WIPO
WORLD
INTELLECTUAL PROPERTY
ORGANIZATION

WIPO PCT Electronic Data Interchange (PCT-EDI)

PCT Technical Cooperation June 2020

Table of Contents

I. INTRODUCTION

The WIPO PCT Electronic Data Interchange Service (PCT-EDI) provides a flexible, secure mechanism for the exchange of intellectual property information between Offices and the International Bureau (IB). This includes record copies, Priority Documents (P- DOCS), and pamphlets, as well as other general-purpose bulk data products.

II. SERVICE OVERVIEW

An important goal in the development of this service is the provision of low-or-no cost software for participating Offices. The IB has elected to support the Secure Shell suite of protocols and services, as it is widely supported in both free and commercial versions. Software packages in each of these categories are discussed below.

From a technical perspective, the Secure Shell Protocol is described in the Secure Shell [secsh] IETF Charter[1] (more commonly known as SSH). Specifically, the preliminary release of the service uses services defined in the SSH File Transfer Protocol Draft.

The XML data structures used for the request and submission of documents are described in other WIPO documents.

A. WHY SSH?

SSH provides a secure means of reliably transferring data over communications networks via the Secure Shell File Transfer Protocol (SFTP). Unlike the standard Internet File Transfer Protocol (FTP), user authentication information is never transmitted in the clear. All communications are encrypted using a user-selected data encryption algorithm. Unlike Transport Layer Security (TLS) enhanced FTP, only a single port is used for bi-directional communications, eliminating some rather difficult firewall and security configuration issues. Unlike Hypertext Transfer Protocol (HTTP), SFTP provides asynchronous, block-oriented data transfer verification, including the facility to restart interrupted transmissions from the point of interruption. Unlike Simple Object Access Protocol (SOAP), no independent software development is required. It should be noted, however, that Offices desiring to develop customized SSH/SFTP applications for integration with existing systems may do so.

SSH is available in many different implementations, both commercial and freely available. Numerous clients with excellent bulk-transfer functionality are available from a number of sources, thus significantly reducing software development and maintenance costs for national Offices and the IB. However, this does not affect the flexibility offered by the service; in fact, several free[2] and commercial development toolkits are available for those Offices wishing to

[1] http://www.ietf.org/html.charters/secsh-charter.html

[2] The OpenSSH toolkit is available from http://www.openssh.org/. Perl modules for SSH and SFTP are available from http://search.cpan.org/~drolsky/Net-SSH-Perl-1.25/lib/Net/SSH/Perl.pm and http://search.cpan.org/~drolsky/Net-SFTP-0.08/lib/Net/SFTP.pm, respectively.

integrate automated priority document access into their existing examination systems and processes.

The modularity of the PCT-EDI service is such that other access mechanisms and protocols can be easily added if needed. These include the protocols mentioned above (TLS-FTP, SOAP, HTTP, etc.), as well as file system oriented services such as samba.

B. THE WIPO SERVER

The IB uses the SSH Communications Security Corporation Tectia SSH2 server for Unix. This server supports the full range of SSH2 services, including public key authentication using both digital certificates and the standard SSH key trust model.

C. CLIENT SOFTWARE

Several clients are available for use with the PCT EDI. Two freely available Windows environment clients, PuTTY[3] and WinSCP[4], are described below. Both clients use the PuTTYgen key generator package. PuTTY is a simple client that is useful for automated tasks initiated from a Windows machine. WinSCP is a user-friendly full- featured client designed for use by an operator. It is suggested that first-time users become familiar with the WinSCP client.

A tested commercial product is the SSH Tectia[5] client for several platforms, including Microsoft Windows servers. This client also supports authentication through the use of industry-standard digital certificates.

Offices working in the Unix environment will find detailed instructions for the use of the OpenSSH secure shell package, including the *ssh* and *sftp* clients, and the *lftp* advanced SFTP client, which includes features such as site mirroring and batch download/upload scheduler.

III. OFFICE ACCOUNTS

A. INTRODUCTION

There are a variety of services available to the more advanced user, including data directory mirroring, the addition of additional authorized users to an Office account, and the development of customized access applications for Unix and Microsoft Windows platforms. These features are discussed in later sections.

B. OFFICE ACCOUNT AND DIRECTORY STRUCTURE

Each participating Office is assigned a unique account. Office accounts have a predefined directory structure as established by the IB:

- **Download** – the directory where the IB systems place documents

[3] http://www.chiark.greenend.org.uk/~sgtatham/putty/

[4] https://winscp.net/eng/index.php

[5] http://www.ssh.com

systematically sent to the Office,

- **Upload** – the directory where the Office places document packages and other agreed files (Search Fee Confirmation) for the IB,
- **Feedback** – the directory where the IB system places feedback files after processing the batch files received from the Office,
- **Upload-npe** – the directory where the Office places National Phase Entry files,
- **Feedback-npe** – the directory where the IB system places feedback files after processing the National Phase Entry files received from the Office.
- **Download-test** – the directory where the IB systems place documents systematically sent to the Office for test purpose,
- **Upload-test** – the directory where the Office places document packages and other agreed files (Search Fee Confirmation) for the IB for test purpose.

C. MAINTENANCE

Offices are required to perform certain basic housekeeping activities with respect to their accounts. Offices should delete old files from their “Download”, “Upload” and “Upload-npe” directories on a daily basis. Files more than 2 weeks old will be deleted automatically by the system. Users are given a warning each day of files due to be soon deleted, and deleted files are maintained in a running list. These lists are automatically placed in the home directory of each Office.

Offices may elect to have multiple users for a single Office account. A local administrator at the Office can control the number and identities of additional users assigned to the Office account. This is discussed in **Advanced Topics**, below.

Individuals who will be working as an additional user under an existing Office account should follow steps under **Setting up and Using WinSCP** and should mail their public keys (**public key only!!**) to their designated administrator. The administrator will rename the key as appropriate and install it on the PCT-EDI system.

D. EXAMPLES IN THIS DOCUMENT

Throughout this document, we will use the country code “xx” for demonstration purposes. We invite your feedback on the utility of the examples in this document. Comments may be sent to pcticd@wipo.int.

Please also note that Offices are free to select encryption algorithms and key types and strengths for the PCT-EDI cryptographic systems. The terms RSA and DSA are both used in the examples.

IV. CLIENT SOFTWARE CONFIGURATION

A. INTRODUCTION

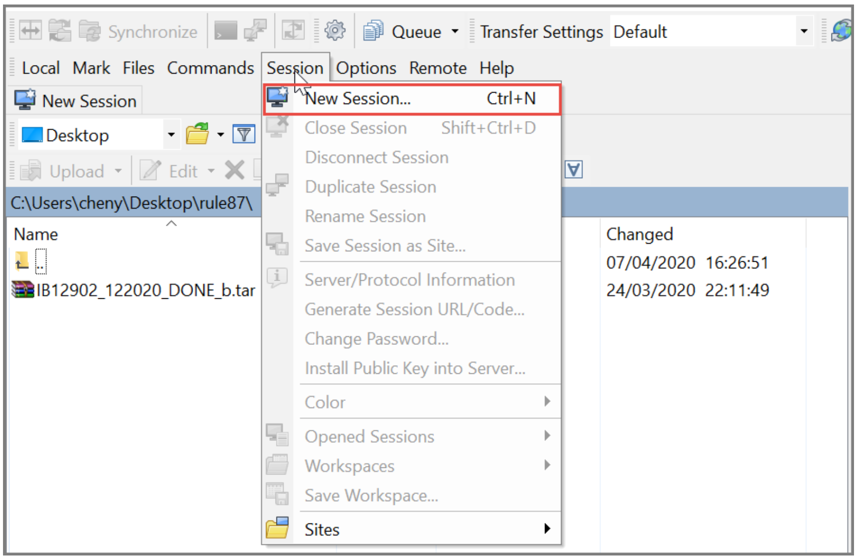
Numerous commercial and free application packages for SSH and SFTP exist. This section provides a “hands-on”, point-by-point reference guide for installing representative packages in the Windows environment as well as a generic Unix environment.

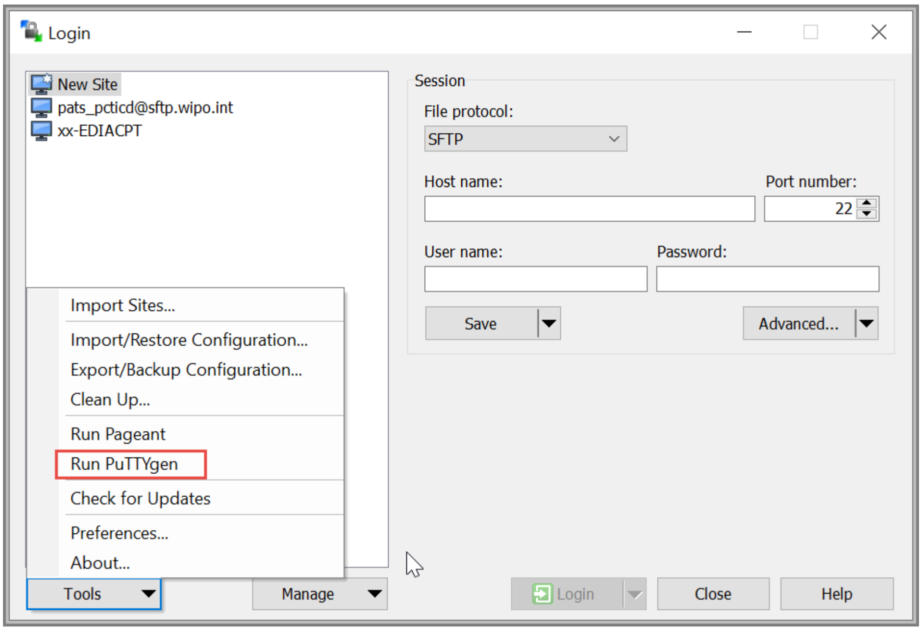
B. MICROSOFT WINDOWS ENVIRONMENT

This section will guide you through the setup of your Office account using two freely available packages, WinSCP and PuTTY, as well as the commercial Tectia SSH2 client.

1. Setting Up and Using WinSCP

- Install the WinSCP Client. It can be downloaded from: **https://winscp.net/eng/download.php**
- After downloading the package, run it. Select “Typical installation.” Follow the installation instructions.
- Start the PuTTYgen application. This application is used to create your authentication keys. Click on “Session” from the top menu and select “New Session.” In the popup window, click on “Tool” and select “Run PuTTYgen.”

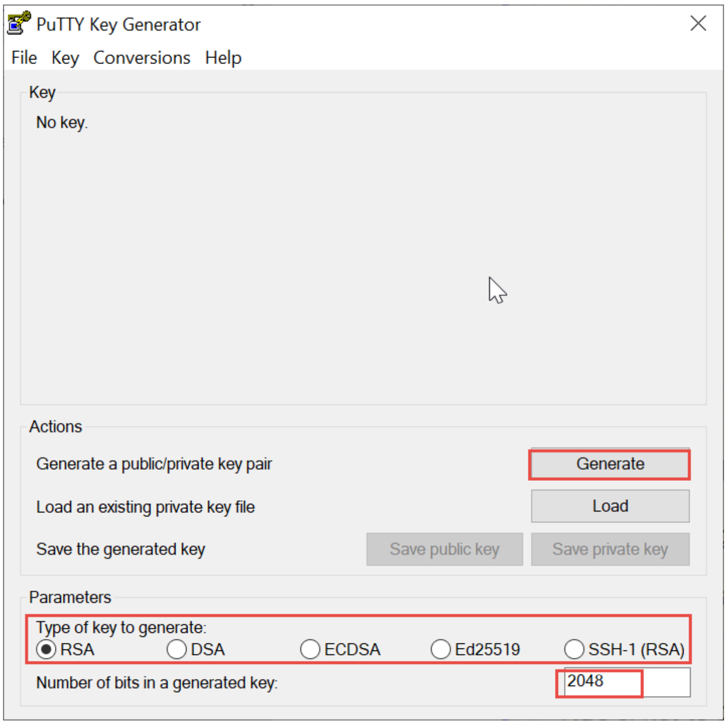




File protocol: SFTP

Port number: 22

Select either “RSA” or “DSA” for the “Type of key to generate”, with 2048 bit key size or greater. Press the “Generate” button and follow the instructions on the screen.



No key.

Generate a public/private key pair

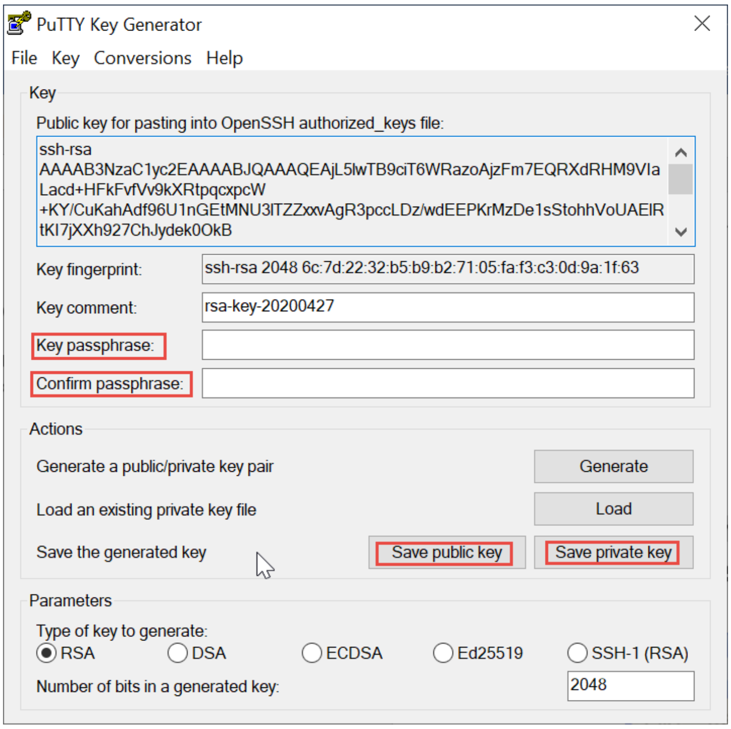
Load an existing private key file

Save the generated key

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ Ed25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

- When key generation is complete, you will be shown the screen below. Enter a good (at least 8 characters, with letters, numbers and punctuation marks) passphrase in the given blocks. You will be prompted for this passphrase whenever you use this key. The passphrase is never sent to the remote machine.



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAjL5lwTB9ciT6WRazoAjzFm7EQRXdRHM9VIa
Lacd+HFkFvfVv9kXRtpqcxpcW
+KY/CuKahAdf96U1nGEtMNU3lTZZxxvAgR3pccLDz/wdEEPKrMzDe1sStohhVoUAElR
tKI7jXXh927ChJydek0OkB
```

Key fingerprint: ssh-rsa 2048 6c:7d:22:32:b5:b9:b2:71:05:fa:f3:c3:0d:9a:1f:63

Key comment: rsa-key-20200427

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

Load an existing private key file

Save the generated key

Type of key to generate:

● RSA ○ DSA ○ ECDSA ○ Ed25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

- Press the “Save public key” button to save your public key with the extension “.pub”. Thus, for our example, the public key would be named “xx.pub”. Then, press the “Save private key” button to save your private key. Give it the same name, but without the “.pub” extension, e.g. enter “xx” in the naming box. You have created a 2048 bit RSA key. Your public key is named “xx.pub”. Your private key is named “xx.ppk”, where “xx” represents the account name for your Office. **Write down where you saved these keys!** You will need this location later.

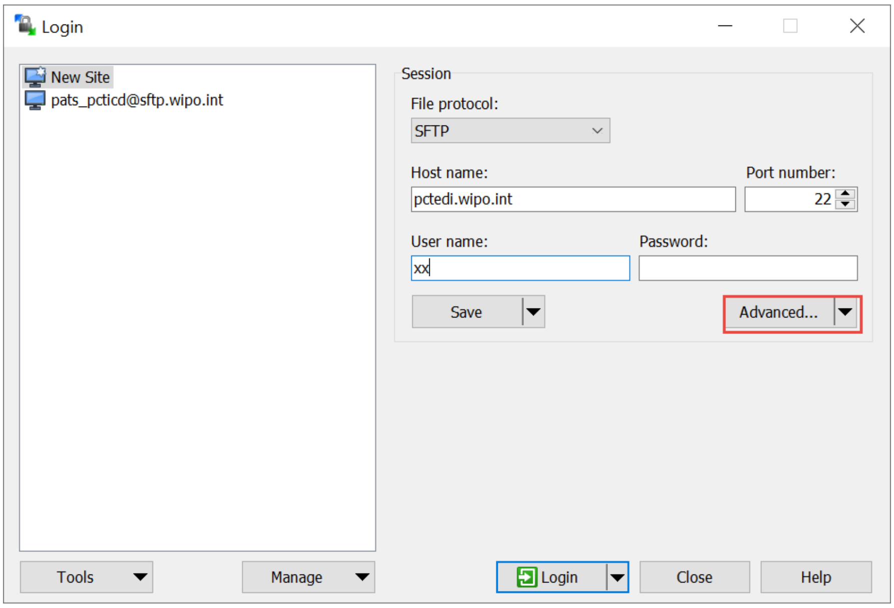
- Email your **public key** to: pcticd@wipo.int

NEVER SEND THE PRIVATE KEY. We will not use any keys where the private key has been transmitted by email.

Wait until you receive an email notifying that your key has been activated for login before proceeding to the next step.

- Start the WinSCP application, open a New Session.

Enter “pctedi.wipo.int” as the host name, and your two-letter country code as the “User name”.



File protocol: SFTP

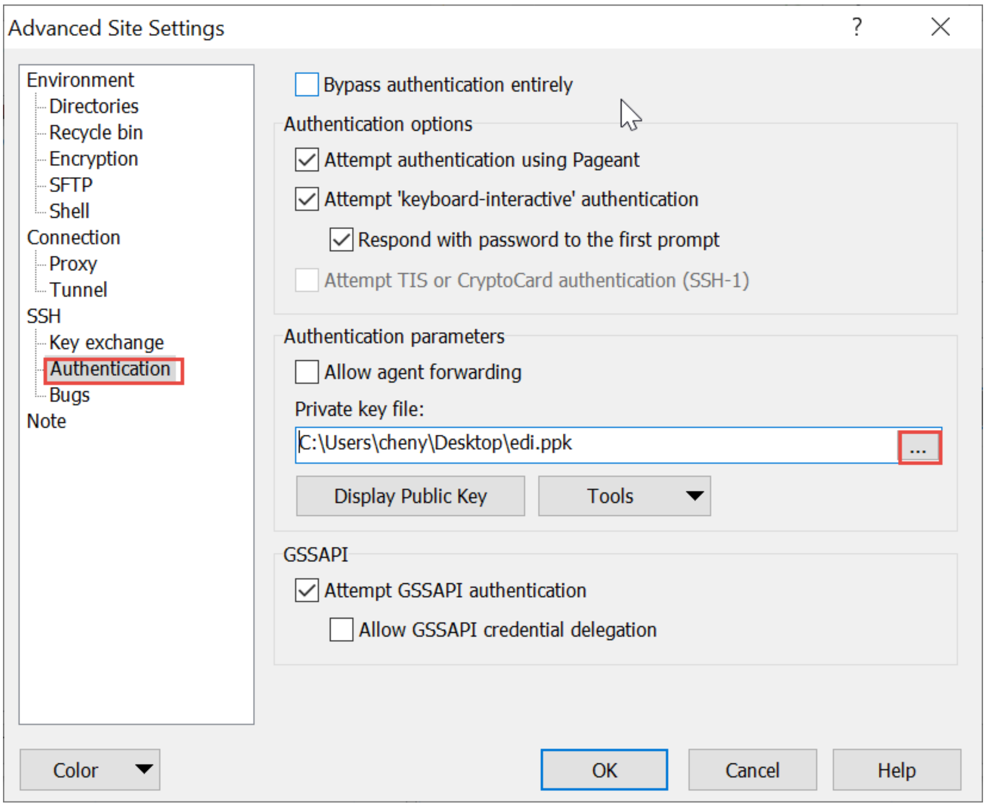
Host name: pctedi.wipo.int

Port number: 22

User name: xx

Password:

Click on “Advanced…” and go to “SHH” – “Authentication.” Under “Private key file” select the private key which you have created and click ok.



□ Bypass authentication entirely

Authentication options

☑ Attempt authentication using Pageant

☑ Attempt 'keyboard-interactive' authentication

☑ Respond with password to the first prompt

□ Attempt TIS or CryptoCard authentication (SSH-1)

Authentication parameters

□ Allow agent forwarding

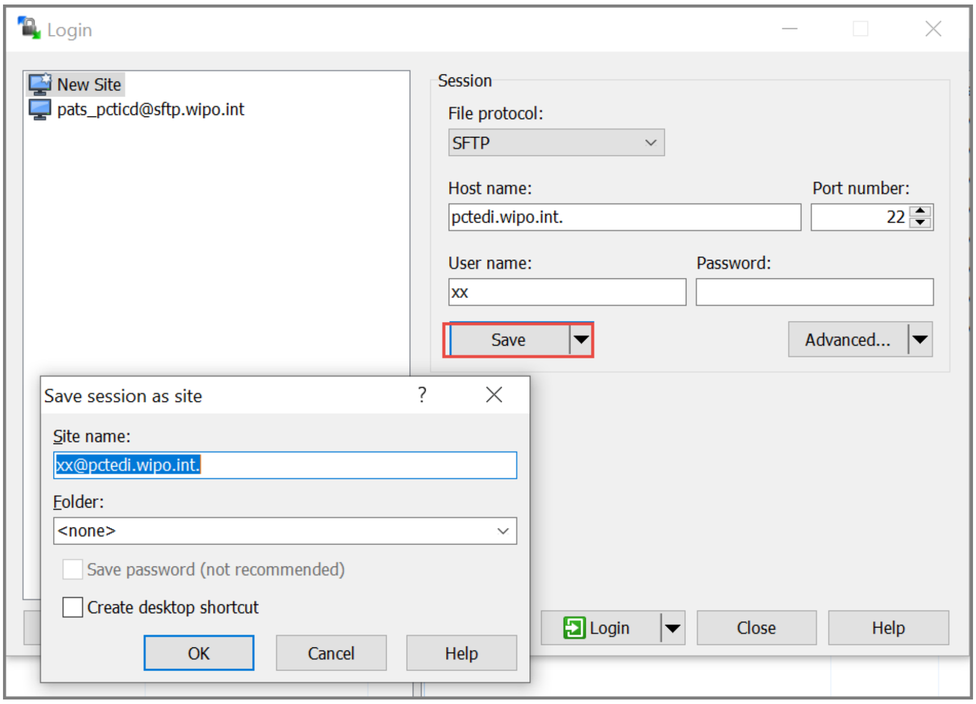
Private key file: C:\Users\cheny\Desktop\edi.ppk

GSSAPI

☑ Attempt GSSAPI authentication

□ Allow GSSAPI credential delegation

Save the session, naming it as ‘xx@pctedi.wipo.int.



Login

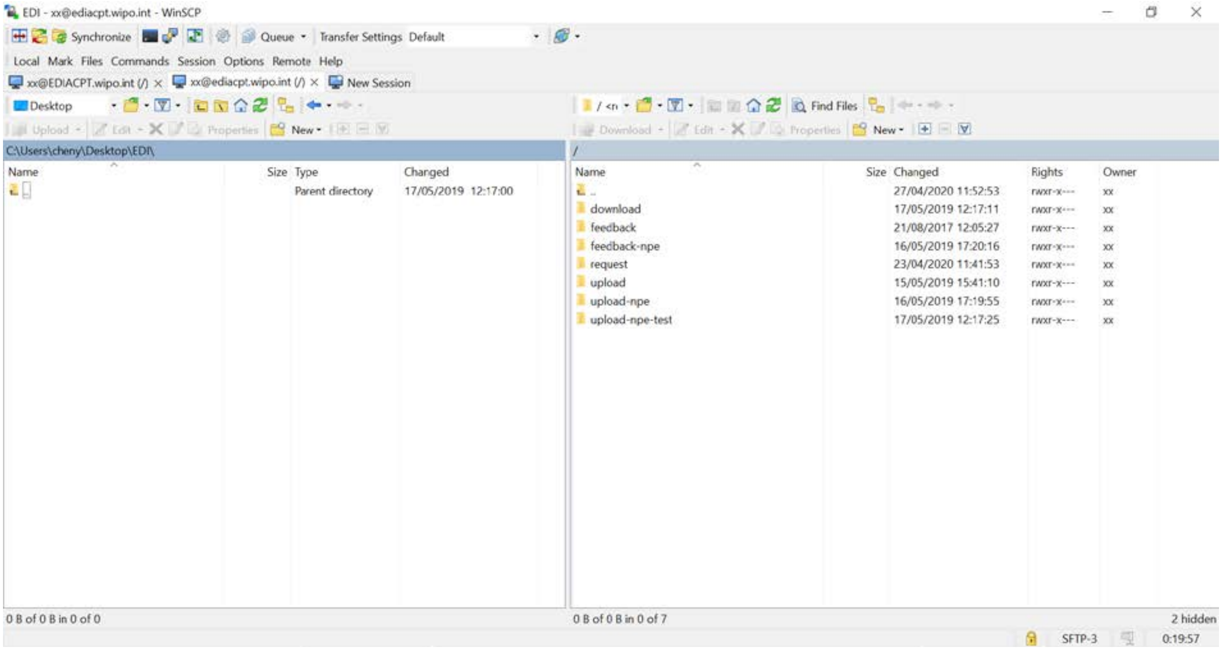
New Site
pats_pcticd@sftp.wipo.int

Session
File protocol: SFTP
Host name: pctedi.wipo.int.
Port number: 22
User name: xx
Password:

Save session as site
Site name: xx@pctedi.wipo.int.
Folder: <none>
□ Save password (not recommended)
□ Create desktop shortcut

- You are now ready to attempt a connection to the system. Press “Login”. You will be asked if you trust this host. Click “Yes”, and then you will be asked for your passphrase (that you input when creating your key-pair).

- Once you are successfully connected, you will see a split screen with the left- hand side containing a list of files in your local directory, and the right-hand side listing files in the remote directory.



EDI - xx@ediacpt.wipo.int - WinSCP

Transfer Settings Default

C:\Users\cheny\Desktop\EDI\

Name	Size	Type	Changed
		Parent directory	17/05/2019 12:17:00

/

Name	Size	Changed	Rights	Owner
..		27/04/2020 11:52:53	rwxr-x---	xx
download		17/05/2019 12:17:11	rwxr-x---	xx
feedback		21/08/2017 12:05:27	rwxr-x---	xx
feedback-npe		16/05/2019 17:20:16	rwxr-x---	xx
request		23/04/2020 11:41:53	rwxr-x---	xx
upload		15/05/2019 15:41:10	rwxr-x---	xx
upload-npe		16/05/2019 17:19:55	rwxr-x---	xx
upload-npe-test		17/05/2019 12:17:25	rwxr-x---	xx

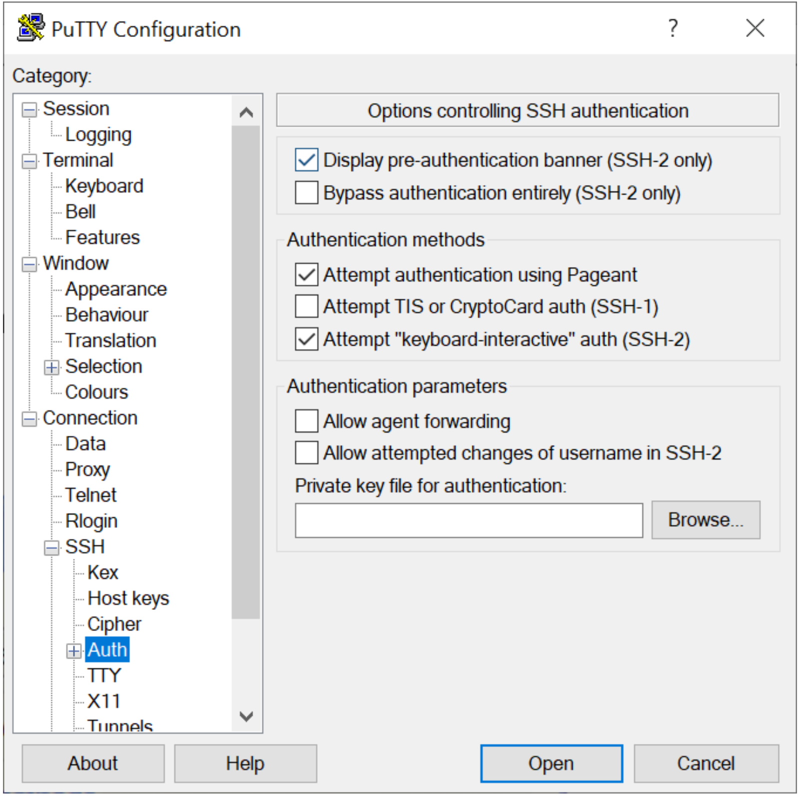
0 B of 0 B in 0 of 0
0 B of 0 B in 0 of 7
2 hidden
SFTP-3
0:19:57

2. Setting Up and Using PuTTY

- Run the installer package and extract the PuTTY components. PuTTY may be downloaded from:

http://www.chiark.greenend.org.uk/~sgtatham/putty/

- Once you install the PuTTY on your machine, you can easily run PuTTYgen. Go to Windows -> Start Menu -> All Programs -> PuTTY -> PuTTYgen. Start the PuTTYgen application. This application is used to create your authentication keys. Please refer to **Setting up and Using WinSCP**.
- Start the PuTTY application. In the left hand panel (Category), find the SSH entry, and click on “Auth”. Under “Private key file for authentication”, click “Browse”, and find the xx.ppk file. Select it.



Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)

□ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant

□ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

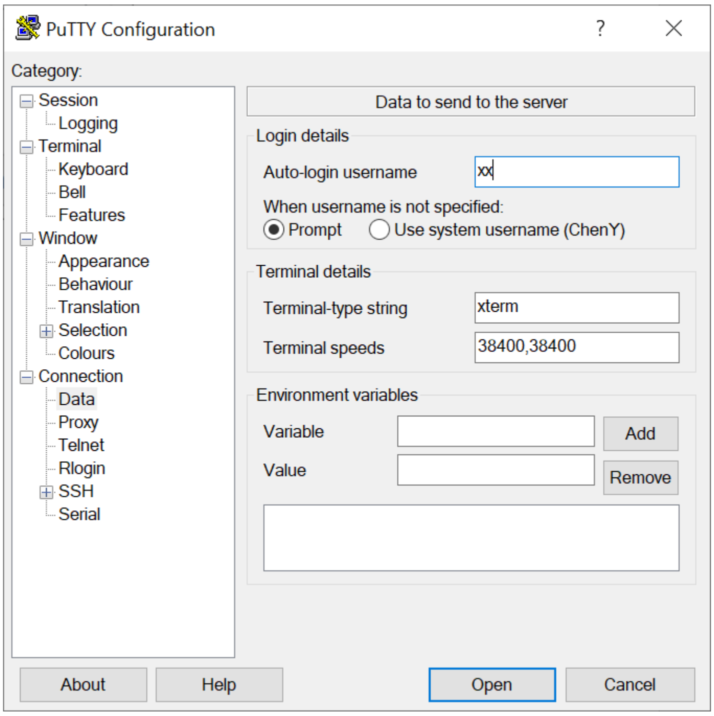
□ Allow agent forwarding

□ Allow attempted changes of username in SSH-2

Private key file for authentication:

Browse...

- Click on “Connection”, then “Data” in the left hand panel, and enter “xx” in the “Auto-login username” field.



Login details

Auto-login username: xx

When username is not specified:
● Prompt ○ Use system username (ChenY)

Terminal details

Terminal-type string: xterm

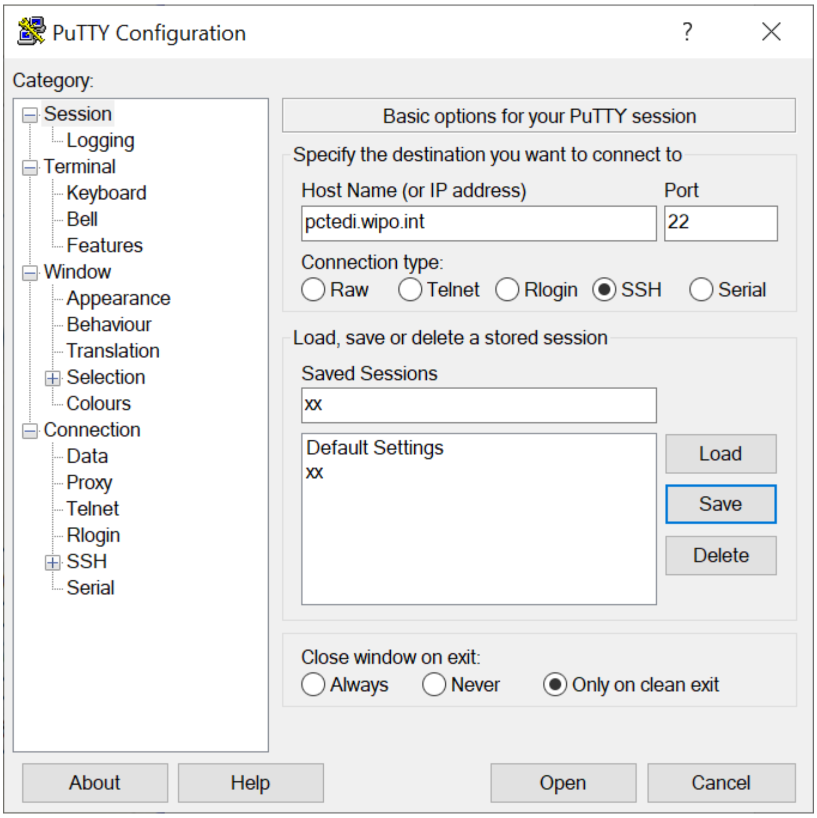
Terminal speeds: 38400,38400

Environment variables

Variable:

Value:

- In the left hand panel (Category) click on "Session" at the top. You will be back at the entry screen with "pctedi.wipo.int" entered as the host name, and "xx" as the session name. Click Save. You have now saved your key information under the session name xx.



Specify the destination you want to connect to

Host Name (or IP address): pctedi.wipo.int

Port: 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

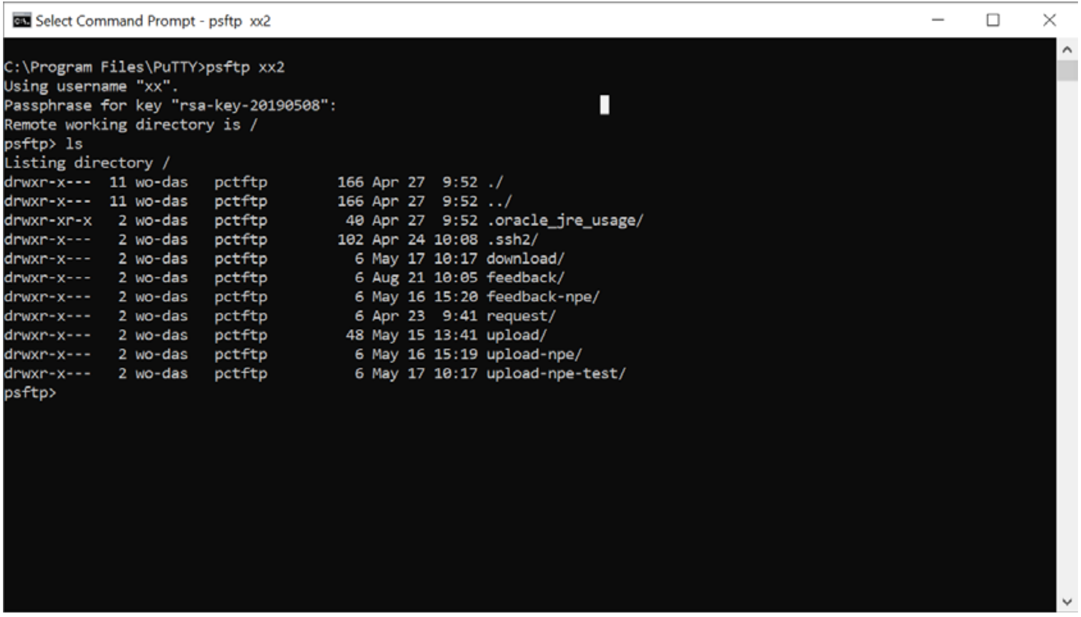
Saved Sessions: xx

Default Settings
xx

Close window on exit:
○ Always ○ Never ● Only on clean exit

- For security reasons, the PCT-EDI server does not allow remote SSH2 shell access. Therefore, the PuTTY SSH terminal client itself will not be used to access your account. Instead, the PuTTY psftp client will be used. It is a very simple command-line based client that uses PuTTY sessions such as the one you just made. For a more user friendly client, see the section that covers "WinSCP".

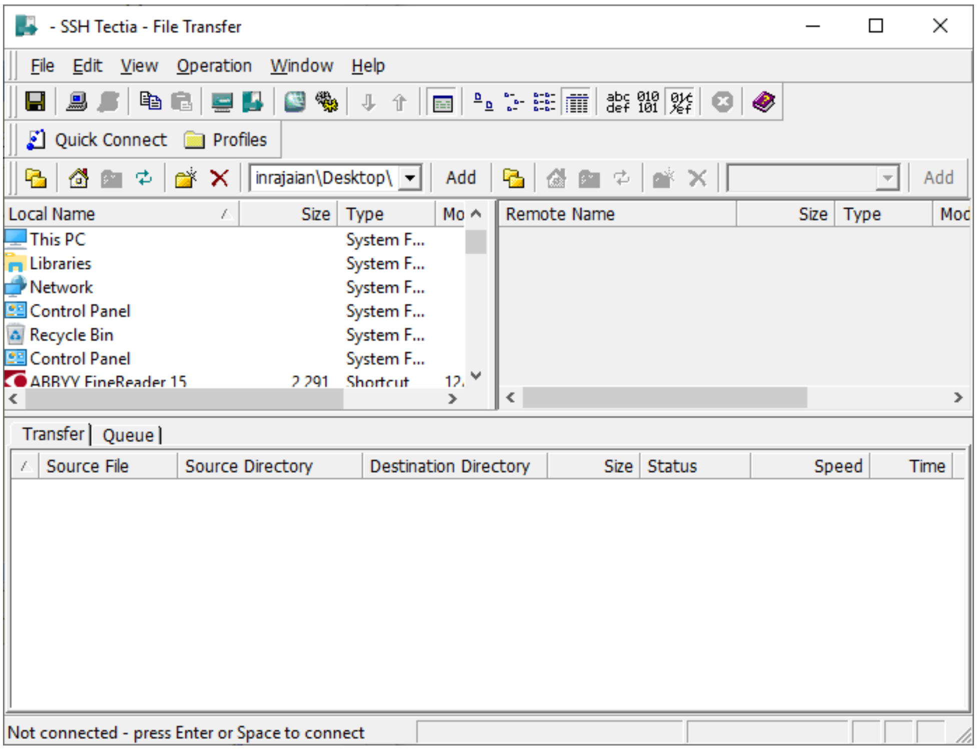
- To begin using psftp, open an MS-DOS window and, if necessary, switch to the PuTTY directory (using the MS-DOS "cd " command). Type "psftp xx" where "xx" is the name of the session you just created. You will be prompted for the passphrase you entered when you created the key, and then will be logged on using your public key. Note that the passphrase is not a password; it is never sent to the remote system. It simply secures your private key in the case of an unauthorized user gaining access to your local computer.



```
C:\Program Files\PuTTY>psftp xx2
Using username "xx".
Passphrase for key "rsa-key-20190508":
Remote working directory is /
psftp> ls
Listing directory /
drwxr-x---  11 wo-das   pctftp       166 Apr 27  9:52 ./
drwxr-x---  11 wo-das   pctftp       166 Apr 27  9:52 ../
drwxr-xr-x   2 wo-das   pctftp        40 Apr 27  9:52 .oracle_jre_usage/
drwxr-x---   2 wo-das   pctftp       102 Apr 24 10:08 .ssh2/
drwxr-x---   2 wo-das   pctftp         6 May 17 10:17 download/
drwxr-x---   2 wo-das   pctftp         6 Aug 21 10:05 feedback/
drwxr-x---   2 wo-das   pctftp         6 May 16 15:20 feedback-npe/
drwxr-x---   2 wo-das   pctftp         6 Apr 23  9:41 request/
drwxr-x---   2 wo-das   pctftp        48 May 15 13:41 upload/
drwxr-x---   2 wo-das   pctftp         6 May 16 15:19 upload-npe/
drwxr-x---   2 wo-das   pctftp         6 May 17 10:17 upload-npe-test/
psftp>
```

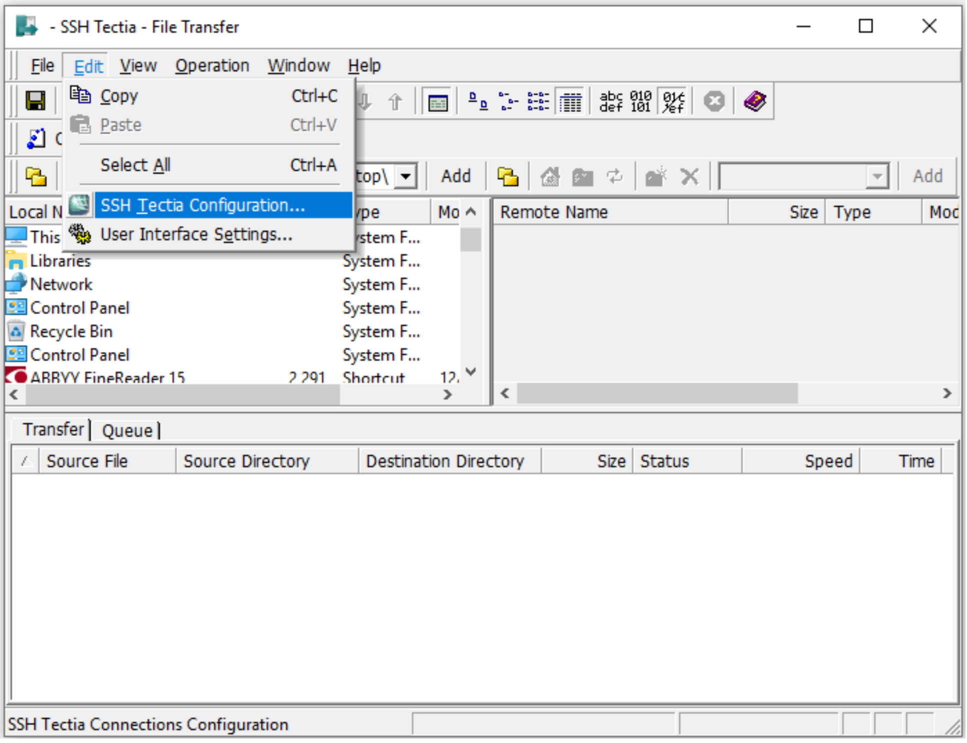
3. Setting Up and Using the Tectia SSH Client

- Download and install the Tectia SSH client for Windows. Follow the installer instructions and start the Tectia file transfer client. The client has the icon of a file folder, not the icon of a terminal screen. The following examples are based upon the freely available version of the commercial Tectia client, without PKI support. You will see the following screen:

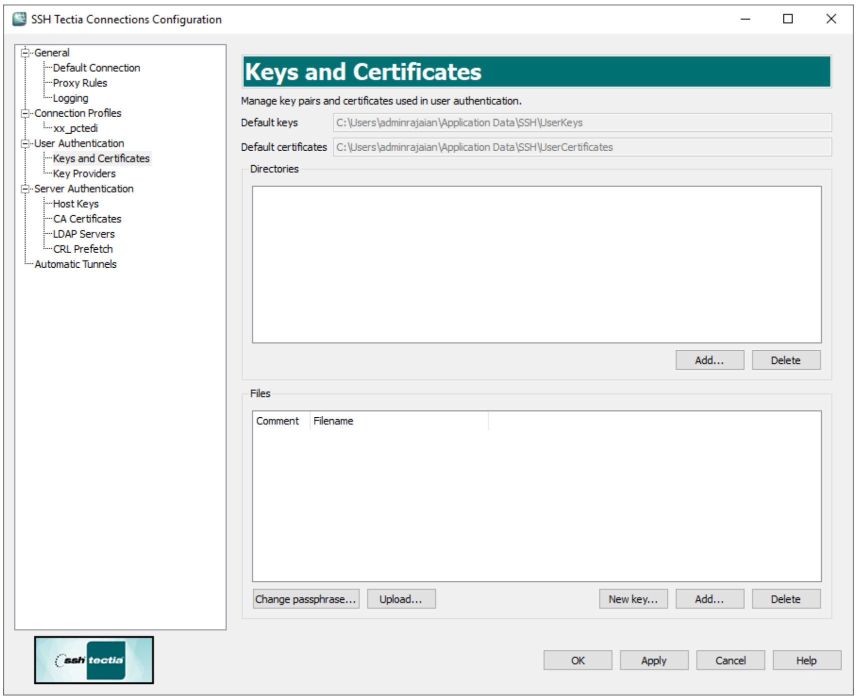


Not connected - press Enter or Space to connect

- Select “Edit-> SSH Tectia Configuration” from the menu. Then on the left-hand side of the Settings control panel, click on “User Authentication-> Keys and Certificates”.



SSH Tectia Connections Configuration



SSH Tectia Connections Configuration

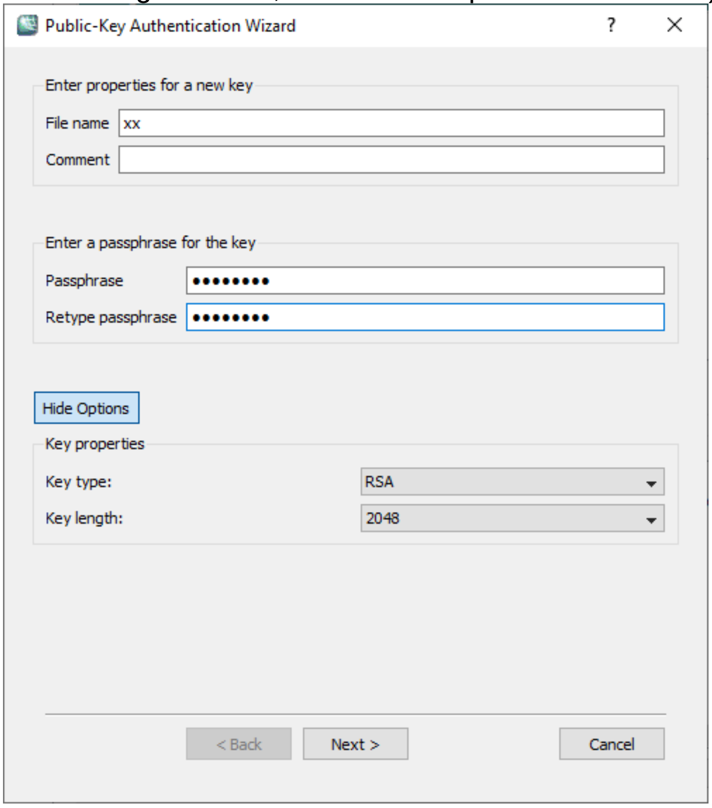
Keys and Certificates

Manage key pairs and certificates used in user authentication.

Default keys C:\Users\adminrajaian\Application Data\SSH\UserKeys

Default certificates C:\Users\adminrajaian\Application Data\SSH\UserCertificates

- Press “New Key…”. You will be prompted to provide a file name, comment and passphrase for the private key. Fill in the blanks as you desire, being sure to pick a good passphrase (minimum of 8 characters, consisting of letters, numbers and punctuation marks). Press “Next”.



Public-Key Authentication Wizard

Enter properties for a new key

File name xx

Comment

Enter a passphrase for the key

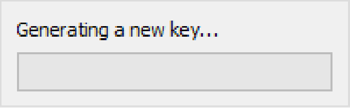
Passphrase ••••••••

Retype passphrase ••••••••

Key properties

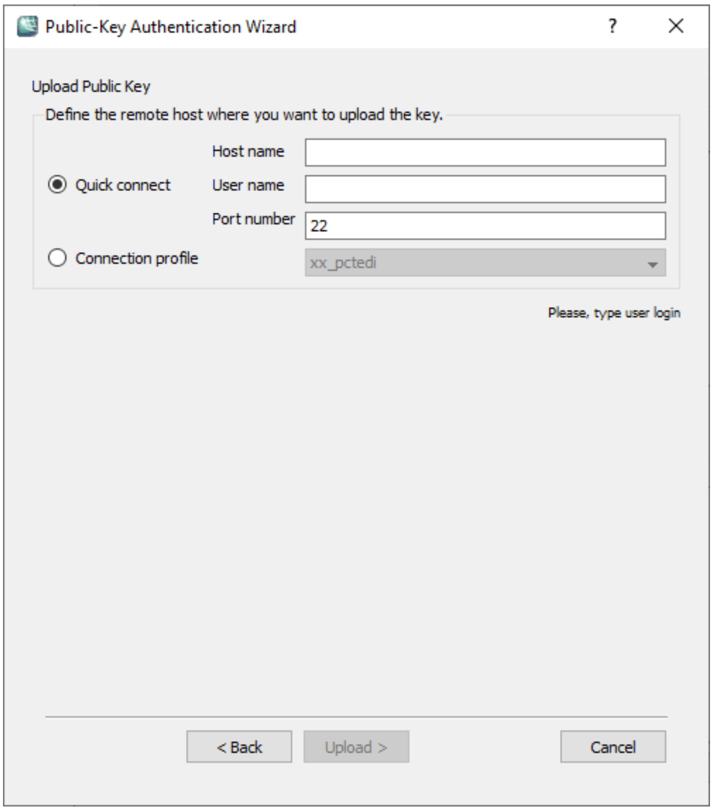
Key type: RSA

Key length: 2048



Generating a new key...

- Click “Next” and wait for the key generation to finish. Click “Next” again.

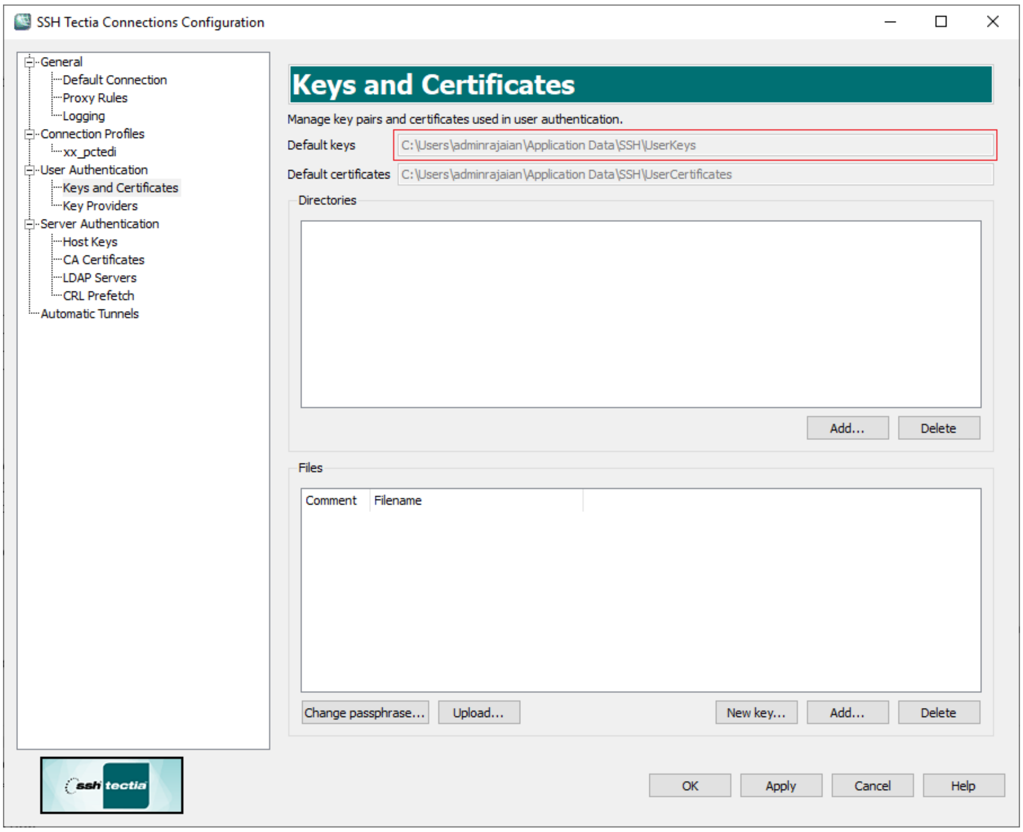


Port number: 22

Connection profile: xx_pctedi

Please, type user login

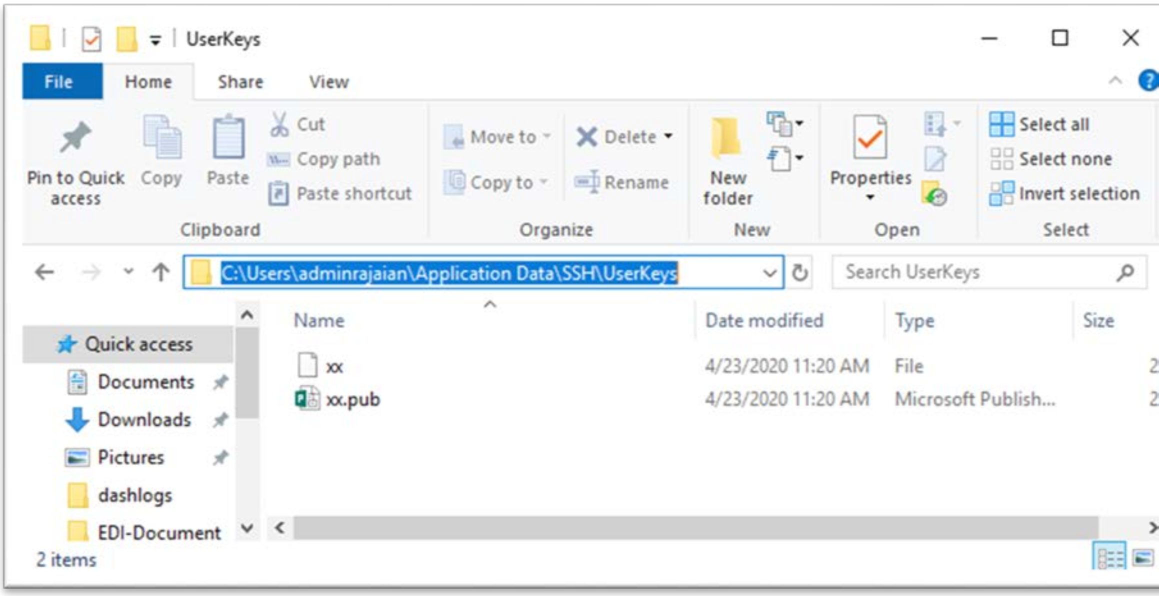
- Click on “Cancel”. The generated key will be in the below highlighted “Default Keys” folder.



Manage key pairs and certificates used in user authentication.

Default keys: C:\Users\adminrajaian\Application Data\SSH\UserKeys

Default certificates: C:\Users\adminrajaian\Application Data\SSH\UserCertificates



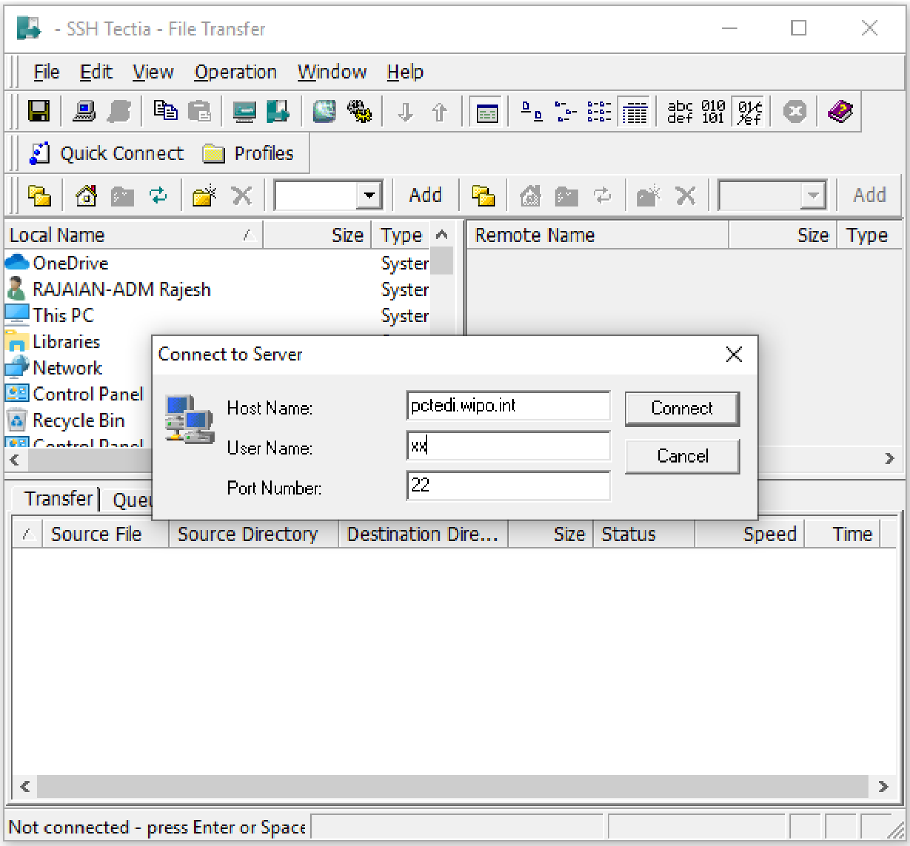
C:\Users\adminrajaian\Application Data\SSH\UserKeys

Name	Date modified	Type
xx	4/23/2020 11:20 AM	File
xx.pub	4/23/2020 11:20 AM	Microsoft Publish...

- Send the xx.pub file to pcticd@wipo.int.

  Wait until you receive an email notifying that your key has been activated for login before proceeding to the next step.

- Start the Tectia file transfer client, then press the “Quick Connect” button and fill in the dialog box fields as below with “pctedi.wipo.int” for the “Host Name”, your account name for the “User Name”. Then, press “Connect”.



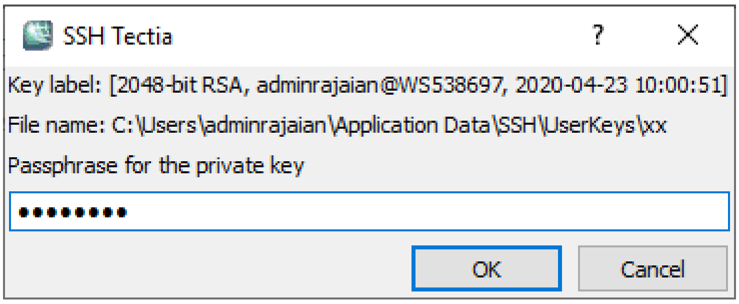
Connect to Server

Host Name: pctedi.wipo.int

User Name: xx

Port Number: 22

- You will be prompted for the key passphrase. Type in the passphrase and click “OK”. You will be connected to the PCT-EDI server.



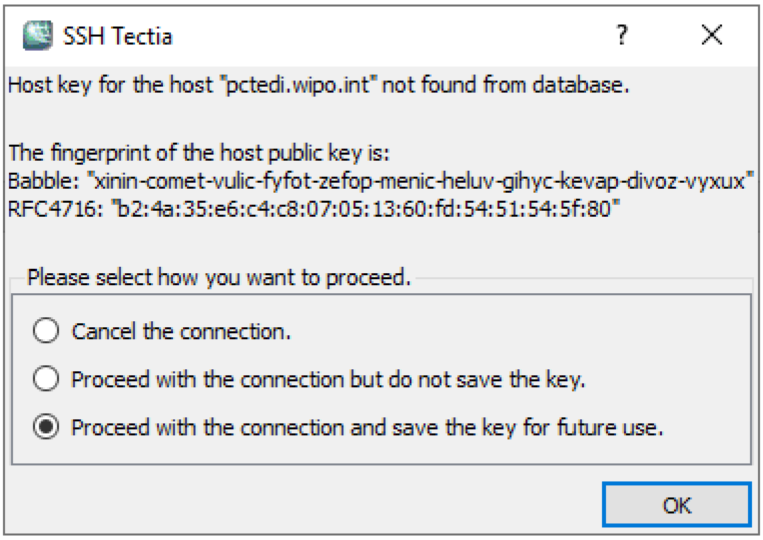
Key label: [2048-bit RSA, adminrajaian@WS538697, 2020-04-23 10:00:51]

File name: C:\Users\adminrajaian\Application Data\SSH\UserKeys\xx

Passphrase for the private key

••••••••

You will then be prompted to accept the host key of the PCT-EDI server. Click“OK”.



Host key for the host "pctedi.wipo.int" not found from database.

The fingerprint of the host public key is:
Babble: "xinin-comet-vulic-fyfot-zefop-menic-heluv-gihyc-kevap-divoz-vyxux"
RFC4716: "b2:4a:35:e6:c4:c8:07:05:13:60:fd:54:51:54:5f:80"

Please select how you want to proceed.

○ Cancel the connection.
○ Proceed with the connection but do not save the key.
● Proceed with the connection and save the key for future use.

Press “Profiles” and “Add Profile”. You will be prompted to name the current profile. Name it “xx-key” and click on “Add to Profiles” to save your key-based connection profile. You may now connect to the PCT-EDI server at any time by selecting the profile “xx-key.”

C. UNIX ENVIRONMENT

The following sections are written for the experienced Unix user or system administrator. It discusses the freely available **OpenSSH** and **lftp** packages for the Unix environment.

OpenSSH is a free, non-commercial version of the SSH protocol suite of network connectivity tools.

The OpenSSH suite includes a variety of utilities of interest to intellectual property Offices requiring secure Internet-based communications. Of particular interest to users of the PCT-EDI system is the *sftp* client, which implements (as of OpenSSH 2.5.0) complete SFTP support.

OpenSSH supports SSH protocol versions 1.3, 1.5, and 2.0, permitting communication with most UNIX, Windows and other commercial ssh implementations. The SSH 2.0 protocol avoids using the patented RSA algorithm and uses the freely useable DH and DSA algorithms instead.

lftp is a file transfer program that allows sophisticated sftp, ftp, http and other connections to other hosts. **lftp** can handle seven file access methods - ftp, ftps, http, https, hftp, fish, sftp and file (https and ftps are only available when lftp is compiled with the openssl library). You can specify the method to use in the “open URL” command, e.g. “open sftp://pctedi.wipo.int”. Sftp is implemented in ssh2 as a call to the external program *sftp* and thus the OpenSSH “ssh” application to be installed and functioning.

Every operation in **lftp** is reliable, that is any non-fatal error is ignored and the operation is repeated. This means that if downloading breaks for any reason, it will be restarted from the point automatically.

lftp has a shell-like command syntax allowing you to launch several commands in parallel in the background (&). It is also possible to group commands within () and execute them in background. All background jobs are executed in the same single process. You can put a foreground job in the background with ^Z (ctl-z) and bring it back with the `wait' command (or `fg' which is an alias to `wait'). To list running jobs, use the command `jobs'. Some commands allow redirecting their output (cat, ls, ...) to file or via a pipe to external command. Commands can be executed conditionally based on the termination status of previous command (&&, ||).

If you exit **lftp** when some jobs are not yet finished yet, **lftp** will automatically place itself in nohup mode in the background.

lftp has a built in mirror function which can download or update a whole directory tree. There is also a reverse mirror function (mirror -R) which uploads or updates a directory tree on the server.

The command “at” launches a job at specified time in the current context, the command “queue” to queue commands for sequential execution with the connected current server, as well as other useful features such as a precious commands history. The best source of usage documentation for **lftp** is the *lftp(1)* man page, installed when the client is installed. It can also be viewed at http://lftp.yar.ru/lftp-man.html

On startup, **lftp** executes */etc/lftp.conf* and then *~/.lftprc* and *~/.lftp/rc*. Individual users can place aliases and `set' commands there. These commands are discussed in the *lftp(1)* man page.

lftp has a number of user-controlled variables. You can use `set -a' to see all of the variables and their values or `set -d' to see list of default values.

1. Installing and Using the OpenSSH Utilities[6]

a) *Installation*

Installation of the OpenSSH utilities can only be performed by your Unix systems administrator, and is dependent upon the target Unix system. Installation instructions, source code, and precompiled binaries for a variety of Unix systems can be found at http://www.openssh.com/portable.html

The following usage instructions are based upon the existence of a properly installed OpenSSH subsystem.

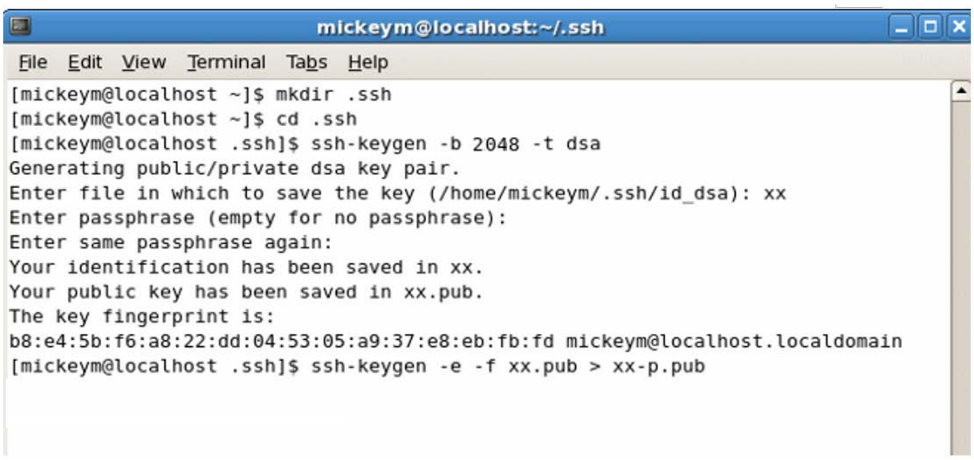
b) *Configuration and Use*

This section describes the configuration and use of the OpenSSH *sftp* client. *Sftp* uses a set of commands similar to those found in normal Internet FTP. It also describes the use of *ssh-keygen* to create your public and private keys.

- Log in to your local Unix machine and create a directory named ".ssh". Change to that directory. You must be in the ".ssh" directory when executing the next commands.
- Enter the command "ssh-keygen -b 2048 -t dsa" to generate a OpenSSH format 2048 bit DSA key. You will be prompted for a file in which to save the key. Enter the two-letter WIPO ST.3 country code for your Office.

When key generation is complete, you will be prompted to enter a passphrase. If this account will be used for non-interactive applications (such as automated scripts and downloading, press "Enter". Otherwise, if the account will be used by humans, enter a good (at least 8 characters, with letters, numbers and punctuation marks) passphrase in the given blocks. You will be prompted for this passphrase whenever you use this key. The passphrase is never sent to the remote machine. The following examples assume the existence of a passphrase.

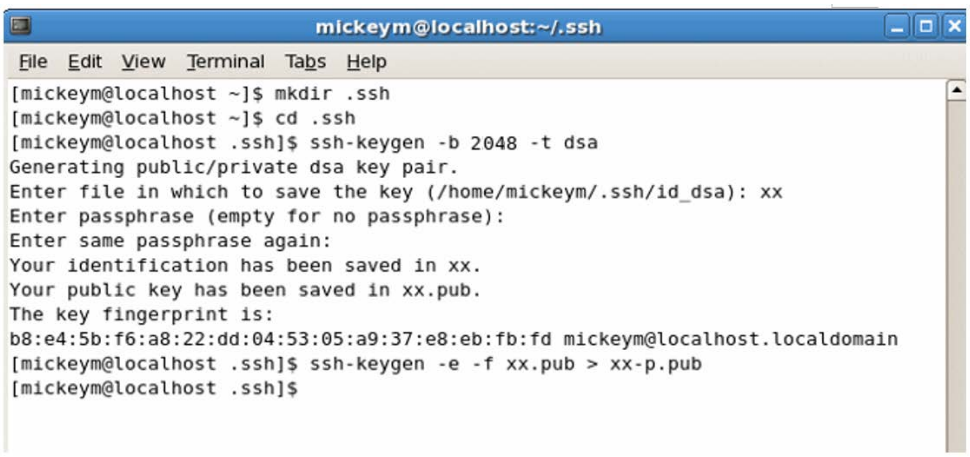
[6] Derived from the OpenSSH web site documentation



```
[mickeym@localhost ~]$ mkdir .ssh
[mickeym@localhost ~]$ cd .ssh
[mickeym@localhost .ssh]$ ssh-keygen -b 2048 -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/mickeym/.ssh/id_dsa): xx
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in xx.
Your public key has been saved in xx.pub.
The key fingerprint is:
b8:e4:5b:f6:a8:22:dd:04:53:05:a9:37:e8:eb:fb:fd mickeym@localhost.localdomain
[mickeym@localhost .ssh]$ ssh-keygen -e -f xx.pub > xx-p.pub
```

This key pair (xx and xx.pub) are suitable for all OpenSSH applications.

- The IB uses the commercial Tectia SSH server, which uses public keys in the newer SECSH Public Key format. It is now necessary to convert your xx.pub public key into this format. Type “ssh-keygen -e -f xx.pub > xx-p.pub”, where “xx” is your account name. This will copy your xx.pub public key file to xx-p.pub, converting it to the SECSH format.



```
[mickeym@localhost ~]$ mkdir .ssh
[mickeym@localhost ~]$ cd .ssh
[mickeym@localhost .ssh]$ ssh-keygen -b 2048 -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/mickeym/.ssh/id_dsa): xx
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in xx.
Your public key has been saved in xx.pub.
The key fingerprint is:
b8:e4:5b:f6:a8:22:dd:04:53:05:a9:37:e8:eb:fb:fd mickeym@localhost.localdomain
[mickeym@localhost .ssh]$ ssh-keygen -e -f xx.pub > xx-p.pub
[mickeym@localhost .ssh]$
```

- Email your converted public key (xx-p.pub) to: pcticd@wipo.int

  NEVER SEND THE PRIVATE KEY! We will not use any keys where the private key has been transmitted by email.

  Wait until you receive an email notifying that your key has been activated for login before proceeding to the next step.

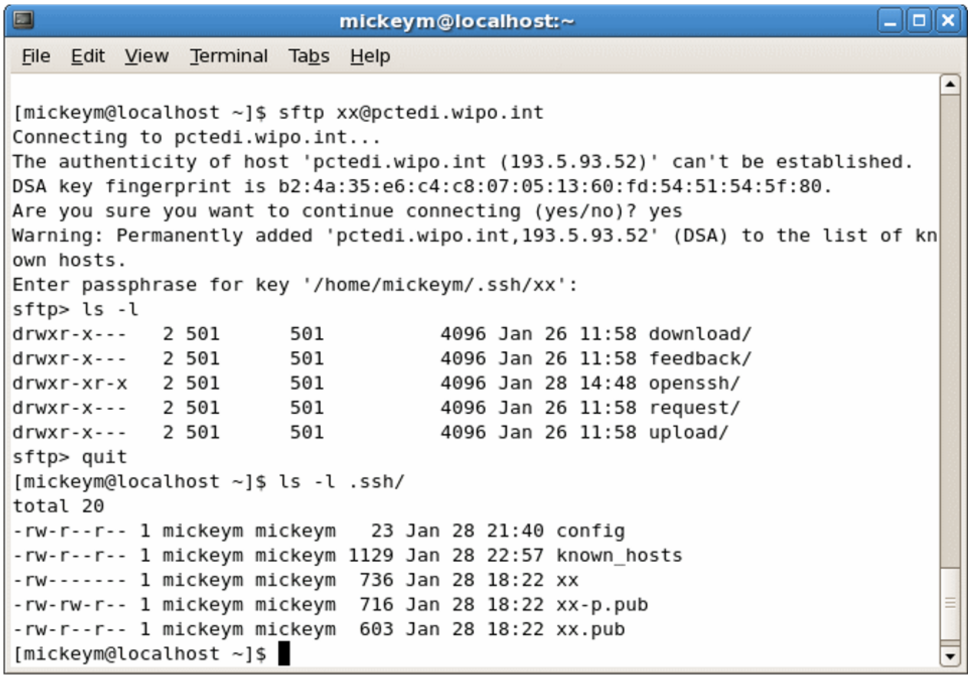
- Ensure that you are in the “.ssh” directory by using the“*pwd*” command. You will now create your OpenSSH configuration file. Use your favorite Unix text editor to create the file “config” containing the line (without quotes) “IdentityFile ~/.ssh/xx”. This will cause your private key “xx” to be used to identify yourself to the PCT EDI server.

Enter the command “*sftp xx@pctedi.wipo.int”* to open a connection to the server using “xx” as the account name.

You will be prompted to accept the authenticity of the pctedi.wipo.int PCT-EDI server. Type “yes”.

If you assigned a passphrase to your private key, you will be prompted for it. Then, you will be logged into your account on the PCT-EDI server.

The screenshot below shows the existence of the “config” file, the results of connecting to the server, and a listing of the “xx” account directory space on the PCT-EDI server.



```
[mickeym@localhost ~]$ sftp xx@pctedi.wipo.int
Connecting to pctedi.wipo.int...
The authenticity of host 'pctedi.wipo.int (193.5.93.52)' can't be established.
DSA key fingerprint is b2:4a:35:e6:c4:c8:07:05:13:60:fd:54:51:54:5f:80.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'pctedi.wipo.int,193.5.93.52' (DSA) to the list of kn
own hosts.
Enter passphrase for key '/home/mickeym/.ssh/xx':
sftp> ls -l
drwxr-x---    2 501      501          4096 Jan 26 11:58 download/
drwxr-x---    2 501      501          4096 Jan 26 11:58 feedback/
drwxr-xr-x    2 501      501          4096 Jan 28 14:48 openssh/
drwxr-x---    2 501      501          4096 Jan 26 11:58 request/
drwxr-x---    2 501      501          4096 Jan 26 11:58 upload/
sftp> quit
[mickeym@localhost ~]$ ls -l .ssh/
total 20
-rw-r--r-- 1 mickeym mickeym   23 Jan 28 21:40 config
-rw-r--r-- 1 mickeym mickeym 1129 Jan 28 22:57 known_hosts
-rw------- 1 mickeym mickeym  736 Jan 28 18:22 xx
-rw-rw-r-- 1 mickeym mickeym  716 Jan 28 18:22 xx-p.pub
-rw-r--r-- 1 mickeym mickeym  603 Jan 28 18:22 xx.pub
[mickeym@localhost ~]$ █
```

2. Installing and Using lftp

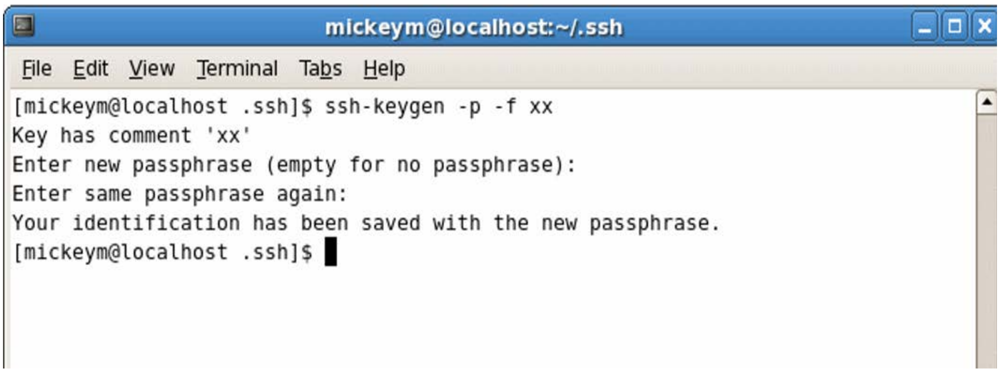
a) *Installation*

Before installing lftp, the OpenSSH suite of utilities must be installed as above. It is also necessary to have configured your local account to support public key authentication with the PCT-EDI server. Once you can successfully connect to the PCT-EDI server using “*sftp*” and public key authentication as above, you may install lftp. As with OpenSSH, lftp should be installed by an experienced Unix systems administrator. Source code and precompiled binaries, along with installation instructions, may be found at http://lftp.yar.ru/

The local configuration and use instructions below assume the presence of a working, correct lftp installation.

b) Configuration and Use

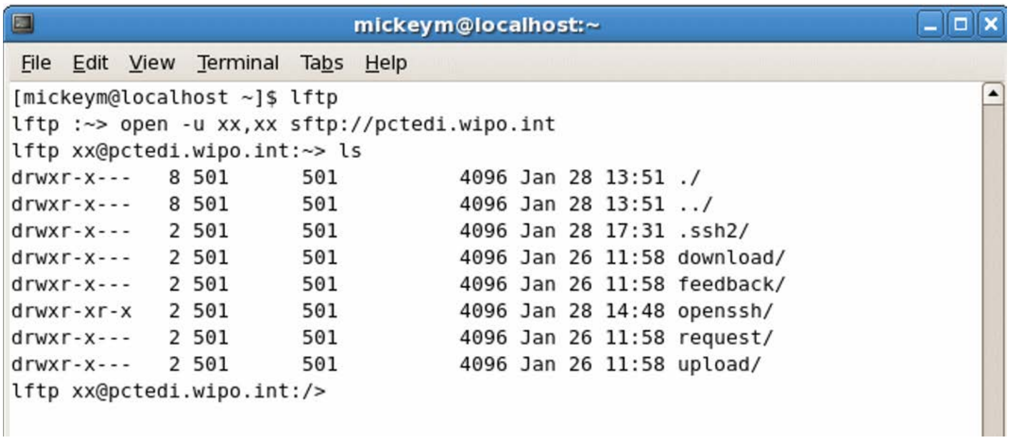
- Lftp requires the use of an unprotected private key file in the user ".ssh" directory. If you current OpenSSH private key is protected, you may unprotect it with the "*ssh-keygen*" application. First, cd to the ".ssh" directory. Then, type "*ssh-keygen-p*" and follow the instructions as shown below. Enter your account name as the file name. You will be prompted for your existing passphrase. You may then change the passphrase to empty.



```
[mickeym@localhost .ssh]$ ssh-keygen -p -f xx
Key has comment 'xx'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
[mickeym@localhost .ssh]$ █
```

- You may now connect to the PCT-EDI system. Type "*lftp*". You will see the lftp prompt. Then, type "open –u xx,xx sftp://pctedi.wipo.int". This specifies an SFTP connection to pctedi.wipo.int, with the username xx. Note that the second "xx" after the –u flag is the account password. No password is needed for this account, as we are using public key authentication. The second "xx" is simply a filler to prevent lftp from asking for a password.

  After entering the "command" you will once again see the lftp prompt. Simply type "*ls*" to make the connection and show a directory listing.



```
[mickeym@localhost ~]$ lftp
lftp :~> open -u xx,xx sftp://pctedi.wipo.int
lftp xx@pctedi.wipo.int:~> ls
drwxr-x---   8 501      501          4096 Jan 28 13:51 ./
drwxr-x---   8 501      501          4096 Jan 28 13:51 ../
drwxr-x---   2 501      501          4096 Jan 28 17:31 .ssh2/
drwxr-x---   2 501      501          4096 Jan 26 11:58 download/
drwxr-x---   2 501      501          4096 Jan 26 11:58 feedback/
drwxr-xr-x   2 501      501          4096 Jan 28 14:48 openssh/
drwxr-x---   2 501      501          4096 Jan 26 11:58 request/
drwxr-x---   2 501      501          4096 Jan 26 11:58 upload/
lftp xx@pctedi.wipo.int:/>
```

- You may now use lftp however you like. The lftp main page contains extensive information on the various lftp features.

V. ADVANCED TOPICS

A. INTRODUCTION

This section covers the advanced topics of managing multiple users within a single Office account, and the development of customized applications for the PCT-EDI service.

B. ADDING AND REVOKING USERS

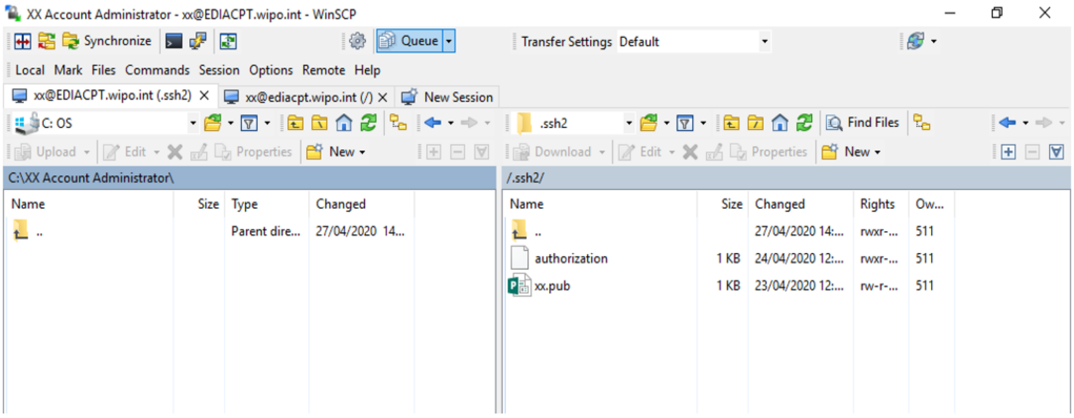
While each Office is assigned only one account, it is quite simple to add additional users to the Office account, and revoke them as necessary. Using public key authentication, no passwords need be circulated when new users are added, or changed when existing users are deleted.

It is suggested that one individual in each Office be responsible for the addition and/or deletion of users. This individual will be referred to as the “account administrator” in this example. The account administrator will be shown using the WinSCP client.

The addition and deletion of users is simply a function of uploading a SECSH-format public key for each new user, and editing the “authorization” file in the .ssh2 directory.

To manage multiple users, the account administrator should create a local “mirror” of the remote “.ssh2” directory. In this example, there is a folder on the local machine called “XX Account Administrator” which shall be used to store copies of the user public keys and the authorization file. The account administrator must also developing a naming system for public keys that allow them to be associated with an individual or machine. The account administrator should recall that any authorized user can view and manipulate the contents of the “.ssh2” directory for the Office account. Thus, the account administrator may wish to implement a naming scheme that distinguishes automated access accounts from individual accounts. In our example, keys will be differentiated by type (P=Personal, A=Automated) and 4 digit serial number. Example, P0001, P0002, A0003.

- Begin by logging into the Office account using the WinSCP client. Switch to the .ssh2 remote directory, and the local “XX Account Administrator” folder. Select the two files in the .ssh2 directory and copy them to the local folder.



XX Account Administrator - xx@EDIACPT.wipo.int - WinSCP

Transfer Settings Default

xx@EDIACPT.wipo.int (.ssh2) | xx@ediacpt.wipo.int (/) | New Session

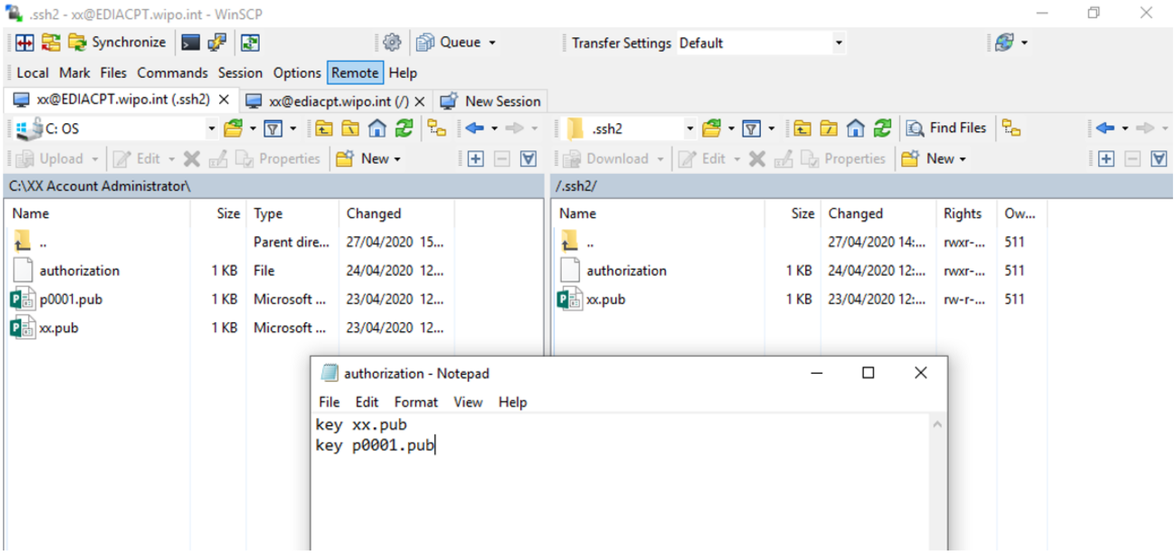
C:\XX Account Administrator\

Name	Size	Type	Changed
..		Parent dire...	27/04/2020 14...

/.ssh2/

Name	Size	Changed	Rights	Ow...
..		27/04/2020 14:...	rwxr-...	511
authorization	1 KB	24/04/2020 12:...	rwxr-...	511
xx.pub	1 KB	23/04/2020 12:...	rw-r-...	511

- Then, drag the properly named public key into the local folder. This public key is called p0001.pub. In the local folder, open the “authorization” file by highlighting it and pressing the F4 key. Add the line “key p0001.pub” as shown below.



.ssh2 - xx@EDIACPT.wipo.int - WinSCP

Transfer Settings Default

xx@EDIACPT.wipo.int (.ssh2) | xx@ediacpt.wipo.int (/) | New Session

C:\XX Account Administrator\

Name	Size	Type	Changed
..		Parent dire...	27/04/2020 15...
authorization	1 KB	File	24/04/2020 12...
p0001.pub	1 KB	Microsoft ...	23/04/2020 12...
xx.pub	1 KB	Microsoft ...	23/04/2020 12...

/.ssh2/

Name	Size	Changed	Rights	Ow...
..		27/04/2020 14:...	rwxr-...	511
authorization	1 KB	24/04/2020 12:...	rwxr-...	511
xx.pub	1 KB	23/04/2020 12:...	rw-r-...	511

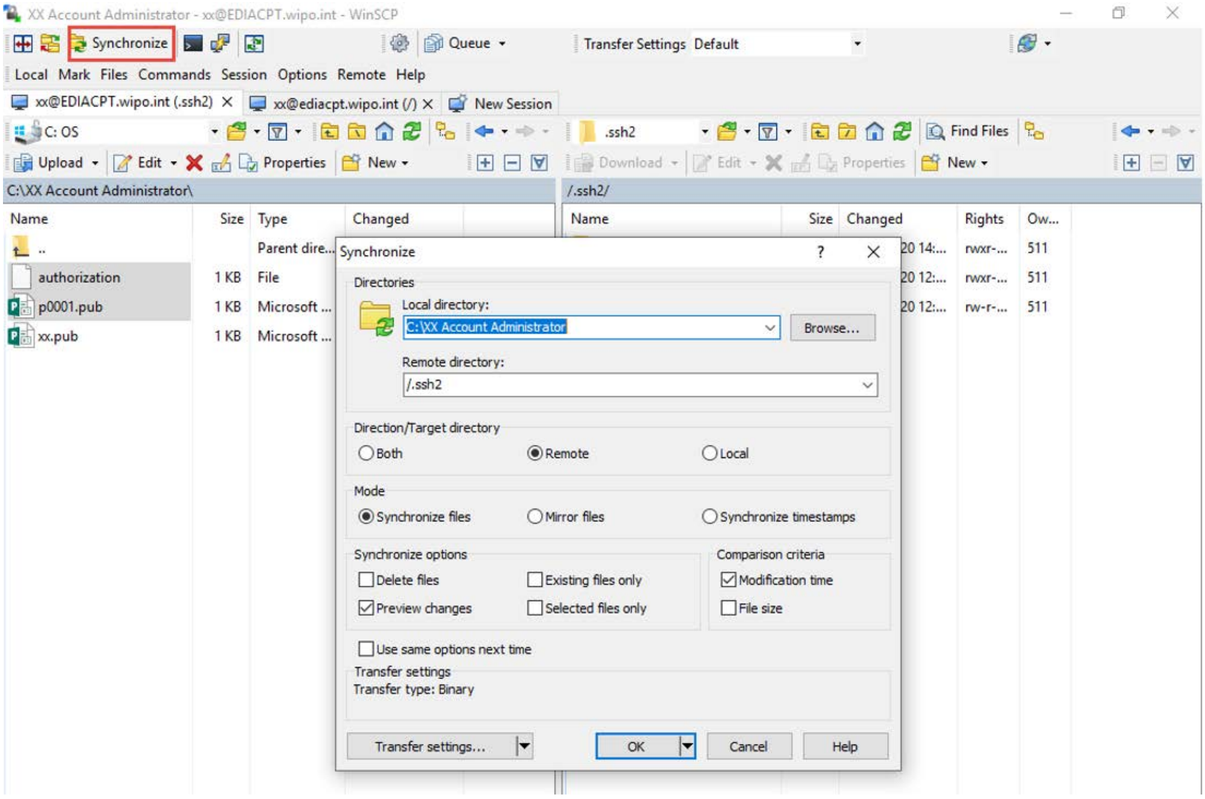
authorization - Notepad

```
key xx.pub
key p0001.pub
```

Synchronize
Queue
Local Mark Files Commands Session Options Remote Help
C: OS
.ssh2
Find Files
Upload
Edit
Properties
New
Download
File Edit Format View Help

- Save the “authorization” file in the local folder. Your local folder now contains two public keys “xx.pub” and “p0001.pub”. Keep track of which person or machine is assigned to which public key!

- When all new keys have been added, click on the “Synchronize” button. Then click on “OK”.


XX Account Administrator - xx@EDIACPT.wipo.int - WinSCP
Synchronize
Queue
Transfer Settings Default
Local Mark Files Commands Session Options Remote Help
xx@EDIACPT.wipo.int (.ssh2)
xx@ediacpt.wipo.int (/)
New Session
C: OS
Upload
Edit
Properties
New
.ssh2
Find Files
Download

C:\XX Account Administrator\

Name	Size	Type	Changed
..		Parent dire...	
authorization	1 KB	File	
p0001.pub	1 KB	Microsoft ...	
xx.pub	1 KB	Microsoft ...	

/.ssh2/

Name	Size	Changed	Rights	Ow...
		20 14:...	rwxr-...	511
		20 12:...	rwxr-...	511
		20 12:...	rw-r-...	511

Synchronize

Directories

Local directory: C:\XX Account Administrator

Remote directory: /.ssh2

Direction/Target directory

○ Both ● Remote ○ Local

Mode

● Synchronize files ○ Mirror files ○ Synchronize timestamps

Synchronize options

□ Delete files □ Existing files only

☑ Preview changes □ Selected files only

Comparison criteria

☑ Modification time

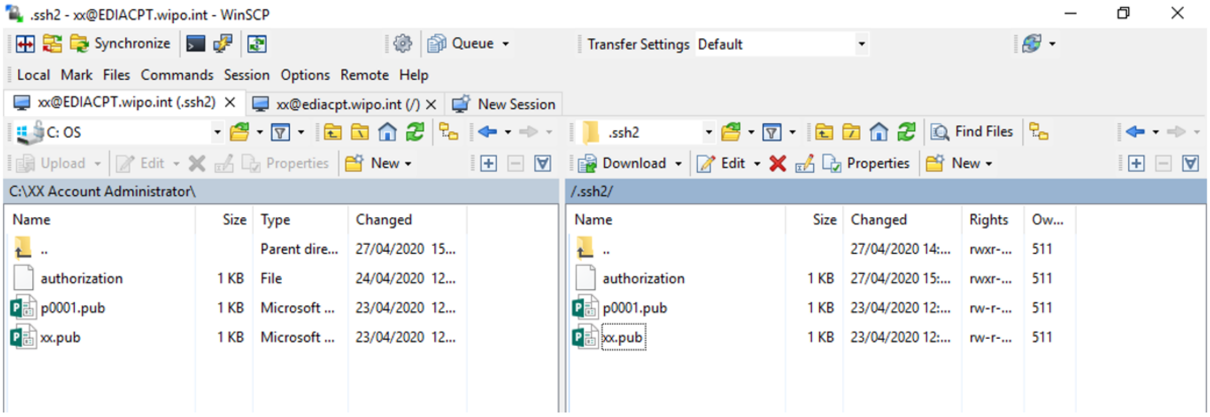
□ File size

□ Use same options next time

Transfer settings

Transfer type: Binary

The .ssh2 directory will be updated with the new p0001.pub key and the edited “authorization” file.


.ssh2 - xx@EDIACPT.wipo.int - WinSCP
Synchronize
Queue
Transfer Settings Default
Local Mark Files Commands Session Options Remote Help
xx@EDIACPT.wipo.int (.ssh2)
xx@ediacpt.wipo.int (/)
New Session
C: OS
Upload
Edit
Properties
New
.ssh2
Find Files
Download

C:\XX Account Administrator\

Name	Size	Type	Changed
..		Parent dire...	27/04/2020 15...
authorization	1 KB	File	24/04/2020 12...
p0001.pub	1 KB	Microsoft ...	23/04/2020 12...
xx.pub	1 KB	Microsoft ...	23/04/2020 12...

/.ssh2/

Name	Size	Changed	Rights	Ow...
..		27/04/2020 14:...	rwxr-...	511
authorization	1 KB	27/04/2020 15:...	rwxr-...	511
p0001.pub	1 KB	23/04/2020 12:...	rw-r-...	511
xx.pub	1 KB	23/04/2020 12:...	rw-r-...	511

- The account administrator has now installed the new user public key and authorized the system to use the key. A duplicate of the entire .ssh2 directory is stored on the local machine.

- To remove a user, simply remove the public key and remove the appropriate “key” line from the “authorization” file in the local directory, and update the .ssh2 directory.

C. DEVELPING CUSTOMIZED APPLICATIONS

If your Office prefers to develop customized software for accessing the PCT-EDI service, this is easily accomplished. Presented is an example of a simple demonstration application written in Perl[7]. Perl is a multiplatform development environment ideally suited for the development of portable code.

Perl itself and all of the modules referenced in this section are freely available. To use this example, you must have an up-to-date version of Perl with the Net::SFTP and Net::SSH-Perl modules referred to in an earlier section installed.

With these modules installed, try the following example. You must first have set up a correct OpenSSH user installation as described above, with working public key authentication. As with the other examples, replace “xx” with your account name.

This demonstration is based upon sample code provided with the Net::SFTP module, and will simply print a directory listing of the user account, and then exit. As outlined in the Net::SFTP documentation (see the footnote above for the URL), the functions to Get a file from the server and Put a file on the server are almost identical to the Ls request below.

D. EXAMPLE OF PERL SOURCE CODE

This example demonstrates the basic concepts of using the Net::SFTP Perl module with the WIPO PCT-EDI server, makes a connection and returns a directory listing. It requires a properly configured OpenSSH public key authentication environment for user "xx", where xx is the account name.

You must be using public key encryption for this example to work. This is the first of many examples

```
#!/usr/bin/perl –w
##########################################
# demo1.pl - demonstrate the basic concepts of using the
# Net::SFTP Perl module with the WIPO PCT-EDI server.
# Makes a connection and returns a directory listing
# Requires a properly configured OpenSSH public key authentication
# environment for user "xx", where xx is the account name.
#
# Arguments:    demo1.pl -v -u=username, both are optional.
# If -u username is not provided the current username for the account is used
# ./demo1.pl pctedi.wipo.int -u xx #
###########################################
use strict;
```

[7] http://www.perl.org

```
use Net::SFTP;
use Getopt::Long;

my %opts;
my $user;
Getopt::Long::Configure('no_ignore_case');
GetOptions(\%opts,"v", 'u=s'=>\$user);

my($host) = @ARGV;
die "usage: demo1 [options] hostname" unless $host;

# set up the arguments based on the command line options
my %args = (ssh_args => []);
$args{debug} = 1 if $opts{v};
push @{ $args{ssh_args} }, user => $user ;

# make our connection
print "Connecting to $host...\n";
my $sftp = Net::SFTP->new($host, %args);
# get a listing of the base directory
$sftp->ls("." , sub { print $_[0]->{longname}, "\n" });
print "Finished\n";
```

[END OF DCOUMENT]