



Tectia Server 6.8 for IBM z/OS

Quick Start Guide

08 April 2026

Tectia Server 6.8 for IBM z/OS : Quick Start Guide

08 April 2026

Copyright © 2007–2026 SSH Communications Security Corporation

This software and documentation are protected by international copyright laws and treaties. All rights reserved.

ssh® and Tectia® are registered trademarks of SSH Communications Security Corporation in the United States and in certain other jurisdictions.

SSH and Tectia logos and names of products and services are trademarks of SSH Communications Security Corporation. Logos and names of products may be registered in certain jurisdictions.

All other names and marks are property of their respective owners.

No part of this publication may be reproduced, published, stored in an electronic database, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, for any purpose, without the prior written permission of SSH Communications Security Corporation.

THERE IS NO WARRANTY OF ANY KIND FOR THE ACCURACY, RELIABILITY OR USEFULNESS OF THIS INFORMATION EXCEPT AS REQUIRED BY APPLICABLE LAW OR EXPRESSLY AGREED IN WRITING.

For Open Source Software acknowledgements, see appendix *Open Source Software License Acknowledgements* in the *Administrator Manual*.

SSH Communications Security Corporation
Karvaamokuja 2D, FI-00380 Helsinki, Finland

Table of Contents

1. About This Guide	5
1.1. Contents	5
1.2. Related Documents	5
1.3. Introduction to Tectia Server for IBM z/OS	6
2. Installing Tectia Server for IBM z/OS	7
2.1. Preparing for Installation	7
2.1.1. System Requirements	7
2.1.2. Permission Requirements	7
2.1.3. Important Directories and Files in USS	9
2.1.4. Uploading Files Required for Installation	9
2.2. Installing the Tectia Server for IBM z/OS Software	11
2.2.1. Installing the Tectia SSH Assistant ISPF Application	11
2.2.2. Installation Settings and Defaults	12
2.2.3. Generating Product Installation Jobs	17
2.2.4. Running the Product Installation Jobs	18
3. Getting Started with Tectia Server for IBM z/OS	21
3.1. Controlling the Server from ISPF	21
3.1.1. Environment Variables	22
3.2. SSH Connection Status	22
3.2.1. Interpreting ApplData format for TCP listener socket	23
3.2.2. Interpreting ApplData format for TCP connection socket	23
3.3. Running Client Programs	25
3.3.1. Under USS	26
3.3.2. Under MVS (Non-Interactive)	26
4. Setting up Non-Interactive Authentication Using JCL	27
4.1. Server Authentication with Public Keys	27
4.2. User Authentication with Public Keys	28

Chapter 1 About This Guide

This document is a Quick Start Guide with minimal instructions for installing and getting started with Tectia Server for IBM z/OS.

This guide is intended for system programmers who are familiar with Unix System Services (USS) and z/OS. Consult the *Tectia Server for IBM z/OS User Manual* and *Admin Manual* for more detailed information and guidance.

1.1 Contents

The following topics are covered in this guide:

- Installing Tectia Server for IBM z/OS ([Chapter 2](#))
- Getting started with Tectia Server for IBM z/OS and its client programs ([Chapter 3](#))
- Setting up Non-Interactive Authentication Using JCL ([Chapter 4](#))

1.2 Related Documents

Tectia Server for IBM z/OS Product Description contains important background information on the Tectia client/server solution.

Tectia Server for IBM z/OS Administrator Manual gives detailed instructions on the installation, configuration, and use of Tectia Server for IBM z/OS.

Tectia Server for IBM z/OS User Manual gives instructions on using the Tectia client tools on z/OS for secure system administration and secure file transfer.

1.3 Introduction to Tectia Server for IBM z/OS

Tectia Server for IBM z/OS is a client/server security solution based on the Secure Shell protocol. It provides secure file transfer, secure shell access, remote shell command execution, and TCP and FTP tunneling. When used with Tectia Client on Windows, it provides transparent secure TN3270 connections.

Tectia Server 6.8 for IBM z/OS is installed to z/OS Unix System Services (USS), but it can be run in the native environment (hereafter MVS) and process files of the native file system. Tectia Server for IBM z/OS supports direct MVS data set access.

Chapter 2 Installing Tectia Server for IBM z/OS

2.1 Preparing for Installation

2.1.1 System Requirements

The following operating system versions are supported as Tectia Server for IBM z/OS platforms:

- z/OS 2.5, 3.1, and 3.2

The following *minimum* hardware is required:

- IBM System z10 (or later)
- 1 GB RAM for hundreds of simultaneous tunnels
- 200 MB free disk space, 200 cylinders (includes client components)

During installation, an additional 300 MB of temporary disk space is required.

- TCP/IP connection

2.1.2 Permission Requirements

Before Installation

Before you start installing Tectia Server for IBM z/OS, make sure the following requirements are met:

File system requirements

Write access to the `/opt` directory is required during the installation.

User account requirements for installing the server

- The user account installing the product must have an OMVS segment, UID 0 and RACF SPECIAL privilege.

User account requirements for running the server

- The user account running the server must have an OMVS segment and the UID 0.
- If the `BPX.DAEMON FACILITY` class profile is defined, the user must have read access to it.

Requirements for user accounts to support access via Tectia Server

- *Required:* An OMVS segment

Users who are to have access to in-bound SFTP or SSH require an OMVS segment defined in their profile. If a shell program is specified, it must be the pathname of an executable z/OS UNIX shell program; if omitted, the default shell program defined in z/OS UNIX customization is used.

- *Optional:* A home directory. It is required if public key user authentication is used or if the account requires user-specific configuration, for example, environment variables for the file transfer subsystem.

Requirements for user accounts that run Tectia client programs

- *Required:* An OMVS segment
- *Optional:* A home directory. It is required if public key user authentication is used or if the account requires user-specific configuration, for example, profiles for remote hosts.

Library requirements

- The Tectia SSH Assistant application requires the Rexx runtime or Rexx alternate libraries to execute. The Rexx Alternate Library SEAGALT (for example, `FAN140 . SEAGALT` or `IBM . REXX . SEAGALT`, etc.), which is shipped as part of z/OS since version 1.9, may be used to satisfy this requirement. SEAGALT must be available in the linklist or in a STEPLIB allocated to your TSO session.
- *Optional:* By default SSZASST requires SDSF to operate. This dependency to SDSF can be removed in the Tectia SSH Assistant's Installation settings and defaults (0 SETM) panel, using option 0 (0.0) and changing the setting. TSO console `====> Y`.

Permissions for storing keys in SAF

If the server host key or the user keys are going to be stored in the System Authorization Facility (SAF), additional permissions are required.

During Installation

The following additional requirements will be handled during the installation with Tectia SSH Assistant:

- The Tectia SSH Assistant ISPF application uses the `extattr` command to make the server program, `/opt/tectia/sbin/sshd2`, program-controlled. To issue the command, the user account running the setup must have read access to the `BPX.FILEATTR.PROGCTL` facility.
- It is recommended that a user account, `SSHD2`, is created for running Tectia Server for IBM z/OS.

- `CEE.SCEERUN` and `CEE.SCEERUN2` libraries must be available in `LPALIB` or `LNKLST`, and `CEE.SCEERUN2` must be program-controlled.
- The server must be allowed to listen to port 22 (or other configured Secure Shell port).

2.1.3 Important Directories and Files in USS

`/opt/tectia` symlink

Symbolic link to the zFS (Space: 200 MB, 200 Cyls, read/write) that contains executable binaries, setup scripts, configuration files, server key files, manual pages, documentation, license agreement, example JCL scripts.

After installation, Tectia Server for IBM z/OS consists of a directory structure under the zFS file system defined for it. The zFS is mounted onto a mount point defined at installation, and pointed to by the `/opt/tectia` symlink.

`/opt/tectia/etc`

Contains global Tectia information and configuration files, such as:

- `sshd2_config`: the main server configuration file
- `ssh-broker-config.xml`: the global Connection Broker configuration file
- `ssh_ftadv_config`: the global file transfer advice profile configuration file
- `hostkey`: the default server host private key

`$HOME/.ssh2`

Contains user-specific information and configuration files.

`/tmp`

Contains server process ID files and default `STDOUT` and `STDERR` files.

`/tmp/ssh-username`

Contains a temporary user-specific file that is valid while the user process is running.

2.1.4 Uploading Files Required for Installation

The following files need to be uploaded (you can find detailed instructions below) in binary mode to your z/OS host:

- *Tectia SSH Assistant XMIT file* `SSZASST.V060800.BXXXX.XMIT`

Extract the XMIT file from `sszasst-6.8.0.XXX.zip` to a temporary location on your local machine.

(Replace the 'x's in the file names with the correct build number.)

Upload the file with the following data set attributes:

Data set organization: RECFM=FB
 Record length: LRECL=80
 Space allocation unit: TRACKS
 Primary space allocation: PRIMARY=15

- *Tectia Server for IBM z/OS product tar archive* `ssh-tectia-server-zos-6.8.0.XXX-ibmzos.tar.Z`

Extract the product tar archive from `tectia-server-zos-6.8.0.XXX-ibmzos-comm.tar` to a temporary location on your local machine.

The product tar archive `ssh-tectia-server-zos-6.8.0.XXX-ibmzos.tar.Z` can be transferred to either the Unix file system or to a sequential data set. Approximately 59M or 130 cylinders are required.

You can use the following data set attributes:

Data set organization: RECFM=U
 Record length: LRECL=0
 Space allocation unit: CYLINDERS
 Primary space allocation: PRIMARY=130

- *Product licenses tar archive*

Uploading Installation Files Using FTP

Connect to your z/OS host and change file transfer mode to binary:

```
ftp USER@zoshost
ftp> binary
```

Upload the Tectia SSH Assistant XMIT file:

```
ftp> quote site RECFM=FB LRECL=80 TRACKS PRIMARY=15
ftp> put SSZASST.V060800.BXXXX.XMIT
```

(Replace the `XXXX` in the XMIT file name with the correct build number.)

Upload the product tar archive (extracted from `tectia-server-zos-6.8.0.XXX-ibmzos-comm.tar`):

```
ftp> quote site RECFM=U LRECL=0 CYLINDERS PRIMARY=130
ftp> put ssh-tectia-server-zos-6.8.0.XXX-ibmzos.tar.Z SSZ.V680xxx.TARZ
```

(Replace the `xxx` in the tar archive and destination data set names with the correct build number.)

Upload the licenses tar archive to the user's home directory:

```
ftp> cd /u/user/
ftp> put licences.tar
ftp> quit
```

2.2 Installing the Tectia Server for IBM z/OS Software

Installing Tectia Server for IBM z/OS consists of the following steps:

1. Installing the Tectia SSH Assistant ISPF application. See [Section 2.2.1](#).
2. Defining the required settings for the installation. See [Section 2.2.2](#).
3. Generating the product installation jobs. See [Section 2.2.3](#).
4. *As the installing user:* submitting the installation jobs. See [Section 2.2.4](#).

2.2.1 Installing the Tectia SSH Assistant ISPF Application

The Tectia SSH Assistant (**SSZASST**) ISPF application provides an interface for installing and configuring Tectia Server for IBM z/OS and its client tools. It is designed to simplify the process of installing the product tar archive appropriately and performing the multiple configuration tasks required using traditional MVS tools (ISPF and JCL), without requiring the use of the Unix shell.

1. On the z/OS host, receive the Tectia SSH Assistant data set via the following command (replace the `xxxx` in the XMIT file name with the correct build number):

```
TSO RECEIVE INDSN(SSZASST.V060800.Bxxxx.XMIT)
```

In response to the `RECEIVE` prompt, you may enter the usual parameters to control the creation of the received data set, or just press `enter` to take the defaults and create a data set called `prefix.SSZASST.PDS`.

2. Inside the restored data set you will find a Rexx script called `$RECEIVE`. EXEC the script to set up the application libraries:

```
TSO EXEC 'prefix.SSZASST.PDS($RECEIVE)'
```

This Rexx will prompt for the HLQ under which the application libraries are to be set up, as well as optional `VOLSER`, if needed.

3. Press **Enter** repeatedly to page through the command output.
4. Set up the Tectia SSH Assistant application to be invoked:

```
TSO EXEC 'prefix.SSZASST.CEXEC(SSZ)'
```

```

SSZ                               Secure Shell Assistant                               2016/04/27 17:05
Option ==> █

0 SETM      Installation settings and defaults submenu
1 GENJ      Generate installation jobs
2 INST      Perform the step-by-step installation
3 CONF      Manage configuration files
4 TASK      Start/stop/modify started tasks
5 UTIL      Utility jobs to manage the installation

```

Figure 2.1. Tectia SSH Assistant main menu

2.2.2 Installation Settings and Defaults

The settings for the installation is done via the Tectia SSH Assistant's **Installation settings and defaults (0 SETM)** panel.

```

SSZ SMITH                               Secure Shell Assistant                               2023/11/19 12:36
Option ==> 0

0 SETM      Installation settings and defaults submenu

```

The settings for installation input ([0.2 SETI](#)) and output ([0.3 SETO](#)) you must define are described in more detail in the following, and you can also consult the online help.

Start from the beginning with **0.0 SETG**:

```

SSZ SMITH                               Settings submenu                               2023/11/19 12:37
Option ==> 0

0 SETG      Define general settings
1 SETS      Define sources for installation
2 SETI      Define settings for installation input
3 SETO      Define settings for installation output
4 SETL      Load settings profile from logged definition

```



Note

At this stage you may return to adjust the settings and regenerate the jobs as often as necessary.

0.0 SETG - Define general settings

A log file of settings options is built based on the selections and input provided. This log file is useful as a record, but it also allows a different user to import your settings and continue the installation, with possibly higher privileges.

For this reason, on a new installation, the name of the log file must be given before proceeding, so you will be prompted for this via the **0.2 SETI** panel.

```

SSZ SMITH          Define Settings and Defaults          2023/11/19 12:37
Command ==>>>

Enter the input settings to use for customizing SSH install jobs.

HLQ              ==>>> SSZ                               High-level qualifier

Version         ==>>> 6.8.0.1       SSZ Version number
Installer      ==>>> U0             UID 0 user who will perform installation

Installer tailored batch jobs output dataset details:
DSName         ==>>> SSZ.INSTALL.CNTL
Logfile        ==>>> SSZ.INSTALL.LOG
Volser         ==>>>              Unit ==>>>          Storclas ==>>>

Default jobcards for install jobs:
//              JOB  ,,CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//              NOTIFY=&SYSUID

```

Just set the logfile details and exit this panel.

The installer will now be able to save settings. You can return to this panel later to complete other details if necessary. On exit, you will be in the originally selected option, **0.0 SETG**.

```

SSZ SMITH          Define Settings and Defaults          2023/11/19 12:37
Command ==>>>

Enter general settings here.

TSO Console     ==>>> N Use TSO console instead of SDSF for Task operations.
Init.d Scripts  ==>>> N Provide old-style init.d wrappers for daemons.

```

The defaults are acceptable for most cases. Refer to the online help here if you require further information.

0.1 SETS - Define sources for installation

Next, choose **0.1 SETS** to specify the location of the product tar file and the licence files supplied by SSH.com. Those files should have been copied to a suitable place in the z/OS USS filesystem or the MVS dataset realm. If not, refer to Tectia Server for IBM z/OS Administrator Manual Chapter 2.1.6 Uploading Files Required for Installation.

As supplied, the product file name includes the version-release number in a form that can conveniently be parsed by the installer, helping to avoid problems. So it is recommended to save it with the same name under

an appropriate path in the filesystem such as, for example: `/tmp/ssh-tectia-server-zos-6.8.0.228-ibmzos.tar.Z`



Note

If the product tar file and the license files are copied to a dataset, the corresponding name should match the MVS-style mask: `**V680228.TAR**`.

```
SSZ SMITH          Define Sources for Installation          2023/11/19 12:37
Command ==>
```

Enter the archive containing the executables and configuration files for the Tectia for z/OS installation:

```
Tarball    ==> /u/work/ssh-tectia-server-zos-6.8.0.228-ibmzos.tar.Z    >
```



Note

If archive resides in the USS file system, you need to specify the path in the form of: `/tmp/ssh-tectia-server-zos-6.8.0.228-ibmzos.tar.Z`



Note

If the archive is copied to a dataset (PSU, Irec1 0, cylinders 130), and specified using a quoted or unquoted dataset name, you need to specify it in the form of: `SSZ.V680228.TARZ`

Next, enter the tar file containing licence files for the ssh server and clients. Specify the USS path name where it may be found. It will be extracted either from a `.tar` file or a gzipped `.tar.gz` file.

```
Licences    ==> /tmp/licences-6.8.tar    >
```

The product version-release number will be set to 6.8.0.228 from this input, so the name of the licences tarball will be set with a reasonable default, but may be changed as needed. This version-release number will also be preset in the version number field of the next screen.

0.2 SETI - Define settings for installation input

Back in the settings submenu, choose **0.2 SETI**. This was the screen we saw when a first install required information to define a log file (**0.0 SETG**). Now we can complete the other fields where the default needs to be changed. Important is to set the correct name of the UID 0 user who will actually run the install jobs.

```
SSZ SMITH          Define Settings and Defaults          2023/11/19 12:38
Command ==>

Enter the input settings to use for customizing SSH install jobs.

HLQ          ==> SSZ          High-level qualifier
```

```

Version    ==> 6.8.0.228    SSZ Version number
Installer  ==> JS            UID 0 user who will perform installation

Installer tailored batch jobs output dataset details:
DSName     ==> 'SMITH.SSZ.INSTALL.CNTL'
Logfile    ==> 'SMITH.SSZ.INSTALL.LOG'
Volser     ==>           Unit ==>           Storclas ==>

Default jobcards for install jobs:
//          JOB , ,CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1) ,
//          NOTIFY=&SYSUID

```



Note

You can refer to the help panel and the field-level help by positioning the cursor in a field and pressing the help PF key.

The following table lists the settings in the **Define settings for installation input (0.2 SETI)** panel used for customizing the install jobs.

Table 2.1. Define settings for installation input (0.2 SETI)

Setting	Description	Example value
HLQ	Data set high-level qualifier for this install. The high-level qualifier will be used to name data sets, such as PARMLIB and SAMPLIB, created during installation.	SSZ
Version	Version number of the SSH package to install. The version number will usually be part of the name of the original product install tar archive, which will contain a directory named <code>./ssh-tectia-server-zos-6.8.0.XXX</code> . <i>Make sure you enter the correct version number! It must correspond with the untarred directory name.</i>	6.8.0.228
Installer	ID of the UID=0 user who will run the installation. A user with the needed authority is required to run jobs and perform other actions to complete the installation. Jobs will be generated to grant this user certain rights.	JS
Installer tailored batch jobs output data set details		
DSName	Name of the tailored batch jobs output data set. If the DSN exists, it must be a partitioned data set. If it does not exist, it will be created.	'SMITH.SSZ.INSTALL.CNTL'
Logfile	Name of the installation log data set. The DSN will not be qualified and will be created as a sequential file, deleting it if it already exists. The optional volser and unit will be used if supplied.	'SMITH.SSZ.INSTALL.LOG'

0.3 SETO - Define settings for installation output

Finally, select the panel 3 SETO for defining various installation settings which specify the names of the output datasets, libraries, users and procedures to be generated.

The ssh server will run as a started task from a procedure library known to JES, which may be customized here in the proclib field. Also supply the parmlib to be used to provide certain parameters to the system if necessary. The name of the ssh server started task and the name of the user under which the started task will be run may be changed here or the default values accepted.

Similarly, the names of the started tasks and userids for the Certificate server and the Socks Proxy may be customized here. Refer to the help screens.

```

SSZ SMITH          Define Settings and Defaults          2023/11/19 12:38
Command ==>

Enter the output settings to use for customizing SSH install jobs.

Installation target details:
ZFS DSName ==> 'SSZ.V680228.ZFS'
Volser      ==>          Unit ==>          Storclas ==>
Mountpoint ==> /u/vendor/ssz-680228          >

System libraries to receive procs and parms:
Proclib     ==> 'USER.Z25B.PROCLIB'
Parmlib     ==> 'USER.Z25b.PARMLIB'

Started task details:
SSH user    ==> SSHD2          UID 0 user to run SSH server
SSH proc    ==> SSHD2          SSH server procedure name
CERT proc   ==> SSHCERTD      Certificate server procedure name
SOXP user   ==> SSHSP          User to run Socks Proxy
SOXP proc   ==> SSHSP          Socks Proxy server procedure name
SOXP home   ==> /u/sshsp          >

SFTP APF    ==> y SFTP server and clients will use filetype=ADDRSSU

```



Note

By default, the product is installed into a separate filesystem in a ZFS dataset which gets mounted on the supplied mountpoint name. This is recommended for its isolation and ease of management. However, the product can also be installed under a chosen directory in an existing filesystem: just supply the word NONE as the ZFS DSN for this case.

The following table lists the settings in the **Define settings for installation output (0.3 SETO)** panel used for customizing the install jobs.

Table 2.2. Define settings for installation output (0.2 SETO)

Setting	Description	Example value
Target zFS details:		
ZFS DSName	Name of the data set for the installation zFS. The DSN will be used to tailor jobs to allocate and define the zFS, as well as mount commands and BPXPRM parameters.	'SSZ.V6802xxx.ZFS'
Mountpoint	Mountpoint name for the installation zFS. The mountpoint name specifies the point in the Unix file system where the installation zFS will be mounted. It is used in generating jobs, commands and parameters.	/u/vendor/ssz-6802xxx
System libraries to receive procs and parms:		
Proclib	Name of the system library to receive procs. The proclib name is used in generating jobs to install cataloged procedures for started tasks for the SSH server and other server tasks.	USER.Z25B.PROCLIB
Parmlib	Name of the system library to receive parms. The parmlib name is used in generating procedures for started tasks such as the SSH server.	USER.Z25b.PARMLIB

2.2.3 Generating Product Installation Jobs

Tailoring of the required installation jobs is performed from the **Generate installation jobs (1 GENJ)** submenu. This step merely generates the jobs; nothing is run at this stage. You should inspect the generated jobs carefully, as well as verify them with `TYPRUN=SCAN` on the generated job cards. The JCLs must be run by the authorized installing user to have effect.

Selecting any of the options in this panel will result in the presentation of a file-tailored JCL job to perform the task in question. You can also use the **99 GENALL** option to generate all the jobs at once.

Table 2.3. Generate Installation Jobs submenu (1 GENJ)

Option	Description
1.1 INSTUSER	Grant permissions to user doing install
1.2 CPGMCTL	Ensure C library program-controlled
1.3 ADDSSHU	Set up SSH Server user
1.4 ADDSOXP	Set up SOCKS Proxy Server user
1.5 CSFSERV	ICSF permissions
1.6 SERVAUTH	Port 22 control
1.7 SAVE	(Save previous installation key data)
1.8 ZFS	Define installation ZFS
1.9 LOAD	Load installation ZFS
1.10 RESTORE	(Restore previous installation key data)
1.11 SYMLINK	Create /opt/tectia symlink
1.12 SSZLIBS	Sample JCL and PARM libraries
1.13 PROCLIB	Set up started task procedures
1.14 LICENCE	Install licenses from supplied tarball
1.15 KEYGEN	Generate server host keys
1.99 GENALL	Generate all jobs

2.2.4 Running the Product Installation Jobs

Once the product installation jobs have been generated as instructed in [Section 2.2.3](#), the installing user must run them.

As the installing user, do the following:

1. EXEC the Tectia SSH Assistant application:

```
TSO EXEC 'prefix.SSZASST.CEXEC(SSZ)'
```

2. In Tectia SSH Assistant, go to **0 SETM** → **3 SETL** and define the installation log data set from which to extract the installation settings:

```
Logfile ===> '<prefix>.SSZ.INSTALL.LOG'
```

3. In **0.1 SETI** and **0.2 SETO**, check that the installation input and output settings are correct.
4. The **Perform the step-by-step installation (2 INST)** submenu provides a member list (**2.1 JOBS**) of generated jobs in the order they should be run. Submit and check the installation jobs one by one in the order they are listed in. You should see a list of jobs that correspond to the actions in [Section 2.2.3](#). Each job run updates the installation log with its results. After successful completion, the SSH server and clients are installed, with all necessary permissions set up.

5. Log off and on.
6. EXEC the Tectia SSH Assistant application.

To start the SSH server, go to **4 TASK** → **1 TSRV** and enter **1 TSRVS**.

To check the version of the running server, enter **5 TSRVQV**.

It remains just to create configuration files, which can be done via the **Configuration Management (3 CONF)** submenu. Currently, default configurations for the SSH server, the Certificate Validator and the SOCKS Proxy are provided in a plain edit panel.

Chapter 3 Getting Started with Tectia Server for IBM z/OS

This chapter provides information on how to get started with Tectia Server for IBM z/OS and its client tools after they have been successfully installed.

The Tectia server component on z/OS consists of two processes:

- **sshd2**: the main Secure Shell server daemon
- **ssh-certd**: the Certificate Validator, a process used by **sshd2** when validating user certificates

3.1 Controlling the Server from ISPF

You can manage SSH tasks via the SSH **Task** submenu. Started tasks may be started, stopped and modified here, provided that the user is properly authorized and that the started task procedure has been generated and installed.

The available actions for controlling the SSH server (**sshd2**) are listed in [Table 3.1](#).

Table 3.1. Controlling the SSH server (4.1 TSRV)

Option	Description
4.1.1 TSRVS	Start the SSH server
4.1.2 TSRVP	Stop the SSH server
4.1.3 TSRVQV	Query the version of the running server
4.1.4 TSRVTR	Turn trace on or off in the running SSH server
4.1.5 TSRVOP	Set options for starting the SSH server
4.1.6 TSRVR	Restart the SSH server
4.1.7 TSRVRF	Restart the SSH server, killing connections
4.1.20 TSRVSC	SSH server configuration settings and info

3.1.1 Environment Variables

The environment variables `_BPXK_AUTOCVT`, `_BPX_SHAREAS` and `_BPX_BATCH_UMASK` must be set as shown below in `SSHENV` (located in `<HLQ>.V680.SAMPLIB`) when running Tectia Server for IBM z/OS (see below). The server startup procedure sets these variables from the `STDENV DD`.

SSHENV:

```
_BPXK_AUTOCVT=ON
_BPX_SHAREAS=NO
_BPX_BATCH_UMASK=0022
SSH_DEBUG_FMT="%W(72)(2) %Dd/%Dt/%Dy %Dh:%Dm:%Ds:%Df %m/%s:%n:%f %M"
_BPXK_JOBLOG=STDERR
_EDC_ADD_ERRNO2=1
```

3.2 SSH Connection Status

The SSH connection status can be found in the z/OS network TCP socket ApplData. `SSHD2` process writes 40 bytes ApplData to TCP socket in EBCDIC encoding. The information is recorded in SMF type 119 sub type 2 (TCP connection termination record) as `SMF119AP_TTAPPLDATA` when the TCP connection terminates.

The SSH connection status can be viewed through the SDSF Network Activity panel or by issuing a TSO command.

Example TSO command:

```
netstat allc (appld SSZ*
```

The example output displays two sets of information. The first is the TCP listener socket information (running server), the second shows the information of the TCP connection socket (an established connection).

```
MVS TCP/IP NETSTAT CS V2R5          TCPIP Name: TCPIP          08:16:20
User Id  Conn      State
-----  ----      -
SSHD2    0000017D Listen
  Local Socket:  0.0.0.0..22
  Foreign Socket: 0.0.0.0..0
  Application Data: SSZS001,002,000
SSHD2    00000C6B Establish
  Local Socket:  192.168.3.90..22
  Foreign Socket: 10.1.70.80..50016
  Application Data: SSZCbb-,bb-rPAuser          6.7.5.56 SSH Se
```

3.2.1 Interpreting ApplData format for TCP listener socket

TCP listener socket string example:

```
SSZS001,002,000
```

Offset 0-3: 'SSZS'

Offset 4-6: the number of concurrent ssh session in hex decimal format.

Offset 7: ','

Offset 8-10: the maximum number of concurrent ssh session in hex decimal format.

Offset 11: ','

Offset 12-14: summary statistics of the accumulated accept() EWOULDBLOCK counter. The counter is kept in software in a 64bits integer. Each character is a base64 digit. Following shows the sequence of base64 digits:

```
0123456789abcdefghijklmnopqrstuv  
wxyzABCDEFGHIJKLMNopQRSTUVWXYZ/+
```

If it is less than g00, it shows the value of the counter in base64 format. If it is greater than or equal to g00, the first character indicates the index of most significant bit of the 64 bits counter. The second and third character show the rightmost 12 bits of the counter in base64 format.

3.2.2 Interpreting ApplData format for TCP connection socket

TCP connection socket string example:

```
SSZCbb-,bb-rPAuser      6.7.5.56 SSH Se
```

Offset 0-3: 'SSZC'

Offset 4-6: Cryptographic algorithms used from server to client direction

Offset 4,8:

```
A: 3des-cbc  
b: aes128-cbc, rijndael-cbc@ssh.com  
#: aes192-cbc  
B: aes256-cbc  
c: aes128-ctr  
+: aes192-ctr  
C: aes256-ctr
```

```
d: gcm128@openssh.com, AEAD_AES_128_GCM
D: gcm256@openssh.com, AEAD_AES_256_GCM
```

Offset 5,9:

```
g: hmac-sha2-512
c: hmac-sha2-256, hmac-sha256-2@ssh.com
b: hmac-sha1
B: hmac-sha1-96
a: hmac-md5
A: hmac-md5-96
d: hmac-sha224@ssh.com
e: hmac-sha256@ssh.com
f: hmac-sha384@ssh.com
g: hmac-sha512@ssh.com
-: when cipher is gcm type.
```

Offset 6,10:

```
-: no compression
z: zlib
Z: zlib@openssh.com
```

Offset 7: Number of session rekey performed

```
,: no session rekey has been performed.
0: 2^0, 1 session rekey has been performed.
1: 2^1, 2 or more session rekey has been performed.
2,3,4,5: 2^2, 2^3, 2^4, 2^5 or more session rekey has been performed.
6,7,8,9: 2^6, 2^7, 2^8, 2^9 or more session rekey has been performed.
a,b,...,z: 2^10, 2^11, .., 2^35 or more session rekey has been performed.
A,B,...,Z: 2^36, 2^37, .., 2^61 or more session rekey has been performed.
+,: 2^62, 2^63 or more session rekey has been performed.
```

Offset 8-10: Cryptographic algorithms used from client to server direction

Offset 11: Key exchange method used

```
a: ecdh-sha2-nistp521
b: ecdh-sha2-nistp384
c: ecdh-sha2-nistp256
d: diffie-hellman-group1-sha1
e: diffie-hellman-group14-sha1
f: diffie-hellman-group14-sha224@ssh.com
g: diffie-hellman-group14-sha256@ssh.com
h: diffie-hellman-group14-sha256
i: diffie-hellman-group15-sha256@ssh.com
j: diffie-hellman-group15-sha384@ssh.com
k: diffie-hellman-group16-sha384@ssh.com
```

```

l: diffie-hellman-group16-sha512@ssh.com
m: diffie-hellman-group16-sha512
n: diffie-hellman-group18-sha512@ssh.com
o: diffie-hellman-group18-sha512
p: diffie-hellman-group-exchange-sha1
q: diffie-hellman-group-exchange-sha256
r: curve25519-sha256,curve25519-sha256@libssh.org
s: ecdh-nistp521-kyber1024-sha512@ssh.com
t: curve25519-frodokem1344-sha512@ssh.com
u: snttrup761x25519-sha512@openssh.com
v: mlkem768nistp256-sha256
w: mlkem1024nistp384-sha384
x: mlkem768x25519-sha256

```

offset 12: Successful user authentication method or status of username

```

P: Password authentication method.
K: User public key authentication method.
I: Username is invalid.
U: Username is undefined.
R: Username is revoked.

```

offset 13: Channel types started

```

A: 'exec' or 'shell'.
B: 'subsystem' including sftp server.
C: both 'A' and 'B'.

```

offset 14: Username requested from client.

offset 25: The first 15 characters of the client ssh identification string.

3.3 Running Client Programs

Tectia Server for IBM z/OS contains three client-side applications: **sshg3** (Tectia SSH terminal client), **scpg3** (SSH file copy client), and **sftpg3** (SSH file transfer client).

The **ssh-broker-g3** (Connection Broker) component handles the client-related tasks associated with the **sshg3**, **scpg3**, and **sftpg3** client programs. The Connection Broker is the most resource-intensive component of the Tectia client tools, because it does all of the cryptographic and authentication-related tasks for the clients. By default, the Connection Broker starts in the background whenever a Tectia client program is started, and stops when the connection is terminated. For more information on the Connection Broker, see *Tectia Server for IBM z/OS User Manual*.



Note

The Connection Broker is only used when processing client operations; it has no relation to the server.

The **ssh-socks-proxy** component is used for transparent FTP tunneling and FTP-SFTP conversion. For more information on the TectiaSOCKS Proxy, see *Tectia Server for IBM z/OS Administrator Manual*.

3.3.1 Under USS

Interactive remote sessions and file transfers can be used from Unix System Services shells. For example, OMVS, Telnet, TN3270, or Secure Shell sessions can be used.

For information on the command syntax and options, see the **sshg3**, **scpg3**, and **sftpg3** man pages or *Tectia Server for IBM z/OS User Manual Appendix Command-Line Tools and Man Pages*.

The environment variables shown in `sshsetenv` (below) must be set in the user-specific `$HOME/.profile` file (or in any other environment variable file you use).

sshsetenv:

```
export _BPXK_AUTOCVT=ON
export _BPX_BATCH_UMASK=0022
export _BPX_SHAREAS=NO
export SSH_DEBUG_FMT="%W(72)(2) %Dd/%Dt/%Dy %Dh:%Dm:%Ds:%Df %m/%s:%n:%f %M"
export _EDC_ADD_ERRNO2=1
```

3.3.2 Under MVS (Non-Interactive)

Tectia client-side applications can be executed from JCL with `BPXBATCH`, `BPXBATSL`, or `oshell`. **scpg3** uses the same syntax for interactive and non-interactive file transfers. **sftpg3** has a batch mode for non-interactive file transfers.

Sample JCL for running clients in batch are provided in the `<HLQ>.V680.SAMPLIB` data set. When running the clients under JCL, `STDENV` must point to `SSHENV` in the JCL (see [Section 3.1.1](#)) to ensure that the necessary environment variables are set.

Chapter 4 Setting up Non-Interactive Authentication Using JCL

The Secure Shell protocol used by Tectia Server for IBM z/OS provides mutual authentication between the client and server.

For more information on the authentication methods available for Tectia Server for IBM z/OS, see the *Administrator Manual Chapter Authentication*.

4.1 Server Authentication with Public Keys

Use the sample JCL HOSTSAVE (shown below) from <HLQ>.V680.SAMPLIB to accept host keys without user interaction. Edit the JCL to suit your needs. You can consult the *Tectia Server for IBM z/OS User Manual* for an explanation of all the available options for the **ssh-keydist-g3** command.

HOSTSAVE:

```
//HOSTSAV EXEC PGM=BXPBATS,REGION=0M,TIME=NOLIMIT
//STDPARM DD *
PGM /opt/tectia/bin/ssh-keydist-g3
  -v ❶ -N ❷ -F plain ❸ -i ❹ -A /tmp/newhosts.log ❺
  host1 host2 host3 ❻
//STDENV DD DSN=<HLQ>.V680.PARMLIB(SSHENV),DISP=SHR ❼
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//STDIN DD DUMMY
//*
```

- ❶ Enable verbose mode.
- ❷ Accept new host keys automatically.
- ❸ Store the accepted host keys in plain file name format.
- ❹ Store the accepted host keys also using the IP addresses of the hosts.
- ❺ The accepted host keys will be listed in the log file specified here.

- ⑥ Your host names or IP addresses go here.
- ⑦ Required environment variables are set here (see [Section 3.1.1](#)).

4.2 User Authentication with Public Keys

The user's public keys are located in the user's `$HOME/.ssh2` directory on the server.

The batch user accesses the remote machine using an account on the remote machine. The remote user name may either be the same as or different from the batch user's RACF user ID.

Each batch user's public key must be distributed to all the remote accounts. The way the public key is set up differs between Tectia and OpenSSH-based products.

ssh-keydist-g3 uses password authentication for this initial access to the remote server. You can store the password for the remote account in a data set as follows:

1. Allocate a data set or a data set member. For example:

```
// 'USERID.PASSWD'
```

2. The data set must only be accessible to the user executing the JCL.
3. Put the user password in the data set. For example:

```
secret
```

Use the sample JCL `KEYDIST` (shown below) from `<HLQ>.V680.SAMPLIB` to distribute user keys. Edit the JCL to suit your needs. The example assumes that the server host key has already been fetched and verified. You can consult the *Tectia Server for IBM z/OS User Manual* for an explanation of all the available options for the **ssh-keydist-g3** command.

Note that `KEYDIST` must be run under the batch user's user ID in order for the file permissions to be set properly.

KEYDIST:

```
//KEYDIST EXEC PGM=BXPBATS,REGION=0M,TIME=NOLIMIT
//STDPARM DD *
PGM /opt/tectia/bin/ssh-keydist-g3
-t rsa -b 2048 ① -P ②
-u userid ③ -p // 'USERID.PASSWD' ④
-U /tmp/my_log_file ⑤
-O host1.example.com ⑥
//STDENV DD DSN=<HLQ>.V680.PARMLIB(SSHENV),DISP=SHR
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//STDIN DD DUMMY
//
```

-
- ❶ Create a new 2048-bit RSA keypair.
 - ❷ Use an empty passphrase.
 - ❸ The user name specified here will be used.
 - ❹ Use a password stored in a data set. Replace `// 'USERID.PASSWD'` with the name of your password data set.
 - ❺ A log file will be written to the (non-default) location specified here.
 - ❻ Connect to a Unix host running OpenSSH. Replace `host1.example.com` with your host.

In `KEYDIST` above the `-O` option is used to connect to an OpenSSH server running on a Unix host. Use the following **ssh-keydist-g3** options when connecting to Tectia Server on different platforms:

- z/OS: `-Z`
- Unix: `-S`
- Windows: `-W`