

Co:Z Co-Processing Toolkit for z/OS

Co:Z SFTP - User's Guide

Steve Goetze
Kirk Wolf

V 2.0.0 Edition

Published December, 2011

Copyright © 2011 Dovetailed Technologies, LLC

Table of Contents

1. Introduction	1
1.1. Features	1
1.2. Supported Environments	3
z/OS Host Requirements	3
Remote System Requirements	3
2. Co:Z SFTP Configuration	4
2.1. Configuring the Co:Z SFTP Server	4
Modifying the SFTP Subsystem	4
Customizing <code>sftp-server.rc</code>	5
2.2. Configuring the Co:Z SFTP Client	7
3. Using the Co:Z sftp server	8
3.1. Setting, displaying and clearing file transfer options	8
Example: Setting and displaying basic options	9
Example: Setting multiple options	9
Example: Showing all options	9
3.2. Reading the error log	10
Example: Getting and displaying the error log	10
3.3. Working with Datasets	12
Navigating Datasets	12
Transferring Datasets	12
Listing datasets and PDS directories	15
3.4. Working with POSIX files	18
Transferring Files	18
3.5. Working with JES jobs and spool files	20
Obtaining JES job status	21
Transferring JES spool files	23
Submitting JES jobs	25
JES related options	26
4. Using the Co:Z sftp client	28
4.1. Starting the Co:Z sftp client on z/OS	28
4.2. Setting, displaying and clearing file transfer options	28
Example: Setting and displaying local (client) transfer options	30
Example: Setting multiple local options	30
Example: Showing all local options	30
4.3. Coordinating Transfer Options with a Co:Z SFTP Server	30
4.4. Working with Datasets	32
Navigating Datasets	32
Transferring Datasets	33
Listing datasets and PDS directories	35
4.5. Working with POSIX files	37
Transferring Files	37
4.6. Using the Co:Z sftp client in batch	39
Example: Co:Z sftp client batch job with a password	39
Example: Co:Z sftp client batch job with a RACF Digital Certificate	40

Example: Using saf-ssh-agent to run remote SSH commands	41
A. Co:Z SFTP options	42
A.1. General transfer options	42
A.2. Miscellaneous options	44
A.3. Dataset allocation options	45
B. Session config files	47
B.1. Specifying fixed (immutable) options	47
B.2. Specifying default options	47
B.3. Specifying file pattern specific options	47
Pattern examples	48
C. Dataset Name Determination	50
D. SMF Information	52
D.1. IBM FTP-compatible SMF 119 record subtypes	52
D.2. New SMF 119 record subtypes	52
D.3. Enabling SMF recording	53
D.4. Using the real-time Co:Z SMF API	53
E. Client Authentication Mechanisms	54
E.1. Interactive password authentication	54
E.2. OpenSSH keypair authentication	54
E.3. OpenSSH SSH_ASKPASS authentication	55
E.4. RACF Digital Certificate authentication	55
F. Restricting OpenSSH users to SFTP	58
G. Setting up a test OpenSSH system on z/OS	59
H. License	61

1. Introduction

The "IBM Ported Tools for z/OS" product includes a port of the popular OpenSSH tools. These tools provide for secure remote login and program execution (**ssh**) and file transfer (**sftp** and **scp**). The **sftp** and **sftp-server** commands implement a file transfer program that is similar to ftp, but use ssh for their underlying secure transport. The sftp specification ¹ accounts only for binary transfers. The IBM z/OS sftp client has been enhanced to support ASCII-EBCDIC conversion. Dataset support is not provided.

The Co:Z Co-Processing Toolkit for z/OS includes Co:Z SFTP - a port of the OpenSSH (v5.0p1) **sftp-server** subsystem and **sftp** command (renamed as **cozsftp**). Extensive enhancements have been made to support z/OS facilities such as MVS datasets and spool files. IBM Ported Tools for z/OS (OpenSSH) (or equivalent), is required since Co:Z does not provide the base ssh and sshd components.

1.1 Features

- Co:Z is compatible with most existing sftp products (see *the section called "Remote System Requirements"*).
- Transfer datasets via the **get** and **put** commands
- Navigate z/OS catalogs via the **cd** command.
- List dataset information and PDS directories via the **ls** command.
- Records SMF 119 records if user has BPX.SMF SAF authorization.
- Support for IBM FTP compatible user exits. A guide to setting up and using these exits can be found here: http://dovetail.com/docs/sftp/coz_sftp_exits.pdf.
- Supports direct access to datasets which can be opened in sequential, record mode by the `fopen()` C-library routine. This includes:
 - MVS sequential datasets (QSAM, BSAM, VSAM)
 - PDS and PDSE members
 - SYSOUT datasets, including the MVS internal reader
- Supports JES2 and JES3 job submission, status and spool file transfer on z/OS 1.9 or later. Future releases of Co:Z SFTP will also support cancel and purge facilities.
- Supports text or binary conversion via flexible line-termination rules:
 - Cr, Lf/Newline, CrLf, Cr and/or Lf, RDW, none
- Supports flexible record padding / overflow rules
- Can specify dataset dynamic allocation (BPXWDYN) keywords

¹<http://tools.ietf.org/wg/secsh/draft-ietf-secsh-filexfer/>

- Can specify name patterns to automatically associate file transfer options to POSIX files and datasets

1.2 Supported Environments

z/OS Host Requirements

- z/OS V1R7 or later
- *IBM Ported Tools for z/OS (SSH)*

Remote System Requirements

- Co:Z SFTP is compatible with a wide variety of operating systems, including Windows, *IX variants, z/OS...
- sftp products supporting the SSH File Transfer Protocol, such as:
 - OpenSSH
 - puTTY psftp
 - winSCP
 - gFTP
 - many commercial implementations

2. Co:Z SFTP Configuration



Important

Before proceeding, ensure that the Co:Z Toolkit for z/OS has been successfully installed according to the instructions provided in the document "Co:Z Toolkit Installation and Release Notes" at <http://www.dovetail.com/docs/cozinstall/index.html>. Be sure to make note of the installation directory.

You do *not* need to install the Co:Z Target System Toolkit on your remote systems to use Co:Z SFTP - a compatible SSH/SFTP product is all that is required.

2.1 Configuring the Co:Z SFTP Server

The configuration discussed here is designed to allow individual users to use either the original sftp-server or the enhanced Co:Z version, depending on their configuration. The default setup makes for an ideal beta testing environment, as only designated users will use the enhanced Co:Z sftp-server.

Modifying the SFTP Subsystem

1. Update the `sshd_config` file, typically located at `/etc/ssh/sshd_config` to modify the sftp subsystem definition:¹

```
#Subsystem      sftp      /usr/lib/ssh/sftp-server      ❶
Subsystem      sftp      /usr/lpp/coz/bin/sftp-server.sh ❷
```

- ❶ The original sftp subsystem line should be commented out.
- ❷ The new subsystem line should point to the `sftp-server.sh` shell script located in the Co:Z installation `bin` directory. This script is designed to run the original sftp-server by default, but will run the Co:Z version if the user has configured it. See [the section called "Customizing sftp-server.rc"](#) for details. The installation process should have marked this file as executable, but this should be verified.

2. If OpenSSH `sshd` was running prior to editing `sshd_config`, it should be reinitialized. This can be done by sending `SIGHUP` to the running process. The pid for this process is typically in the file `/var/run/sshd.pid`:

```
kill -HUP `cat /var/run/sshd.pid`
```

¹It is sometimes convenient to set up a *test* OpenSSH server where this subsystem can be easily modified. To do this see: [Appendix G, Setting up a test OpenSSH system on z/OS](#).

Customizing `sftp-server.rc`

By default, the `sftp-server.sh` script discussed above will execute the IBM version of `sftp-server`. Individual users can activate the Co:Z version of `sftp-server` by copying a special profile shell to their home `.ssh` directory:

```
# if the user's .ssh does not exist:
mkdir $HOME/.ssh
chmod 700 $HOME/.ssh

cp /usr/lpp/coz/samples/sftp-server.rc $HOME/.ssh
chmod u+x $HOME/.ssh/sftp-server.rc
```

Note: Removing or renaming this file will re-enable the IBM version of `sftp-server`.

Making Co:Z `sftp-server` the default for all users

In order to make the Co:Z `sftp-server` the default for *all* users, place a copy of `sftp-server.rc` in `/etc/ssh`:

```
cp /usr/lpp/coz/samples/sftp-server.rc /etc/ssh
chmod +x /etc/ssh/sftp-server.rc
```

In this version of the profile, explicitly set the environment variable `USE_COZ_SFTP`:

```
USE_COZ_SFTP=true
```

Restricting OpenSSH users to SFTP

Some installations prefer to restrict `ssh` users to a certain set of commands like the `sftp-server`, rather than giving them interactive shell access. See [Appendix F, Restricting OpenSSH users to SFTP](#) for a technique to enforce this restriction.

Sample `sftp-server.rc`

A sample version of `sftp-server.rc` can be found in the `samples` directory of the installed Co:Z Toolkit. This file can be used as a template for individual users to modify and place in their `$HOME/.ssh` directory.

```
#!/bin/sh
# The presence of this executable script in $HOME/.ssh/sftp-server.rc
# will cause the COZ version of sftp-server to be used

# You may uncomment and set the following options to override the defaults:
#export SFTP_ZOS_OPTIONS="mode=text" ❶
#export SFTP_ZOS_INITIAL_DIR="// ❷
#export SFTP_LOGFILE=$HOME/sftp.log ❸

# The Co:Z support team may request that you uncomment the following options
# to enable tracing:
#export SFTP_SERVER_OPTIONS="-e -l debug3"
#export COZ_LOG=T
```

-
- ❶ The `SFTP_ZOS_OPTIONS` environment variable can be used to set the default options for the user. Multiple options may be specified, separated by commas. The options are described here: [*Appendix A, Co:Z SFTP options*](#).
 - ❷ The `SFTP_ZOS_INITIAL_DIR` environment variable can be used to override the home directory on the server. By default this is the user's USS home directory. If the string `//` is supplied, the user's MVS top level qualifier is used. Otherwise an absolute path (USS or MVS dataset space) may be supplied.
 - ❸ Logfiles are created for every sftp server session; these files are of particular interest in case a problem is encountered and additional error detail is needed. By default, logfiles are written to the `/tmp` directory (or the directory specified by the `TMPDIR` environment variable, if it is set). To change this default for all users, modify **`sftp-server.sh`** as needed. Individual users can override this setting by exporting `SFTP_LOGFILE` in the copy of `sftp-server.rc` in their individual `.ssh` directory.

2.2 Configuring the Co:Z SFTP Client

When the `cozsftp` command is invoked, the contents of the files `/etc/ssh/cozsftp_client.rc` and `$HOME/.ssh/cozsftp_client.rc` are dotted into the environment at the start of the command.

These files may be used to automatically provide command line arguments to the `cozsftp` client without having to explicitly code them every time the client is invoked. The desired command line arguments should be made available in the environment variable `COZSFTP_CLIENT_OPTS`.

For example, assume that the file `/etc/ssh/cozsftp_client.rc` is created with the following content:

```
#!/bin/sh
COZSFTP_CLIENT_OPTS="-S /u/vendor/oshxl/bin/ssh"
```

In this example, the `-S` option directs all invocations of **cozsftp** to use the *OpenSSH Accelerator* for its **ssh** program.

3. Using the Co:Z sftp server

3.1 Setting, displaying and clearing file transfer options

Unlike standard FTP, SFTP has no **site** command for setting platform specific options. Co:Z SFTP file transfer options are set with a special **ls** command request of the form: **ls /+<name=value>**. They can be cleared with a request of the form: **ls /+NO<name>**.

Multiple options can be set by separating the key=value pairs with commas. An error is returned if one or more of the options was incorrectly specified, but the remaining options are set as requested.

The options directory **/+ /** is a pseudo directory on the server, and it is possible to make it the working directory via the **cd /+** command. From this directory, options may be set and listed without the **/+** prefix.

The active options and their settings can be displayed by issuing the command **ls /+**.

Co:Z sftp server file transfer options may be specified interactively or via configuration files. The active options are determined in the following priority order:

1. The **fixed:** section of `/etc/ssh/cozsftp_server_config` (highest priority and non-modifiable)
2. The first matching pattern (if any) from `$HOME/.ssh/cozsftp_server_config`
3. The first matching pattern (if any) from `/etc/ssh/cozsftp_server_config`
4. Previous interactive commands: `ls /+` (described below) in the same session
5. The environment variable `SFTP_ZOS_OPTIONS`
6. The **default:** section of `/etc/ssh/cozsftp_server_config` (lowest priority)

For a list of available options, see [Appendix A, Co:Z SFTP options](#).

For a description of the `cozsftp_server_config` file format, including how to specify file name patterns, see [Appendix B, Session config files](#).

All examples in the following sections can be run by most sftp clients, either from z/OS or from other platforms (Windows, linux, etc..). **Note:** There are some differences in the way clients interact with the server, so the output shown in the examples below (performed with the OpenSSH sftp client) may not match your output exactly.

Example: Setting and displaying basic options

```
sftp> ls /+mode=text           ❶
/+mode=text
sftp> ls /+                   ❷
/+/clientcp=iso8859-1         /+/error.log
/+/loglevel=I                 /+/mode=text
/+/servercp=IBM-1047         /+/trim
```

- ❶ The option command `ls /+mode=text` is used to set the transfer mode to text. **mode=binary** is the default.
- ❷ The option list command `ls /+` shows the options currently in effect. In this case, the codepages `clientcp` and `servercp` are set to the defaults.

Example: Setting multiple options

```
sftp> ls /+lrecl=80,recfm=fb,space=trk.3.2 ❶
/+lrecl=80,recfm=fb,space=trk.3.2
```

- ❶ Multiple options can be specified, separated by commas. Note that the `SPACE` parameter uses periods for commas to avoid ambiguity.

Example: Showing all options

```
sftp> ls /+showall           ❶
/+showall=true
sftp> cd /+                 ❷
sftp> ls                    ❸
NOblksize                   NObufno                       clientcp=ISO8859-1
conddisp=catlg              NOcopies                       NOdataclas
NOdest                       NOdir                          NOdisp
NOdsntype                   NOdsorg                        error.log
estsize                     NOforms                        NOgdgnt
NOhold                      NOjesjobname                   NOjesjobwait
jeslrecl=80                 jesowner=KIRK                 jesrecfm=f
jesstatus=*                 NOlabel                        NOlike
linerule=flexible          loglevel=I                     lrecl=80
NOMaxvol                    NOmgmtclas                     mode=text
NOMount                     NONorecall                     NOoutdes
overflow=wrap              NOPad                          recfm=fb
NOrelease                   Noreplace                      NOreqexits
NOreset                     NOretpd                        NOsequence
servercp=IBM-1047          showall                         smf
space=trk.3.2              NOspin                         NOSTORCLAS
NOSysout                    trim                            NOTrtch
NOucount                    NOunit                          NOvol
```

```

NOwriter
sftp> ls noshowall,norecfm      ❶
noshowall,norecfm
sftp> ls
clientcp=ISO8859-1      error.log      loglevel=I
lrecl=80                mode=text      servercp=IBM-1047
space=trk.3.2          trim

```

- ❶ The option command **ls /+showall** is used to set the option listing mode to show all options, even those that are not active.
- ❷ Since the options are treated as entries in a pseudo directory, the **cd** command can be used to make that directory the working remote directory.
- ❸ Issuing the **ls** from the options directory will show all of the options. Those that are not active are prefixed with the string **NO**. Note that the options can be listed even if the current working directory is not the options pseudo dir with the command **ls /+**.
- ❹ Active options can be de-activated by prefixing the option with the string **NO**. In this example, the **showall** option is cleared, as well as the **recfm** option.

3.2 Reading the error log

Most implementations of the sftp specification, including OpenSSH, do not allow for transmission of detailed information from the server to the client in the event of an error. Adding dataset transfers to the mix only increases the need for better error reporting. To help alleviate this problem, the Co:Z sftp implementation provides a comprehensive logging facility that can be enabled and tuned by each user session.

Several of the above option listing examples show `error.log` as one of the options. This is actually an alias for the running session's log file, which is usually written to the `/tmp` directory (See [Chapter 2, Co:Z SFTP Configuration](#) for more information on where this file is written). This alias can be used to easily retrieve the log at anytime and examine it from the client.

This feature makes it possible to examine detailed error information from the client without having to abandon the active sftp session. Users of graphical clients such as **winSCP** and **gFTP** see an even greater advantage in that the `error.log` file can be viewed simply by selecting the file and transferring it in `view` mode.

Example: Getting and displaying the error log

```

sftp> rm //user.coz.sampjcl      ❶
Removing //user.coz.sampjcl
Couldn't delete file: Failure    ❷
sftp> get /+/error.log          ❸
Fetching /+/error.log to error.log
/+/error.log                    100%  68      0.1KB/s  00:00
sftp> !cat error.log           ❹
ZosUtil[E]: Dataset "USER.COZ.SAMPJCL" is a PDS. Use rmdir instead.
sftp>

```

- ❶ This command attempts to delete a PDS with the **rm** which is not allowed.
- ❷ The request fails, but the standard sftp error message is not very helpful.

- ③ To get better information, the `error.log` from the options directory is requested.
- ④ Using the local shell command `cat` to display the log gives detailed error information.

3.3 Working with Datasets

The Co:Z implementation of sftp accepts two prefix strings to identify MVS datasets as absolute paths. The first (//) is consistent with IBM's common usage. A secondary form (/-/) is also available, as not all sftp clients will allow double slash characters to be sent.

Navigating Datasets

The sftp **cd** command can be used to navigate around the z/OS dataset space. Using the dataset prefix // or /-/, the dataset space can be entered. Once there, traversal up and down various dataset levels can be performed similarly to hierarchical file systems.

Partitioned datasets are treated as directories as well. Once a PDS is made the current working directory, its members can be listed and retrieved like normal files.

Just as listing the entire catalog from the root is not allowed, it is not possible to make the catalog root the current working directory. As such, the command **cd //** will fail.

Example: Navigating the dataset space

```
sftp> cd //user           ❶
sftp> pwd                ❷
Remote working directory: //USER
sftp> cd coz.testjcl     ❸
sftp> pwd
Remote working directory: //USER.COZ.TESTJCL
sftp> cd ..             ❹
sftp> pwd
Remote working directory: //USER.COZ
```

- ❶ Using the dataset prefix //, the high level qualifier `user` is specified. For `cd` commands, the dataset name is case insensitive.
- ❷ The `pwd` command will list the current working dataset level. Note that the name is properly displayed in uppercase
- ❸ Multiple levels can be traversed at a time. Instead of using the normal separator (`.`), a slash can be used: `cd coz/testjcl`.
- ❹ The `cd ..` command will move up a level, as expected.

Transferring Datasets

The `get` and `put` commands are used to transfer datasets and PDS members. By default, the transfer mode is binary, and when storing new datasets, the DCB defaults are determined by the system and are often `RECFM=U`.

Any options previously set via the `ls /+option=value` are in effect for any given transfer.



Note

When using the `put` command to write datasets, the target name is used to determine the actual dataset name written. In most cases this determination is straight forward, but in certain circumstances, name

determination is more involved. See [Appendix C, Dataset Name Determination](#) for complete details.

Example: Get a text sequential dataset

```
$ sftp user@zos.com      ❶
Connecting to zos.com...
user@zos.com's password:
sftp> ls /+mode=text     ❷
/+mode=text
sftp> get //USER.LOG.MISC  ❸
Fetching //USER.LOG.MISC to USER.LOG.MISC
```

- ❶ This example shows the full connection process, using keyboard-interactive password authentication.
- ❷ The default transfer mode of binary is overridden and set to `text`.
- ❸ The `get` command uses the dataset path prefix `//` (or, optionally `/-/`) to specify that a dataset is being requested.

Example: Get PDS members

```
sftp> ls /+              ❶
/+/clientcp=ISO8859-1    /+/mode=text      /+/servercp=IBM-1047
sftp> get //user.ssh.jcl(sshd)  ❷
Fetching //user.ssh.jcl(sshd) to user.ssh.jcl(sshd)
```

- ❶ If this transfer is performed after the prior example, the transfer mode will still be `text`. Using the `ls /+` command quickly confirms the active options.
- ❷ The `get` command uses the dataset path prefix `//` and pds member name in parentheses to identify the member to get. Note again that the dataset name for transfers is case insensitive.

Example: Get a PDS member

```

sftp> get //user.coz.sampjcl(cozproc) cozproc.txt ❶
Fetching //user.coz.sampjcl(cozproc) to cozproc.txt

sftp> cd //user.coz.sampjcl ❷
sftp> get runcoz ❸
Fetching //USER.COZ.SAMPJCL/runcoz to runcoz

sftp> get * ❹
Fetching //USER.COZ.SAMPJCL/@@README to @@README
Fetching //USER.COZ.SAMPJCL/BPXBATCH to BPXBATCH
Fetching //USER.COZ.SAMPJCL/BPXBATSL to BPXBATSL
Fetching //USER.COZ.SAMPJCL/COZCFGD to COZCFGD
Fetching //USER.COZ.SAMPJCL/COZPROC to COZPROC
Fetching //USER.COZ.SAMPJCL/DTLSPAWN to DTLSPAWN
Fetching //USER.COZ.SAMPJCL/GPGDSN to GPGDSN
Fetching //USER.COZ.SAMPJCL/GREPDSN to GREPDSN
Fetching //USER.COZ.SAMPJCL/GREPSD to GREPSD
Fetching //USER.COZ.SAMPJCL/OFFLDSMF to OFFLDSMF
Fetching //USER.COZ.SAMPJCL/RUNCOZ to RUNCOZ
Fetching //USER.COZ.SAMPJCL/RUNCOZ2 to RUNCOZ2
Fetching //USER.COZ.SAMPJCL/RUNCOZ3 to RUNCOZ3
Fetching //USER.COZ.SAMPJCL/RUNSPAWN to RUNSPAWN
Fetching //USER.COZ.SAMPJCL/RUNSPWN2 to RUNSPWN2
Fetching //USER.COZ.SAMPJCL/TDIRK to TDIRK
Fetching //USER.COZ.SAMPJCL/WGET2DSN to WGET2DSN

```

- ❶ The **get** can be used to get a member from a fully qualified dataset.
- ❷ The **cd** command is used to make a PDS the current working "directory".
- ❸ The **get** command uses just the member name to retrieve the desired member.
- ❹ The **get *** command can be used to retrieve all members at once.

Example: Put a text MVS dataset, overriding DCB attributes

```

...
sftp> ls /+mode=text,lrecl=80,recfm=fb ❶
/+mode=text,lrecl=80,recfm=fb
sftp> put afile.txt //USER.AFILe.TXT ❷
Uploading afile.txt to //USER.AFILe.TXT

```

- ❶ The option command `ls /+mode=text,lrecl=80,recfm=fb` is used to set the transfer mode to text, and set the DCB attributes for the new dataset `USER.AFILe.TXT`. This overrides the system default for new datasets. Input lines will be broken on CR, LF, or CRLF and lines longer than allowed by the dataset will be wrapped onto multiple records. The options `linerule` and `overflowrule` can be used to override those settings.
- ❷ The **put** command uses the specialized path prefix `//` (or, optionally `/- /`) to specify the dataset name.

Listing datasets and PDS directories

MVS datasets can be listed using the sftp **ls** command. Partitioned datasets are treated as directories with their members as entries.

In order to support existing sftp clients, several considerations have to be made when listing datasets:

- The **ls** lists multiple dataset levels (by default), and therefore can return a large amount of information. As such, listings that would involve searching the entire catalog, such as **ls //** or **ls //A*** are not allowed. Furthermore, because of the way sftp clients interact with the server, the following style of command is not supported: **ls //USER.LVL1***. However, the same effect can be produced by either using directory notation for searching the catalog (**ls //USER/LVL1***) or changing to the desired level and issuing a relative listing command:

```
cd //USER
ls LVL1*
```

- Where possible, dataset names are treated as case insensitive. A **get** or **put** can specify the name in either lower or upper case and it will be found. However, any globbed (wildcard) **ls** command requires upper case characters. Individual datasets can be listed in either upper or lower case. To be safe, it is a good idea to use upper case on all list requests.
- As mentioned above, partitioned datasets (PDS) are treated as directories. the long list form **ls -l** can be used to list member statistics, if statistics exist.
- The **ls** accepts dataset names prefixed either by **//** or **/-/**. The second form should be used for the few sftp clients that do not allow a double slash to sent to the server.

Example: Listing datasets

```

sftp> cd //USER
sftp> ls -al ❶
Volume Referred Ext Tracks Used Recfm Lrecl BlkSz Dsorg Dsname
WORK84 2008/09/05 1 1 1 FB 80 27920 PS USER.AFILF.TXT
WORK81 2008/09/08 1 30 ? U 0 6144 PO-E USER.COZ.LOADLIB
WORK81 2008/09/11 1 15 4 FB 80 27920 PO USER.COZ.SAMPJCL
WORK84 2008/09/11 1 1 1 U 0 6144 PS USER.COZ.TEST.SEQ
WORK81 2008/09/09 1 15 3 FB 80 27920 PO USER.COZ.TESTJCL

sftp> cd //USER ❷
sftp> ls CO*
COZ.LOADLIB/ COZ.SAMPJCL/ COZ.TEST.SEQ COZ.TESTJCL/

sftp> ls //USER/CO* ❸
//USER/COZ.LOADLIB/ //USER/COZ.SAMPJCL/ //USER/COZ.TEST.SEQ
//USER/COZ.TESTJCL/

sftp> ls //USER.CO* ❹
Couldn't get handle: Failure
Can't ls: "//USER.CO*" not found

sftp> ls // ❺
Couldn't stat remote file: No such file or directory
Can't ls: "/" not found

sftp> cd //user
sftp> ls co* ❻
Can't ls: "//USER/co*" not found
sftp>

```

- ❶ The long form of the list command **ls -al** will list detailed information from the catalog about each dataset.
- ❷ Relative listing requests can be performed by first navigating to the desired level, then issuing the list request without any prefix.
- ❸ When using wildcards, the desired result can be achieved by using a slash (/) in place of the traditional level separator (.).
- ❹ Due to existing sftp client design, this list request would require the entire catalog to be searched, then filtered with the pattern `USER.CO*`. It is therefore disallowed.
- ❺ Lists that would involve the entire catalog are not supported. The `openssh` sftp client reports this as shown.
- ❻ The same command with a lower case pattern will fail (as described above).

Example: Listing a PDS directory

```

...
sftp> cd //user.coz.sampjcl ❶
sftp> ls -al ❷
Name Size Created Changed ID
@@README
BPXBATCH 13 2008/04/04 2008/04/04 17:18:09 USER
BPXBATSL 16 2008/04/03 2008/04/03 10:36:52 USER
COZCFGD 65 2008/03/27 2008/05/12 14:28:54 USER
COZPROC 30 2008/03/27 2008/03/27 11:54:48 USER
DTLSPAWN 40 2008/05/05 2008/05/05 09:31:08 USER

```

GPGDSN	15	2008/05/05	2008/05/05	10:40:05	USER
GREPDSN					
GREPSED	12	2008/05/05	2008/05/05	09:30:51	USER
OFFLDSMF					
RUNCOZ	20	2008/03/27	2008/05/12	14:08:02	USER
RUNCOZ2	15	2008/05/05	2008/05/05	10:02:51	USER
RUNCOZ3	8	2008/05/05	2008/05/06	08:50:37	USER
RUNSPAWN	54	2008/05/12	2008/05/12	14:25:37	USER
RUNSPWN2	20	2008/05/12	2008/05/12	13:19:05	USER
TDIRK	18	2008/04/03	2008/04/03	10:19:20	USER
WGET2DSN					

- ❶ The **cd** command is used to make a PDS the current working "directory".
- ❷ The **ls -al** command (long list form) is used to display the members of the PDS, including available statistics.

3.4 Working with POSIX files

This section describes how to use the Co:Z implementation of sftp with POSIX files (HFS, zFS) on z/OS. Standard sftp implementations (including IBM's ported tools version) support only binary mode file transfers. The Co:Z implementation provides binary transfer mode by default, but also supports text mode transfers. Text mode transfers are controlled via the following options:

- `mode`: when set to `text` causes file transfers to be text based.
- `clientcp` and `servercp`: When text mode is active, these settings determine the codepage translation that will take place. The default client code page is ISO8859-1. The default server code page is the current z/OS locale.
- `linerule`: When text mode is active, this setting determines how line separators are converted between the client and server.

Transferring Files

The `get` and `put` commands are used to transfer POSIX files (either on HFS or zFS filesystems).

The options (listed above) that have been previously set via the `ls /+option=value` are in effect for any given transfer. All other options (used for dataset support) are ignored for POSIX file transfers.

Example: Get a text POSIX file

```
sftp> ls /+mode=text,clientcp=UTF-8      ❶
/+mode=text,clientcp=UTF-8
sftp> ls /+                               ❷
/+/clientcp=UTF-8      /+/mode=text      /+/servercp=IBM-1047
sftp> get .ssh/sftp-server.rc           ❸
Fetching /u/user/.ssh/sftp-server.rc to sftp-server.rc
/u/user/.ssh/sftp-server.rc      100% 234      0.2KB/s    00:00
sftp>
```

- ❶ The default transfer mode of binary is overridden and set to `text`. Additionally, the client code page is explicitly set to `UTF-8`.
- ❷ Displays the active options. Note that the server code page, if not explicitly set, defaults to the current z/OS locale.
- ❸ The `get` command requests the transfer of the POSIX file using the options in effect.

Example: Put a text POSIX file

```
sftp> put sftp-server.rc .ssh           ❶
Uploading sftp-server.rc to /home/user/.ssh/sftp-server.rc
sftp-server.rc      100% 234      0.2KB/s    00:00
sftp>
```

- ❶ The client text file `sftp-server.rc` is put to the remote directory `.ssh` under the current working remote

directory. The active file transfer options are used.

3.5 Working with JES jobs and spool files

This section describes how to use Co:Z SFTP to submit jobs, query job status and access spool files on z/OS. Future releases of Co:Z SFTP will also support enhanced job cancel and purge facilities.

Note: Co:Z JES spool access supports both JES2 and JES3, but is currently limited to the primary JES subsystem.

Obtaining JES job status

To query the status of z/OS jobs, you simply list the "//-JES" pseudo-directory:

```
sftp> cd //-JES ❶
sftp> ls ❷
JOB00434 JOB00561 TSU00560 TSU00562
sftp> ls -al ❸
JOBNAME JOBID OWNER STATUS CLASS
KIRKL JOB00434 KIRK OUTPUT A RC=0000
TOMCAT JOB00561 KIRK ACTIVE A
KIRK TSU00560 KIRK OUTPUT TSU RC=0000
KIRK TSU00562 KIRK ACTIVE TSU
sftp> ls /+jesjobname=kirk ❹
/+jesjobname=kirk
sftp> ls -al
JOBNAME JOBID OWNER STATUS CLASS
KIRKL JOB00434 KIRK OUTPUT A RC=0000
KIRK TSU00560 KIRK OUTPUT TSU RC=0000
KIRK TSU00562 KIRK ACTIVE TSU
sftp> ls /+jesjobname=kirk. ❺
/+jesjobname=kirk.
sftp> ls -al
JOBNAME JOBID OWNER STATUS CLASS
KIRK TSU00560 KIRK OUTPUT TSU RC=0000
KIRK TSU00562 KIRK ACTIVE TSU
sftp> ls /+nojesjobname ❻
/+nojesjobname
sftp> ls /+jesowner=goetze ❼
/+jesowner=goetze
sftp> ls -al
JOBNAME JOBID OWNER STATUS CLASS
GOETZEB JOB00601 GOETZE OUTPUT A RC=0000
GOETZE TSU00505 GOETZE OUTPUT TSU RC=0000
GOETZE TSU00515 GOETZE ACTIVE TSU
sftp> ls /+jesstatus=active ❽
/+jesowner=goetze
sftp> ls -al
JOBNAME JOBID OWNER STATUS CLASS
GOETZE TSU00515 GOETZE ACTIVE TSU
```

- ❶ Change to the //-JES pseudo-directory.
- ❷ Listing the contents of the //-JES directory will by default display a list a job ids whose owner is the same as the current user.
- ❸ Requesting a detailed listing of the //-JES directory produces a formatted list of the same jobs. Note here how jobs are sorted lexically by jobid - this is actually being done by the sftp client. Sorting on most sftp clients can be disabled; in the case of OpenSSH, use the `-f` switch on the `ls` command, eg: `ls -alf` will display the jobs in the order returned by the JES subsystem interface.
- ❹ By default, all jobs owned by the current user are displayed. The `jesjobname` setting may be used to set a jobname filter.
- ❺ Terminating the `jesjobname` setting with a period filters on an exact jobname match, rather than a prefix.

- ⑥ The `jesjobname` setting is turned off.
- ⑦ By default, the `jesowner` setting is set to the current userid. Here it is changed to a different userid.
- ⑧ The `jesstatus` setting may be used to filter job listings by one of the following categories: `input`, `output`, or `active`.

Co:Z SFTP uses the unauthorized "Extended Status" subsystem interface to obtain job status. This facility is only available if you are running z/OS 1.9 or later. A SAF(RACF) SECLABEL dominance check may be used by the IBM extended status subsystem interface to control access to this facility; refer to RACF or your security product documentation for more information.

Transferring JES spool files

Job spool files may be transferred using normal SFTP "get" commands from your SFTP client.

```
sftp> cd //-JES
sftp> ls -al
JOBNAME  JOBID    OWNER    STATUS   CLASS
KIRKL    JOB00434 KIRK     OUTPUT   A        RC=0000
TOMCAT   JOB00561 KIRK     ACTIVE   A
KIRK     TSU00560 KIRK     OUTPUT   TSU      RC=0000
KIRK     TSU00562 KIRK     ACTIVE   TSU
sftp> cd JOB00434 ❶
sftp> ls ❷
102  2    3    4
sftp> ls -al ❸
DSID  STEPNAME  PROCSTEP  DDNAME    C OWNER    RECFM  LRECL  BYTES
 102  LOGDEF    SYSPRINT  A KIRK     FBA     133    5195
   2   JES2     JESMSGLG  A KIRK     FA      133    911
   3   JES2     JESJCL    A KIRK     V       136    271
   4   JES2     JESYSMSG  A KIRK     VA      137    839
sftp> ls -alf ❹
DSID  STEPNAME  PROCSTEP  DDNAME    C OWNER    RECFM  LRECL  BYTES
   2   JES2     JESMSGLG  A KIRK     FA      133    911
   3   JES2     JESJCL    A KIRK     V       136    271
   4   JES2     JESYSMSG  A KIRK     VA      137    839
 102  LOGDEF    SYSPRINT  A KIRK     FBA     133    5195
sftp> ls /+mode=text ❺
/+mode=text
sftp> get 102 logdef.text ❻
Fetching //-JES.JOB00434/102 to logdef.text
//-JES.JOB00434/102
sftp>
sftp> get * ❼
Fetching //-JES.JOB00434/102 to 102
//-JES.JOB00434/102
Fetching //-JES.JOB00434/2 to 2
//-JES.JOB00434/2
Fetching //-JES.JOB00434/3 to 3
//-JES.JOB00434/3
Fetching //-JES.JOB00434/4 to 4
//-JES.JOB00434/4
sftp>
sftp> get all concat.txt ❽
Fetching //-JES.JOB00434/all to concat.txt
sftp> get jesysmsg ❾
Fetching //-JES.JOB00434/jesysmsg to jesysmsg
sftp> get logdef.sysprint
Fetching //-JES.JOB00434/logdef.sysprint to logdef.sysprint
```

- ❶ Jobs are represented in Co:Z SFTP as *directories* under the //-JES pseudo-directory. Here we change the current directory to a specific job.

- ② Job spool files are represented as file names with the numeric JES DSID identifier.
- ③ A detailed listing displays a formatted list of spool files.
- ④ Many sftp clients will sort the files lexically by name (dsid). The `ls -f` switch on the OpenSSH sftp client will preserve the natural ordering, which is by numerical dsid.
- ⑤ The transfer mode is set to `text`.
- ⑥ The sftp `get` command can be used to download spool files.
- ⑦ A wildcard `get` command can be used to download all spool files in the job directory.
- ⑧ The special `ALL` file name can be used to transfer all spool files to a concatenated output file.
- ⑨ Spool files may also be referenced by `[step . [procstep .]]ddname`

Co:Z SFTP uses the unauthorized interface to the JES "Spool Browse" facility, which is only available if you are running z/OS 1.9 or later. As with IBM FTP, the SAF(RACF) JESSPOOL resource class is used to control access to spool files through the Spool Browse facility.

Submitting JES jobs

Jobs may be submitted to the JES internal reading using SFTP "put" commands from your SFTP client into a special pseudo-directory named "//-JES.INTRDR".

```
sftp> ls /+mode=text
/+mode=text
sftp> !cat jcl.txt ❶
//SLEEP3 JOB (),'Kirk Wolf',MSGCLASS=H
//UNIX EXEC PGM=COZBATCH
//STEPLIB DD DISP=SHR,DSN=KIRK.COZ.LOADLIB
//STDIN DD *
for i in 1 2 3
do
  echo "Sleeping..."
  sleep 1
done
//
sftp> cd //-jes.intrdr ❷
sftp> put jcl.txt myjob ❸
Uploading jcl.txt to //-JES.INTRDR/myjob

sftp> ls -al ❹
ALIAS      JOBNAME  JOBID     OWNER     STATUS    CLASS     COMPL
MYJOB      SLEEP3   JOB01941 KIRK      ACTIVE    A         
```

ALIAS	JOBNAME	JOBID	OWNER	STATUS	CLASS	COMPL
MYJOB	SLEEP3	JOB01941	KIRK	ACTIVE	A	

```

sftp> ls -al
ALIAS      JOBNAME  JOBID     OWNER     STATUS    CLASS     COMPL
MYJOB      SLEEP3   JOB01941 KIRK      OUTPUT    A         RC=0000

sftp> cd myjob ❺
sftp> ls -alf
DSID STEPNAME PROCSTEP DDNAME   C OWNER     RECFM LRECL BYTES
   2 JES2          JESMSG LG H KIRK     FA     133 1316
   3 JES2          JESJCL  H KIRK     V      136 373
   4 JES2          JESYSMSG H KIRK     VA     137 824
  102 UNIX          SYSOUT  H KIRK     FBA    121 308

sftp> get 2 2.txt
Fetching //-JES.INTRDR.MYJOB/2 to /tmp/2.txt

sftp> get all jobout.txt
Fetching //-JES.INTRDR.MYJOB/all to /tmp/jobout.txt

sftp> cd ..
sftp> put jcl.txt myjob2
sftp> ls -al
ALIAS      JOBNAME  JOBID     OWNER     STATUS    CLASS     COMPL
MYJOB      SLEEP3   JOB01941 KIRK      OUTPUT    A         RC=0000
MYJOB2     SLEEP3   JOB01943 KIRK      ACTIVE    A         
```

```
sftp> ls /+jesjobwait ❻
```

```

/+jesjobwait
sftp> cd myjob2
sftp> get all jobout2.txt
Fetching //-JES.INTRDR.MYJOB2/all to /tmp/jobout2.txt

sftp> ls /+jesjobwait=10.1 ⑦
sftp> ls /+nojesjobwait

```

- ❶ Run the `cat` command on the sftp client to display a file containing JCL.
- ❷ `//-JES.INTRDR` is a special pseudo-directory that contains any jobs submitted by the current session.
- ❸ The JCL is submitted by uploading it using the sftp client's `put` command. The target file name `MYJOB` is a *handle* that can be used to refer to this job later in the same session.
- ❹ Listing the `//-JES.INTRDR` directory displays all of the jobs that have been submitted in this session.
- ❺ `//-JES.INTRDR.MYJOB` is a directory that contains all of the spool files for the job referenced by this handle.
- ❻ The `jesjobwait` setting can be used to cause Co:Z sftp server to wait until the job completes before listing or transferring the jobs spool files.
- ❼ The default time limit to wait is 60 seconds with a polling interval of 2 seconds (60.2), but this can also be changed.

JES related options

The following table describes options that affect JES submit, status, and spool file transfer.

Table 3.1. JES related options

Name	Value	Notes
jesjobname	<input type="text" value="<pattern>"/>	The value of this setting is used as a filter when listing jobs. If the value doesn't end in a period, then the value is used as the job prefix. The default for this setting is <code>nojesjobname</code> , which means that jobs are not filtered by name.
jesowner	<input type="text" value="<userid>"/>	This setting specifies the userid used to filter job listings by job owner. The default for this setting is the current MVS userid, but may be set to <code>nojesowner</code> or <code>jesowner=*</code> to disable filtering by owner.
jesstatus	<input type="text" value="input output active"/>	This setting is used to filter job listings by job status. The default for this setting is <code>nojesstatus</code> .
jesjobwait	<input type="text" value="secs[.intvl]"/>	If enabled, this setting specifies the time in seconds to wait for a job to complete before listing or transferring its spool files. For most sftp clients, a <code>cd</code> to the job's spool file directory will also wait. If no value is specified, the default is <code>60.2</code> , which means to wait up to 60 seconds, polling every 2 seconds. The default for this setting is <code>nojesjobwait</code> .

Name	Value	Notes
jeslrecl	<input data-bbox="399 317 699 380" type="text" value="<numeric>"/>	This setting specifies the lrecl used when submitting jobs to the internal reader. The default for this setting is 80.
jesrecfm	<input data-bbox="399 451 699 514" type="text" value="f fb v vb"/>	This setting specifies the record format used when submitting jobs to the JES internal reader. The default for this setting is F.

4. Using the Co:Z sftp client

An enhanced sftp client (**cozsftp**) for z/OS is also included in the Co:Z toolkit. This client can be used to initiate transfers with a remote host and supports the same set of file transfer options as the Co:Z sftp server. The **cozsftp** command is installed in the **\$COZ_HOME/bin** directory.

4.1 Starting the Co:Z sftp client on z/OS

```
$ export PATH=/opt/dovetail/coz/bin:$PATH ❶
$ cozsftp user@host
Co:Z sftp version: 1.1.0 (5.0p1) 2008-10-20
Copyright (C) Dovetailed Technologies, LLC. 2008. All rights reserved.
Connecting to host...
user@host's password: *****
cozsftp>
```

- ❶ Add the Co:Z binaries directory to your PATH. This is not necessary if symbolic links from **/bin** were created during installation.

4.2 Setting, displaying and clearing file transfer options

The enhanced client introduces two new commands:

```
lzopts [-a] [option=value, ...]
```

The **lzopts** command is used to set local (client) file transfer options. These options are set prior to initiating file/dataset transfers from z/OS to a remote host.

```
zopts [-a] [option=value, ...]
```

The **zopts** command is used to set server file transfer options -- if the server is a Co:Z sftp server. The **zopts** command is functionally equivalent to the **ls /+<option_list>** command used by existing clients to set Co:Z sftp-server file transfer options.

Multiple options can be set by separating the option=value pairs with commas. An error is returned if one or more of the options was incorrectly specified, but the remaining options are set as requested.

The active options and their settings can be displayed by issuing the commands without arguments. The **-a** option can be specified to list *all* available options, even those that are not active.



Note

For compatibility with IBM Ported Tools SFTP, the **cozsftp** command recognizes the following additional subcommands: **ascii** and **binary**. these subcommands are treated as synonyms for **lzopts mode=text** and **lzopts mode=binary** respectively.

Client session options are determined in the following priority order:

1. The `fixed:` section of `/etc/ssh/cozsftp_config` (highest priority and non-modifiable)
2. The first matching pattern (if any) from `$HOME/.ssh/cozsftp_config`
3. The first matching pattern (if any) from `/etc/ssh/cozsftp_config`
4. Previous interactive commands: `lzopts` (described below) in the same session
5. The environment variable `SFTP_ZOS_OPTIONS`
6. The `default:` section of `/etc/ssh/cozsftp_config` (lowest priority)

For a list of available options, see [*Appendix A, Co:Z SFTP options*](#).

For a description of the `cozsftp_config` file format, including how to specify file name patterns, see [*Appendix B, Session config files*](#).

Example: Setting and displaying local (client) transfer options

```
cozsftp> lzopts mode=text ❶
mode=text
cozsftp> lzopts ❷
clientcp=IBM-1047 loglevel=I mode=text
servercp=IBM-1047
```

- ❶ The local option command `lzopts mode=text` is used to set the transfer mode to text. **mode=binary** is the default.
- ❷ The local option list command `lzopts` shows the options currently in effect. In this case, the codepages `clientcp` and `servercp` are set to the defaults.

Example: Setting multiple local options

```
sftp> lzopts lrecl=80,recfm=fb,space=trk.3.2 ❶
lrecl=80,recfm=fb,space=trk.3.2
```

- ❶ Multiple options can be specified, separated by commas. Note that the `SPACE` parameter uses periods for commas to avoid ambiguity.

Example: Showing all local options

```
cozsftp> lzopts -a ❶
clientcp=IBM-1047 linerule=flexible loglevel=I lrecl=80
mode=text overflow=wrap recfm=fb servercp=IBM-1047
space=trk.3.2 NOallowmount NOblksize NObufno
NOCopies NOdataclas NOdest NODir
NODisp NODsorg NOforms NOgdgnt
NOhold NOlabel NOlike NOMaxvol
NOMgmtclas NONorecall NOoutdes Norelease
Noret pd NOsequence NOshowall NOspin
NOSTORCLAS NOSysout NOTrim NOTrtch
NOucount NOunit NOvol NOWriter
```

- ❶ The option command `lzopts -a` is used to show all of the available options, even those that are not currently active.

4.3 Coordinating Transfer Options with a Co:Z SFTP Server

The enhanced Co:Z sftp client can connect to any sftp server, including a Co:Z SFTP server. In this case, there are two sets of transfer options in effect; the enhanced client's and the server's. Client side (local) options are controlled via the `lzopts` command. Server side (remote) options are controlled via the `zopts` command.

When transferring POSIX files between a z/OS server and z/OS client, using the default `mode=binary` transfer option both locally and remotely will usually yield the desired results. If codepage translations need to take place, the desired `clientcp`, `servercp` and `mode=text` can be set either locally (via **lzopts** command) or remotely (via the **zopts** command). The other side can be left in `mode=binary`.

When transferring datasets between a z/OS server and z/OS client, it is generally recommended that `linerule=rdw` be used for binary transfers so that record mode boundaries are preserved.

When converting from dataset to POSIX file between a z/OS server and z/OS client, the transfer options should be set where the dataset resides.

4.4 Working with Datasets

The Co:Z implementation of sftp accepts two prefix strings to identify MVS datasets as absolute paths. The first (//) is consistent with IBM's common usage. A secondary form (/-/) is also available.

Navigating Datasets

The sftp **lcd** command can be used to navigate around the z/OS dataset space. Using the dataset prefix // or /-/, the dataset space can be entered. Once there, traversal up and down various dataset levels can be performed similarly to hierarchical file systems.

Partitioned datasets are treated as directories as well. Once a PDS is made the current working directory, its members can be listed and retrieved like normal files.

Just as listing the entire catalog from the root is not allowed, it is not possible to make the catalog root the current working directory. As such, the command **lcd //** will fail.

Example: Navigating the dataset space

```
sftp> lcd //user           ❶
sftp> lpwd                ❷
Local working directory: //USER
sftp> lcd coz.testjcl     ❸
sftp> lpwd
Local working directory: //USER.COZ.TESTJCL
sftp> lcd ..             ❹
sftp> lpwd
Local working directory: //USER.COZ
```

- ❶ Using the dataset prefix //, the high level qualifier `user` is specified. For `lcd` commands, the dataset name is case insensitive.
- ❷ The **lpwd** command will list the current working dataset level. Note that the name is properly displayed in uppercase
- ❸ Multiple levels can be traversed at a time. Instead of using the normal separator (`.`), a slash can be used: **lcd coz/testjcl**.
- ❹ The **lcd ..** command will move up a level, as expected.

Transferring Datasets

The `get` and `put` commands are used to transfer datasets and PDS members.

Any options previously set via the `lzopts` are in effect for any given transfer.

Example: Get a file to a text sequential dataset

```
$ sftp user@linux.com ❶
Connecting to linux.com...
user@linux.com's password:
sftp> lzopts mode=text ❷
mode=text
sftp> lzopts
clientcp=IBM-1047 loglevel=I mode=text
servercp=ISO8859-1
sftp> get /tmp/GPGDSN //USER.GPGDSN ❸
Fetching /tmp/GPGDSN to //USER.GPGDSN
ZosDataset[I]: Opening dataset USER.GPGDSN for write with options: new catalog
/tmp/GPGDSN 100% 1215 1.2KB/s 00:00
ZosDataset[I]: Closing dataset //USER.GPGDSN - 1215 bytes received, 15 records written
ZosSmf119Record[I]: SMF Type119 recording not enabled; SMF recording disabled
```

- ❶ This example shows the full connection process, using keyboard-interactive password authentication to a remote linux system.
- ❷ The default transfer mode of binary is overridden and set to `text`.
- ❸ The `get` command uses the dataset path prefix `//` (or, optionally `/- /`) to specify that a dataset is to be written. At the default log level of `I` (INFO), information is emitted about the transfer process. Note also that in this case, SMF recording is disabled because the FTP SMF records (type 119) are not currently configured for recording.

Example: Get a text file to a PDS member

```
sftp> lzopts ❶
clientcp=IBM-1047 loglevel=I mode=text
servercp=ISO8859-1
sftp> lcd //user.coz.testjcl
sftp> lpwd
Local working directory: //USER.COZ.TESTJCL
sftp> get /tmp/GPGDSN ❷
Fetching /tmp/GPGDSN to //USER.COZ.TESTJCL/GPGDSN
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(GPGDSN) for write with options: old
/tmp/GPGDSN 100% 1215 1.2KB/s 00:00
ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(GPGDSN) - 1215 bytes received, 15 records written
```

- ❶ If this transfer is performed after the prior example, the transfer mode will still be `text`. Using the `lzopts` command quickly confirms the active options.
- ❷ The `get` command uses the dataset path prefix `//` and pds member name in parentheses to identify the member to create.

Example: Put PDS members

```

sftp> lpwd
Local working directory: //USER.COZ.TESTJCL
sftp> put ONETEST /tmp/ONETEST ❶
Uploading ///USER.COZ.TESTJCL(ONETEST) to /tmp/ONETEST
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(ONETEST) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(ONETEST) - 38 records read, 3078 bytes sent

sftp> put //USER.coz.testjcl ❷
Uploading //USER.COZ.TESTJCL(@@README) to /tmp/@@README
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(@@README) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(@@README) - 34 records read, 2754 bytes sent
Uploading //USER.COZ.TESTJCL(ALLOCDS) to /tmp/ALLOCDS
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(ALLOCDS) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(ALLOCDS) - 6 records read, 486 bytes sent
Uploading //USER.COZ.TESTJCL(CHKENVD) to /tmp/CHKENVD
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(CHKENVD) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(CHKENVD) - 1 records read, 81 bytes sent
Uploading //USER.COZ.TESTJCL(CHKPOST) to /tmp/CHKPOST
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(CHKPOST) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(CHKPOST) - 6 records read, 486 bytes sent
Uploading //USER.COZ.TESTJCL(CHKPRE) to /tmp/CHKPRE
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(CHKPRE) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(CHKPRE) - 72 records read, 5832 bytes sent
Uploading //USER.COZ.TESTJCL(COZCFGO) to /tmp/COZCFGO
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(COZCFGO) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(COZCFGO) - 1 records read, 81 bytes sent
Uploading //USER.COZ.TESTJCL(GPGDSN) to /tmp/GPGDSN
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(GPGDSN) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(GPGDSN) - 15 records read, 1215 bytes sent
Uploading //USER.COZ.TESTJCL(ONETEST) to /tmp/ONETEST
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(ONETEST) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(ONETEST) - 38 records read, 3078 bytes sent
Uploading //USER.COZ.TESTJCL(TESTPROC) to /tmp/TESTPROC
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(TESTPROC) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(TESTPROC) - 111 records read, 8991 bytes sent
Uploading //USER.COZ.TESTJCL(USERTEST) to /tmp/USERTEST
ZosDataset[I]: Opening dataset USER.COZ.TESTJCL(USERTEST) for read with options: shr

ZosDataset[I]: Closing dataset //USER.COZ.TESTJCL(USERTEST) - 187 records read, 15147 bytes sent

```

- ❶ The **put** command can be used to transfer a member from a fully qualified dataset.
- ❷ When the **put** command is used on a PDS, all of the members are uploaded.

Listing datasets and PDS directories

MVS datasets can be listed using the sftp **lls** command. Partitioned datasets are treated as directories with their members as entries.

When listing z/OS datasets locally with the **lls** command, catalog search filter keys are in effect for any wildcard requests. The catalog search wildcards *****, ******, and **%** used in the examples below are described in the IBM manual *DFSMS: Managing Catalogs - SC26-7409*. Note that this is different behavior from sftp clients that connect to the Co:Z sftp-server and list datasets with the **ls**. In that case, regular file globbing rules are in effect.

Example: Listing datasets

```
sftp> lcd //USER
sftp> lls -al ❶
Volume Referred Ext Tracks Used Recfm Lrecl BlkSz Dsorg Dsname
WORK84 2008/09/05 1 1 1 FB 80 27920 PS USER.AFILE.TXT
WORK81 2008/09/08 1 30 ? U 0 6144 PO-E USER.COZ.LOADLIB
WORK81 2008/09/11 1 15 4 FB 80 27920 PO USER.COZ.SAMPJCL
WORK84 2008/09/11 1 1 1 U 0 6144 PS USER.COZ.TEST.SEQ
WORK81 2008/09/09 1 15 3 FB 80 27920 PO USER.COZ.TESTJCL

sftp> lls -al //user.coz.t* ❷
Volume Referred Ext Tracks Used Recfm Lrecl BlkSz Dsorg Dsname
WORK81 2008/10/20 1 15 4 FB 80 27920 PO USER.COZ.TESTJCL

sftp> lls -al //user.c*.* ❸
Volume Referred Ext Tracks Used Recfm Lrecl BlkSz Dsorg Dsname
WORK81 2008/10/20 1 30 ? U 0 6144 PO-E USER.COZ.LOADLIB
WORK81 2008/10/20 1 15 4 FB 80 27920 PO USER.COZ.SAMPJCL
WORK84 2008/09/25 1 1 1 U 0 6144 PS USER.COZ.TEST.SEQ
WORK81 2008/10/20 1 15 4 FB 80 27920 PO USER.COZ.TESTJCL
```

- ❶ The long form of the list command **lls -al** will list detailed information from the catalog about each dataset.
- ❷ Using the catalog search filter key syntax, a single asterisk can be used to as a wildcard for a single dataset level.
- ❸ Using the catalog search filter key syntax, a double asterisk can be used to perform a *deep* listing. In this example, the single and double asterisk syntax is combined to list all of the datasets beginning with the prefix **USER.C**.

Example: Listing a PDS directory

```
...
sftp> lcd //user.coz.sampjcl ❶
sftp> lls -al ❷
Name Size Created Changed ID
@@README
BPXBATCH 13 2008/04/04 2008/04/04 17:18:09 USER
BPXBATSL 16 2008/04/03 2008/04/03 10:36:52 USER
COZCFGD 65 2008/03/27 2008/05/12 14:28:54 USER
COZPROC 30 2008/03/27 2008/03/27 11:54:48 USER
DTLSPAWN 40 2008/05/05 2008/05/05 09:31:08 USER
GPGDSN 15 2008/05/05 2008/05/05 10:40:05 USER
```

GREPDSN						
GREPSED	12	2008/05/05	2008/05/05	09:30:51	USER	
OFFLDSMF						
RUNCOZ	20	2008/03/27	2008/05/12	14:08:02	USER	
RUNCOZ2	15	2008/05/05	2008/05/05	10:02:51	USER	
RUNCOZ3	8	2008/05/05	2008/05/06	08:50:37	USER	
RUNSPAWN	54	2008/05/12	2008/05/12	14:25:37	USER	
RUNSPWN2	20	2008/05/12	2008/05/12	13:19:05	USER	
TDIRK	18	2008/04/03	2008/04/03	10:19:20	USER	
WGET2DSN						

- ❶ The **lcd** command is used to make a PDS the current working local "directory".
- ❷ The **lls -al** command (long list form) is used to display the members of the PDS, including available statistics.

4.5 Working with POSIX files

This section describes how to use the enhanced client with POSIX files (HFS, zFS) on z/OS. Standard sftp implementations (including IBM's ported tools version) support only binary mode file transfers. The Co:Z implementation provides binary transfer mode by default, but also supports text mode transfers. Text mode transfers are controlled via the following options:

- `mode`: when set to `text` causes file transfers to be text based.
- `clientcp` and `servercp`: When text mode is active, these settings determine the codepage translation that will take place. The default client code page is ISO8859-1. The default server code page is the current z/OS locale.
- `linerule`: When text mode is active, this setting determines how line separators are converted between the client and server.

Transferring Files

The `get` and `put` commands are used to transfer POSIX files (either on HFS or zFS filesystems).

The options (listed above) that have been previously set via the `lzopts` are in effect for any given transfer. All other options (used for dataset support) are ignored for POSIX file transfers.

Example: Get a text POSIX file

```
$ sftp user@linux.com           ❶
Connecting to linux.com...
user@linux.com's password: *****
sftp> lzopts mode=text,servercp=UTF-8           ❷
  mode=text          servercp=UTF-8
sftp> lzopts           ❸
clientcp=IBM-1047    loglevel=I                mode=text
servercp=UTF-8
sftp> pwd
Remote working directory: /tmp
sftp> get msgs.txt           ❹
Fetching /tmp/msgs.txt to msgs.txt
/tmp/msgs.txt                               100%  19KB  19.0KB/s  00:00
ZosPosixFile[I]: Closing file msgs.txt - 19488 bytes received, 19488 bytes written❺
sftp>
```

- ❶ This example shows the full connection process, using keyboard-interactive password authentication to a remote linux system.
- ❷ The default transfer mode of binary is overridden and set to `text`. Additionally, the server (linux) code page is explicitly set to `UTF-8`.
- ❸ Displays the active options. Note that the client code page, if not explicitly set, defaults to the current z/OS locale.
- ❹ The `get` command requests the transfer of the POSIX file using the options in effect.
- ❺ Upon completion, an informational message is written that describes the number of bytes received from the server and the number of bytes written to the local file. These counts are commonly the same, but changes in line separators and codepages can result in different counts.

Example: Put a text POSIX file

```
sftp> put sftp-server.log /tmp ❶
Uploading sftp-server.log to /tmp/sftp-server.log
sftp-server.log                100% 127      0.1KB/s   00:00
ZosPosixFile[I]: Closing file sftp-server.log - 127 bytes read, 127 bytes sent
ZosSmf119Record[I]: SMF Type119 recording not enabled; SMF recording disabled
```

- ❶** The client text file `sftp-server.log` is put to the remote directory `/tmp`. The active file transfer options are used.

4.6 Using the Co:Z sftp client in batch

The **cozsftp** client command can be conveniently used in a z/OS batch job without user interaction. The **COZBATCH** batch utility, also installed as part of the Co:Z toolkit, makes it easy to run **cozsftp** (or other Unix shell scripts) directly as z/OS batch jobs.

The authentication with the remote system must be set up so as not to require any user interaction. There are three ways to do this with OpenSSH:

- Use the `SSH_ASKPASS` environment variable to point to a program that will read a password.
- Use an OpenSSH public/private keypair.
- Use a RACF Digital Certificate.

For details on these three authentication options, see [Appendix E, Client Authentication Mechanisms](#). Note that instructions in this appendix must be followed in order to run the examples described below.

Example: Co:Z sftp client batch job with a password

The **RUNSFTP** sample JCL distributed with the Co:Z toolkit can be used as a tailorable model for running the sftp client in a batch job.

```
//RUNSFTP EXEC PGM=COZBATCH ❶
//STDIN DD * ❷

# Customize these ... ❸
coz_bin="/opt/dovetail/coz/bin"
ruser="uid"
server="remote.host.name"
servercp="ISO8859-1"
rfile="/path/to/file"

# These can be used to read the ssh password from a (secured) dataset
# if you don't want to setup public/private keypairs ❹
export PASSWD_DSN='//HLQ.PASSWD(SITE1)'
export SSH_ASKPASS=$coz_bin/read_passwd_dsn.sh
export DISPLAY=none

ssh_opts="-oBatchMode=no" # allows ssh to use SSH_ASKPASS program
ssh_opts="$ssh_opts -oConnectTimeout=60"
ssh_opts="$ssh_opts -oServerAliveInterval=60"
ssh_opts="$ssh_opts -oStrictHostKeyChecking=no" # accept initial host keys

# Invoke the Co:Z sftp client with an in-line batch of commands
# that downloads a remote file to a local DD.
# Note that "-oBatchMode=no" must be specified before "-b"
# since ssh opts are first-sticky

$coz_bin/cozsftp $ssh_opts -b- $ruser@$server <<EOB ❺
lzopts mode=text,servercp=$servercp
get $rfile //DD:DOWNLOAD ❻
```

```

EOB
//DOWNLOAD DD DSN=&&DOWNLOAD,DISP=(NEW,DELETE),
//          DCB=(...),SPACE=(...)
//

```

- ❶ The **COZBATCH** utility is similar to IBM's BPXBATCH, but runs a Unix login shell in the original address space.
- ❷ The **STDIN** DD is used to read a shell script as input to the z/OS Unix shell. This script will contain the setup and execution of Co:Z sftp in batch.
- ❸ Customize these variables for your needs.
- ❹ The following variables are set so that the ssh password is read from a z/OS dataset. Alternatively, an OpenSSH keypair can be used.
- ❺ The Co:Z sftp client is invoked with the **-b** option, which indicates that a "batch" of commands should be read from stdin. The shell "here" document redirection is used to read stdin from the shell script itself.
- ❻ The sftp **get** command is used to get a remote file and write it to DD **DOWNLOAD**. Alternatively, the syntax **//HLQ.MY.DATASET** could have been used to download to a dynamically allocated dataset.

Example: Co:Z sftp client batch job with a RACF Digital Certificate

The **RUNSFTPK** sample JCL distributed with the Co:Z toolkit can be used as a tailorable model for running the sftp client in a batch job with certificates.

Note: A RACF Digital Certificate must be setup for the user prior to running this example. See the [Co:Z customization](#) section of the Co:Z Toolkit User's Guide for details.

```

//RUNSFTP EXEC PGM=COZBATCH
//STDIN DD *

# Customize these ...
coz_bin="/opt/dovetail/coz/bin"
ruser="uid"
server="remote.host.name"
servercp="ISO8859-1"
rfile="/path/to/file"

ssh_opts="-oConnectTimeout=60"
ssh_opts="$ssh_opts -oServerAliveInterval=60"
ssh_opts="$ssh_opts -oStrictHostKeyChecking=no" # accept initial host keys

# Invoke the Co:Z sftp client with an in-line batch of commands
# that downloads a remote file to a local DD.
# Note that "-k MY-RING" will cause cozsftp to use the Co:Z
# saf-ssh-agent to authenticate the ssh client using the user's
# RACF Digital Certificate in MY-RING

$coz_bin/cozsftp $ssh_opts -k MY-RING -b- $ruser@$server <<EOB ❶
lzopts mode=text,servercp=$servercp
get $rfile //DD:DOWNLOAD
EOB

//DOWNLOAD DD DSN=&&DOWNLOAD,DISP=(NEW,DELETE),

```

```
//          DCB=(...),SPACE=(...)
//
```

- ❶ The `-k MY-RING` option is used to specify the RACF keyring to use for certificate authentication.

Example: Using saf-ssh-agent to run remote SSH commands

The **RUNSFTPS** is similar to the previous example, but performs a put rather than a get. After the result of the put is checked, the `saf-ssh-agent` command is used to invoke `ssh` to process the new remote file.

Note: A RACF Digital Certificate must be setup for the user prior to running this example. See the [Co:Z customization](#) section of the Co:Z Toolkit User's Guide for details.

```
//RUNSFFTP EXEC PGM=COZBATCH
//STDIN DD *

# Customize these ...
coz_bin="/opt/dovetail/coz/bin"
ruser="uid"
server="remote.host.name"
servercp="ISO8859-1"
rfile="/path/to/file"

ssh_opts="-oConnectTimeout=60"
ssh_opts="$ssh_opts -oServerAliveInterval=60"
ssh_opts="$ssh_opts -oStrictHostKeyChecking=no" # accept initial host keys

# Invoke the Co:Z sftp client with an in-line batch of commands
# that uploads a local dataset to a remote file and then runs
# ssh to invoke a remote command to process the uploaded file
# Note that "-k MY-RING" will cause cozsftp to use the Co:Z
# saf-ssh-agent to authenticate the ssh client using the user's
# RACF Digital Certificate in MY-RING

$coz_bin/cozsftp $ssh_opts -k MY-RING -b- $ruser@$server <<EOB ❶
lzopts mode=text,servercp=$servercp
put //DD:UPLOAD $rfile
EOB

if [ $? -eq 0 ]; then
    $coz_bin/saf-ssh-agent -c MY-RING ssh $ruser@$server /bin/chk $rfile ❷
fi

//UPLOAD DD DSN=HLQ.UPLOAD.DATA,DISP=SHR
//
```

- ❶ The `-k MY-RING` option is used to specify the RACF keyring to use for certificate authentication.
- ❷ After checking for a successful upload, the `saf-ssh-agent` command is used to invoke the `ssh` client to perform some action `/bin/chk` on the remote file.

Appendix A. Co:Z SFTP options

A.1 General transfer options

The following table describes the general transfer options. The usage columns describe when the option (if active) will apply during transfer. Usage *Read* means using Co:Z SFTP (server or client) to read a z/OS file. Usage *Write* means using Co:Z SFTP (server or client) to write a z/OS file.



Note

Options that have a blank *value* column are on/off options. They are activated by supplying the option name by itself (no values allowed) and deactivated by prefixing the option name with the prefix `NO`. For example: `trim` and `notrim`.

Table A.1. General transfer options

Keyword		Usage			
Name	Value	Datasets	POSIX	Read	Write
<code>clientcp</code>	<codepage>	X	X	X	X
<code>linerule</code>	<code>cr</code> <code>crlf</code> <code>crnl</code> <code>lf</code> <code>nl</code> <code>rdw</code> <code>mfrdw</code> <code>flexible</code> <code>0xbb[bb..]</code> <code>none</code>	X	X	X	X
<code>mode</code>	<code>binary</code> <code>text</code>	X	X	X	X
<code>overflow</code>	<code>error</code> <code>flow</code> <code>trunc</code> <code>wrap</code>	X			X
<code>pad</code>	<pad_char> <code>0xbb</code>	X			X
<code>replace</code>		X	X		X
<code>servercp</code>	<codepage>	X	X	X	X
<code>trim</code>		X		X	

`clientcp`

Specifies the name of the client codepage used when performing text mode transfers. Data will be converted between the server codepage (`servercp`) and this code page. The codepage must either be a single byte codepage or any multi-byte codepage that has single-byte line terminators (e.g. UTF-8). The z/OS Unix command `iconv -l` lists the available codepages. The default, if not specified, is ISO8859-1.

`linerule`

The values `cr`, `crlf`, `crnl`, `lf`, `nl` specify that, for text mode transfers, lines will be terminated with the given characters in the client codepage. `flexible` may be used when writing to files or datasets to indicate

that any combination of `cr`, `lf`, or `newline` will be recognized as a line terminator. `rdw` specifies that IBM-style RDWs are used as prefixes. `mfrdw` indicates that Micro-focus file and records headers are used. `0xbb[bb. .]` may be used to specify a sequence of one or more bytes in the source codepage. The default is `flexible` for writing and `lf` for reading.

mode

Specifies whether transfers are as-is (binary) or subject to codepage/linerule/overflow/pad processing (text). The default is `binary`.

overflow

For text-mode dataset write processing, controls the treatment of lines longer than the maximum dataset record length.

pad

For text-mode dataset write processing, specifies the character to use when padding lines into fixed-length dataset records. If given as `0xbb`, it specifies (in hex) a single-byte character in the source codepage. If not specified, the default is a space character in the server codepage.

replace

This setting allows for existing datasets or files to be replaced. The default, if not specified, is `replace`, which allows for replacement. `noreplace` can be set to prevent an existing dataset or file from being replaced. Note: if `noreplace` is set, you may not create PDS members, regardless of whether the member exists, and you may also not write to GDG datasets using a positive (+n) relative reference.

servercp

Specifies the name of the server codepage used when performing text mode transfers. Data will be converted between the client codepage (`clientcp`) and this code page. The default, if not specified, is `default codepage` for the server's process.

trim

For text-mode dataset read transfers, enabling this options will cause pad characters to be trimmed from the dataset records as they are read.

A.2 Miscellaneous options

The following table describes the miscellaneous options. These options do not apply to transfer operations, but affect the behavior of Co:Z SFTP.

Table A.2. Miscellaneous options

Name	Value	Notes
loglevel	E W N I D T F	Sets the logging level of the Co:Z sftp-server. The UPPER CASE values correspond to the list: (Error, Warning, Info, Notice, Debug, Trace, Fine).
reqexits	exit[.exit]...	(server only) For each exit listed, the corresponding loadmodule must be available and loaded. If not, an error message will be written to the log and the server session immediately terminated. The exit names that may be specified are: CZCHKCMD, CZCHKIP, CZCHKPWD, and CZPOSTPR. Installations that use exits will typically set this option in the <code>fixed:</code> section of <code>/etc/ssh/cozsftp_server_config</code> .
showall		If active, all options will be shown on option display (ls /+). Inactive options are shown with a prefix of NO .
smf		If active (the default), SMF 119 records will be written for file transfer events. if a failure occurs because SMF is disabled, this option will be automatically set to NOsmf and no further attempts will be made. To completely disable SMF recording supply "nosmf" in one of the config files: Appendix B, Session config files . This option may not be set or changed after the session has started.
ssh-le-options		(Co:Z SFTP client only) This option may be used to supply z/OS Language Environment options for the IBM Ported Tools ssh command when it is invoked by cozsftp . This option may not be set or changed once the session has started. Referring to IBM APAR OA34819, we suggest that customers adopt the value for this setting that is in the sample <code>cozsftp_config</code> file supplied with the distribution.

A.3 Dataset allocation options

The following table describes options that apply when transferring MVS datasets. The z/OS BPXWDYN service is used for dataset allocation and these options correspond to keywords available with BPXWDYN with similar syntax except that:

- `keyword=value` is used rather than `keyword(value)`
- periods are used in place of commas
- other minor differences as described below

The usage columns below describe when the option (if active) will apply during dataset transfer (none of these options, with the exception of `conddisp`, apply during POSIX file transfers). For more information on BPXWDYN, see *Using REXX and z/OS UNIX System Services - SA22-7806*



Note

The `conddisp` option is not a BPXWDN keyword, but is supported by Co:Z SFTP to handle dataset / POSIX file deletion in the result of a transfer error. If this option is set to `delete`, Co:Z SFTP server will attempt to delete any file or dataset that is being written to (on the client or the server) if the transfer is interrupted. In the OpenSSH sftp client, a Ctrl-C (SIG-INT) is caught in the client and it just closes the file, so there is no way for the server to see this as an interruption. In this case, it is still the client's responsibility to clean up the file.

Table A.3. BPXWDYN options

Keyword		Usage			
Name	Value	Read	Write New	Write Existing	Sysout
blksize	<numeric>		X		X
bufno	<numeric>	X	X	X	
conddisp	catlg delete	X	X	X	X
copies	<numeric>				X
dataclas	<alphanum>		X		
dest	dest[.user]				X
dir	<numeric>		X		
disp	old shr mod new	X	X	X	
dsntype	library pds large extreq extpref basic		X		
dsorg	ps po da		X		
forms	<alphanum>				X

Keyword		Usage			
Name	Value	Read	Write New	Write Existing	Sysout
gdgnt			X		
hold					X
label	nl sl nsl sul blp ltm al aul		X		
like	<Dataset Name>		X		
lrecl	<numeric>		X		X
maxvol	<numeric>		X		
mount		X	X	X	
mgmtclas	<alphanum>		X		
norecall		X		X	
outdes	<alphanum>				X
recfm	<alphanum>		X		X
release		X	X	X	
retpd	<numeric>		X		
sequence	<numeric>		X		
space	[blk. trk. cyl.]primary[.secondary]		X		
spin	unalloc				X
storclas	<alphanum>		X		
sysout	<sysout_class>				X
trtch	noncomp comp c e et t		X		
ucount	<numeric>	X	X	X	
unit	<alphanum>		X		
vol	<alphanum>		X		
writer	<alphanum>				X

Appendix B. Session config files

The files `/etc/ssh/cozsftp_config` and `/etc/ssh/cozsftp_server_config` can be used to customize the options available for Co:Z SFTP client and server sessions respectively. The permissions for each of these files should be `0644`.

Samples of both of these files are located in the `/usr/lpp/coz/samples` directory. These samples may be customized and placed at the above locations to make them active.

Each file has the sections `fixed:`, `default:` and `pattern:` which are described below. Additionally, individual users can provide their own file patterns (but not fixed or default options) in copies of these files in `$HOME/.ssh`

The individual Co:Z SFTP client and server options are described in: [Appendix A, Co:Z SFTP options](#).

B.1 Specifying fixed (immutable) options

Use the `fixed:` section to specify site-wide options that *cannot* be overridden by individual users. Multiple options may be specified on a single line if separated by commas.

In the example below, the `smf` option is activated for all users, and because it is fixed, may not be overridden by any user.

```
fixed:
smf
```

B.2 Specifying default options

Use the `default:` section to specify site-wide options that *can* be overridden by individual users. Multiple options may be specified on a single line if separated by commas.

In the example below, the `mode` option is set to `text` default. Because this option is set in the `default:` section, it can be easily overridden by individual users.

```
default:
mode=text
```

B.3 Specifying file pattern specific options

It is often useful to have a set of custom options associated with specific files and/or datasets. For example, transferring all files with the `.pax` extension in binary mode. The `pattern:` section(s) of the configuration files enables file and dataset names matching a specific POSIX *glob pattern* to automatically have specific options applied regardless of the options currently in place.

`pattern:` sections can be supplied in the site (`/etc/ssh`) versions of the config files and may also be supplied in copies of these files located in the user's `$HOME/.ssh` directory. A specific pattern may only be defined once;

subsequent definitions read from the config file(s) are ignored.

When a `put` or `get` command is issued, the file or dataset name is checked against the patterns in the order that they were originally read. The options associated with the *first* matching pattern (if any) are applied to that specific file transfer, overriding the current options. Once the transfer completes, the previous options are reactivated.

Pattern sections have the following syntax:

```
pattern: [//]<glob_pattern>
```

If double slashes (`//`) precede the pattern, it is used to match dataset names, otherwise it is used to match POSIX pathnames. Matching is performed on the name supplied on the transfer request after it has been normalized (e.g. embedded slashes in a dataset name are converted to periods and the characters are converted to uppercase). Please note that DD names will not be *not* resolved to their catalog name prior to matching. Patterns follow the UNIX *glob pattern* syntax, where the wildcard characters `?` (match exactly one character) and `*` (match zero or more characters) can be used in conjunction with literal characters to provide a match pattern. For a complete description of the pattern syntax, see the “File name generation” section of the `sh` command documentation in the [z/OS Unix System Services Command Reference](#).

Pattern examples

Setting text mode transfer for all members of a PDS

In the following example, a user specifies in `$HOME/.ssh/cozsftp_config`:

```
pattern: //*.JCL(*)
mode=text
```

And in a `cozsftp` session issues the following:

```
$ cozsftp user@host
Co:Z SFTP version: 1.9.3 (5.0p1) 2011-09-01
Copyright (C) Dovetailed Technologies, LLC. 2011. All rights reserved.
Connecting to host...
user@host's password: *****
cozsftp>lzopts mode=binary
cozsftp>get myjcl //HLQ.DEV.JCL(FOO)
```

Because the target name matches the pattern, the file `myjcl` will be transferred as in text mode even though the current mode setting is binary.

Automatically set dataset allocation parameters

In the following example, consider the Co:Z SFTP server configuration file `/etc/ssh/cozsftp_server_config`:

```
pattern: //*.PARTNER.TRANS*
space=cyl.3.2,recfm=fb,lrecl=80
```

And a remote sftp session issues the following:

```
sftp> put trans0923 //HLQ.PARTNER.TRANS0923
```

Assuming the dataset HLQ.PARTNER.TRANS0923 doesn't already exist, a new dataset with that name will be allocated with allocation parameters associated with the pattern. This example shows how a server can be setup to automatically allocate incoming datasets based on a predefined name pattern.

Pattern selection determined by first match

For the examples that follow, consider the following configuration files excerpts:

```
(from $HOME/.ssh/cozsftp_server_config)
pattern: *.txt
mode=text,clientcp=1252,linerule=crlf
```

```
(from /etc/ssh/cozsftp_server_config)
pattern: *.zip
mode=binary
pattern: *.pax
mode=binary
pattern: *.txt
mode=text,linerule=lf
```

```
sftp> get myarchive.pax      ❶
sftp> get mynotes.txt      ❷
```

- ❶ The file will be transferred in binary mode because it matches the site specified pattern (via /etc/ssh/cozsftp_server_config) for files with a .pax extension.
- ❷ The file myfile.txt will be transferred in text mode with a client code page of 1252 and a linerule of crlf. While the .txt extension could match two of the specified patterns, the one processed first (via \$HOME/.ssh/cozsftp_server_config) is selected. This is an example of how an individual user can override site behavior for a specific need (e.g. a Windows client platform).

Appendix C. Dataset Name Determination

When issuing a `put` command to create a dataset, or `get` to a local dataset using the `cozsftp` client, the resulting dataset name is determined as follows:

```
put myfile //HLQ.LEVEL
get myfile //HLQ.LEVEL (using the cozsftp client)
```

Table C.1. Dataset Name determination

Case	Condition	Dataset Name	Notes
1	HLQ.LEVEL is a Sequential Dataset	HLQ.LEVEL	Replaces existing SEQ dataset
2	HLQ.LEVEL is a PDS	HLQ.LEVEL(MYFILE)	Creates or replaces member named MYFILE in PDS
3	HLQ.LEVEL is not a dataset, but HLQ.LEVEL.XXX names exist in catalog	HLQ.LEVEL.MYFILE	Create or replace SEQ dataset
4	HLQ.LEVEL is not a dataset, and no HLQ.LEVEL.XXX names exist in catalog	HLQ.LEVEL	Creates new SEQ dataset

In most cases, this is acceptable behavior. However, there are cases where the supplied name should be treated as a dataset rather than a "directory" (as in Case 3 above). If this is the required behavior, a different dataset prefix can be supplied: `///!` or `/-!:`

```
put myfile ///!HLQ.LEVEL
get myfile ///!HLQ.LEVEL (using the cozsftp client)
```

Table C.2. Dataset Name determination

Case	Condition	Dataset Name	Notes
1	HLQ.LEVEL is a Sequential Dataset	HLQ.LEVEL	Replaces existing SEQ dataset
2	HLQ.LEVEL is a PDS	HLQ.LEVEL(MYFILE)	Creates or replaces member named MYFILE in PDS

Case	Condition	Dataset Name	Notes
3	HLQ.LEVEL is not a dataset	HLQ.LEVEL	Creates new SEQ dataset

NOTE: If you intend to always use a fully qualified dataset name, it is preferable to use the // ! or / - / ! syntax.

Appendix D. SMF Information

D.1 IBM FTP-compatible SMF 119 record subtypes

Co:Z SFTP supports recording SMF type 119 records that are compatible with the following IBM FTP records:

- Subtype 3 - FTP client transfer completion
- Subtype 70 - FTP server transfer completion

For more information on type 119 records, see: *z/OS Communications Server: IP Programmer's Guide and Reference*.

D.2 New SMF 119 record subtypes

In addition to standard FTP completion records above, Co:Z SFTP 2.0.0 and later will also create the following SMF 119 record subtypes:

- Subtype 192 - Co:Z SFTP server log messages
- Subtype 193 - Co:Z SFTP client log messages

These new records follow the structure of other SMF type 119 records. Six triplets are allocated, although only the first three triplets are currently used.

- The first section is a copy of the TCP/IP identification section, as it appeared in the related client transfer completion (subtype 3) or server transfer completion (subtype 70) record.
- The second section describes the socket connection.

Offset	Length	Format	Description
0	16	Binary	Remote IP address
16	16	Binary	Local IP address
32	2	Binary	Remote port number
34	2	Binary	Local port number
36	15	EBCDIC	FTP session ID
51	1	BINARY	Unused

- The third section contains Co:Z SFTP messages, informational level or above, that were associated with the previous transfer. This section contains one or more message sub-sections, each with the following layout:

Offset	Length	Format	Description
0	4	Binary	Time (in local time)

Offset	Length	Format	Description
4	4	Packed	Date (in local time)
8	2	Binary	Length of message that follows
10	variable	EBCDIC	Message text

D.3 Enabling SMF recording

In order to enable recording of Co:Z SFTP SMF 119 records, you must:

1. configure SMF to allow recording these records and subtypes. See *z/OS MVS System Management Facilities (SMF)* for more information.
2. permit the users running Co:Z SFTP client or server jobs READ access to the BPX.SMF SAF/RACF class.
3. the `nosmf` configuration option must not be set. See http://dovetail.com/docs/sftp/options.html#options_misc for more information.
4. in order to get accurate local and remote host/port information for these records, the program `COZ_HOME/bin/ssh-socket-info` is called by Co:Z once the child ssh session is established.

This program uses the IBM EZBNMIFR network management API, which requires the `ssh-socket-info` program to be APF authorized. The Co:Z installer will attempt to set the "+a" extattr bit on this program, but will only succeed if the installing userid has READ access to the BPX.FILEATTR.APF SAF resource. If for some reason, this program is not APF authorized, Co:Z SFTP will operate properly, but the SMF socket information is not guaranteed to be accurate.

D.4 Using the real-time Co:Z SMF API

The Co:Z SFTP client and server will also write SMF 119 records to a Unix datagram socket if it is available. By default, the name of the socket is `/var/log/cozsftp.smf.sock` unless overridden by the `SFTP_SMF_SOCKET` environment variable. This API is useful in managed file transfer environments that need real-time access to file transfer events. The API is independent of actual SMF recording - you may use either real SMF recording, the datagram socket API, or both.

To use this facility, you must write a program that creates this Unix-domain socket and receives datagram messages from it. Each message will be a SMF record image from a Co:Z SFTP client or server running on the same system. A sample C++ program, `CoZSmfServer.C`, demonstrates how to use this facility. See the documentation and build instructions in `$COZ_HOME/samples/smfapi/CoZSmfServer.C`.

Appendix E. Client Authentication Mechanisms

Running the Co:Z Launcher and/or the Co:Z SFTP client requires that the z/OS ssh client can authenticate with the Target System ssh server. Several authentication choices are available from z/OS; site policies will usually dictate which is best.

One of the following authentication mechanisms should be performed on z/OS from **each** userid that will be used to execute the Co:Z Batch jobs.

- Interactive password: *Section E.1, “Interactive password authentication”*. **Note:** this mechanism requires user keyboard interaction, so it will not work in batch. It should only be used for command line invocations of the Co:Z SFTP client.
- OpenSSH ASK_PASS (read a password from a dataset): *Section E.3, “OpenSSH SSH ASKPASS authentication”*.
- Conventional OpenSSH keypairs: *Section E.2, “OpenSSH keypair authentication”*.
- RACF Digital Certificates: *Section E.4, “RACF Digital Certificate authentication”*.

E.1 Interactive password authentication

This is the simplest form of OpenSSH client authentication and requires no additional setup. It can only be used from a terminal connected shell where the user can supply the Target System password. Due to this requirement, it is not suitable for z/OS batch programs and is therefore not an option for running the Co:Z Launcher or batch Co:Z SFTP. It *is* suitable for shell invocations of Co:Z SFTP `cozsftp`.

E.2 OpenSSH keypair authentication

This is the conventional mechanism for performing OpenSSH client authentication. A public/private key pair is generated on z/OS. The private key is kept (protected) in the user's `/.ssh` directory. The public key is stored on each target system in the user's `/.ssh/authorized_keys` file. The following steps describe how to generate and use an OpenSSH keypair:

Note: a z/OS shell invoked under telnet, rlogin, or ssh must be used for key generation. Don't attempt to do this under an OMVS shell, since the `ssh` commands are generally not supported under OMVS.

Note: Proceed with caution if you have more than one userid mapped to the same `uid` number (an unfortunately common occurrence on z/OS USS). The default key storage home directory is hard to predict.

1. Generate a keypair using `ssh-keygen`:

```
$ mkdir ~/.ssh
$ chmod 700 ~/.ssh
$ ssh-keygen -t dsa
Generating public/private dsa key pair.
```

```

Enter file in which to save the key (/home/<userid>/ssh/id_dsa): <enter>
Enter passphrase (empty for no passphrase): <enter>
Enter same passphrase again: <enter>
Your identification has been saved in /home/<userid>/ssh/id_dsa.
Your public key has been saved in /home/<userid>/ssh/id_dsa.pub.
The key fingerprint is:
dd:ff:00:87:43:11:fa:7b:0d:84:3a:19:3b:7f:5d:2e <userid>@<host>

```

The private key file `id_dsa` will be generated without a passphrase so that Co:Z can run in batch. It is therefore important that this file is protected with file permissions and/or ACLs that only allow the owning userid to read the file.

2. Move a copy of the public key to the target system:

```

ZOS$ sftp -oPort=<port> cozuser@linux1.myco.com
Connecting to n.n.n.n...
cozuser@linux1.myco.com's password: *****
sftp> ascii
Sets the file transfer type to ASCII.
sftp> cd .ssh
sftp> put -p id_dsa.pub authorized_keys
Uploading id_dsa.pub to /home/sgoetze/.ssh/authorized_keys
id_dsa.pub          100% 601    0.6KB/s   00:00
sftp> quit

```

Note: If you are adding public keys from more than one z/OS userid to `authorized_keys`, then you must append each key rather than replacing the file as shown above.

E.3 OpenSSH SSH_ASKPASS authentication

OpenSSH supports the use of the `SSH_ASKPASS` environment variable to point to a program that will read a password, without keyboard interaction.

A dataset member (e.g.) `//HLQ.PASSWD(SITE1)` must be created that contains a single line with the password starting in the first column and *without* line numbers.

E.4 RACF Digital Certificate authentication

Traditional OpenSSH keypairs and `SSH_ASKPASS` are convenient, but some sites have strict policies about keeping user credentials in a SAF facility. The z/OS Communications Server FTP command can exploit RACF Digital Certificates for authentication and encryption. The Co:Z toolkit provides a similar capability via its `saf-ssh-agent` which can be used in conjunction with a user RACDCERT RSA certificate to provide OpenSSH client authentication.

An existing SAF/RACF Keyring and client certificate set up for use with the z/OS FTP client may be used with

Co:Z Launcher and the Co:Z SFTP client.

The following steps describe how to create an RSA RACF Digital Certificate, export its public key in OpenSSH compatible format, and transfer the public key to the target system.

1. Create a Keyring and RSA Digital Certificate:

Note: In order to create RACF Digital Certificates, certain RACF permissions must be held. This step is typically performed by an administrator; the permissions required are *not* required for the user to access the certificate (see below). For details, see the chapter *RACF and Digital Certificates z/OS Security Server RACF Security Administrator's Guide (SA22-7683)*.

This JCL is located in RACDCERT member of the COZ.SAMPJCL PDS. It will create an RSA Digital Certificate labeled MY-CERT held in the keyring MY-RING

```
//COZUSERJ JOB ( ), ' ',MSGCLASS=H,NOTIFY=&SYSUID
//*
// EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *

/* Generate a self-signed RSA certificate to use */
/* for SSH client authentication. */
/* A certificate signed by your CA will also work. */
RACDCERT ID(COZUSER) GENCERT + ❶
    SUBJECTSDN(
        CN('First Lastname' ) + ❷
        O('My Company' ) + ❷
        OU('Development' ) + ❷
        C('US' ) + ❷
    ) + ❷
    WITHLABEL('MY-CERT' )

/* Create a KEYRING for the user */
RACDCERT ID(COZUSER) ADDRING(MY-RING) ❶

/* Connect the certificate to the ring */
RACDCERT ID(COZUSER) CONNECT ( + ❶
    ID(COZUSER) + ❶
    LABEL('MY-CERT' ) +
    RING(MY-RING) +
    DEFAULT + ❸
    USAGE(PERSONAL) )

/* Refresh to activate */
SETROPTS RACLIST(DIGTCERT, DIGTRING) REFRESH

/* List the user's certs */
RACDCERT ID(COZUSER) LIST ❶
//
```

- ❶ Change the string COZUSER to the MVS userid that will own and use the certificate.
 - ❷ Change the subject DSN fields according to your company's standards.
 - ❸ Makes this certificate the default in the ring. This allows the user to specify just the keyring name in order to access the certificate.
2. Export an OpenSSH version of the certificate's public key:

Note: This and the remaining steps are performed by the user. In order to access the keyring and certificate, the user must have READ access to the FACILITY class resources:

- IRR.DIGTCERT.LIST
- IRR.DIGTCERT.LISTRING

Public key extraction is performed using Co:Z's `saf-ssh-agent` and the `-x` option. If the `-f` option is specified, the key is extracted to the specified filename. Otherwise it is written to `stdout`.

```
$ saf-ssh-agent -x -f cozuser_saf.pub MY-RING:MY-CERT
```

Note: An administrator may export the key of a another user by prefixing the keyring name with `USERID/`. In order to do this, the administrator must have UPDATE access to the SAF classes listed above.

3. Move a copy of the public key to the target system:

```
ZOS$ sftp -oPort=<port> cozuser@linux1.myco.com
Connecting to n.n.n.n...
cozuser@linux1.myco.com's password: *****
sftp> ascii
Sets the file transfer type to ASCII.
sftp> cd .ssh
sftp> put -p cozuser_saf.pub authorized_keys
Uploading cozuser_saf.pub to /home/cozuser/.ssh/authorized_keys
cozuser_saf.pub          100% 601    0.6KB/s   00:00
sftp> quit
```

Note: If you are adding public keys from more than one z/OS userid to `authorized_keys`, then you must append each key rather than replacing the file as shown above.

Appendix F. Restricting OpenSSH users to SFTP

The common technique for restricting ssh capabilities is to change the user's default shell (the "default program" in the OMVS segment) to a shell that only allows certain commands and no interactive access. The sample script below can be customized and used as the user's "restricted shell". Put this script somewhere in your Co:Z bin install directory and make its permissions `u=rwx,go=rx` (i.e 755). Use the full path name of the script as the users shell. Make sure that the script has the full path names that you used for the Co:Z SFTP executable. You can remove the `cozserver` entry from the list if you don't want to allow remote Dataset Pipes commands via ssh.

```
#!/bin/sh
# A shell script which can be set as a users default shell
# to only allow certain commands or ssh subsystems to run,
# disallowing full shell logins.

if [ $# -ge 2 -a "$1" = "-c" ]; then
  case $2 in
    # Update this list to match what you have in /etc/ssh/sshd_config
    # or add any other commands that you would like to allow
    # from ssh.
    /opt/dovetail/coz/bin/sftp-server.sh) exec $2;;
    /opt/dovetail/coz/bin/cozserver) exec $2;;
    *);;
  esac
fi

# Write out whatever messages you want your users to see
# if they try something else
echo "Only sftp and dataset pipes file transfers are allowed\
from this account."
exit 1
```

Note: this will not only restrict the user from using an interactive shell under OpenSSH, but will prevent them from running an interactive shell under TSO as well. Also, any batch jobs that run with their userid will also use this as the default Unix shell (BPXBATCH or COZBATCH).

Appendix G. Setting up a test OpenSSH system on z/OS

It's sometimes convenient to create your own z/OS SSHD server on an alternate port for testing purposes. You can do this without any special privileges, and the SSHD server will run fine, except that it will only allow logins for the userid that it is running under.

This is especially handy if your Systems Programmer doesn't understand immediately that adding an SSH user subsystem doesn't introduce any new security risks.

Procedure G.1. General outline for adding a test SSHD server

1. Create your own ssh directory, say ~/sshd, and copy the file /etc/ssh/sshd_config into it:

```
zos$ mkdir ~/sshd
zos$ cp /etc/ssh/sshd_config ~/sshd
```

2. In this directory, generate your DSA and RSA host keys, as directed in the *IBM Ported Tools for z/OS User's Guide*.

If you can copy the keys in /etc/ssh directory, then you will avoid "host key" mismatch problems if you switch your SSH client from the production to the test server. If you do copy the production host keys, make sure that you change the file permissions to 600 so that they can't be read by others.

3. Edit your copy of sshd_config:
 - a. Find the line "Subsystem" which defines the sftp subsystem
 - b. Add a new line after this line:

```
Subsystem dspipes /usr/lpp/coz/bin/dspipes
```

(where /usr/lpp/coz is the directory where Co:Z Toolkit is installed).

- c. Uncomment the Port line and set it to an available port
- d. Uncomment / add the following lines (to use the private keys generated in the previous step):

```
HostKey ./ssh_host_rsa_key
HostKey ./ssh_host_dsa_key
```

(where /usr/lpp/coz is the directory where Co:Z Toolkit is installed).

4. From a z/OS shell, change to the directory that you created and start your copy of SSHD:

```
/usr/sbin/sshd -e -D -f ./sshd_config
```

Note: If you are unable to execute `/usr/sbin/sshd`, you may be able to copy it to your local directory, add the execute bit (`chmod +x ~/sshd/sshd`) and run the above command using this local copy.

5. To connect to your test SSHD server from a client, don't forget to use the `-ssh -p port` SSH option on your `ssh`, **fromdsn** or **todsn** commands.

Appendix H. License

The Co:Z Co-Processing Toolkit, comprised of Co:Z Launcher, Co:Z Dataset Pipes, Co:Z SFTP, Co:Z Batch, and the Co:Z Target System Toolkits is distributed under the Co:Z Community License Agreement (see below).

Note: This community license is superseded for Co:Z Toolkit Enterprise License and Support customers. All components are distributed in binary form. The Co:Z Target Systems Toolkit and the Co:Z FTP-SSH Proxy and are also available in source form under the [Apache V2 License](#).

For more information, see our licensing [frequently asked questions](#).

CO:Z COMMUNITY LICENSE AGREEMENT

PLEASE READ THIS COMMUNITY LICENSE AGREEMENT (THIS "AGREEMENT") CAREFULLY. THIS AGREEMENT SETS FORTH THE TERMS ON WHICH DOVETAILED TECHNOLOGIES, LLC ("DOVETAIL"), A MISSOURI LIMITED LIABILITY COMPANY, MAKES AVAILABLE THE CO:Z CO-PROCESSING TOOLKIT FOR z/OS AT NO CHARGE FOR DOWNLOAD, INSTALLATION AND USE BY THE COMMUNITY. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE, YOU ACKNOWLEDGE THAT YOU HAVE READ, UNDERSTAND, AND AGREE TO BE LEGALLY BOUND BY THIS AGREEMENT.

1. DEFINITIONS. As used in this Agreement, the following capitalized terms shall have the following meanings:

"Documentation" means Dovetail's accompanying user documentation for the Software, as may be updated by Dovetail from time to time, in print or electronic form.

"Software" means the Co:Z Co-Processing Toolkit for z/OS, comprised of Co:Z Launcher, Co:Z Dataset Pipes, Co:Z SFTP, Co:Z Batch, and the Co:Z Target System Toolkit in object code form only, together with certain sample code and scripts in source form.

"Update" means any bug fix, enhancement, or other modification to or update for the Software issued by Dovetail for general release to the Software community.

"You" means the person or entity downloading, installing or using the Software. If you are downloading, installing or using the Software on behalf of a company or organization, the term "You" refers to both you and your company or organization, and you represent and warrant that you have authority to bind your company or organization to the provisions hereof.

2. SOFTWARE LICENSE. During the term of this Agreement, and subject to the provisions hereof, Dovetail hereby grants to You, and You hereby accept, an enterprise-wide, non-exclusive, non-transferable, royalty-free and fully paid-up license to install and use the Software on an unlimited number of Your servers, solely for Your internal business purposes, in accordance with the Documentation, and in compliance with all applicable laws and regulations.

3. LICENSE RESTRICTIONS. You shall not, nor shall You authorize any other

person or entity to: (a) distribute, rent, lease, lend, sell, sublicense or otherwise make the Software available to any third party; (b) modify, adapt, alter, translate, or create derivative works of the Software; (c) use the Software in or as part of a service bureau, timesharing or outsourcing capacity; (d) develop an alternative to the Software that is based on or derived from, in whole or in part, the Software or Documentation; (e) remove or obscure any copyright, trademark or other proprietary rights notices or designations on the Software, the Documentation or any copies thereof; or (f) reverse engineer, decompile, disassemble, or otherwise attempt to derive the source code for the Software, except where such reverse engineering is expressly permitted under applicable law, but then only to the extent that Dovetail is not entitled to limit such rights by contract.

4. **UPDATES.** From time to time, Dovetail may make available Updates for the Software as a general release to the Software community. All such Updates (whether posted by Dovetail on the Dovetail website or included with the Software) shall be deemed part of the Software, and are licensed to You under the license and other provisions of this Agreement, together with any supplementary license terms that Dovetail may provide for such Updates.

5. **YOUR RESPONSIBILITIES.** You are responsible for: (i) installation of the Software and any Updates; (ii) selecting and maintaining all third party hardware, software, peripherals and connectivity necessary to meet the system requirements for the Software; (iii) creating a restore point for Your systems and backing up and verifying all data; and (iv) adopting reasonable measures to ensure the safety, security, accuracy and integrity of Your facilities, systems, networks and data. Dovetail shall have no responsibility or liability arising out of or resulting in whole or in part from Your failure or delay to perform any such responsibilities, or for acts or omissions of third parties, Internet or telecommunications failures, or force majeure or other events beyond Dovetail's reasonable control.

6. **SUPPORT.** This Agreement does not include, and Dovetail shall have no obligation under this Agreement to provide, any technical support or other professional services for the Software. If You are interested in purchasing a support plan for the Software, You should visit the Dovetail website to review Dovetail's then current offerings.

7. **TERM; TERMINATION.** This Agreement and Your license rights hereunder shall continue unless and until terminated as set forth herein. You may terminate this Agreement for convenience at any time by uninstalling, erasing all copies of, and ceasing all use of the Software and Documentation. This Agreement shall terminate immediately and automatically if You violate the license terms or restrictions for the Software, or materially breach any other provision of this Agreement and fail to cure such breach within ten (10) days after receiving notice thereof from Dovetail. Upon the expiration or termination of this Agreement for any reason: (i) Your license to the Software shall automatically and immediately terminate; and (ii) You shall discontinue use of the Software, promptly (within 5 days) uninstall and remove any remnants of the Software and Documentation from Your computers, network and systems, and destroy (or return to Dovetail) all tangible copies of the Software and Documentation in Your possession. Sections 1, 3, 5, 7, 8, 9, 10 and 11 of this Agreement shall survive the expiration or termination of this Agreement for any

reason, and shall be binding on and inure to the benefit of the parties and their permitted successors and assigns.

8. **DISCLAIMER.** THE SOFTWARE AND DOCUMENTATION ARE PROVIDED TO YOU UNDER THIS AGREEMENT "AS IS" WITHOUT REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, AND ALL USE IS AT YOUR OWN RISK. WITHOUT LIMITING THE FOREGOING, DOVETAIL AND ITS SUPPLIERS HEREBY disclaim any IMPLIED OR STATUTORY warranties of MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT. THE SOFTWARE IS NOT INTENDED OR LICENSED FOR USE IN ANY HAZARDOUS OR HIGH RISK ACTIVITY. DOVETAIL DOES NOT WARRANT THAT THE SOFTWARE WILL OPERATE UNINTERRUPTED OR ERROR-FREE, OR MEET YOUR BUSINESS, TECHNICAL OR OTHER REQUIREMENTS. No employee or agent has authority to bind DOVETAIL to any representations or warranties NOT EXPRESSLY SET FORTH IN THIS AGREEMENT.

9. **PROPRIETARY RIGHTS.** Dovetail and its suppliers shall retain exclusive right, title and interest in and to the Software, including the object code, source code, program architecture, design, coding methodology, Documentation, screen shots, and "look and feel" therefor, all Updates thereto, all goodwill associated therewith, and all present and future copyrights, trademarks, trade secrets, patent rights and other intellectual property rights of any nature throughout the world embodied therein and appurtenant thereto. All rights and licenses to the Software not expressly granted to You in this Agreement are reserved by Dovetail and its suppliers. From time to time, You may submit suggestions, requests or other feedback for the Software. Dovetail shall be free to commercialize and use such feedback, including for developing improvements to its products and services, free of any claims, payment obligations, or proprietary, confidentiality or other restrictions of any kind.

10. **LIMITATIONS ON LIABILITY.** IN NO EVENT SHALL DOVETAIL BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, SPECIAL, PUNITIVE OR SIMILAR DAMAGES ARISING OUT OF OR RELATED TO THE SOFTWARE OR THIS AGREEMENT, INCLUDING LOSS OF BUSINESS, PROFITS OR REVENUE, LOSS OR DESTRUCTION OF DATA, BUSINESS INTERRUPTION OR DOWNTIME. THE TOTAL CUMULATIVE LIABILITY OF DOVETAIL ARISING OUT OF AND RELATED TO THE SOFTWARE AND THIS AGREEMENT SHALL NOT, REGARDLESS OF THE NUMBER OF INCIDENTS OR CAUSES GIVING RISE TO ANY SUCH LIABILITY, EXCEED TEN U.S. DOLLARS (\$10). THE LIMITATIONS ON LIABILITY IN THIS SECTION SHALL APPLY TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, REGARDLESS OF THE CAUSE OF ACTION OR BASIS OF LIABILITY (WHETHER IN CONTRACT, TORT OR OTHERWISE), AND EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS ON LIABILITY ARE AN ESSENTIAL PART OF THIS AGREEMENT, AND SHALL BE VALID AND BINDING EVEN IF ANY REMEDY IS DEEMED TO FAIL OF ITS ESSENTIAL PURPOSE.

11. MISCELLANEOUS

Governing Law. This Agreement shall be governed and interpreted for all purposes by the laws of the State of Missouri, U.S.A., without reference to any conflict of laws principles that would require the application of the laws of a different jurisdiction. The United Nations Convention on Contracts for the International Sale of Goods and the Uniform Computer Information Transactions Act (as enacted in any jurisdiction) do not and shall not apply to this Agreement, and are hereby specifically excluded.

Jurisdiction; Venue. Any dispute, action or proceeding arising out of or related to the Software or this Agreement shall be commenced in the state courts of St. Louis County, Missouri or, where proper subject matter jurisdiction exists, the United States District Court for the Eastern District of Missouri. Each party irrevocably submits and waives any objections to the exclusive personal jurisdiction and venue of such courts, including any objection based on forum non conveniens.

Notices. All notices under this Agreement shall be in writing, and shall be delivered personally or by postage prepaid certified mail or express courier service, return receipt requested. Notices to You may be delivered to the most current address on file. Notices to Dovetail shall be directed to the following address, unless Dovetail has provided an alternative notice address:

Dovetailed Technologies, LLC
305 Willowpointe Drive
St. Charles, MO 63304

Assignments. You may not assign or transfer this Agreement, or any rights or duties hereunder, in whole or in part, whether by operation of law or otherwise, without the prior written consent of Dovetail. Any attempted assignment or transfer in violation of the foregoing shall be null and void from the beginning and without effect. Dovetail may freely assign or transfer this Agreement, including to a successor in interest upon Dovetail's merger, acquisition, corporate reorganization, or sale or other transfer of all or substantially all of its business or assets to which this Agreement relates.

Relationship; Third Party Beneficiaries. The parties hereto are independent contractors. Nothing in this Agreement shall be deemed to create any agency, employment, partnership, fiduciary or joint venture relationship between the parties, or to give any third party any rights or remedies under or by reason of this Agreement; provided, however, the disclaimers and limitations on liability in this Agreement shall extend to Dovetail and its directors, officers, shareholders, employees, agents and affiliates. All references to Dovetail in connection therewith shall be deemed to include the foregoing persons and entities, who shall be third party beneficiaries of such contractual disclaimers and limitations and entitled to accept all benefits afforded thereby.

Equitable Relief. The Software comprises the confidential and proprietary information of Dovetail and its suppliers, and constitutes a valuable trade secret. You acknowledge that Your breach of the license or ownership provisions of this Agreement would cause irreparable harm to Dovetail, the extent of which would be difficult and impracticable to assess, and that money damages would not be an adequate remedy for such breach. Accordingly, in addition to all other remedies available at law or in equity, and as an express exception to the jurisdiction and venue requirements of this Agreement, Dovetail shall be entitled to seek injunctive or other equitable relief in any court of competent jurisdiction.

U.S. Government Restricted Rights. The Software and Documentation are licensed with RESTRICTED RIGHTS as "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial

Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation is licensed (if at all) to U.S. Government end users only as Commercial Items, and with only those rights as are granted to other licensees pursuant to this Agreement.

Export Control. The Software and underlying information and technology may not be accessed or used except as authorized by United States and other applicable law, and further subject to compliance with this Agreement. The Software may not be exported or re-exported into any U.S. embargoed countries, or to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Department of Commerce Denied Person's List or Entity List. You represent and warrant that You and Your end users are not located in, under the control of, or a national or resident of any country or on any such list.

Amendment; Waiver. This Agreement may be amended only by a written instrument signed by an authorized representative of Dovetail. No rights shall be waived by any act, omission or knowledge of a party, except by an instrument in writing expressly waiving such rights and signed by an authorized representative of the waiving party. Any waiver on one occasion shall not constitute a waiver on subsequent occasions.

Severability; Construction. If any provision of this Agreement is determined to be invalid or unenforceable under applicable law, such provision shall be amended by a court of competent jurisdiction to accomplish the objectives of such provision to the greatest extent possible, or severed from this Agreement if such amendment is not possible, and the remaining provisions of this Agreement shall continue in full force and effect. The captions and section headings in this Agreement are for reference purposes only and shall not affect the meaning or interpretation of this Agreement. The term "including" as used herein means "including without limitation." The terms "herein," "hereto," "hereof," and similar variations refer to this Agreement as a whole, rather than to any particular section.

Entire Agreement. This Agreement sets forth the entire agreement of the parties and supersedes all prior agreements and understandings, whether written or oral, with regard to the subject matter hereof. Any additional or conflicting terms proposed by You in any purchase order, request for proposal, acknowledgement, or other writing shall not be binding, and are hereby objected to and expressly rejected.