

HOST LINKS



GSFTP

TM

***Gateway
between FTP
and SFTP***

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



Microsoft, Windows, MS, MS-DOS are registered trademarks of Microsoft Corp.
IBM and PC are registered trademarks of IBM Corp.
UNIX is a registered trademark in the United States and other
countries, licensed exclusively through X/Open Company, Ltd.

Any other product names are trademarks of their respective owners.

Version 6.6
© Gallagher & Robertson as 1990-2013
All Rights Reserved

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 Gsftp | 1 |
| Installation | 1 |
| Host Links Product Overview | 1 |
| Terminal environment | 1 |
| Server environment | 2 |
| Scope of the product..... | 3 |
| Prerequisites | 3 |
| Run-time licenses | 3 |
| | |
| Gftp – the FTP server | 5 |
| Overview | 5 |
| Command line parameters | 5 |
| Server start-up | 7 |
| Supported FTP commands | 7 |
| | |
| Gsftp – the SFTP server | 9 |
| Overview | 9 |
| SSH configuration | 9 |
| Command-line parameters..... | 10 |
| Logging on to the FTP host | 11 |
| Supported SFTP requests | 12 |
| | |
| The custom SSH program | 13 |
| OpenSSL | 13 |
| OpenSSH..... | 13 |
| Applying patches..... | 15 |
| Patch description | 15 |
| Using the customized program..... | 16 |
| | |
| Appendix: Host Links Manuals | 17 |

Contents

Host Links Gsftp

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 network. It avoids all need for Front-ends (MainWay/Datanet) 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/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.

The gateway consists of two programs:

- `gftp` which provides an FTP server and an SFTP client
- `gsftp` which provides an SFTP server and an FTP client

In addition, a custom-made copy of SSH is used (see page 13).

Prerequisites

G&R/Gsftp is only available on UNIX/Linux.

A functioning installation of OpenSSH (see <http://www.openssh.com/>) is needed to run Gsftp.

Functioning installations of both OpenSSL (see <http://www.openssl.org/>) and OpenSSH are needed to run Gftp. A C compiler must also be available to compile the Gftp-specific version of SSH (see page 13).

Run-time licenses

In order to run Gsftp, the following license keys must be present in your `/usr/gar/config/licenses` file:

| | |
|--------------------|----------------------------------|
| <code>basic</code> | For the base G&R run-time system |
| <code>gsftp</code> | For Gsftp |

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 how many simultaneous sessions the product is licensed for. If a limitation is specified in the license, only the licensed number of sessions can be active at any time.

Gftp – the FTP server

Overview

The G&R/Gftp program is a server to which FTP clients can connect. The program will require the client to supply a user name, a password and a host name – this information is used to establish an SFTP connection to the SFTP host.

Command line parameters

Gftp accepts a number of parameters.

| Parameter | Description |
|--|---|
| <code>-listen</code> <code>[host][:port]</code> | Defines the IP address and port number the server will listen on. Default: all IP interfaces on the current system, and the ftp port, number 21 |
| <code>-gssh path-to-special-SSH</code> | Defines the path to the specially compiled SSH program (see page 13) |
| <code>-mi Mode-Id</code> | Defines the Mode Id for this server. Default value: def |
| <code>-pid path</code> | Defines the path to the file where the program PID is stored. Default: the file <code>pid.MID</code> where MID is the server's Mode Id |
| <code>-sshopt string</code> | Defines additional parameters for the ssh task. Default: None. Example: <code>-sshopt "-v -v"</code> which adds verbosity (for debug purposes). |
| <code>-min NNN</code> <code>-max NNN</code> | Defines the lowest and/or highest port number to be used when Gftp prepares connections used with Passive data transfers. Default port range: 1024-65536 |

| Parameter | Description |
|--------------------|---|
| -dbg <i>path</i> | Defines the path of a file where run-time debug information is stored. Default: none. The process id (pid) is appended to the <i>path</i> when the file is created. |
| -dbgmax <i>LEN</i> | Defines the max length of a buffer written to the debug file, if active. Default: 0, indicating that the complete buffer is dumped on the debug file. |
| -dbgopt <i>OPT</i> | <p>Defines what should be logged. The <i>OPT</i> parameter is the sum of the following values:</p> <ul style="list-style-type: none"> 1 - Sftp protocol 2 - Ftp command channel 4 - Ftp data channel 8 - Force flush of debug data to disk <p>Default setting: 15 (all options enabled)</p> |
| -to <i>NNN</i> | Defines the time-out value. A connection will automatically be terminated when it has been idle for <i>NNN</i> seconds. Default value: 0, indicating no timeout. |
| -acks <i>NNN</i> | Defines the max number of SFTP status (acknowledgement) packets that can be left unprocessed during file transfers, to reduce file transfer time. Default value: 128. A value of 0 disables this feature. |
| -fips <i>VALUE</i> | Defines the value for the OPENSSH_FIPS environment variable passed to the ssh program (see the -gssh parameter). If the ssh program is Fips capable, setting this parameter to 1 will cause a Fips compliant session. Default: not set. |

Server start-up

When you start the Gsftp server it automatically forks into the background. The current directory is changed to `/usr/gar/servers/gsftp/`, and the server writes its pid in the pid file.

To stop the server you normally execute the command

```
kill $(cat /usr/gar/servers/gsftp/pid.def)
```

The server logs its activities onto a file named `_logfile.def`.

If the server is set up to use a privileged port (a port numbered lower than 1024), it must be started by root. Otherwise, a user account can be used.

Supported FTP commands

Gsftp supports a subset of the FTP protocol as defined in [RFC 959](#).

The following FTP client commands are supported:

| Command | Description |
|--------------|---|
| USER | Supply the SSH login name |
| PASS | Supply the SSH password |
| ACCT | Supply the SFTP host name and options (see below) |
| SYST | Request FTP server version |
| PWD/CWD/CDUP | Print/change working directory |
| MKD/RMD | Create/delete directory |
| RNFR/RNTO | Rename file or directory |
| DELE | Delete file |
| LIST/NLST | Show directory formatted/unformatted |
| TYPE | Set transfer type (Ascii or Binary) |

Gsftp

| Command | Description |
|-----------|--------------------------------------|
| PORT/PASV | Set transfer method (active/passive) |
| STOR | Store file |
| RETR | Retrieve file |
| NOOP | Null operation |

Only the four commands at the top of the table (USER, PASS, ACCT and SYST) can be used until the SSH connection is established.

The SSH connection will be established when USER, PASS and ACCT have been supplied by the client.

The format of the ACCT parameter is `host[:[port]][:[options]]`, where

- `host` is the DNS name or IP address of a machine running an SFTP server
- `port` is the port number (default 22)
- `options` is a subset of the command line parameters, where you can use the parameters `-dbg`, `-dbgmax`, `-dbgopt`, `-sshopt`, `-to`, `-min` and `-max`

Gsftp – the SFTP server

Overview

The G&R/Gsftp program is a server to which SFTP clients can connect. The program will examine the file names supplied by the SFTP client, and will regard all file names containing a colon (:) character as a special log-on path. The information in this special path is extracted (a user name, a password and a host name) and used to establish an FTP connection to an FTP host.

Gsftp supports [version 3 of the SFTP protocol](#).

SSH configuration

The Gsftp server is started by the system's SSH daemon. The `sshd` configuration file (normally `/etc/ssh/sshd_config`) defines which actions are to be taken when incoming SSH sessions are tagged as using the SFTP protocol.

To make Gsftp the program that is activated as the SFTP server, edit the `sshd` configuration file by changing the `Subsystem sftp` line to

```
Subsystem sftp /etc/ssh/gar_sftp
```

Because the `sshd_config` file does not accept parameters in `Subsystem` directives, the Gsftp program must be launched via a shell script; in this example: `/etc/ssh/gar_sftp`. Typical content of this shell script is:

```
/usr/gar/bin/gsftp [command-line-parameters]
```

When an SSH connection marked as an SFTP Subsystem is accepted by the systems' `sshd` daemon, the Gsftp server is started and the current directory is changed to `/usr/gar/servers/gsftp/`.

The server logs its activity in a file named `_logfile.def`.

Command-line parameters

Gsftp accepts a number of parameters.

| Parameter | Description |
|------------------------------------|---|
| -du <i>USER</i> | Default user name to be sent to the FTP server. No initial value. |
| -d? <i>PASS</i> | Default password to be sent to the FTP server. No initial value. |
| -da <i>ACCT</i> | Default account to be sent to the FTP server. No initial value. |
| -dh <i>HOST</i> | Default host name or IP address of the FTP server. Initial value: localhost. If the HOST parameter is a single period (.) it means that the ftp host has the same IP address as the one the SFTP client connected to. |
| -dp <i>PORT</i> | Default port number for the FTP server. Initial value: 21 |
| -ascii -binary | Default file transfer mode. Initial value: Depends on type of FTP host; for Gcos8 hosts: ascii, for all others: binary. |
| -pasv +pasv | Disables (-pasv) or enables (+pasv) use of passive mode. Initial value: Passive mode enabled. |
| -mi <i>Mode-Id</i> | Defines the Mode Id for this server. Default value: def |
| -min <i>NNN</i> -max <i>NNN</i> | Defines the lowest and/or highest port number to be used when Gsftp prepares connections used with Passive data transfers. Default port range: 1024-65536 |
| -dbg <i>path</i> | Defines the path of a file where run-time debug information is stored. Default: none. The process id (pid) is appended to the path when the file is created. |

| Parameter | Description |
|--------------------------|--|
| <code>-dbgmax LEN</code> | Defines the max length of a buffer written to the debug file, if active. Default: 0, indicating that the complete buffer is demped on the debug file. |
| <code>-dbgopt OPT</code> | Defines what should be logged. The OPT parameter is the sum of the following values: <ul style="list-style-type: none"> 1 - Sftp protocol 2 - Ftp command channel 4 - Ftp data channel 8 - Force flush of debug data to disk Default setting: 15 (all options enabled) |
| <code>-to NNN</code> | Defines the time-out value. A connection will automatically be terminated when it has been idle for NNN seconds. Default value: 0, indicating no timeout. |

Logging on to the FTP host

Gsftp will examine the file names supplied by the SFTP client, and will regard all file names containing at least one colon (:) character as a special logon path. Until the client has supplied log-on information, the Gsftp server will return a negative status for all requests.

The format of the logon path is as follows:

```
username:password:account:ftphost:ftpport:options
```

The elements of the logon path are:

- `username` is the FTP logon user name. Default value: see `-DU`.
- `password` is the FTP logon password. Default value: see `-D?`.
- `account` is the FTP logon account. Default value: see `-DA`.
- `ftphost` is the FTP host name or IP. Default value: see `-DH`.
- `ftpport` is the FTP host port number. Default value: see `-DP`.
- `options` is a subset of the command line parameters, where you can use the parameters `-ascii`, `-binary`, `-dbg`, `-dbgmax`, `-dbgopt`, `-to`, `-min` and `-max`.

Gsftp

All elements are optional. Default values are defined using Gsftp command-line parameters. The simplest command the can lead to a successful FTP server logon is

```
cd :
```

Gsftp will examine all file names supplied from the client on all supported SFTP requests. If a logon path is found, but the client is already logged on the FTP server, the logon info is silently stripped from the path.

Supported SFTP requests

The following SFTP client requests are supported:

| Command | Description |
|----------------|---|
| INIT, VERSION | Initial protocol handshake, version set to 3. |
| REALPATH | Expand relative path name |
| OPEN, OPENDIR | Open file or directory |
| READDIR | Get directory contents |
| READ, WRITE | Read from or write to a file |
| CLOSE | Close file or directory |
| REMOVE | Delete file |
| MKDIR, RMDIR | Create or remove directory |
| STAT, LSTAT | Get details of file or directory |
| RENAME | Rename file or directory |

Note: The SFTP protocol extension mechanism (SSH_FXP_EXTENDED) is neither used nor supported.

The custom SSH program

An SSH program is started by Gsftp to be used as an encrypted communication tunnel between Gsftp and the target SFTP server.

Unfortunately, the SSH program always asks for the connection password from the terminal. Since Gsftp needs to pass the SSH connection password (which is supplied in the ftp client's `PASS` command) programmatically, the SSH program has to be modified to support this. The modified SSH should be stored on disk under a different name, normally `/usr/gar/bin/garssh`, and the path to this program should be supplied to Gsftp in the `-gssh` parameter (see page 5).

OpenSSL

As SSH uses the OpenSSL libraries, OpenSSL must be installed on the system. If OpenSSL is already installed you can skip this step.

The software can be downloaded from www.openssl.org. A typical command to compile and install OpenSSL is something like this:

```
./Configure no-asm no-shared -no-dso
make
make install
```

OpenSSH

Gsftp should be compatible with any version of OpenSSH. Releases that are known to work are 4.6 and 4.7. If SSH is already installed on your system, the customized SSH program should be the same release.

You must download the portable OpenSSH source from the OpenSSH website at <http://www.openssh.com/portable.html> and unpack the files.

Gsftp

First compile the software without modification. A typical command is this:

```
export LDFLAGS="-L/openssl"  
export CPPFLAGS="-I/openssl/include"  
./configure --prefix=/usr --libexecdir=/usr/sbin  
make
```

The resulting `ssh` program should at this point work exactly as the preinstalled program.

Note:

- The path `/openssl` is the directory where the OpenSSL libraries (`lib-ssl.a`, `libcrypto.a`) reside
- The path `/openssl/directory` is the directory where the OpenSSL header files (i.e. `openssl/conf.h`) reside
- The path `/usr/sbin` is the directory where the OpenSSH header files (i.e. `openssl/conf.h`) reside

Applying patches

A Gsftp delivery includes a pre-created patch file that can be used to apply the necessary patches to the SSH source. The file can be used as an input file to the Unix patch utility, and resides in `/usr/gar/install/garssh.diff`. To use this file:

```
cd /path/to/openssh-4.6p1
patch -i /usr/gar/install/garssh.diff -b
```

This should make the necessary changes in the files involved (`readpass.c` and `sshconnect.c`). The original (unpatched) files will be renamed with a `.orig` file name extension.

Patch description

If, for some reason, you cannot use the patch utility as outlined above, you must manually patch the two files in question. This is described here.

There is one change to the `readpass.c` source file. A diff between the original and the patched version looks like this:

```
*****
*** 113,124 ****
--- 113,126 ----
char *
read_passphrase(const char *prompt, int flags)
{
    char *askpass = NULL, *ret, buf[1024];
    int rppflags, use_askpass = 0, ttyfd;

+   if (NULL != (askpass = getenv("SSH_PASSPHRASE")))
+       return xstrdup(askpass);
    rppflags = (flags & RP_ECHO) ? RPP_ECHO_ON : RPP_ECHO_OFF;
    if (flags & RP_USE_ASKPASS)
        use_askpass = 1;
```

In other words; the lines:

```
if (NULL != (askpass = getenv("SSH_PASSPHRASE")))
    return xstrdup(askpass);
```

should be inserted as the initial statements in the `read_passphrase` function.

Gsftp

There is also a change to the `sshconnect.c` source file. A diff between the original and the patched version looks like this:

```
*** 500,513 ****
--- 500,514 ----
    static int
    confirm(const char *prompt)
    {
        const char *msg, *again = "Please type 'yes' or 'no': ";
        char *p;
        int ret = -1;

+   if (getenv("SSH_PASSPHRASE")) return 1;
        if (options.batch_mode)
            return 0;
```

In other words; the line:

```
if (getenv("SSH_PASSPHRASE")) return 1;
```

should be inserted as the initial statement in the `confirm` function.

Using the customized program

After the patches have been applied, either using the `patch` utility or manually, the source must be recompiled and the customized SSH utility must be linked. A new `make` in the `OpenSSH` directory will create a new version of the `ssh` executable. Copy this file to the location indicated by the `-gssh` parameter to the `gftp` program.

Gftp will now be able to supply the password to the SSH connection.

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 |
| GIAPI | 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 |

