BARNARD SOFTWARE, INC.

IPv6/VSE

SSH Secure Copy Supplement Guide

Current Build

Table of Contents

Preface

About this Publication

This is the **IPv6/VSE SSH Secure Copy Supplement Guide**. The manual will introduce you to the SSH Secure Copy facility and provide information on using the SSH Secure Copy facility.

Trademarks

The following are lists of the trademark and products referenced in this manual. Symbols for trademarks and registered trademarks do not appear in subsequent references.

Barnard Software, Inc.

IPv6/VSE is a registered trademark of Barnard Software, Inc.

International Business Machines Corporation

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.

Copyrights

This software and documentation is covered by the following copyright:

Contacting Barnard Software, Inc.

Technical Support is available from Barnard Software, Inc. by phone, mail or email:

Barnard Software, Inc.

Phone: 1-407-323-4773

Support: bsiopti@bsiopti.com

Sales: bsisales@bsiopti.com

Support is available from 9:00 a.m. through 5:00 p.m. EST, Monday through Friday.

If a TSR (Technical Support Representative) is not available at the time of your call, please leave a message and a TSR will return your call as soon as possible. Please provide the following information: name, company, phone number, product name, product release level, and a short description of the problem.

BSIUsers Announcement List Server

When new releases of IPv6/VSE are available BSI will post an announcement on its BSIUsers announcement list.

To subscribe to the BSIUsers announcement list send an email to this email address

BSIUsers-subscribe@yahoogroups.com

To unsubscribe to the BSIUsers announcement list send an email to this email address

BSIUsers-unsubscribe@yahoogroups.com

Problem Determination

If you have a problem using a IPv6/VSE application always check the SYSLST output for additional information and messages. Most messages are written to SYSLST and not to the VSE/ESA system console.

When contacting BSI for technical support always have the applications JCL/commands, console and SYSLST output available for problem determination. The SYSLST output is very important.

While a IPv6/VSE application is running, you can issue the **AR CANCEL XX,PARTDUMP** command to terminate IPv6/VSE application and dump the partition to SYSLST. Using the VSE/POWER Flush (F) command cancels the IPv6/VSE application partition without a dump.

If the IPv6/VSE application partition stops responding to its console interface, use the **AR DUMP XX** command to obtain a dump of the partition.

BSTTSCPY SSH Secure Copy Facility

Over the years, Barnard Software, Inc., has received a number of requests to provide SSH or SSH like functionality. However, VSE/ESA and z/VSE does not provide the basic foundation for this type of function.

At the same time we have wondered “What exactly would you do with SSH on z/VSE?” It is a good question since z/VSE does not have a 'shell' or interactive command environment. When we ask this question more often that not we hear “Well, we have to transfer data to someone that requires we use SSH.” For this we can provide a solution.

The IPv6/VSE BSTTSCPY SSH Secure Copy Facility uses a Linux Pass-through image to facilitate an SSH connection to remote hosts providing for secure file transfer using SSH to and from z/VSE.

SSH

SSH is the standard world wide for secure access to systems.

Secure Shell, or SSH, is a cryptographic (encrypted) network protocol for initiating text-based shell sessions on remote machines in a secure way.

This allows a user to run commands on a machine's command prompt without them being physically present near the machine. It also allows a user to establish a secure channel over an insecure network in a client-server architecture, connecting an SSH client application with an SSH server. Common applications include remote command-line login and remote command execution, but any network service can be secured with SSH. The protocol specification distinguishes between two major versions, referred to as SSH-1 and SSH-2.

The most visible application of the protocol is for access to shell accounts on Unix-like operating systems, but it sees use on Windows as well. In 2015 Microsoft announced that they would include native support for SSH in a future release.

SSH was designed as a replacement for Telnet and other insecure remote shell protocols such as the Berkeley rsh and rexec protocols, which send information, notably passwords, in plaintext, rendering them susceptible to interception and disclosure using packet analysis. The encryption used by SSH is intended to provide confidentiality and integrity of data over an unsecured network, such as the Internet.

Secure Copy

Secure copy or SCP is a means of securely transferring computer files between a local host and a remote host. It is based on the Secure Shell (SSH) protocol.

SFTP vs. FTPS

FTPS (also known as FTP-ES, FTP-SSL and FTP Secure) is an extension to the commonly used File Transfer Protocol (FTP) that adds support for the Transport Layer Security (TLS) and the Secure Sockets Layer (SSL) cryptographic protocols.

FTPS should not be confused with the SSH File Transfer Protocol (SFTP), an incompatible secure file transfer subsystem for the Secure Shell (SSH) protocol. It is also different from FTP over SSH, the practice of tunneling FTP through an SSH connection.

In the past, CSI and IBM have written manuals describing a “Secure FTP Facility” for z/VSE. This facility is FTPS (FTP using SSL). It is not SFTP (FTP over SSH).

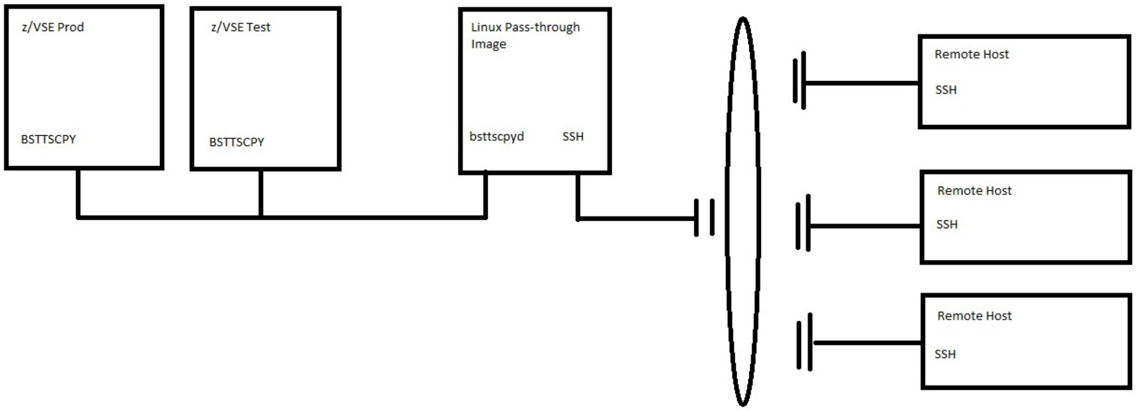
IPv6/VSE provides FTPS (FTP over SSL) also.

Secure Copy Concepts

The following diagram shows how the BSTSTSCPY Secure Copy Facility transfers data to and from z/VSE using a Linux Pass-through image.

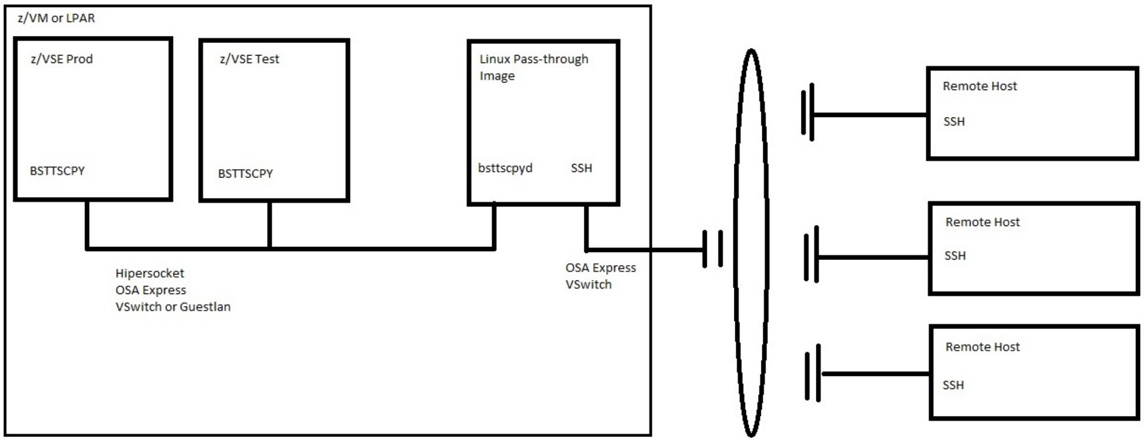
BSTTSCPY using a Linux Pass-through Image

This is the basic overview of the IPv6/VSE Secure Copy Facility and the Linux Pass-through Image.



The BSTTSCPY application running on z/VSE connects to the bsttscpyd (BSTTSCPY Daemon) running on the Linux Pass-through image. From there, the bsttscpyd uses SSH to connect to the destination remote host. Data transferred from BSTTSCPY running on z/VSE to the bsttscpyd is clear text. The data transferred by SSH is, of course, encrypted.

BSTTSCPY Using Linux on System z



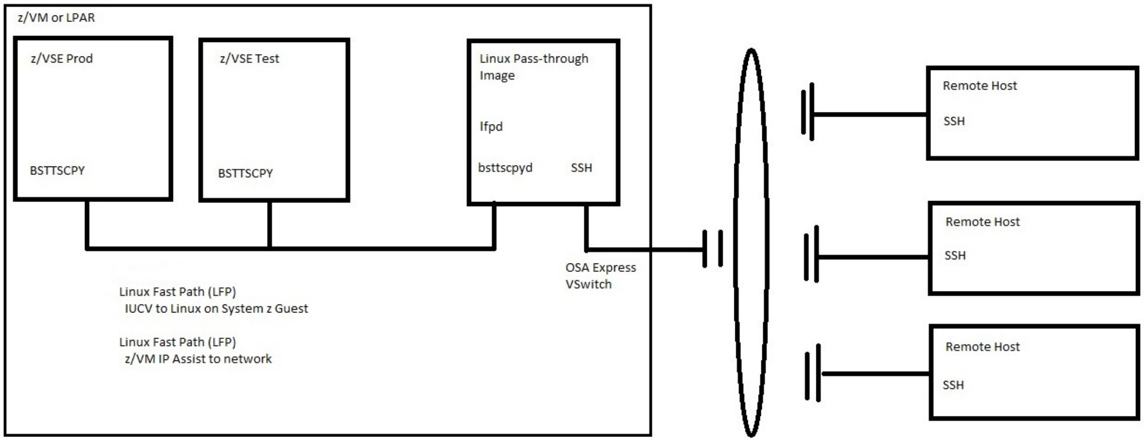
This is the recommended configuration.

In this configuration we suggest using a Hipersockets connection between z/VSE and the Linux Pass-through image. This is very fast. This configuration also guarantees no clear text data ever leaves the System z machine.

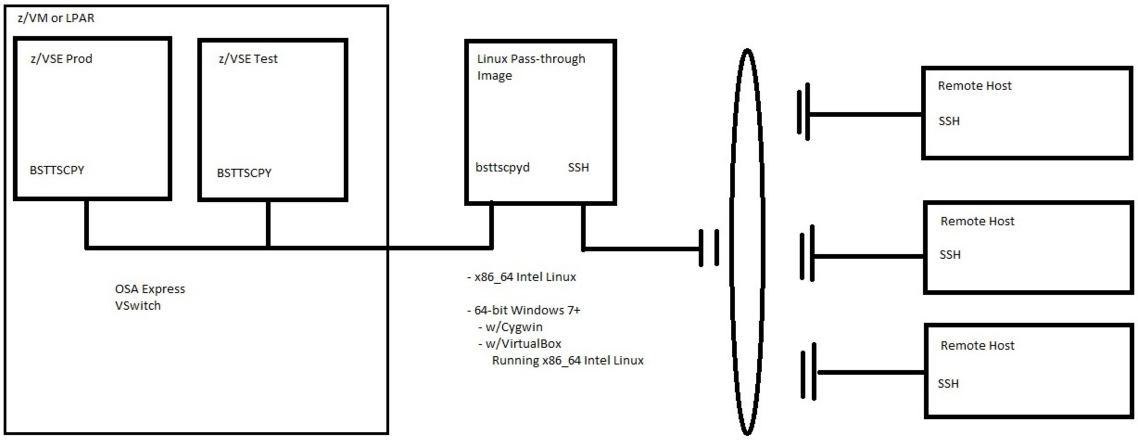
Linux Fast Path (LFP)

IBM's Linux Fast Path (LFP) can also be used in this configuration. Using LFP, BSTTSCPY can communicate with bsttscpyd running on the Linux Path-Tthough image using IUCV.

LFP also provides access to z/VM IP Assist which can be used to access the network on supported System z hardware, providing access to bsttscpyd running on an x86_64 Linux Pass-through image.



BSTTSCPY Using x86_64 Intel



If you do not have a Linux on System z machine available to run the bsttscpyd, you still can use this feature. You can use one of these options.

1. An x86_64 Intel Linux machine
2. A 64-bit Windows 7 (or newer) machine
   Running either ...
   1. 64-bit Cygwin
   2. Virtual Box
      Running an x86_64 Intel Linux image
3. A 64-bit Windows 10 machine
   1. Using Windows Subsystem for Linux
   2. x86_64-bsttscpyd runs in bash command line

Some customers have suggested that this is not a 'secure' configuration and I have been mystified by these comments. A good network administrator can easily make this configuration completely secure.

First, the subnet used by the BSTTSCPY facility in z/VSE would be different than the usual production subnet. E.g., If the production subnet is 192.168.0.0/16 then the subnet used by the BSTTSCPY facility might be 172.16.1.0/24.

Second, the NIC's used by the System z machine and the PC would be connected to the same layer 2 switch. This means traffic from these systems would never go outside of the switch being used.

Next, traffic from these systems would use a special/unique VLAN.

And, this is the key. By using a special VLAN for this traffic, it is physically separate from all other traffic on the LAN. This provides excellent security for the data transfers.

Why Use a Linux Pass-through Image?

The SSH connections from the Linux Pass-through image use public key authentication. Public key authentication allows you to login to a remote host via the SSH protocol without a password and is more secure than password-based authentication.

Password authentication is not supported and can not be used with the BSTTSCPY Secure Copy facility.

There are several benefits to using a Linux Pass-through image.

1. SSH is basic to all Linux OS installations.
2. SSH and Linux are Open Source
3. Support and updates are provided by the Linux distribution
   E.G., SUSE, Red Hat.
4. FIPS 140-2 Certification of OpenSSH and OpenSSL
5. All cryptographic overhead is offloaded to the Linux Pass-through image.
   CPU overhead of cryptographic functions can be very high.
6. No data is stored on the Linux Pass-through image.

The last item is critical. The Linux Pass-through image is used only for SSH (and its functionality). No data is stored on the Linux Pass-through image at any time.

The Linux Pass-through image can be a Linux on System z (zLinux) image, an x86-64 Intel Linux image or a Windows system hosting a Linux Pass-through image. When using a Windows host Cygwin, VirtualBox and Windows Subsystem for Linux images are supported.

Cygwin

Cygwin is a Unix-like environment and command-line interface for Microsoft Windows. Cygwin provides native integration of Windows-based applications, data, and other system resources with applications, software tools, and data of the Unix-like environment. Thus it is possible to launch Windows applications from the Cygwin environment, as well as to use Cygwin tools and applications within the Windows operating context.

Cygwin consists of two parts: a dynamic-link library (DLL) as an API compatibility layer providing a substantial part of the POSIX API functionality, and an extensive collection of software tools and applications that provide a Unix-like look and feel.

Cygwin was originally developed by Cygnus Solutions, which was later acquired by Red Hat. It is free and open source software, released under the GNU General Public License version 3.

Cygwin can be download using this link ...
https://www.cygwin.com/

VirtualBox

Oracle VM VirtualBox (formerly Sun VirtualBox, Sun xVM VirtualBox and Innotek VirtualBox) is a hypervisor for x86 computers from Oracle Corporation. Innotek GmbH first developed the product before a Sun Microsystems acquisition in 2008. Oracle has continued development since 2010.

VirtualBox may be installed on an existing host operating system; it can create and manage guest virtual machines, each with a guest operating system and its own virtual environment. Supported host operating systems include Linux, OS X, Windows XP and later, Solaris, and OpenSolaris; there are also ports to FreeBSD and Genode. Supported guest operating systems include versions and derivations of Windows, Linux, BSD, OS/2, Solaris, Haiku and others. Since release 3.2.0, VirtualBox also allows limited virtualization of OS X guests on Apple hardware, though OSx86 can also be installed using VirtualBox.

VirtualBox can be downloaded using this link ...
https://www.virtualbox.org/wiki/Downloads

Linux Pass-through Image

Once you have access to the Linux Pass-through image, you will want to create the user that will run the bsttscpyd daemon. This can be root but it is not required. Since no data is stored on the Linux Pass-through image the user used can be a normal user.

Authentication

The SSH connections from the Linux Pass-through image to destination remote hosts use public key authentication. Public key authentication allows you to login to a remote host via the SSH protocol without a password and is more secure than password-based authentication.

Password authentication is not supported and can not be used with the BSTTSCPY Secure Copy facility.

SSH keys provide a more secure way of logging into a virtual private server with SSH than using a password alone. While a password can eventually be cracked with a brute force attack, SSH keys are nearly impossible to decipher by brute force alone. Generating a key pair provides you with two long string of characters: a public and a private key. You can place the public key on any server, and then unlock it by connecting to it with a client that already has the private key. When the two match up, the system unlocks without the need for a password. You can increase security even more by protecting the private key with a passphrase.

Setting Up SSH Keys

Create an RSA Key Pair

Login to the Linux Pass-through image using the userid that will be running the BSTTSCPY daemon (it is not necessary to use root for this function). Enter this command and press enter at the prompts.

```
ssh-keygen -t rsa
```

The result will look something like this ...

```
ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/demo/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/demo/.ssh/id_rsa.
Your public key has been saved in /home/demo/.ssh/id_rsa.pub.
The key fingerprint is:
4a:dd:0a:c6:35:4e:3f:ed:27:38:8c:74:44:4d:93:67 demo@a
The key's randomart image is:
+--[ RSA 2048]----+
|          .oo.   |
|         .  o.E  |
|        + .  o   |
|     . = = .     |
|      = S = .    |
|     o + = +     |
|      . o + o .  |
|           . o   |
|                 |
+-----------------+
```

Copy the Public Key

Once the key pair is generated, it's time to place the public key on the destination remote host(s) that we want to use.

```
ssh-copy-id user@ip_address
```

This command will use a special ssh copy utility to connect to the specified remote host with the specified userid (you will have to enter a password here ... one last time) and copy the public key to a file on the remote host. You can use this same command for each of your destination remote hosts.

```
The authenticity of host '12.34.56.78 (12.34.56.78)' can't be
established.
RSA key fingerprint is
b1:2d:33:67:ce:35:4d:5f:f3:a8:cd:c0:c4:48:86:12.
Are you sure you want to continue connecting (yes/no)? Yes
Warning: Permanently added '12.34.56.78' (RSA) to the list of known
hosts.
user@12.34.56.78's password:
Now try logging into the machine, with "ssh 'user@12.34.56.78'",
and check in:

  ~/.ssh/authorized_keys

to make sure we haven't added extra keys that you weren't
expecting.
```

Test the setup

Now, ssh into the destination remote host from the Linux Pass-through image. You should connect without a password being required.

Disabling Password Login

Once you have created your keys and confirmed password-less authentication you may wish to disable password authentication to an SSH daemon.

In the /etc/ssh/sshd_config

```
ChallengeResponseAuthentication no
PasswordAuthentication no
UsePAM no
```

Bsttscpyd Programs

The various bsttscpyd program are located in a file called bsttscpyd.tar.gz. The bsttscpyd.tar.gz file is part of the IPv6/VSE zip download.

IBM IPv6/VSE customers can download the bsttscpyd.tar.gz file from the BSI web site at http://www.bsitcpip.com/ftp/IP-IPv6_VSE-bsttscpyd.tar.gz

x86_64-bsttscpyd is the x86_64 (64-bit) Intel Linux daemon

s390x-bsttscpyd is the s390x (Linux on System z) Linux daemon

cygwin-bsttscpyd is the Cygwin on Windows Linux daemon

Linux on System z

The BSTTSCPY daemon for Linux on System z is called s390x-bsttscpyd.

Linux on x86_64 Intel

The BSTTSCPY daemon for x86_64 Intel Linux is called x86_64-bsttscpyd.

Linux on Windows

Many z/VSE customers I talk with simply do not have a Linux image available. This is not a problem. You can use a 64-bit Windows system also. It must be Windows 7 (or newer) and 64-bit.

There are three supported methods for running the Linux BSTTSCPY daemon under Windows.

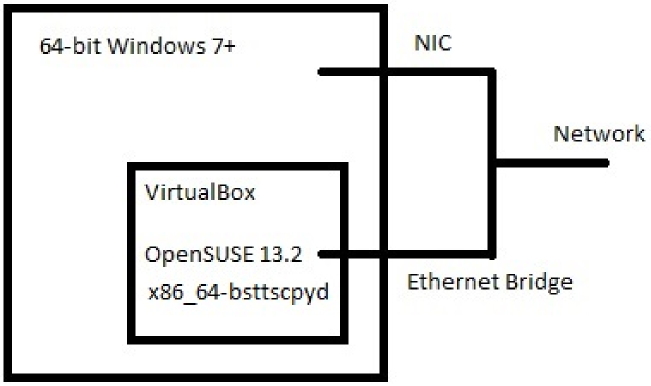
Windows 10 Windows Subsystem for Linux (WSL)

The Windows 10 Windows Subsystem for Linux (WSL) will run the x86_64-bsttscpyd daemon in a bash shell. This is the easiest and simplest method of running the secure copy daemon under Windows.

Instructions for installing WSL can be found on the Microsoft web site. Only the basic installation is required. Once installed a bash command prompt will execute the ssh commands required to exchange ssh keys just as any Linux command prompt would do.

VirtualBox

The first method is to use Oracle's Virtual Box software to run a Linux Pass-through image virtual machine on one of your existing Windows machines. You can do this with other visualization products also but we used VirtualBox because it is free, simple, performs well and it works.



Installing

It would be great if I could write amazing documentation on how to install VirtualBox. However, the Oracle VM Virtual Box web site has a truly great User's Guide.

The Oracle VM Virtual Box User's Guide is here ...
https://www.virtualbox.org/manual/UserManual.html

I did a full installation.

Tip: Use the Bridged Networking for your Linux Pass-through image.

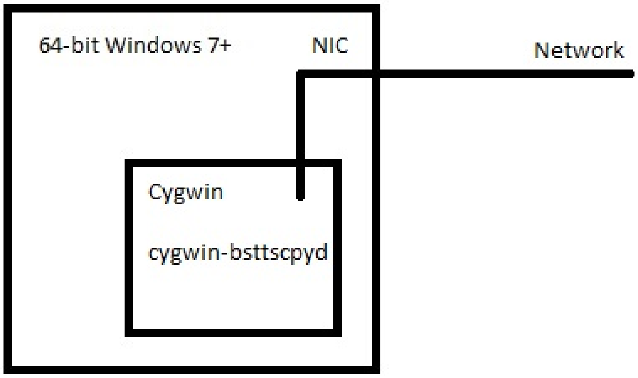
With bridged networking, VirtualBox uses a device driver on your host system that filters data from your physical network adapter. This driver is therefore called a "net filter" driver. This allows VirtualBox to intercept data from the physical network and inject data into it, effectively creating a new network interface in software. When a guest is using such a new software interface, it looks to the host system as though the guest were physically connected to the interface using a network cable: the host can send data to the guest through that interface and receive data from it. This means that you can set up routing or bridging between the guest and the rest of your network.

The BSTTSCPY daemon for x86_64 Intel Linux is called x86_64-bsttscpyd.

Once I completed the VirtualBox installation, I started a virtual box and installed the OpenSUSE 13.2 DVD. I did a standard install and during setup I enabled the SSH server and opened the SSH port in the firewall. After the installation was complete, it was ready to use.

Gygwin

Another option for running the Linux BSTTSCPY daemon on a Windows machine is to use Gygwin. Cygwin is also a good option. It is free, performs well and is also easy to use.



Installing

Again, it would be great if I could write documentation for installing Cygwin but thankfully this is not necessary. The Oracle manual for their “Enterprise Manager Cloud Control Basic Installation Guide” Chapter 5 has a wonderful description of how to install Cygwin and configure the SSH daemon.

The Oracle documentation is here ... http://docs.oracle.com/cd/E24628_01/install.121/e22624/preinstall_req_cygwin_ssh.htm#EMBSC150

When installing, select the x86_64 (64-bit) version.

The BSTTSCPY daemon for Cygwin is named cygwin-bsttscpyd.

BSTTSCPY

The basic structure of the z/VSE BSTTSCPY application is similar to the IPv6/VSE BSTTFTPC application. Remember, SSH transfers all data in binary form. So, if translation of the data is necessary you must tell BSTTSCPY to handle this function.

BSTTSCPY requires IPv6/VSE Build 256pre17 (or later).

Feature code 'S' is required for use of the IPv6/VSE BSTTSCPY application. If your IPv6/VSE license key does not have feature code 'S' in it, you will need to contact Barnard Software, Inc. for an updated license key.

The IPv6/VSE BSTTSCPY application (like BSTTFTPC, BSTTMTPC, etc.) requires a minimum 8M partition for execution.

BSTTSCPY can use the IPv6/VSE BSTTINET/BSTT6NET TCP/IP stacks as well as the TCP/IP for VSE TCP/IP stack.

The Basic Process

Identify the stack and connect to the bsttscpyd you want to access.

Define the INPUT or OUTPUT data.

Specify options. E.g., TYPE A (Convert to ASCII) etc. Most of the options used for a BSTTFTPC FTP client data transfer can be used with BSTTSCPY also.

Define the destination remote host, userid and port.

STOR or RETR the data. Or, EXEC a command/script.

And, finally QUIT. Only one file can be transferred per execution of BSTTSCPY.

Basic JCL

```
*  STOR data to a Remote Host
// EXEC BSTTSCPY,SIZE=BSTTSCPY
ID 00
OPEN ip_address or name
*
INPUT
*
TYPE A | E | I
*
PORT 22
HOST user@host
STOR
*
QUIT
/*

*  RETR data from a Remote Host
// EXEC BSTTSCPY,SIZE=BSTTSCPY
ID 00
OPEN ip_address or name
*
OUTPUT ...
*
TYPE A | E | I
*
PORT 22
HOST user@host
RETR
*
QUIT
/*
* EXEC data from a Remote Host
// EXEC BSTTSCPY,SIZE=BSTTSCPY
ID 00
OPEN ip_address or name
*
OUTPUT ...
*
TYPE A | E | I
*
PORT 22
HOST user@host
EXEC
*
QUIT
/*
```

Just like BSTTFTPC, BSTTSCPY commands are used in pairs. The INPUT command is paired with the STOR command and the OUTPUT command paired with the RETR command.

Commands

ASA

```
ASA ON|OFF
```

The ASA command is used with ASCII (TYPE A) transfers to/from the VSE/POWER LST Queue. This command enables (ON) or disables (OFF) setting/adding the ASA carriage control character as the first character of each print line. The default is ON.

CRLF

```
CRLF ON|OFF
```

The CRLF command is used with ASCII (TYPE A) transfers. This command, combined with the NL command, determines if a <CRLF>, <NL> or no characters mark end-of-line. The default is CRLF ON, NL OFF. Setting CRLF OFF and NL OFF results in no end-of-line characters. Setting CRLF ON and NL ON is invalid and will cause data transfer errors.

DBCS

```
DBCS name EBCDIC 300 ASCII 301
```

The DBCS command is used by the FTP client and FTP server to identify the name of the Double Byte Character Set translation table to be used. There is no default table. This command is issued to the FTP server using a SITE command.

Table	EBDCIC	ASCII
JAPAN	300	301 941 941C
CHINA	835	837 927 947 1380 1385 4933
KOREA	834	951 1362 4930

ID

```
ID nn
```

The ID command identifies the TCP/IP partition to be used during socket processing. The default is 00. The ID command must specify a two digit decimal number. This must be the first command read from SYSIPT and must be placed before any OPEN commands.

INPUT

```
INPUT NULL
INPUT POWER queue name number class userid password segment
INPUT LIBRARY lib sublib member type mode data
INPUT VSAM dlbl
INPUT XRDS dlbl
INPUT SLT tlbl BLKSZ nnnnn RECSZ nnnnn RECFM F|FB|VB option
INPUT NLT tlbl BLKSZ nnnnn RECSZ nnnnn RECFM F|FB|VB option
INPUT SAM dlbl BLKSZ nnnnn RECSZ nnnnn RECFM F|FB|VB
INPUT EXIT phase
INPUT EXIT BSTTPZIO
INPUT EXIT BSTTVTIO
INPUT SYSIPT
```

The INPUT command is used to inform the FTP client of the location and access method to be used to access the data to be stored on the FTP server. Examples of each type of INPUT are available in the Examples chapter.

Keyword	Description
queue	VSE/POWER Queue Id (RDR, LST, PUN)
name	VSE/POWER Queue Member Name
number	VSE/POWER Queue Member Number (or zero)
class	VSE/POWER Queue Class
userid	VSE/POWER Queue Userid
password	VSE/POWER Queue Password
lib	VSE Library Name
sublib	VSE Sublibrary Name
member	VSE Library Member Name
type	VSE Library Member Type
mode	VSE Library Access Mode (Fixed or String)
option	UNLOAD (default), NOREW, REWIND
phase	Phase Name of the User Exit Program

JSEP

```
JSEP ON|OFF
```

The JSEP command is used to tell the BSTTFTPC, BSTTMPTC BSTTLPRC programs to use VSE/POWER separators. This command must precede the INPUT/OUTPUT command to be effective. JSEP OFF is the default.

LF

```
LF ON | OFF

LF ON is equivalent to:
CRLF OFF
NL ON
TRANSLATE ASCII 10 21
TRANSLATE EBCDIC 21 10

LF OFF is equivalent to:
CRLF ON
NL OFF
TRANSLATE ASCII 10 37
TRANSLATE EBCDIC 21 133
```

The LF command is used to enable (ON) or disable UNIX LineFeed mode in the FTP client and server.

MPWD

```
MPWD xxxxxxxx
```

The MPWD command defines the VSE/POWER Master Password.

NL

```
NL ON|OFF
```

The NL command is used with ASCII (TYPE A) transfers. This command, combined with the CRLF command, determines if a <CRLF>, <NL> or no characters mark end-of-line. The default is CRLF ON, NL OFF. Setting CRLF OFF and NL OFF results in no end-of-line characters. Setting CRLF ON and NL ON is invalid and will cause data transfer errors.

OPEN

```
OPEN ipaddr|name port
```

The OPEN command opens a connection on the specified host on the specified port. The IP address can be specified in standard dotted decimal notation or can be specified as a character name. If a name is specified, the name will be used in a GETHOSTBYNAME call to TCP/IP. The default port address for bsttscpyd is 1807.

OUTPUT

```
OUTPUT NULL
OUTPUT POWER queue name number class disp form dest
OUTPUT LIBRARY lib sublib member type mode
OUTPUT KSDS dlbl RECSZ nnnnn NORESET password KEYOS nnnn
OUTPUT ESDS dlbl RECSZ nnnnn NORESET|* password|* V
OUTPUT XRDS dlbl RECSZ nnnnn NORESET|* password|*
OUTPUT SLT tlbl BLKSZ nnnnn RECSZ nnnnn RECFM F|FB|VB
OUTPUT NLT tlbl BLKSZ nnnnn RECSZ nnnnn RECFM F|FB|VB
OUTPUT SAM dlbl BLKSZ nnnnn RECSZ nnnnn RECFM F|FB|VB
OUTPUT EXIT phase
OUTPUT EXIT BSTTPZIO srclib.sublib phase.PHASE dstlib.sublib
OUTPUT EXIT BSTTVTIO
OUTPUT SYSLST
OUTPUT SYSPCH
```

The OUTPUT command is used to inform the FTP client of the location and access method to be used to access the data to be retrieved from the FTP server. Examples of each type of OUTPUT are available in the Examples chapter.

Keyword	Description
queue	VSE/POWER Queue Id (RDR, LST, PUN)
name	VSE/POWER Queue Member Name
number	VSE/POWER Queue Member Number (or zero)
class	VSE/POWER Queue Class
tlbl	VSE TLBL Name
dlbl	VSE DLBL Name
NORESET	Do not reset VSAM cluster
disp	VSE/POWER Disposition
dest	VSE/POWER Destination Node Name
form	VSE/POWER Form Id
lib	VSE Library Name
sublib	VSE Sublibrary Name
member	VSE Library Member Name
type	VSE Library Member Type
mode	VSE Library Access Mode (Fixed or String)
Password	The password for the VSAM file
phase	Phase Name of the User Exit Program
KEYOS	KSDS Key OffSet (used with NORESET to delete existing records when new records are loaded into a file)
V	Variable length

PAD

```
PAD ON | OFF
```

The PAD command is used to enable or disable padding of output data records. Variable length ASCII input records can be padded to fixed length using this command. The PAD command can be used in the BSTTFTPC (Batch FTP) and as a SITE command with the BSTTFTPS (FTP server).

PADCHAR

```
PADCHAR nnn
```

The PADCHAR command is used to define the pad character value. This value is specified as a decimal number. PADCHAR 64 would define a pad character of a space.

SBCS

```
SBCS name
```

The SBCS command is used by the FTP client and FTP server to identify the name of the Single Byte Character Set translation table to be used. The default table is US_ENG_03. This command is issued to the FTP server using a SITE command.

US_ENG_01	-	US English	EBCDIC	037	ASCII	437
US_ENG_02	-	US English	EBCDIC	037	ASCII	850
US_ENG_03	-	US English	EBCDIC	037	ASCII	1252
UK_ENG_01	-	UK English	EBCDIC	285	ASCII	437
UK_ENG_02	-	UK English	EBCDIC	285	ASCII	850
UK_ENG_03	-	UK English	EBCDIC	285	ASCII	1252
GERMAN_01	-	Germany/Austria	EBCDIC	273	ASCII	437
GERMAN_02	-	Germany/Austria	EBCDIC	273	ASCII	850
GERMAN_03	-	Germany/Austria	EBCDIC	273	ASCII	1252
FRANCE_01	-	France	EBCDIC	297	ASCII	437
FRANCE_02	-	France	EBCDIC	297	ASCII	850
FRANCE_03	-	France	EBCDIC	297	ASCII	1252
ITALY_01	-	Italy	EBCDIC	280	ASCII	437
ITALY_02	-	Italy	EBCDIC	280	ASCII	850
SPAIN_01	-	Spain/Latin America	EBCDIC	284	ASCII	437
SPAIN_02	-	Spain/Latin America	EBCDIC	284	ASCII	850
SPAIN_03	-	Spain/Latin America	EBCDIC	284	ASCII	1252
DN_01	-	Denmark/Norway	EBCDIC	277	ASCII	437
DN_02	-	Denmark/Norway	EBCDIC	277	ASCII	850
DN_03	-	Denmark/Norway	EBCDIC	277	ASCII	1252
FS_01	-	Finland/Sweden	EBCDIC	278	ASCII	437
FS_02	-	Finland/Sweden	EBCDIC	278	ASCII	850
FS_03	-	Finland/Sweden	EBCDIC	278	ASCII	1252
BELGIUM_01	-	Multilingual	EBCDIC	500	ASCII	437
INTER_01	-	Multilingual	EBCDIC	500	ASCII	850
INTER_02	-	Multilingual	EBCDIC	500	ASCII	1252
OS_01	-	Multilingual	EBCDIC	1047	ASCII	1252
OS_02	-	Multilingual	EBCDIC	1047	ASCII	1252
OS_03	-	Multilingual	EBCDIC	1047	ASCII	437
OS_04	-	Multilingual	EBCDIC	1047	ASCII	437
OS_05	-	Multilingual	EBCDIC	1047	ASCII	850

SOSI

```
SOSI NONE
SOSI KEEP
SOSI XLATE
SOSI CONVERT
SOSI BLANK
```

The SOSI command specifies how to handle DBCS Shift-In (SI) and Shift-Out (SO) characters. The SOSI command is passed to the FTP server as a SITE command.

SOSI CONVERT
This specification (the default) indicates that the data stream will be converted and that SO/SI characters will be added or removed as appropriate.

SOSI KEEP
SO/SI characters will be retained as place-holders in the ASCII stream.

SOSI XLATE
SO/SI characters will be retained as place-holders in the ASCII stream but will be translated to their ASCII equivalents.

SOSI BLANK
SO/SI characters will be retained as place holders in the ASCII stream but will be translated to ASCII spaces.

SOSI NONE
No SO/SI characters are expected. Instead the complete data stream is expected to be pure DBCS only.

TERMINATE

```
TERMINATE
```

The TERMINATE command terminates processing. This command can be issued at anytime during the transfer process and the application will terminate its processing and go to end-of-job. If an FTP client transfer hangs you can use this command to terminate the job.

TRANSLATE

```
TRANSLATE ASCII xxx yyy
TRANSLATE EBCDIC xxx yyy
```

The TRANSLATE command is used to modify the default ASCII to EBCDIC or ECBDIC to ASCII translate table. The TRANSLATE ASCII command translates an ASCII decimal xxx to EBCDIC decimal yyy. The TRANSLATE EBCDIC command translates an EBCDIC decimal xxx to ASCII decimal yyy. xxx and yyy must be specified in decimal and in the range 0 to 255.

TRCMD

```
TRCMD ON|OFF
```

The TRCMD command is used to tell the FTP client or server to translate ASA printer control code to ASCII. This allows better viewing of VSE/POWER LST output by word processors.

Examples

VSE/Power

```
// EXEC BSTTSCPY,SIZE=BSTTSCPY
ID 00
OPEN 192.168.1.60
*
INPUT POWER LST IDCAMS  00091 H BARNARD
TYPE A
* Convert output to ASCII text file
ASA ON
TRCMD ON
*
PORT 22
HOST jcb@192.168.1.12
STOR idcams.txt
*
QUIT
/*
```

SAM

```
// ASSGN SYS001,DISK,TEMP,VOL=SYSWK2,SHR
// DLBL FTP1TST,'FTP1.TEST.FILE',0,SD
// EXTENT SYS001,SYSWK2,,,00015,10000
// DLBL FTP2TST,'FTP2.TEST.FILE',0,SD
// EXTENT SYS001,SYSWK2,,,15015,10000
/*
*  Store a SAM file on a Remote Host
// EXEC BSTTSCPY,SIZE=BSTTSCPY
ID 00
OPEN 192.168.1.60
*
INPUT SAM FTP1TST BLKSZ 27600 RECSZ 80 RECFM FB
TYPE A
*
PORT 22
HOST jcb@192.168.1.12
STOR /tmp/ftp1tst.txt
*
QUIT
/*
*  Retrieve a SAM file from a Remote Host
// EXEC BSTTSCPY,SIZE=BSTTSCPY
ID 00
OPEN 192.168.1.60
*
OUTPUT SAM FTP2TST BLKSZ 27600 RECSZ 80 RECFM FB
TYPE A
*
PORT 22
HOST jcb@192.168.1.12
RETR /tmp/ftp1tst.txt
*
QUIT
/*
```

EXEC

```
*  Execute a command or script on a Remote Host
// EXEC BSTTSCPY,SIZE=BSTTSCPY
ID 00
OPEN 192.168.1.60
*
OUTPUT SYSLST
TYPE A
LF ON
*
PORT 22
HOST jcb@192.168.1.12
EXEC <command>
*
QUIT
/*
```

The EXEC command does NO return code or completion checking. The command or script name is limited to 75 characters (columns 6-80 of the command line). No continuation is supported.

However, if completion checking is required the OUTPUT command can be directed to a VSAM ESDS or SAM file. This OUTPUT file can be read using a simple REXX EXEC to determine completion.

If the command is longer than 75 characters, a script can be used instead.

If the path to the command/script is too long, a symbolic link on the remote host can be used to shorten the path.