more work defining a keyboard. VT220 clones with programmable shifted F-keys should certainly use this solution, and program the shifted f-keys 1-12 to send recognized SS3 or CSI extensions so that they can be assigned to emulator functions (SF1-SF12).

For a complete description of the video handlers and the profiles directives that allow you to customize them, see the *Installation and Configuration Guide*.

G5250 Function keys

The PF keys on the synchronous IBM5250 family are defined as transmitting the screen data to the host, preceded by a one byte flag (AID) to show which PF key was pressed. There are 24 PF keys. The PA keys are defined as sending a single byte to the host, but not the data from the screen. There are 3 PA keys. This means that many of the asynchronous terminals and PC-based emulations will not have enough function keys to allocate a unique key to each. The solution in G5250 is to allocate internal functions F1 to F12 as PF1 to PF12, and to make SHIFT/F1 to SHIFT/F3 lead to a selection. For example SHIFT/F1 leads to a selection of F1 for PA1 or SHIFT/F1 again for PF13. If you are defining a stranger keyboard you would put PA1 on a key by defining it as a macro (LF K) in which you quoted the internal function value of SHIFT/F1 F1'.

The type-ahead facility

Option `-TA ON` turns on type-ahead. Text typed while you don't have the 'turn' is stored until it is your turn, and then delivered to the emulator. The data will go into the variable fields if there is a form present. You can type several commands with Xmit after each. They will be delivered one at a time, whenever you get the turn. All `LF` commands clear the type-ahead buffer and are executed immediately. An `LF B` command will always send a break, and an `LF SPACE` or `LF CR` command will simply clear the type-ahead buffer.

Screen-scraping applications

In many cases the 'terminal' running a G&R emulator is not a terminal at all, but an application that simulates a VTnnn terminal in order to access the mainframe data that is painted on the VTnnn 'screen' by the G&R emulator. In this way the application can use a commodity terminal emulation (VTnnn) to access mainframe data otherwise available only to proprietary terminals (VIP, DKU, 3270, 5250). These applications are referred to as 'screen-scraping' applications. The VTnnn protocol has no 'turn' signal whereby the screen-scraping application can recognize that the output of a particular screen is finished. They therefore generally wait for a pause in output, and assume that the screen is complete. To make such an application work you will need to disable the normal dynamic update of the terminal screen with the -DSU parameter, so that the complete screen is updated in a single pass after the mainframe output is complete.

Printing

Hard copy

Host Links has a generic hard copy function in the video interface, CTL/P that may be used from any *Host Links* product to copy the screen to your screen.sav file. The LF P command enters a print menu of options for manipulating this file. However, if you always take hard copy in the same way and want it printed immediately via the print spooler, or direct to a device, then the LF W command may be used to write the screen content on to your <product>.sav. If you have configured a write command (-WC parameter), then this will be executed.

e.g. `-WC 'lp -dpr1'`

Options for the UNIX spooler lp may be used within the quotes, and the path name of the print will be supplied automatically to the command as a final argument.

You can also use your own shell script to process/print the file, and the path name of the print will be supplied automatically to the command as a final argument.

e.g. `-WC myprintscript`

Commands

G&R emulators have an extensive range of commands that you can select with the sequence `LF x` where `LF` (`CTRL/J`) is the command key.

Command overview

The following keys may be typed in immediately after the command key.

`1-99`	set count for command
`!`	start UNIX shell
`"`	quote count as a key
`#`	change context (1-9)
`$`	directory administrator
`+`	Field + key (5250)
`-`	Field - key (5250)
?	product information
`B`	send break to the host
`C`	connect to host
`D`	disconnect from host
`E`	execute UNIX command
`H`	5250 Help key
`I`	set insert mode
`K`	start key definition
`M`	start macro definition
`N`	end key/macro definition
`O`	set overlay mode
`P`	print menu
`Q`	quit (terminate emulator)
`R`	sysreq key (5250)
`S`	set emulator/line parameter
`T`	call editor with screen
`U`	unlock keyboard
`W`	write screen, and print
`X`	include CTL in macro
`Z`	test key (5250)
`Fn`	execute macro F1-F12
`S/Fn`	execute macro SF1-SF12
`Home`	clear screen
`Del`	reset screen
`DOWN`	roll up (5250)
`UP`	roll down (5250)
`XMIT`	send page/all (DKU/V78)
`EOL`	erase user input (DKU)

The command key is F16 (Do) on a VT220, and the GRAY- key on a PC with *Glink*. The X is some other key denoting the desired command. You can type these commands or include them in macros assigned to various control keys on your keyboard. Some functions may be inhibited using *Host Links* profiles directives:

```
NODOLLAR, NOEXEC, NOEDIT, NOMACRO, NOSETUP
```

Set number for following command (LF n)

Many of the products in the *Host Links* range use the LF command key followed by a single character to select a function. If the function has several options then these may be selected by setting a number first, using the LF NUMBER command.

Examples are the context switch command, and the execute command:

`LF #`	Switch to context manager
`LF 2 #`	Switch to context 2
`LF 6 #`	Switch to context 6
`LF E`	Disconnect video, show/execute command, wait for CR
`LF 1 E`	Disconnect, cursor to 24/1, execute, no wait for CR
`LF 2 E`	No disconnect, execute command, refresh variables
`LF 3 E`	No disconnect, execute command, no refresh
`LF 4 E`	No disconnect, execute command, refresh whole screen

Start UNIX shell (LF !)

If you are working within the *Host Links* systems and are an advanced user, a UNIX specialist, or the system administrator you might find it convenient to start a command line shell in order to work with UNIX commands for a period without loosing your *Host Links* context. Thus you can maintain several host connections or other activities while you temporarily work at command level.

`LF !`	Start a command line shell

This may be inhibited in your `profiles` file (`NOEXEC`).

Quote count as key (LF nnn ")

In several of the *`Host Links`* products there is a need to enter a key that is not on the keyboard. This can be achieved using the `LF NNN "` command. The count (NNN) represents the key value (in decimal), and is followed by `"` to quote the key. Combined with knowledge of the character set this allows entry of any valid display character. Combined with knowledge of the assignment of functions to control keys for a given product, it allows selection of functions otherwise unavailable from the keyboard. This is particularly important when defining keys or macros (e.g. in the macro facility).

`LF 209`	Set count 209 (hex D1)
`"`	Quote 209 as key Ñ (Upper case N with tilde)
`LF 304`	Set count 304 (hex 130)
`"`	Quote 304 as a key (F1) to select internal function F1

See section on macros for a table of internal functions.

Change context (LF #)

The *`Host Links`* multi-context feature is a set of utilities for allowing user friendly access to the UNIX multitasking capability such that a user may have several facilities in use at the same time, and jump between them as desired without terminating or restarting. For example, you may have several terminal emulators running, each of them logged on to a different host application and then switch between these applications as desired.

Each facility started by the *`Host Links`* multi-context feature is defined as an active context. If you have installed this software, then the `LF #` command allows you to jump directly to another active context.

A context number (1-n) is assigned to each active context. Use a repetition count to set the desired context number:

`LF #`	Jump to context manager

`LF N #` Jump to context n

You should also note that *Host Links* supports UNIX shell level job control as supported in all UNIX shells except old versions of the Bourne shell.

Call the Host Links file manager (LF $)

The *Host Links* file manager, *G&R/Gdir*, is a fully compatible interactive menu driven replacement for the standard file commands in UNIX. It provides facilities for navigating around the file and directory structure, and for creating and releasing files.

Additionally it has a facility for activating the alphanumeric keyboard with user-oriented functions that can then be executed with a single keystroke. These functions may contain any command, and the name of the file or directory being pointed to with the cursor in the file display is available as a parameter to the commands. The file manager is integrated with a configurable editor and print system, so the file pointed at may be read in to the editor or printed. For a detailed explanation, enter the manager and type `LF H`.

G&R/Gdir is included in the *G&R/Basic* package of utilities bundled with all *Host Links* products. Your `profiles` file may inhibit this function (`NODOLLAR`).

Field Plus (LF +)

Executes the 5250 Field Plus command.

Field Minus (LF -)

Executes the 5250 Field Minus command.

Get product information (LF ?)

Product name, version number and platform code is displayed in line 24.

Set FKC-addressing in macro (LF A n)

FKC-addressing is only really relevant for the DKU emulation (*Qsim*).

The DKU-family of terminals allows the FKC-sequence defined on an F-key to be addressed in three different ways. The sequence may be sent directly to the host (FKC-0), the sequence may be sent to the host and followed by transmission of whatever is transmittable on the screen (FKC-1), or the sequence may be merely displayed on the screen (FKC-2). On the hardware DKU terminals FKC-0 is the default; the sequence is sent directly to the host.

You program the FKC-sequences by redefining the F-keys as macros using the `LF K` command. You can set a function code first using `LF F` before defining FKC addressing using `LF A`. The `LF A` sequence asks you to enter `0`, `1` or `2` to define how the macro is to be addressed. FKC-0 and FKC-1 both transmit data to the host, and ASCII/CTL characters can be included in the macro, see `LF X`. In G&R emulations FKC-2 (display macro) is default, but a display macro may include a transmit, and send to the host.

In *G3270* this command is added for functional compatibility with other emulators in the family, and intended for specialists because use of macros that are transmitted directly to the host is complex in the 3270 environment. Macros are by default delivered through the emulator and cover the needs of ordinary users who need to define macros that automate dialog steps with the host.

The 3270 transmits a single byte direct to the host for PA-keys. It transmits a single byte (AID) followed by the cursor position, followed by transmission of whatever is transmittable on the screen for PF-keys.

In *G3270* PA-keys are simply defined using `LF A` to set FKC-0 addressing, and then entering the desired control byte. You enter this in ASCII. If you also define PF-keys that turn on the `-FC` parameter you should start the PA-key macro with `LF S` and set `-FC off` before the `LF A` to set FKC-0 addressing.

PF-keys are defined by first using `LF S` and setting parameter `-FC ON,` which adds the current cursor position after the first byte of an FKC-addressed macro. You use FKC-1 addressing. The `LF A` sequence will ask you to enter `0`, `1` or `2` to define how the macro is to be addressed, and you select 1. You then enter the AID byte in ASCII and terminate the macro. The FKC-1 addressing causes transmission of the transmittable fields following your macro.

ASCII/CTL characters can be included in the macro, see `LF X`. It is therefore theoretically possible to use FKC-0 addressing and define the complete 3270-Xmit sequence of AID, cursor position, and then several variable fields, each with cursor position, control byte and content, but this is not really practical.

For GCOS8 applications using 3270 presentation the necessary `'Esc'` is added automatically to the start of FKC-addressed macros.

Send a break command to the host application (LF B)

Several of the mainframe systems you may access have situations where they keep the initiative in a dialogue, and if you want to regain it you are required to transmit a break signal. Please do not confuse this function with the BREAK key on your asynchronous terminal.

Because the host has the initiative, your keyboard is locked for normal typing. However you may send a break to the host with `LF B`, which will normally cause the host to give you the initiative.

You may of course unlock the keyboard with `LF U`, and type normally. A break command may in this case be sent typing:

```
$*$BRK <TRANSMIT>
```

Disconnect from host application (LF D)

To disconnect from a host application, enter the log-off sequence required by the host and transmit. The current session will then be closed by the host system and a 'Disconnected' message will appear on your screen.

If this log-off procedure for some reason fails, you may send a session disconnect request to the host by entering the command `LF D`. Normally you will then receive a disconnect confirmation message. Or you may type manually:

```
$*$DIS <Xmit>
```

Execute command (LF E)

This command allows you to execute a UNIX command without exiting from *Host Links*. The following variations are available:

LF E COMMAND	Disconnect video, show/execute, wait for CR
LF E *COMMAND	Disconnect, cursor to 24/1, execute, no wait
LF E **COMMAND	No disconnect, execute, refresh variables
LF E ***COMMAND	No disconnect, execute, no refresh
LF E ****COMMAND	No disconnect, execute, refresh whole screen

In some products the variations can be produced using the LF N command to set a count prior to LF E. Normally the LF E command clears the screen, displays the command, executes the command, then waits for a CR before continuing. The display and wait may be avoided by preceding the command with an asterisk. Two asterisks should be used when executing commands from the file display that might change the directory. Three asterisks may be used when starting commands that do not update the screen at all. Four are used when starting another *Host Links* product, and cause the first to refresh the screen completely on return.

Your profiles file may inhibit this function (NOEXEC).

Call the Host Links help facility (LF H)

Typing LF H calls the *G&R/Gmenu* help facility. This is an independent system that executes outside of the products. The menu facility navigates through a series of menus and/or information pages, and most *Host Links* products have a set of help menus that can be consulted using the system. Other applications may also use it.

Menus and information pages may be added or changed by the user or system administrator. The kind of information you add can be information about different applications and a description of how to run them. Creating and maintaining the menus and information pages is described in the *G&R/Gmenu* facility manual.

Remember that you may take a hard copy of help pages with the hard copy control keys CTRL/P and CTRL/T, see the Host Links Screen handling section.

G&R/Gmenu is included in the *G&R/Basic* package of utilities bundled with all *Host Links* products.

Enter insert or overwrite mode (LF I, LF O)

Synchronous terminals normally operate in overlay mode, i.e. new characters, typed by you, overwrite the character at the cursor position on the screen.

In *Host Links* emulators you can choose to use insert mode instead, which will place a character at the cursor position and 'push' the character that was there one position to the right. This makes for easier editing in some circumstances.

Note that there are several other functions available using the control keys, all designed to be useful when working with a host editor that works in full-screen mode.

Start/End a key definition (LF K, LF N)

Any control key on the terminal that is visible to the emulator may be redefined as a macro, i.e. one or more other keystrokes. When you try to redefine the key you will be told its symbolic name, and asked to verify that you wish to redefine it, this means that you can safely use LF K to test if a key is visible, and find its name.

If you then continue and redefine the key as a macro the new definition is stored on disk as a file with the symbolic key name, and is available the next time you start the emulator. During macro definition each key is remembered as well as executed until you terminate with LF N.

Thereafter any time you press the control key the macro will be executed. System administrators please note that there are advanced functions available for use in macros; see LF X command.

Your profiles file may inhibit this function (NOMACRO).

Start/End a macro definition (LF M, LF N)

Up to twenty-four different user macros may be defined (as opposed to redefining control keys). `F-KEY` in the text below should be interpreted as one of the function keys `F1` to `F12`, or `SHIFT/F1` to `SHIFT/F12`.

`LF M F-KEY` sets the emulator in macro definition mode. This means that all user input is remembered, as well as being executed. `LF N` terminates the remembered key sequence. The sequence may then be executed at any time by entering `LF F-KEY`. The defined macro is saved in your macro directory with a name of type `M1-M12` or `N1-N12` and is therefore available the next time you start the emulator.

System administrators please note that there are advanced functions available for use in macros; see `LF X` command. Also any control key including `F-KEYS` may be redefined (`LF K`) to send a sequence when pressed (without a preceding `LF`). A macro may be defined and then renamed, and the new name given as the `-MI` parameter to cause it to be executed as an initialization macro when the emulator is started.

Your `profiles` file may inhibit this function (`NOMACRO`).

Call print menu (LF P)

Whenever you execute the `LF P` command, the emulator shows its print menu, which allows you to print screen dumps, print and remove screen dumps or simply remove the screen dumps.

Quit, leave the emulator (LF Q)

This command is used to terminate the emulator. The host log off sequence should be sent and a disconnect confirmation received before entering this command. However, if you leave with a session still active the network software will do an abnormal disconnect to close it. If the emulator is configured with the 'Disconnect Quit' (`-DQ`) parameter then it will always terminate if the host terminates the session or a connect fails. On unexpected disconnects and connection failures the emulator will generate a non-zero return code.

SYSREQ (LF R)

Executes the 5250 System Request command.

Set parameter (LF S)

Parameters are normally set when you start the emulator, either from the command line or preferably from the `<product>.cfg` file, but can be set afterwards by entering the `LF S` command. You will then be asked if the parameter is for the emulator or the line module. You answer `E` or `L` and then enter the parameter. More than one parameter may be entered. The following format should be used:

```
-XX parameter1 -YY parameter2
```

For an overview of the emulator and line module parameters available see the Startup/Configuration section.

Start your editor with screen image (LF T)

The `LF T` command starts the editor you have configured in your *`Host Links`* profile (`DIRTED`), the default is `Gedit`, but the experienced UNIX user may prefer `vi`. The editor workspace is loaded with the content of `screen.sav`, which has the screen image you had when you used the command, and any others saved previously with the video handler `CTRL/P` function. While in the editor you may manipulate the screen image as normal text, merge it with other text files, write it back to a file, print it out and then return to the *`Host Links`* product.

You may of course start any editor with the `LF E` option to get an empty workspace.

Unlock the keyboard (LF U)

Several of the host systems you access may have situations where they keep the initiative in a dialogue. Because the host has the initiative your keyboard is locked for normal typing, however you may send a break to the host with `LF B`, which will normally cause the host to give you the initiative.

You may unlock the keyboard with `LF U`, and type normally. A break command may in this case be sent typing:

```
$*$BRK <TRANSMIT>
```

Write screen and execute print command (LF W)

As noted elsewhere, you may at any time within any *`Host Links`* product use the `CTL/P` command to take a hard copy of your screen image onto the file `screen.sav` on your home directory. You can access this via the `LF P` command, which puts you into the print menu.

However, if your hard copy printing is always done in exactly the same way, then you may use the `LF W` command, which first writes your screen image to the `<product>.sav` file, and then executes a pre-configured command (`-WC` parameter). The command may be any legal command, generally a spooling command to put the image into a queue for printing. Note that rename and delete parameters should be used, so that you may make successive requests, even though the printer is busy.

Include ASCII control character in macro (LF X)

Macro sequences to be sent directly to the host system (`LF A`) may need to include ASCII control characters. These special characters cannot be typed directly into the emulator, since it would interpret them as keyboard functions.

To include one in a macro, use the `LF X` command while in macro definition mode. You will be asked to press the desired control character, BUT WITHOUT HOLDING DOWN THE CONTROL KEY.

ASCII	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
KEY	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

ASCII	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
KEY	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_

Test (LF Z)

Executes the 5250 Test request command.

Execute a defined macro (LF F-key)

You can execute a predefined macro by entering `LF FN`, where `FN` is one of the function keys `F1..F12`, `SHIFT/F1..SHIFT/F12`. The emulator will check for a macro file with a name in the range `M1` to `M12` or `N1` to `N12` depending on the key. Macros are stored in the `<product>_mac` directory under `/usr/gar` (unless overridden with `-MP`) and the emulator looks first in the user's personal macro directory (unless overridden with `-MD`), and then in a directory with name `default`. If you have not defined the macro, an error message will be displayed.

Clear screen (LF Home)

The clear key is not available on some terminals, so `LF HOME` has been defined to duplicate the clear function. For Qsim/V78sim all variable fields are erased. In normal mode this is all data on the screen. For 3270/5250 all data on the screen is erased, and the host is notified.

Reset screen (LF Del)

The reset key is not available on some terminals, so `LF DEL` has been defined to duplicate the reset function. For Qsim/V78sim reset erases all fixed and variable data, cancels all field definitions, and resets tab positions. The emulator is set to normal mode. For 3270/5250 reset unlocks the keyboard and turns off insert mode.

Send page/all (LF Xmit)

The `TRANSMIT` key in Qsim/V78sim causes the emulator to transmit data to the host in a way depending on the mode of the emulator at the time. On a PC with *`G&R/Glink`* the `XMIT` key is `GRAY +`, and on VTnnn terminals it is probably configured as the `ENTER` key on the numeric keypad. In forms mode all variable data is sent from the fields in the form. In normal mode data is transmitted from the point on the screen at which the first character was entered, up to, but not including, the position of the cursor.

While a DKU is in normal mode it is possible to force transmission of all data on the screen, using the 'send page' function. On the DKU this is done using `CTL/XMIT` but in Qsim you use `LF XMIT`.

The VIP7800 terminals have a 'transmit all' function to force transmission of all data on the screen (including field definitions). This function is reached using `LF XMIT`. On the terminal this is done using `Esc t`, and this is supported as an alternative to `LF XMIT`.

Erase last user input (LF Eop)

In some situations where a user fills out some of the fields in a form it can be convenient to be able to erase the data from the last user input, without erasing other variable data filled out by the host or user previously. In Qsim `LF EOL` gives this function.

National character transliteration

IBM hosts use EBCDIC internally while UNIX systems use ASCII. This means that all data characters must be transliterated by G5250 when sent to or received from the host.

Use the –EA parameter to enable correct national language transliteration for your host applications. Some languages are implemented in more than one version, and some experimentation with these may be required for all national characters to be displayed correctly.

Note that Scandinavian users may have traditionally 'fixed' the mapping of the three Scandinavian characters Æ Ø Å between their EBCDIC presentation of # @ $ and their ASCII presentation of [\ ] in which case they will need the `-NA` option to duplicate this in G5250.

Host Links Screen handling

Control key functions

Besides standardizing screen handling, the various terminal specific drivers provide a selection of user-functions. The commands are as follows:

CTRL/F Toggle language key. The mapping to/from the extended character set will be stepped through the available ISO substitution sets. The chosen set is displayed in line 24 of the screen. US ASCII gives no mapping.

CTRL/I Horizontal tab. For terminals with no TAB key.

CTRL/J Line Feed. For terminals with no LF key.

CTRL/K Copy field from screen. The characters between the cursor and the next attribute change on the screen (or the end of the line) are copied.

CTRL/L Redisplay the entire screen.

CTRL/N Select between different function and keypad modes in the VT100 and VT200 handlers.

CTRL/P Hard copy of screen. The screen content is saved on your home directory with file name `screen.sav` in plain text format.

CTRL/U Paste field. The field previously copied with CTRL/K is inserted as keyboard input at the cursor position. You should be at a place where it would be logical for you to type the content of the field you have copied.

CTRL/Z Backtab. For terminals with no backtab key.

Hard copy in video handler

While in any *Host Links* product hard copy of the screen is made with CTRL/P. The screen content is saved on your home directory with file name `screen.sav`.

The screen content is added to any existing data on the `screen.sav` file. That means that while running a remote application you may take a hard copy of each screen image in which you are interested, and thereafter print the `screen.sav` file.

To facilitate the manipulation of the `screen.sav` file, the `LF P` command brings you into a print menu.

Host Links field editing

The *Host Links* system will often require input of a field from the user, rather than a single key. The system uses a common method of editing this user input throughout the system using the following keys:

BACKSPACE	Delete the character to the left of the cursor
DELETE	Delete the character at the cursor position
CR	Finished with input: the whole field is used
ERASE EOF	Delete all characters from the cursor on
INSERT CHAR	Opens a space
CLEAR	Erase the whole field being input
CURSOR LEFT/RIGHT	Move cursor over typed text
TAB/BACK TAB	Move 10 positions (or to end)
CURSOR UP/DOWN	Go to start/end of typed data
LINE FEED	Call context sensitive help for the field

Host Links User profiles

Profiles file structure

User profiles are called profiles and maintained by the *`Host Links`* administrator.

```
/usr/gar/config/default/profiles
/usr/gar/config/$LOGNAME/profiles
/usr/gar/config/system/profiles
```

These files are read one by one in the sequence given above. Suggested commands should be given in `config/default/profiles`, user selected commands in `config/$LOGNAME/profiles` and finally the administrator may use the `config/system/profiles` to override all other settings.

All files are organized in sections, which may apply either to all terminals (Default), a specific user (User), a specific terminal (Terminal), a combination (Termuser), or a specific terminal type (Termtype). Each section is prefaced with a header line:

```
DEFAULT

USER        person.project.mode

TERMINAL    terminal_name

TERMUSER    terminal_name person.project.mode

TERMTYPE    terminal_type
```

Commands may be put into any section to specify the *`Host Links`* environment for a user or terminal. A full explanation of the facility is in the *Host Links Installation and Configuration* manual.

Startup/Configuration

```
g5250 -ta -li tcp -am tn5250 -tm IBM-3179-1
```

'`-PP`' can be a path to a device, file or '`*`' for an attached printer.

Parameters for the G&R emulators and gateways are divided into two categories, parameters for the product itself and parameters for the line handler. Parameters for the product must precede the `-LI` argument that signals that the rest of the parameters are for the line handler module.

The `-USER` and `-HOST` parameters may later be used to switch between product and line parameters. Parameters may be pre-configured on:

```
/usr/gar/config/$LOGNAME/<product>.cfg
```

The `user` (`$LOGNAME`) directory may be replaced with `default` to give a configuration for users with no private directory.

Configuration file (<product>.cfg)

The configuration file is grouped into sections.

`Default`	Parameters apply to all users
`User user(.accnt.mode)`	Parameters apply to this user only
`Terminal terminal-name`	Parameters apply to this terminal only
`TermUser terminal user`	Parameters apply to this combination only
`ContextUser ctx user`	Parameters apply to user in this context

The 'User' identifier may be in the standard star format, or the .accnt.mode part can be omitted altogether. Comment lines are denoted by an asterisk or hash (*, #) in column 1, and may appear anywhere in the file. A section header must appear before the first parameter, but sections may be in any order. Both section headers and parameters can be indented for clarity. The file is processed sequentially and subsequent parameters will override any previous parameters of the same type.

Parameters for the product must be preceded by -USER if any line parameters have been delivered previously, e.g. in the Default section. Line parameters must always be preceded by -LI XXX.

In the following there are line parameters in the Default section, so the parameters in the user sections must be preceded with -USER, and the user's line parameters must be preceded with -LI XXX again.

e.g. /usr/gar/config/default/g5250.cfg

```
Default
    -LI TCP -DN IBM1
ContextUser 1 Peter
    -USER -TA -LI TCP -DN IBM2
ContextUser 2 Peter
    -USER -MD OTHERMAC -LI TCP -DN IBM3
User Mary
    -USER -MD MARYMAC -LI TCP -DN IBM4
```

G5250 parameters

Overview of operating parameters

Defaults in upper case.

Parameter		Description
-CN	on/OFF	Connect automatically to host application (defined by line handler parameters) at startup.
-CRX	on/OFF	The CR key will act as transmit.
-DBG	on/OFF	Turns on an internal trace in G5250.
-DSU	On/OFF	Normally the screen is updated as data arrives from the line, provided the emulator gets time to do it. This parameter inhibits screen update until the host application indicates the end of the data. It may give a more pleasing impression by smoothing the screen update at the cost of waiting for all the data before starting. If the 'terminal' is actually a screen scraping application on another system then this parameter may be important to stop a premature reaction to a pause in the incoming screen image.
-DQ	on/OFF	Disconnect causes G5250 to quit, normal disconnect gives a return code of zero, abnormal disconnect gives a return code equal to the return code from the line. G5250 also quit on failing connection attempts with a non-zero return code.
-EA	INT	EBCDIC-ASCII conversion (default INT)
	INT	International
	UK	English (UK)
	US	English (US)
	AG	Austrian/German
	AG1	Austrian/German (alt)
	BRA	Brazilian
	BE	Belgian
	CFR	Canadian French
	FR	French

Parameter		Description
	ICE	Icelandic
	SFI	Finnish/Swedish
	SF1	Finnish/Swedish (alt)
	DN	Danish/Norwegian
	DN1	Danish/Norwegian (alt)
	IT	Italian
	JAP	Japanese English
	PO	Portuguese
	SP	Spanish
	SPS	Spanish Speaking
	SP1	Spanish (alt)
-LC	ON/off	Lower case is transmitted to the host. Can be turned off so that all data to the host is transmitted in upper case (although it is displayed in lower case on the screen).
-MD	name	Macro directory. The default is the user name in the directory 5250_mac in the /usr/gar directory.
-MI	name	The macro with this name is executed on G5250 startup.
-MP	path	Change the path to the main macro directory, the default is 5250_mac in the /usr/gar directories.
-NA	On/OFF	Norwegian ASCII (default OFF)
-TA	on/OFF	Allow type ahead. User input is collected while the host has the turn.
-WC	'cmd'	The write command is executed after LF W to print the users' g5250.sav, with the path name as the last parameter in the command.
-XX	hxhx	Any incoming character from the host can be translated into any other for display purposes. Both are expressed in hex, and the first becomes the second. This is done after the EBCDIC => ASCII transliteration.

Some line handler parameters

See the *Gline* manual for a complete description of all line handler parameters.

Parameter	Value	Explanation
-LI	TCP	Use TCP/IP.
-AM	TN5250	Use the Telnet/5250 protocol.
-DN	Hostname	Remote host name. Numeric/symbolic IP-addr.
-HO	Hostname	Remote host name. Numeric/symbolic IP-addr.
-DEV	Devicename	The Device name is a symbolic name that can be appended to the terminal type parameter when doing TN5250 connections.
-TM		Terminal type response.
	IBM-3179-2 IBM-3180-3 IBM-3196-A1 IBM-3477-FC IBM-3477-FG IBM-5251-11 IBM-5291-1 IBM-5292-2 IBM-5555-B01 IBM-5555-C01 IBM-3812-1 IBM-5553-B01	When connecting to IBM hosts over TCP/IP, the TN5250 protocol negotiates the terminal type. You must choose one that is acceptable to the IBM TCP/IP front end. The default is IBM-3179-2.

Troubleshooting

If you experience any kind of problem when using an emulator or gateway to access your host application, the product trace file and/or the line handler trace file will provide useful documentation of the problem, for your own use, to the G&R distributor or to G&R if it turns out to be caused by an error in the product. See the appendix *Host Link Trace* for a full discussion of how to generate G&R/Host Links trace files.

Product trace files

UNIX/Linux	`/usr/gar/debug/`**`XXX`**`/`**`YYY`**`.dbg`

XXX = user name
YYY = product identifier g32, g52, qsi, v78, pth

This traces details from internal processing. Enable it by adding `-DBG` to the start-up command or the `<product>.cfg` configuration file:

```
-USER
      -DBG ON
```

Line handler trace files

UNIX/Linux	`/usr/gar/debug/`**`XXX`**`/`**`YYY`**`-gli.dbg`

XXX = user name
YYY = product identifier g32, g52, qsi, v78, pth

This traces details of line handler operation. Enable it by adding one or both of `–D_` and `–S_` to the start-up command or the `<product>.cfg` configuration file:

```
-LI ZZZ
      -S_ -D_
```

(ZZZ =line handler identification, i.e. DSA, DIWS, TCP or X25)

When connecting through Ggate

UNIX/Linux	`/usr/gar/debug/`**`ZZZZ`**`/gga`**`NN-PPPP`**`.dbg`
Windows	`C:\gar\debug\`**`ZZZZ`**`\gga`**`NN-PPPP`**`.dbg`

(ZZZZ = DSA node name, e.g. EN06 or PH13)
(NN =Instance number, starting at 01)
(*PPPP =IP-address of the client system, running the emulator*)

When any G&R or customer application based on GlAPI connects through Ggate to the host application, the line handler trace will be generated on the Ggate system, with the name and location showed in the table above. In this case the product start-up command or `<product>.cfg` file would look like this:

```
    -LI YYY:PPPP
          -S_ -D_
```

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

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
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: Host Links License Keys

All G&R products require a license key to run. If you are a G&R distributor you need a license key from G&R. If you are a customer you should have received the license keys from your distributor together with the software. The licenses are stored in text format in a file named `licenses`.

If `licenses` is delivered with the product files, it is merged with any existing licenses in the configuration directory when you run the install procedure.

UNIX/Linux	`/usr/gar/config/licenses`
Windows server	`C:\gar\config\licenses`

Glicense

The `Glicense` program is included in every software delivery, and it can be used even though no license key is installed. This allows you to create or modify your own licenses from a license card. You must execute `Glicense` from a user-id that has permission to write in the configuration directory (i.e. the Host Links administration user `gar` for Host Links). When executed with no parameters, `Glicense` will check for an existing `licenses` file. If found it will skip directly to the command dialog, but if there is no license then it will prompt you for distributor name, customer name and the main license key. Be careful to type the names and the key exactly as given to you by your distributor. It is important that you respect case and spaces between words.

To tell `Glicense` explicitly where the license file is, or where it should be written, supply the full path as an option. For example:

```
glicense /usr/gar/config/licenses
```

Enter the license information, text and keys, exactly as specified on the supplied license card.

Once the first time installation has been done, you can simply run `Glicense` without any options and it will automatically find the `licenses` file.

When started `Glicense` gives you the following prompt:

```
Enter command or '?':
```

If you enter '?' a list of the available commands is returned:

Use these commands to define/modify and save the `licenses` file.

A = Add	Add a new product to the license file.
D = Delete	Delete a product from the license file.
R = Read	Read in a new license file.
M = Merge	Merge in license keys from a license file
W = Write	Save the license file.
P = Print	Display a list of configured products.
N = New	Create a new license file.
H = Hardware	Change the hardware platform
X = eXpiration	Define an expiration date for all products.
V = Version	Set new version for all products
Q = Quit	Quit the `Glicense` program.
ENTER	The ENTER key quits `Glicense`

If you enter the Print command, the result will be something like this:

```
Enter command or '?': p
Distributor: Bull A/S   Customer: Arbeidsdirektoratet
Product: Basic
Product: Gline
Product: Ggate
Product: Gspool
Product: Qsim
```

License keys

Below you find a complete list of all Host Links and Glink for Java license keys:

License key	Products that require this key
`basic`	All products.
`ggate`	Ggate.
`gspool`	Gspool.
`guft`	GUFT server.
`Guftc`	GUFT client.
`gproxy`	Gproxy.
`qsim`	Qsim.
`v78sim`	V78sim.
`g3270`	G3270.
`g5250`	G5250.
`pthru`	Pthru.
`sdkglapi`	GlAPI SDK.
`Glapi`	GlAPI run-time.
`Telnet`	Enable Ggate Telnet support
`Tnvip`	Enable Ggate TNVIP support
`ssl`	Enables SSL support
`Marben`	Marben OSI Stack for Windows NT 4
`Marben2K`	Marben OSI Stack for Windows 2000
`MarbenXP`	Marben OSI Stack for Windows XP
`Marben03`	Marben OSI Stack for Windows 2003
`Gweb`	Gweb Professional Edition All terminal emulations
`Gljopen`	Glink for Java Open (VTnnn/ANSI/Minitel).
`Gljpro`	Glink for Java Professional Edition.

License key	Products that require this key
`Gljent`	Glink for Java Enterprise Edition.
`Gljall`	Glink for Java All terminal emulations.
`Gljsa`	Glink for Java Standalone (No configuration server).
`Gljsrv`	Glink for Java Server (Configuration and License).
`Gljcnx`	Glink for Java Gconnect (J2EE connector)
`Gljdsa`	Glink for Java DGA (native DSA comms stack)
`Gljggen`	Glink for Java Gargen

Appendix: 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 behaviour.

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. OSI Transport, or 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 or DIWS)
(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_
gli-gli.diw	DIWS 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 debug	`-dbg`
`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_