Tectia Server 6.6 for IBM z/OS

Quick Start Guide

06 July 2022

Tectia Server 6.6 for IBM z/OS : Quick Start Guide

06 July 2022

SSH Communications Security Corporation
Kornetintie 3, FI-00380 Helsinki, Finland

Table of Contents

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 and ConnectSecure on Windows, it provides transparent secure TN3270 connections.

Tectia Server 6.6 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 2.2, 2.3 and 2.4

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.V060611.B`*`XXXX`*`.XMIT`

  Extract the XMIT file from `sszasst-6.6.11.`*`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-6.6.11.`*`XXX`*`-ibmzos.tar.Z`

  Extract the product tar archive from `tectia-server-zos-6.6.11.`*`XXX`*`-ibmzos-comm.tar` to a temporary location on your local machine.

  The product tar archive `ssh-tectia-server-zos-6.6.11.`*`XXX`*`-ibmzos.tar.Z` can be transferred to either the Unix file system or to a sequential data set. Approximately 55M or 120 cylinders are required.

  You can use the following data set attributes:

  Data set organization: `RECFM=U`
  Maximum block size: `BLKSIZE=32256`
  Record length: `LRECL=0`
  Space allocation unit: `CYLINDERS`
  Primary space allocation: `PRIMARY=120`

- *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.V060611.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-6.6.11.`*`XXX`*`-ibmzos-comm.tar`):

```
ftp> quote site RECFM=U BLKSIZE=32256 LRECL=0 CYLINDERS PRIMARY=120
ftp> put ssh-tectia-server-zos-6.6.11.xxx-ibmzos.tar.Z SSZ.V6611xxx.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.V060611.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)'
   ```


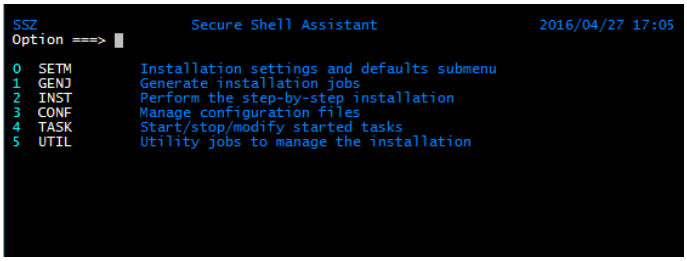

Figure 2.1. Tectia SSH Assistant main menu

2.2.2 Installation Settings and Defaults

Set up the settings for the installation in the Tectia SSH Assistant's **Installation settings and defaults (0 SETM)** panel. The settings for installation input (0.1 SETI) and output (0.2 SETO) you must define are described in more detail in the following, and you can also consult the online help.

The settings will be saved in the installation log file, as well as in the user's application profile. The facility for another user to load these settings from the log file is provided by the **Load settings profile from logged definition (0.3 SETL)** option.

`0.1 SETI` - Settings for installation input

The following table lists the settings in the **Define settings for installation input (0.1 SETI)** panel used for customizing the install jobs.

Table 2.1. Define settings for installation input (0.1 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 `./ssh-tectia-server-zos-6.6.11.`*`XXX`*. *Make sure you enter the correct version number!* It must correspond with the untarred directory name.	`6.6.11.123`
`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.	`BERT`

`Tarball`	Name of the TAR file from which to install Tectia Server for IBM z/OS. The tar archive may be a Unix file or a data set, distinguished by the presence or absence of a leading Unix path symbol. The upload of the tar archive should have been done in binary mode, and for a data set target, its attributes should be `RECFM=0 BLKSIZE=32256 LRECL=0 CYLINDERS PRIMARY=120`.	Unix file:
		`/tmp/ssz-6.6.11.123.tar.Z`
		
		Data set:
		`'FRED.SSZ.V6611100.TARZ'`

`Licenses`	Name of the TAR file containing the product licenses. The tar archive may be a Unix file or a data set, distinguished by the presence or absence of a leading Unix path symbol. If the tar archive is gzipped, gzip must be available.	Unix file:
		`/tmp/licenses-6.6.tar.gz`, `/u/fred/licenses-6.6.tar`
		
		Data set:
		`//'FRED.SSZ.LIC.TAR'`

`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.	`'FRED.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.	`'FRED.SSZ.INSTALL.LOG'`

`0.2 SETO` - Settings for installation output

The following table lists the settings in the **Define settings for installation output (0.2 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:`		
`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.V66110`*xxx*`.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-66110`*xxx*
`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.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.PARMLIB`

`0.3 SETL` - SSH Settings Log

Use this panel to indicate the SSH settings log to use.

SSH settings defined by another user may be loaded from the log file where they have been saved, enabling another user to continue the installation using those settings.

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 ADDSSHDU`	Set up SSH Server user
`1.4 ADDSOXPU`	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 TSRVR`	Restart the SSH server
`4.1.4 TSRVRF`	Restart the SSH server, killing connections
`4.1.5 TSRVQV`	Query the version of the running server
`4.1.6 TSRVTR`	Turn trace on or off in the running SSH server
`4.1.7 TSRVOP`	Set options for starting the SSH server

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>.V6611.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 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*.

3.2.1 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
```

3.2.2 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>.V6611.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>.V6611.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=BPXBATSL,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>.V6611.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.

❻ Your host names or IP addresses go here.

❼ 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>.V6611.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=BPXBATSL,REGION=0M,TIME=NOLIMIT
//STDPARM  DD  *
PGM /opt/tectia/bin/ssh-keydist-g3
-t rsa -b 2048 ❶ -P ❷
-u userid ❸ -p //'USERID.PASSWD' ❹
-U /tmp/my_log_file ❺
-O host1.example.com ❻
//STDENV   DD  DSN=<HLQ>.V6611.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`

