**Revision Sheet**

|  |  |  |
| --- | --- | --- |
| **Release No.** | **Date** | **Revision Description** |
| Rev. 0 | 12/10/2017 | First version. |
| Rev. 1 | 12/27/2017 | Second version   * adding new section on use of the three new flags “—search-root” “—validate-customer” “—no-backup”, after two times setup at HiSilicon Co. Ltd(Huawei) |
| Rev.2 | 1/24/2018 | Typo fixes; Updated with latest information. |
| Rev.3 | 8/2/2018 | Reviewed docs and fix misinformation. |
|  |  |  |
|  |  |  |
|  |  |  |

**User’s Manual Authorization Memorandum**

I have carefully assessed the User’s Manual for PDF TDS RELAY XFER System. This document has been completed in accordance with the requirements of System Development Methodology.

MANAGEMENT CERTIFICATION - Please check the appropriate statement.

\_\_\_\_\_\_ The document is accepted.

\_\_\_\_\_\_ The document is accepted pending the changes noted.

\_\_\_\_\_\_ The document is not accepted.

HiSilicon deployment and use of this utility is subject to the terms and

conditions of the SOFTWARE LICENSE & RELATED SERVICES AGREEMENT between PDF Solutions, Inc. and Huawei Technologies Co. Ltd. This utility is

Software, as defined in the foregoing Software License Agreement.b .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME DATE

HiSilicon Contact Person

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME DATE

HiSilicon Project Owner

**PDF TDS RELAY XFER USER'S MANUAL**

Table of Contents

[1. General information 4](#_Toc502149147)

[2. System installation requirements 4](#_Toc502149148)

[3. System installation procedure 4](#_Toc502149149)

[4. The command options 5](#_Toc502149150)

[5. How to setup a folder for data transfer ( within subfolders) 6](#_Toc502149151)

[6. Trouble Shooting 7](#_Toc502149152)

[6.1 How to use python logging facility 7](#_Toc502149153)

[6.2 About the .lock file 8](#_Toc502149154)

[6.3 The pysftp (python module) dependencies 8](#_Toc502149155)

[7. Appendix 9](#_Toc502149156)

# General information

This document is about how to deploy tds-relay FTP/SFTP code into production environment for Hisilicon.

The code is not packaged into module for the reason that the code may be revised and monitored by the Hisilicon staff and the PDF Solution.

# System installation requirements

|  |  |
| --- | --- |
| OS | RHEL or Centos 5.x, 6.x |
| Software | * Shell Interpreter for running shell programs * **Python3** interpreter for executing functional program * Python tool ‘pip’ for install module online |
|  |  |

# System installation procedure

* Modify the variables in the file *tds-relay-run.sh*, for example

PYTHON=/usr/local/bin/python3

RUNDIR=/home/pdfs/tds-relay

DATADIR=/home/pdfs/data

* PYTHON, the variable should be the absolute path of the python interpreter
* RUNDIR, where the scripts locate
* DATADIR, where the data to be transferred. See “How to setup a folder for data transfer (without or within subfolders)” for explanation.
* Offline installation.

See “The pysftp (python module) dependencies”.

* [Optional]Online installation. Run the *INSTALL.sh* program to install Python dependent module(s)

**Tip:** The program can be used even if pysftp (the python module) is not installed. The installation of pysftp may need to take extra mile if the machine can’t install module from Internet. See “The pysftp (python module) dependencies”

**Warning**: If there are multiple Python versions installed on the machine, the *INSTALL.sh* must run using pip which corresponds to the python, otherwise errors like 'No module named XXX' will occur.

> ./INSTALL.sh # or sh INSTALL.sh

TIP: Check pip and python versions under your python environment

> pip --version   
> python --version

* Add the job to the cron.

> crontab -e

# Please copy and paste the content from tds-relay-crontab.txt

# Eg：\*/5 \* \* \* \* /home/pdfs/tds-relay/tds-relay-run.sh

Note: Please be advised that don’t set the interval less than 300 seconds for cron task (For estimation, the interval should be greater than the seconds for program to recognized stable data just copied in, plus the seconds for transferring duration.)

# Up and running

To test out if the script is working correctly, you can see the ./Logs/<timestamp, e.g.20180802>.log for any errors resulting from script errors or FTP errors.

Don’t hesitate to reach for support from PDF, see “Appendix” for maintainers’ contact information.

# The command options

> python tds\_relay.py --root-dir <data\_directory> --search-root --no-validate\_customer --backup –all\_pass --delay 120

* *--root-dir <data\_directory>*, or *-r <data\_directory>*, is to specify where the data files needs to transfer.
* *--search-root,* default is false when not specified*.* Once specified, the program will not scan for subfolders of the data directory. This is used in relay server where there is no idea of subfolders of data for different vendors.
* *--no\_validate\_customer,* default is false when not specified. Once specified, the program will compare the customer’ information in the envelope.xml (as in the .dat file) and the .config.ini file.  
  “multiple customers in the .config.ini” is supported .   
  **TIP:** For files of different format, or abnormal naming or packaging, this flag can be used.
* *--backup,* default is false when not specified, the data will be removed after successful transferring. Once this flag is specified, the data will be moved to the folder named ‘transferred’ of the same directory
* *--delay [seconds],* if not specified, default is 120 seconds. For instant transfer, set to 0 is OK.
* *--all-pass,* default is false, meaning only .dat files will be transferred. If specified, all the files under the directory except the .config.ini will be transferred.  
  TIP: even the *--all-pass* is specified, the files with suffix .tmp will not be transferred in case of incomplete files in the making. Neither of the .config.ini

# How to setup a folder for data transfer (without or within subfolders)

## The .config.ini file

Whether the data to be transferred is put under the one root folder or multiple subfolders of the root folder, the .config.ini should be present in order to make FTP/SFTP work.

The *.config.ini* file should be configured for each FTP/SFTP connection. Here is an example with description, replace *<value>* with your FTP/SFTP site information and also of the receiver.

[Destination]

customer = <data\_receiver\_assigned>

# host - FQDN or IP address of destination FTP/SFTP server to which data will be forwarded.

host = <IPv4\_address>

# mode - Transport mode, of either FTP or SFTP.

mode = SFTP

# user / password - User and password for destination server to which data will be forwarded.

user = <FTP or SFTP account name>

passwd = <FTP or SFTP account password>

# outdir - Directory on destination server to which data will be deposited.

outdir = <remote\_FTP\_site\_relative\_directory>

**TIP:** One can copy example.config.ini to /your\_data\_folder/.config.ini and modify after that.

**TIP:** If the *<value>* is empty, please leave the value empty instead of deleting the key. If the key is missing, the FTP/SFTP will not be functional.

**TIP:** The *outdir* should be an existing relative path to your FTP receiver’s folder, not the absolute path on the server.

**TIP:** Multiple destinations transferring is supported. To transfer data to two different FTP/SFTP destinations, create another sections of configuration in the same .config.ini, REMEMBER to rename the new section name to something different, such like ‘[Destination2]’.

## Data directory without subfolders

If you don’t want to use subfolders layout, i.e. some relay servers only have one folder containing all data files, you can just call the program with ‘*--search-root*’, which will not go through all the subfolders looking for data files.

The root folder should contain one *.config.ini* specifying the FTP/SFTP information (be careful about the ‘.’ ( a dot ) before the ‘config.ini’, otherwise not recognized.)

To run the program use following flags,

> python tds\_relay.py --root-dir <data\_directory> --search-root

# OR

> python tds\_relay.py --root-dir <data\_directory> --search-root --all-pass --delay 0

**TIP:** for this kind of data transfer, you may at the same time use flags such like “--all-pass” and “--delay 0” for transferring all files, don’t save the copy to local directory and do it right away.

## Data directory within subfolders

For normal use, you should use following flags *without* flag ‘--search-root’

> python tds\_relay.py --root-dir <data\_directory> --validate-customer –delay 0

If using the subfolders layout, In the first step of “System installation procedure”, the *$DATADIR* is where all your sub data folders reside. Each sub data folder is supposed to be a folder of data and contain a .config.ini.

In this case, your *$DATADIR* should look like this,

DATADIR\_base\_directory\_of\_all\_sub\_data\_folder, e.g. UPLOAD\_DATA

|-----sub\_data\_folder\_for\_ftp\_sftp\_site\_1, e.g. PDF\_to\_COM

|-----.config.ini

|-----data\_1.dat

|-----data\_2.dat

|-----data\_N.dat

|-----sub\_data\_folder\_for\_ftp\_sftp\_site\_2, e.g. REMOTE\_CLIENT\_ID\_NAME

|-----….

|-----sub\_data\_folder\_for\_ftp\_sftp\_site\_N, e.g. SFTP\_SERVER\_N

Each sub data folder should contain

* One *.config.ini* (be careful about the ‘.’ (a dot) before the ‘config.ini’, otherwise not recognized)
* The data files (if the flag ‘--all-pass’ is not specified, only .dat files will be recognized, and place the data file at the same directory of the .ini file. Enclosing data in deeper subfolders will not be processed.)

# Trouble Shooting

## How to use python logging facility

The process of the program will log its actions into logging file of the day, for example, *./logs/tdsrelay-<datestamp>.log*. All errors will be written into the log files. If the data is never transferred, or other unexpected thing happens, the log file should provide information for your diagnosis.

For further debugging, you may need to change the logging level to debug. In the tds\_relay.py, ca. line line 97, *“level=logging.DEBUG”* instead of *level=logging.INFO”*.

## About the .lock file

If you try to restart the program, or run the program via cron task, please make sure that there isn’t a .lock file under the program/script directory, the $RUNDIR.

The .lock file under the program directory will keep itself from running more than one instance. The program is *NOT* designed to run multiple instances and please do NOT run two instances of this program to watch for the same data folder, problems caused by racing conditions are not expected or handled.

## The pysftp (python module) dependencies

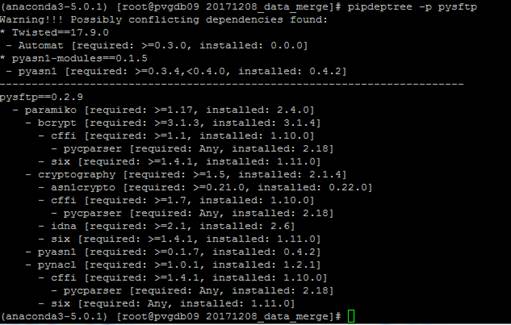
Warn: this tasks might takes some time and trial-and-errors. Be patient!

When installing python modules offline, e.g. pysftp, you may need to download all the required python modules unto the hard disk and initiate the installation command like,

> pip install --no-index --find-links=d:\downloaded\_pkg\ -r requirements.txt

Please download the Python modules from <https://pypi.python.org/pypi> according to this dependency graph for pysftp. And then copy them to the offline server.

PDF is planning to use Docker for easier deployment, please be patient until a verified container is provided.



During python module installation, there will be on-demand building of cartography module. Please also install the following .rpm for CentOS, libffi-devel, python-devel, openssl-devel, pkgconfig. Be aware that you must use the correct version for your operation system as latest version of .rpm packages may conflict with installed .rpm package of the operation system.

# Known Issues

* The logging files may take some spaces if not clean. Please clean up the ./logs folder under the program/scripts folder manually and regularly.
* Currently there is no UI or webpage to indicating the status of the transferring. PDF is going to take advantage of buildbot for monitoring.   
  Usually it is recommended to use PDF’s DataPower tool suite to create reporting template for graphs and files counting.

# Appendix

For technical support or for consulting, please reach

Jiacheng(Alien) Zhu (PDF) <jiacheng.zhu@pdf.com>

Wanyang (Emily) Cao (PDF) <wanyang.cao@pdf.com>

Bijian (Rebecca) Lan (PDF) <bijian.lan@pdf.com>