

How to Enable SFTP on MPE

an Allegro Consultants Whitepaper

Let's begin this discussion with some acknowledgements and a list of prerequisites.

That we can run SFTP on MPE is due to the outstanding efforts of folks like Mark Klein, Mark Bixby and Ken Hirsch and others. Their skill and contributions have improved MPE beyond measure.

The prerequisites include:

- Running MPE 6.5 or better.
- Your system is patched as well as possible – especially for networking!
- Having DNS configured. This means `/etc/resolv.conf` (aka `RESLVCNF.NET.SYS`) is properly configured or `/etc/hosts` (aka `HOSTS.NET.SYS`) has been modified so that your 3000 can resolve the target system by name.

Most of the “what to type” examples presented below can be simply copy-n-pasted from this document into your 3000 session. Use this hint to your advantage 😊

On Allegro's web server, you can obtain the following files:

- `sftpstd.wrq` – a store-to-disc file in Reflection “Labels” format which contains
 - `openssh 3.7.1p2`
 - `openssl 0.9.8e`
 - `gzip 1.2.4`
- `PERLSTD.gz` – another store-to-disc file that has been gzip'ed which contains
 - Perl 5.8.8 along with...
 - ...many helpful modules included but especially the interface to the SHA-1 encryption algorithm.

One more thing – **this paper shows how to enable *out-bound* sftp only (from your 3000 to some other server (called `<remotesys>` in the rest of this paper)) because that's all the MPE systems support.** MPE system cannot do *in-bound* sftp file transfers!

“Installing” SFTP

The good news is – there is no actual install! Rather, it is just a simple restore. ☺

The SFTPSTD file looks like this on an MPE system:

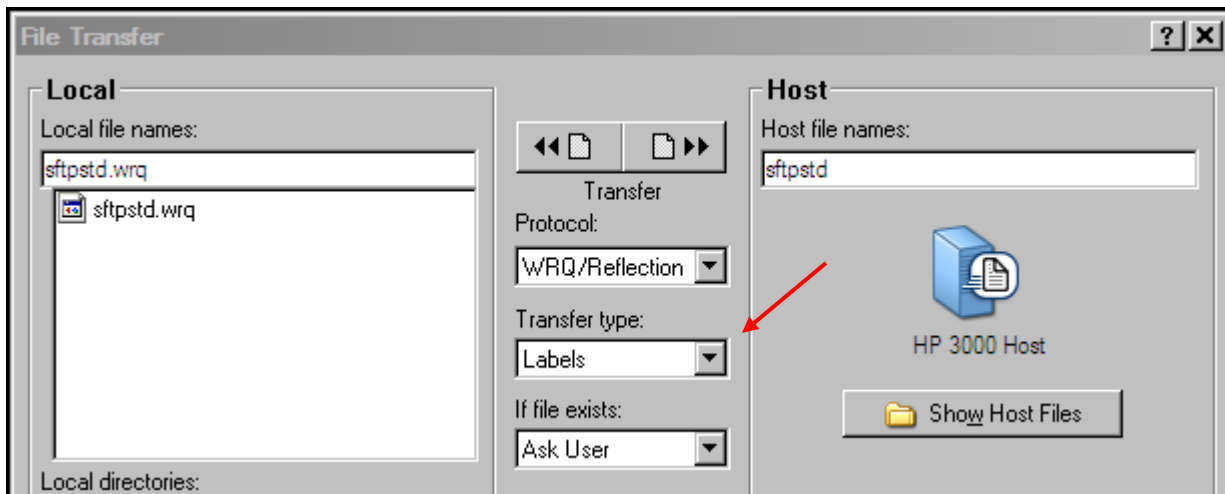
```
:listfile sftpstd,2
ACCOUNT=  SYS          GROUP=  PUB

FILENAME  CODE  -----LOGICAL RECORD-----  ----SPACE----
          SIZE  TYP          EOF          LIMIT R/B  SECTORS #X MX
SFTPSTD   STORE  128W  FB          229363   16776959   1   229376  *  *
```

When this file is on (finally) on your system, it should look similar.

Start by logging on as **MANAGER.SYS,PUB**.

When you upload SFTPSTD to your system, the Reflection File Transfer window should look something like the following:



If you don't have Reflection, please let us know!

Use the following commands to restore the files:

```
:file sftpstd;dev=disc
:restore *sftpstd;/;show;create
```

You'll see...

```
RESTORE *sftpstd;/;/SHOW;CREATE

WED, DEC 14, 2011, 12:23 PM

'CREATE' OPTION DEFAULTING TO 'CREATE=ACCOUNT, GROUP, CREATOR, PATH'
BASED ON YOUR CAPABILITIES (S/R 502)
RESTORE CREATED ACCOUNT "OPENSSL "
RESTORE CREATED GROUP "A0100 " IN ACCOUNT "OPENSSL "

WILL RESTORE      122 HFS DIRECTORIES; HFS DIRECTORIES ON MEDIA      122
WILL RESTORE      148 SYMBOLIC LINKS ; SYMBOLIC LINKS ON MEDIA      148
WILL RESTORE      2342 FILES ; NUMBER OF FILES ON MEDIA      2342

VOLUME RESTRICTIONS          SECTORS CODE  MEDIA FILENAME
DISC                          :C       16          1 /OPENSSL/A0100/.profile
DISC                          :C       32          1 /OPENSSL/A0100/INSTALL
DISC                          :C       32          1 /OPENSSL/A0100/README
DISC                          :C         0 HFSDIR    1 /OPENSSL/A0100/bin/
DISC                          :C       16          1 /OPENSSL/A0100/bin/c_rehash
DISC                          :C      7872 NMPRG     1 /OPENSSL/A0100/bin/openssl
DISC                          :C         0 HFSDIR    1 /OPENSSL/A0100/certs/
<<snip>>
```

The restore will create one new account – OPENSSL – and one new user – MGR.OPENSSL. Secure these logons with passwords as necessary.

Create the following links:

```
:newlink /usr/local/bin/gunzip,/usr/local/bin/gzip
:newlink /usr/local/bin/sftp,/opt/openssh-3.7.1p2/sftp
:newlink /usr/local/bin/ssh,/opt/openssh-3.7.1p2/ssh
:newlink /usr/local/bin/ssh-keygen,/opt/openssh-3.7.1p2/ssh-keygen
:newlink /usr/local/libexec/ssh-rand-helper,/opt/openssh-3.7.1p2/ssh-rand-helper
:newlink egdctl,/opt/openssh-3.7.1p2/EGDCTL
```

Regarding the last two newlink commands...

1. You may not have a `/usr/local/libexec` directory. If newlink fails because the directory is missing, do the following:

```
shell/iX> cd /usr/local
shell/iX> mkdir libexec
shell/iX> chmod 755 libexec
```

Do an “ll” to confirm and then redo the newlink command.¹

```
drwxr-xr-x  3 MANAGER.SYS      SYS          416 Dec 13 08:39 libexec
```

2. Presuming you are logged on as **MANAGER.SYS,PUB**, then “egdctl” will be created in the PUB group. You may want to put this in a more secure location.

¹“ll” (ell-ell) is a common alias for the shell command “ls -l” Most people have this alias configured in the shell. If “ll” isn’t working. Check for the following – 1) typing error, 2) that the shell is being run properly (`run sh.hpbin.sys;info="-L"`) and 3) that “ll” is a predefined alias in `/etc/profile`.

"Installing" Perl

The good news is – this is just a restore as well. The not-so-good news is – the Perl file is really big. So, getting it onto your 3000 will be a bit more challenging.

Once the file is downloaded from Allegro to your PC, we recommend using FTP to transfer it to your MPE system. In order for FTP to your 3000 to work, JINETD must be running. Open a 'cmd' window on your PC and 'cd', as necessary, to the folder where PERLSTD.gz resides and start ftp:

```
C:\Users\myself\Documents>cd ..\Downloads
```

```
C:\Users\myself\Downloads>dir PER*
Volume in drive C has no label.
Volume Serial Number is B867-3BED
```

```
Directory of C:\Users\myself\Downloads
```

```
07/17/2013  12:10 PM           39,940,648 PERLSTD.gz
                1 File(s)          39,940,648 bytes
                0 Dir(s)  158,407,720,960 bytes free
```

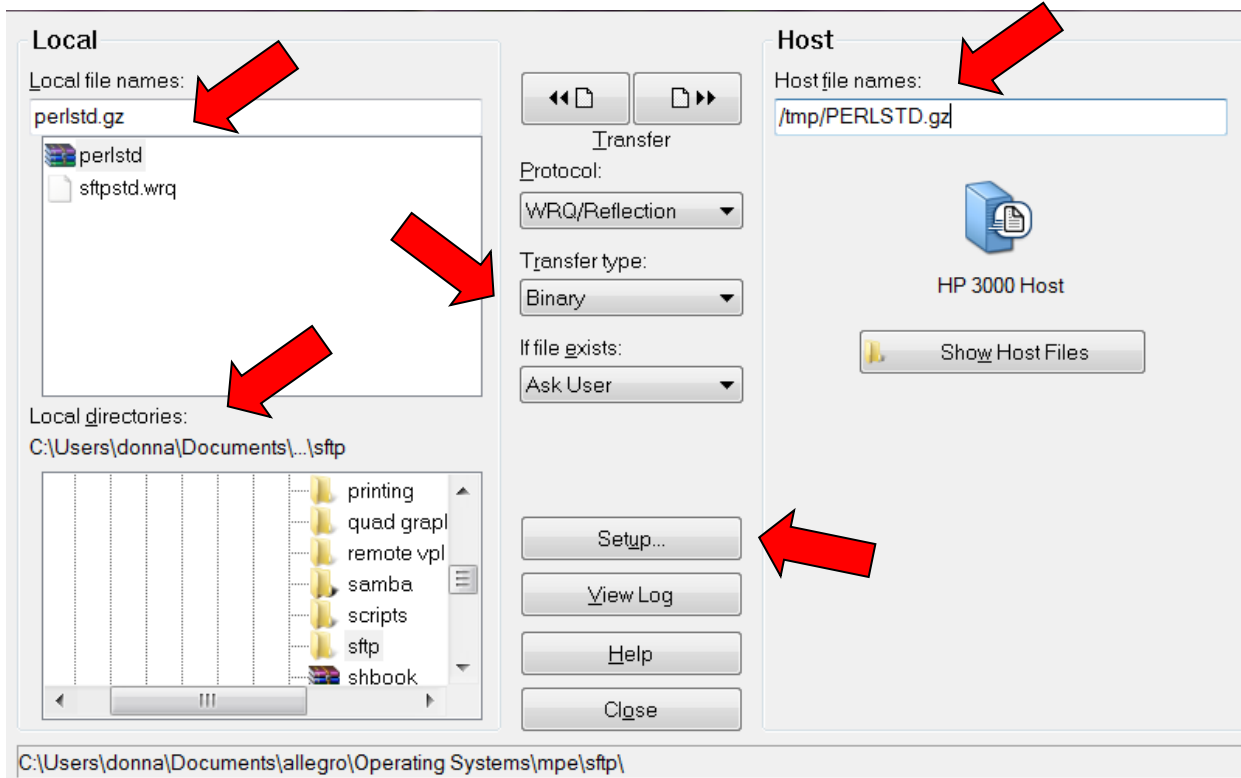
```
C:\Users\myself\Downloads>ftp my3000
```

```
Connected to my3000.allegro.com.
220 HP ARPA FTP Server [A0012I16] (C) Hewlett-Packard Co. 2000 [PASV SUPPORT]
User (my3000.allegro.com:(none)): manager.sys
331 Password required for MANAGER.SYS.  Syntax: userpass
Password:
230 User logged on
ftp> bin
ftp> put PERLSTD.gz /tmp/PERLSTD.gz;rec=-1,,b,ascii;disc=2147483647
ftp> quit
```

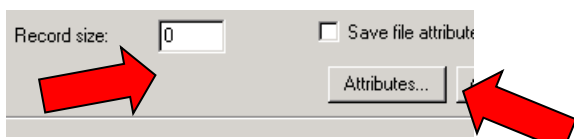
The command format for 'ftp' is ftp [host name or IP address]. If you have both an account and user password set for manager.sys, enter it as [account password],[user password].

If FTP doesn't work as shown above due to needing "passive" enabled. You can use Refection's FTP Client (which supports "passive"). If you need help with this, please let us know.

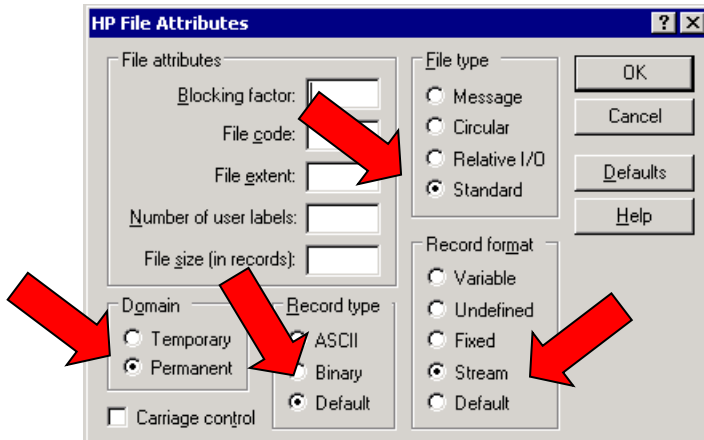
If you can only use Reflection to transfer the Perl file, here are the steps:



1. Navigate to the "Local Directory" where you saved perlstd.gz.
2. Fill-in the "Local file name" field with perlstd.gz
3. Change the Transfer type to Binary
4. Fill-in the "Host file name" field with /tmp/PERLSTD.gz
5. Click on the Setup button to make more changes...



1. Make sure the "Record size" is zero (0).
2. Click on the Attributes button to make more changes...



1. Set the File type to Standard
2. Set the Domain to Permanent
3. The Record type should be Default
4. The Record format should be stream

With all the Reflection “tweaks” made, click the OK button to back up to the main transfer screen, then click on the right-pointing File transfer button and go find something to do...

Again, if you don’t have Reflection, please let us know!

Now with the file on the 3000, the first step is to unzip it. Go into the shell and...

```
shell/iX> cd /tmp
shell/iX> ll
-rw----- 1 MANAGER.SYS 0 39940648 Jul 18 10:24 PERLSTD.gz
shell/iX> gunzip PERLSTD.gz
shell/iX> exit
```

(If the shell has been invoked properly, /usr/local/bin/ should be in your path.)

Next, rename it into MPE namespace and check it (your EOF value may be different):

```
:rename /tmp/PERLSTD,perlstd
:listfile perlstd,2
ACCOUNT= SYS          GROUP= PUB

FILENAME  CODE  -----LOGICAL RECORD-----  ----SPACE----
          SIZE  TYP      EOF      LIMIT R/B  SECTORS #X MX
PERLSTD   1B   BA   238818304 2147483647  1   932896  *  *
```

Finally, restore the files:

```
:file perlstd;dev=disc
:restore *perlstd;/;show;create
```

You'll see...

```
RESTORE *perlstd;/;SHOW;CREATE
```

```
WED, DEC 14, 2011, 1:49 PM
```

```
'CREATE' OPTION DEFAULTING TO 'CREATE=ACCOUNT, GROUP, CREATOR, PATH'
BASED ON YOUR CAPABILITIES (S/R 502)
RESTORE CREATED ACCOUNT "PERL"
```

```
WILL RESTORE      1255 HFS DIRECTORIES; HFS DIRECTORIES ON MEDIA      1255
WILL RESTORE              5 SYMBOLIC LINKS ; SYMBOLIC LINKS ON MEDIA      5
WILL RESTORE      7529 FILES ; NUMBER OF FILES ON MEDIA      7529
```

```
VOLUME RESTRICTIONS          SECTORS CODE  MEDIA FILENAME
DISC                          :C          0 HFSDIR    1 /PERL/PUB/.cpan/
DISC                          :C          0 HFSDIR    1 /PERL/PUB/.cpan/Bundle/
<<snip>>
```

Because the Perl account was created with default capabilities, you may need to add “PH” capability to the PERL account and PUB group:

```
:altacct perl;cap=+PH
:altgroup pub.perl;cap=+PH
```

Additionally, you may wish to secure the Perl account with passwords.

Test Perl to make sure it runs:

```
:perl.pub.perl "-v"
```

```
This is perl, v5.8.8 built for PA-RISC1.1
```

Set up a link for Perl:

```
:newlink /usr/local/bin/perl,/PERL/PUB/PERL
```


Step to make SFTP work

Finally, we're to the point of actually trying to use SFTP!

First, the EGDJOB needs to be started. Do this by typing:

```
:egdctl start
```

This job needs to remain running in order to make SFTP to work properly. In most cases, this means that the job should be started when the system comes up and halted when the system is being brought down.

Cleverly enough – `:egdctl stop` – will stop the job 😊

Next, add “PH” capability to the target user and account (this is the user.account that will be initiating your SFTPs) if they don't already have it:

```
:altacct donna;cap=+ph  
:altuser mgr.donna;cap=+ph
```

(This only needs to be done once, of course.)

Log on as the user that will be doing the SFTPs. You cannot create one set of “id_rsa” files and copy them to different locations on your MPE system and expect SFTP to work from a different logon. Each user that will be doing SFTP needs to have their own unique “id_rsa” files. Additionally, “id_rsa” files cannot be edited. “ssh-keygen” is easy enough to run...just do it!

Go into the shell and run “ssh-keygen” without any parameters just to make sure it works. You’ll be shown the help text for the program:²

```
shell/iX> ssh-keygen
You must specify a key type (-t).
Usage: ssh-keygen [options]
Options:
  -b bits      Number of bits in the key to create.
  -c           Change comment in private and public key files.
  -e           Convert OpenSSH to IETF SECSH key file.
  -f filename  Filename of the key file.
  -g           Use generic DNS resource record format.
  -i           Convert IETF SECSH to OpenSSH key file.
  -l           Show fingerprint of key file.
  -p           Change passphrase of private key file.
  -q           Quiet.
  -y           Read private key file and print public key.
  -t type      Specify type of key to create.
  -B           Show bubblebabble digest of key file.
  -C comment  Provide new comment.
  -N phrase    Provide new passphrase.
  -P phrase    Provide old passphrase.
  -G file      Generate candidates for DH-GEX moduli
  -T file      Screen candidates for DH-GEX moduli
```

² If the shell has been invoked properly, `/usr/local/bin/` should be in your path.

The first step is to generate a key file. In this example, the passphrase (-N "") is blank. You may have different, local requirements regarding the passphrase. *Note: the passphrase is **not** a password.*

```
shell/iX> ssh-keygen -t rsa -N ""
Generating public/private rsa key pair.
Enter file in which to save the key (/DONNA/PUB/.ssh/id_rsa): <return>
Created directory '/DONNA/PUB/.ssh'.
Your identification has been saved in /DONNA/PUB/.ssh/id_rsa.
Your public key has been saved in /DONNA/PUB/.ssh/id_rsa.pub.
The key fingerprint is:
23:aa:e1:78:7a:bc:5c:64:9a:ae:74:b8:91:71:ad:48 MGR.DONNA@mpesys
```

Let's see what we did!

```
shell/iX> ll -d .*
total 2
-rw-rw----  1 MGR.DONNA      DONNA      137 Dec 14 14:14 .sh_history
drwx-----  2 MGR.DONNA      DONNA      416 Dec 14 14:13 .ssh
```

(You may see more than just these two "dot" files. You **must** see the ".ssh" directory.)

```
shell/iX> ll .ssh/
total 3
-rw-----  1 MGR.DONNA      DONNA      883 Dec 14 14:13 id_rsa
-rw-r-----  1 MGR.DONNA      DONNA      228 Dec 14 14:13 id_rsa.pub
```

If the read-write-execute (rwx) permissions for .ssh/ is not "drwx-----", correct it by doing:

```
shell/iX> chmod 700 .ssh/
```

If the permissions for id_rsa.pub is not "-rw-r-----", correct it by doing:

```
shell/iX> chmod 640 .ssh/id_rsa.pub
```

At this point you can test SFTP by doing:

```
shell/iX> sftp someuser@remotesys
Connecting to remotesys...
The authenticity of host 'remotesys (192.168.1.100)' can't be established. RSA key
fingerprint is b1:60:dd:0e:2d:ab:4f:be:67:ff:cf:7b:b6:19:d8:7d.
Are you sure you want to continue connecting (yes/no)?
y
e
s
```

```
Warning: Permanently added 'remotesys,192.168.1.100' (RSA) to the list of known
hosts.
```

```
someuser@remotesys's password:
```

Things to note:

- You run SFTP by specifying “<someuser>@<remotesys>” on the command line – that’s a change from regular FTP.
- The “authenticity of host” message is normal the *first time* you connect.
- You have to type “yes” (and <return>) to the “are you sure” question. And, yes – each letter will appear on a new line. (It’s weird but it’s OK.)
- The known hosts warning is also normal and **necessary!**
- If you provide the password, each character of the password will also generate a line feed.

You can actually stop at this point. (Yeah!)

However, one of the most desirable features of SFTP is the ability to do password-less transfers. This is accomplished by copying one of the key (rsa) files to <remotesys>. To do that, use the following steps.

On <remotesys>, confirm that the `.ssh/` directory in *someuser’s* **home directory** exists and security for the directory is set to 700 and that it’s owed by *someuser*. If necessary, make the directory then `chown` and `chmod` it.

Presuming, you’ve completed your SFTP logon to <remotesys> put `id_rsa.pub` onto <remotesys>:

```
Remote working directory is /home/someuser
sftp> cd .ssh
Remote directory is now /home/someuser/.ssh
sftp> put .ssh/id_rsa.pub id_rsa.pub
local:id_rsa.pub => remote:/home/someuser/.ssh/id_rsa.pub
sftp> exit
```

If you cannot use SFTP to send `id_rsa.pub` to <remotesys> there are other ways of doing this. Downloading the file to your PC (via Reflection) so that you can email it, or even plain FTP will work.

If there is already an `id_rsa.pub` on <remotesys>, *your* rsa file can be named differently. For example, it can be name **`id_rsa.mpe`**. If you do this, the following steps will have to be adapted accordingly.

Once the file is on <remotesys>, look to see what's in .ssh/. You should see something like the following:

```
[someuser@remotesys ~]$ ll .ssh/
total 4
-rw-rw-r-- 1 someuser user 228 May 24 10:15 authorized_keys
-rw-rw-r-- 1 someuser user 228 Dec 14 13:46 id_rsa.pub
```

→If there is already an authorized_keys file present, id_rsa.pub needs to be appended to it:

```
[someuser@remotesys ~]$ cd .ssh/
[someuser@remotesys ~]$ cat id_rsa.pub >> authorized_keys
```

→If there is **no** authorized_keys file, simple rename id_rsa.pub to authorized_keys:

```
[someuser@remotesys ~]$ cd .ssh/
[someuser@remotesys ~]$ mv id_rsa.pub authorized_keys
```

Correct access:

```
[someuser@remotesys .ssh]$ chmod 640 authorized_keys
```

Back on the MPE system, do:

```
shell/iX> sftp user@remotesys
Connecting to remotesys...
sftp>
```

Verily tis cool!

Legal Notices

OpenSSH is developed by [the OpenBSD Project](#). The software is developed in countries that permit cryptography export and is freely useable and re-useable by everyone under a BSD license.

The **OpenSSL** toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. Both licenses are BSD-style Open Source licenses.

Perl and **GZIP** are available under the [GNU GENERAL PUBLIC LICENSE](#).