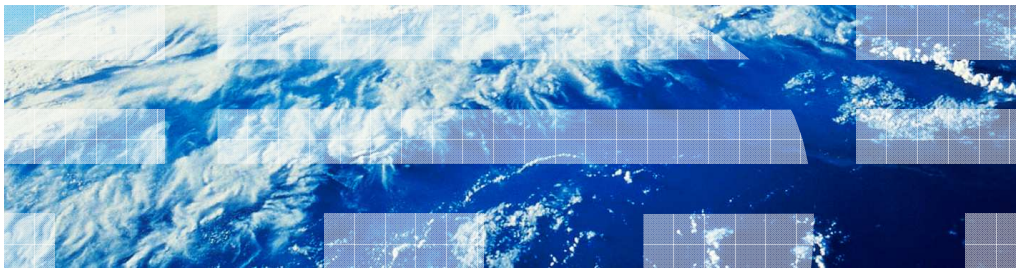
# WebSphere Partner Gateway V6.2.0:

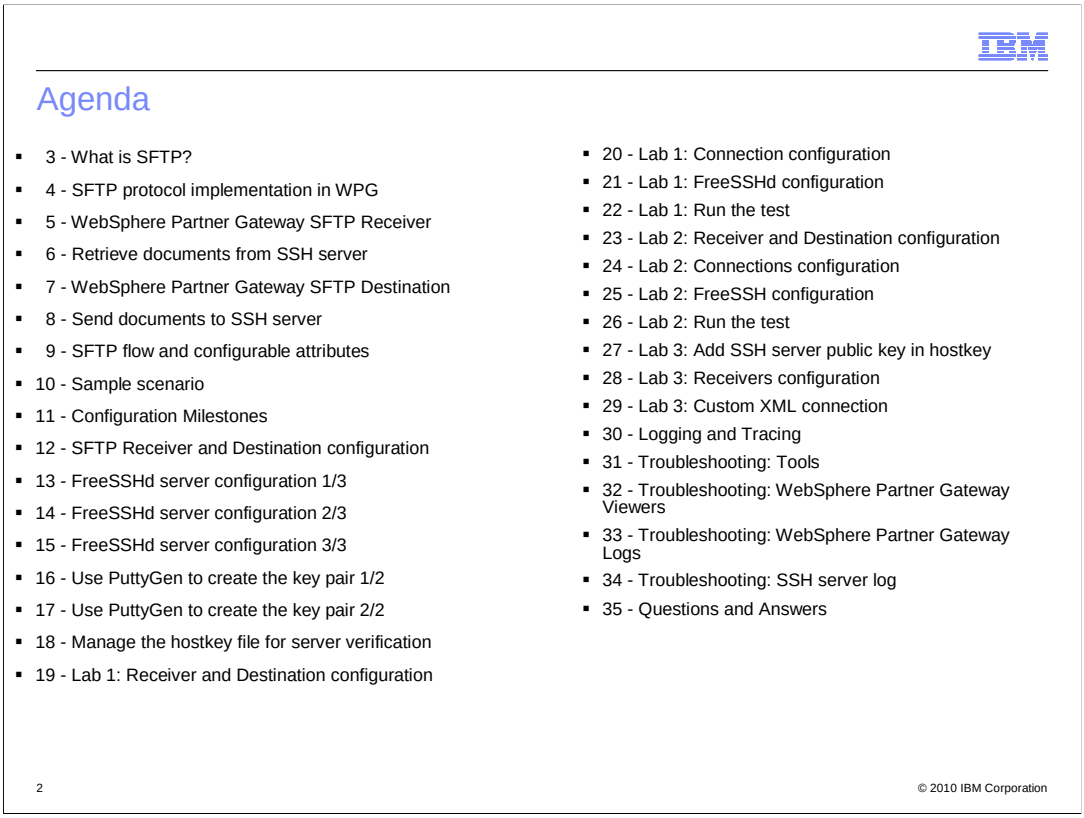
## Configuring to use the SFTP protocol



WebSphere. software



The SFTP protocol is one of the new features released with WebSphere Partner Gateway 6.2.0. This presentation provides details on how to configure WebShpere Partner Gateway to use the SFTP protocol.




This is the agenda, which helps summarizing the presentation contents as follows:

Slide 3 starts with some general information about SFTP

Slides 4-9 show how this protocol has been implemented in WebSphere Partner Gateway .

Slides 10-29 describe three sample scenarios of SFTP transaction flows using:

1 - private key authentication

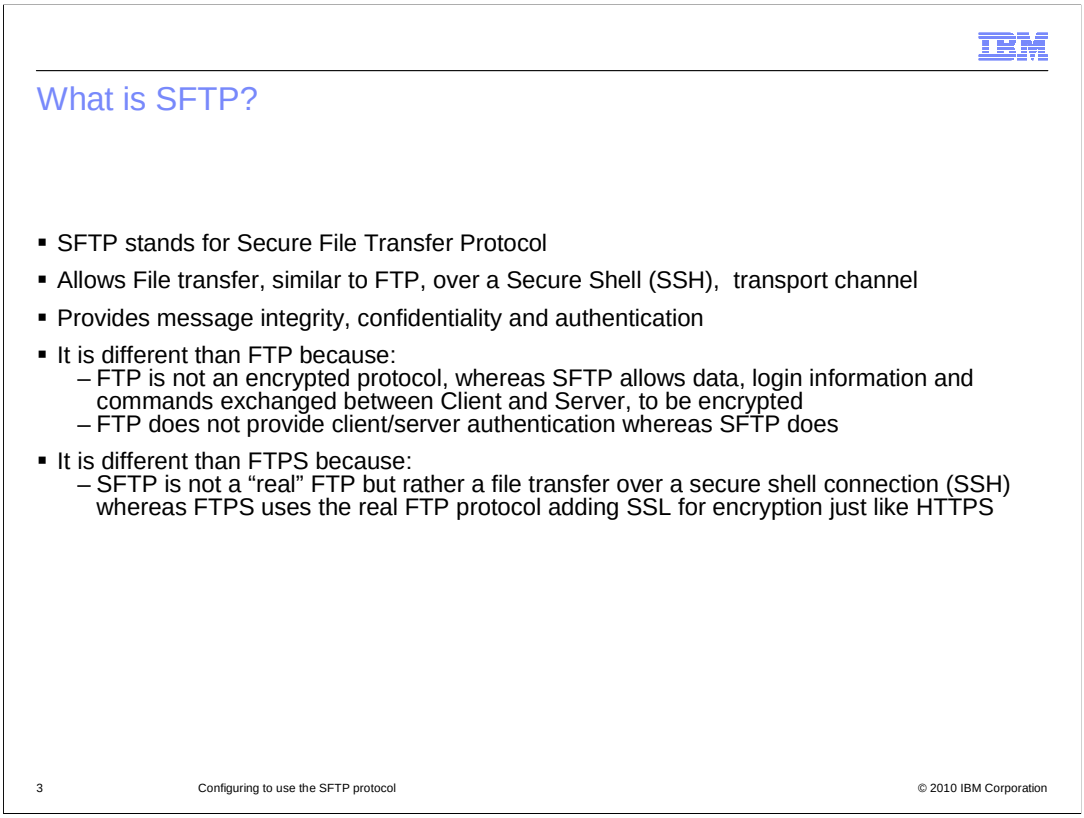
2 - user/password authentication

3 - server authentication

Configuration details for both the SSH server and the WebSphere Partner Gateway SFTP Receivers/Destinations are provided as well.

Slide 30 is about the logging and tracing for an SFTP flow

Slides 31-34 Provides troubleshooting tips and techniques



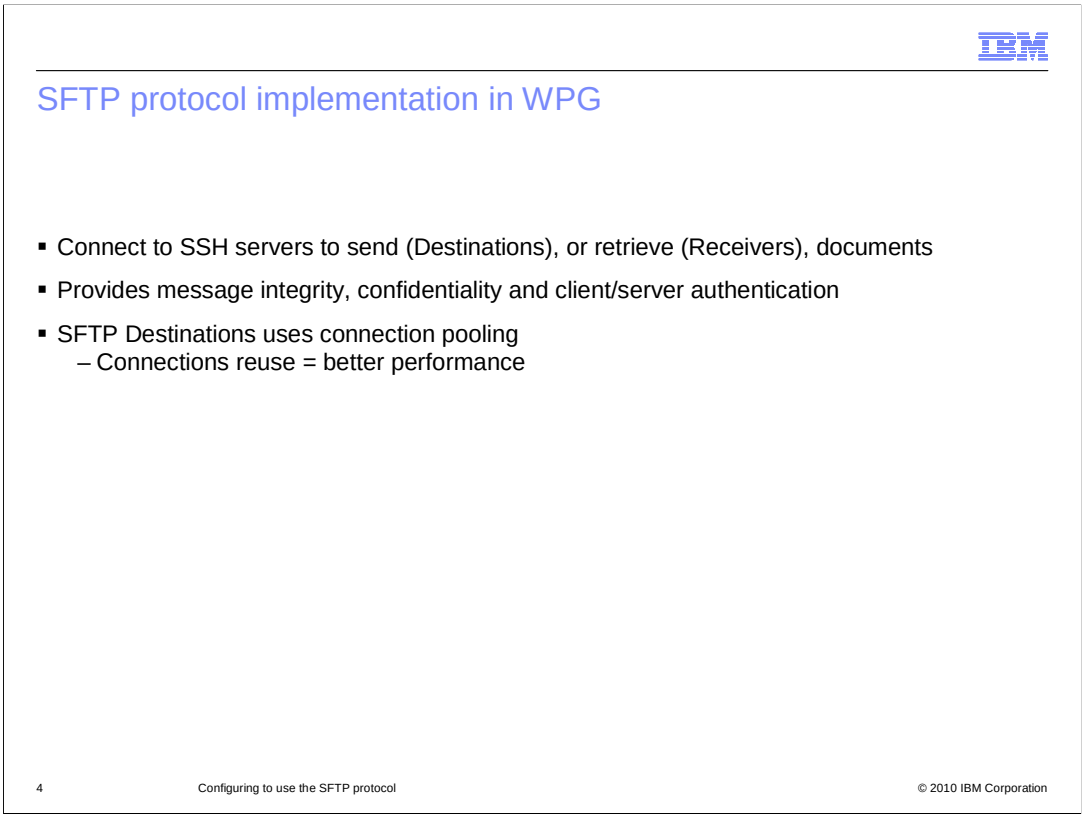

Here is a little background about SFTP itself.

The acronym stands for Secure File Transfer Protocol.

It allows file transfer, similar to FTP, but over a secure shell transport channel called SSH, which allows for the confidentiality, authentication and integrity of the message.

Although the name contains the "FTP" word, SFTP is not really FTP. For one, FTP is not an encrypted protocol or allows for client/server authentication, things that SFTP does. In fact, the login information, data and the commands exchanged between server and client are encrypted. Client authentication and server verification are also SFTP features that cannot be performed using regular FTP.

FTPS is also different than SFTP. because it uses the real FTP protocol with the addition of SSL for encryption, very much like HTTPS. SFTP does not use the "real" FTP protocol, it uses an SSH channel.

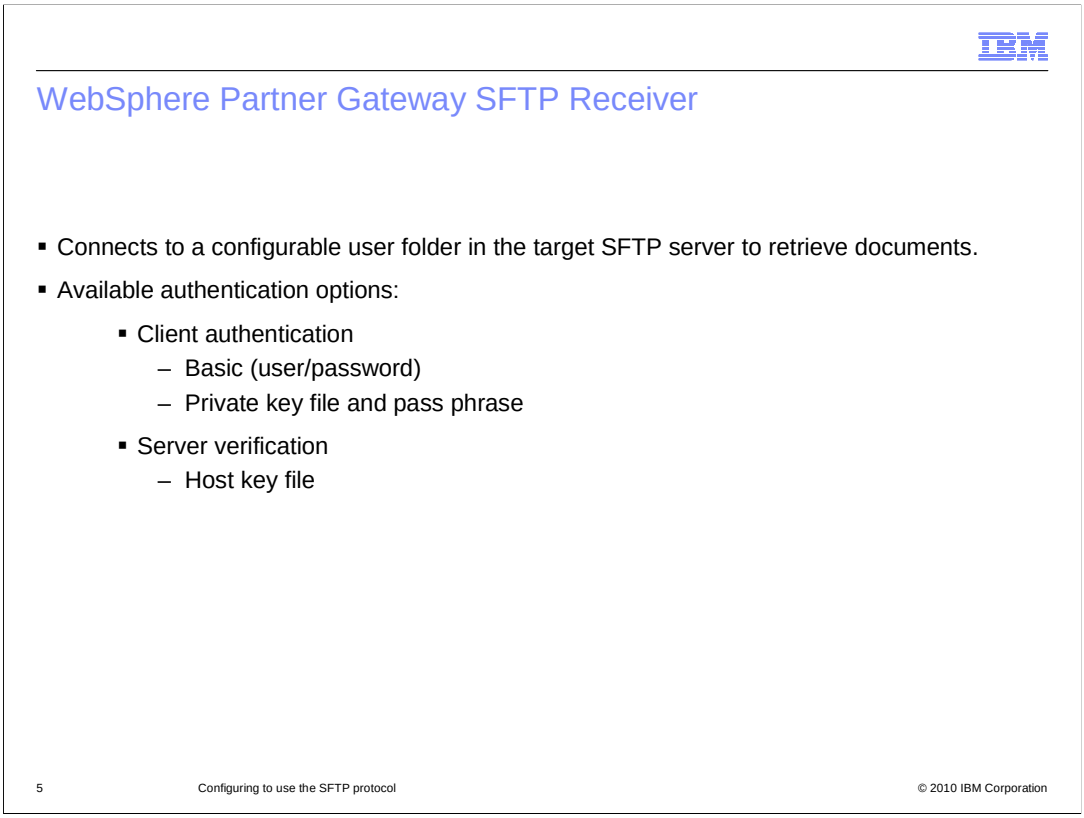



This slide enters the subject of how the SFTP protocol has been implemented in WebSphere Partner Gateway .

There are two ways you can use SFTP in WebSphere Partner Gateway

Inbound, to retrieve files from the server, using a SFTP Receiver

Outbound, to send files to the server using a SFTP Destination (which makes use of “connection pooling”, to enhance performance).




Here are the specifics about how the SFTP Receiver works.

You can choose to authenticate the client either using "user and password" or "Private key and pass phrase".

Optionally you can enable server verification which requires the setup of the so called "host key" file, which stores the server key.

Once the connection is established, the WebSphere Partner Gateway Receiver looks in the SSH server user folder, as specified in the configuration, to pick up documents and submit them in the transaction flow.

## Retrieving documents from the SSH server

- 1 - WebSphere Partner Gateway Receiver connects to the SSH server
- 2 - Client authentication (basic or private key)
- 3 - Optionally, server verification takes place
- 4 - WebSphere Partner Gateway receiver polls server remote event folder
- 5 - File retrieved and placed in local event folder
- 6 - File placed in archive folder to be processed




This slide lays down the logical steps of the WebSphere Partner Gateway Receiver actions:

First it connects to the SSH server

Next the Client authentication happens and, optionally, server verification

Then it Polls the configured folder in the server machine (also called "Remote Event" folder)

And If present, the files are retrieved and placed in the "Local Event" folder, in the WebSphere Partner Gateway Receiver machine

From there the files are then placed in the "Archive" folder to be processed

## WebSphere Partner Gateway SFTP Destination

- Connects to a configurable user folder in the target SFTP server to deliver documents.
- Available authentication options:
  - Client authentication
    - Basic (user/password)
    - Private key file and pass phrase
  - Server verification (APAR JR31639)
    - Host key file




More or less the same items mentioned for the Receiver are also valid for the WebSphere Partner Gateway Destination.

Notice that you need to install APAR JR31639 to enable the "server verification" option for the destination as well. That APAR also enables the "autoqueue" functionality for the SFTP destination. Neither of these options were available in the GA version.

Client authentication offers the same "user/password" or "private key and pass phrase" options as the Receivers, and when the authentication process is completed, the Destination drops the output document in the chosen folder of the server.

## Sending documents to the SSH server

1 - WebSphere Partner Gateway destination connects to the SSH server

2 - Client authentication (basic or private key)

3 - Optionally, server verification takes place

4 - File is dropped into the server user folder

5 - WebSphere Partner Gateway destination disconnects from the server




This slide reports the logical steps of the WebSphere Partner Gateway Destination actions, when sending documents to the server:

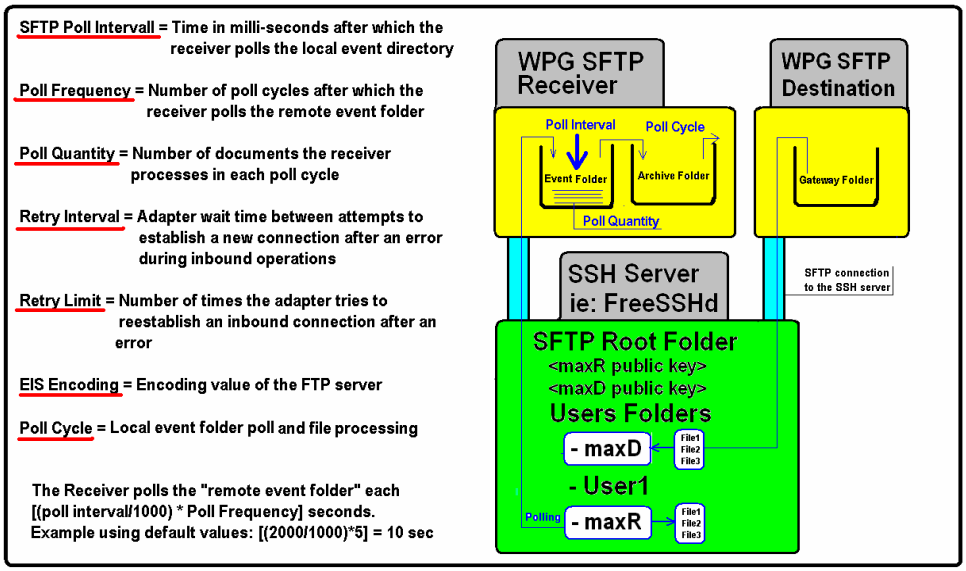
It first connects to the SSH server

Then client authentication occurs and, optionally, server verification

Then the outbound files are dropped in the configured user folder of the server machine.

And finally the Destination disconnects from the server.

## SFTP flow and configurable attributes







This is a picture showing the flow and the relevant attributes you can configure.

Looking at the flow first, in yellow are reported the WebSphere Partner Gateway Receiver and Destination elements. In green is the server.

The FreeSSHd server has been used for the tests reported in this presentation.

The server picture shows the WebSphere Partner Gateway user folders "maxR" and" maxD", where "R" and "D" indicates Receiver and Destination.

The user's public keys are also placed in the server which will be used to authenticate the client.

The picture shows what happens during the flow:

The WebSphere Partner Gateway Receiver connects to the server, goes through client authentication and then polls the configured folder "maxR".

If there are files to process, then these are retrieved and transmitted through the SFTP channel to the WebSphere Partner Gateway Receiver and placed in the "Event" folder.

From the “Event” folder the files are then moved to the "Archive" folder, to follow the normal processing.

Now that you have visualized the flow, the attributes on the left side of the slide, might make more sense. Take a quick look:

Poll Interval: Is the time after which the receiver polls the "local event folder“

Poll Frequency: Is the number of "poll cycles" after which the receiver polls the "remote event folder“

Poll Quantity: Defines the number of files the receiver retrieves from the remote event folder, each cycle

Retry Interval/Retry Limit: Define the wait between connection attempts and the total number of attempts

Poll Cycle: Is the local event folder poll and file processing

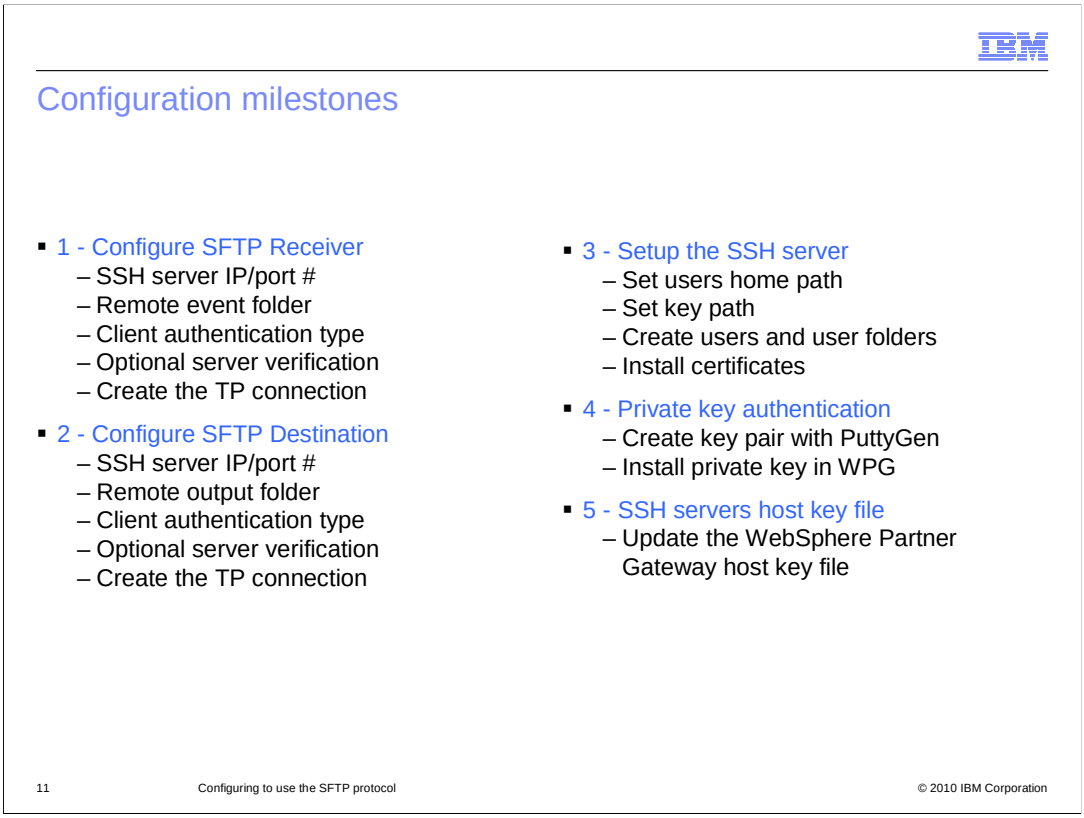

After the lecture talk, take a look at some practical applications of the concepts you went through the previous slides.

The three lab scenarios that are presented in the next charts are comprehensive of all items discussed so far.

The first two labs show a flow where the WebSphere Partner Gateway SFTP Receiver picks a file from a folder in the FreeSSHd server and uses a None,EDI-X12,ISA connection to pass it to a SFTP Destination which drops the file in a different folder of the same server.

The only difference between Lab1 and Lab2 is that you use "private key and passphrase" authentication in the first and "user/password" authentication in the second.

The third lab, uses a similar flow (with perhaps the option of a custom XML connection instead of EDI), but also enables the "server authentication" feature.




In this slide the configuration milestones needed to accomplish the labs mentioned earlier.

There are five main tasks to tackle, and each of them have sub-tasks, that are :

1 - Configure the SFTP Receiver, which includes these configuration sub-tasks:

1a - SSH server IP/port #
1b - Remote event folder
1c - Client authentication type
1d - Optional server verification
1e - Create the connection with the trading partner

2 - Configure the SFTP Destination, which includes the same sub-steps as the Receiver configuration

2a - SSH server IP/port #
2b - Remote output folder
2c - Client authentication type
2d - Optional server verification
2e - Create the connection with the trading partner

3 - Setup the SSH server, which includes the following configuration sub-tasks:

3a - Set users home path
3b - Set key path
3c - Create users and user folders
3d - Install certificates

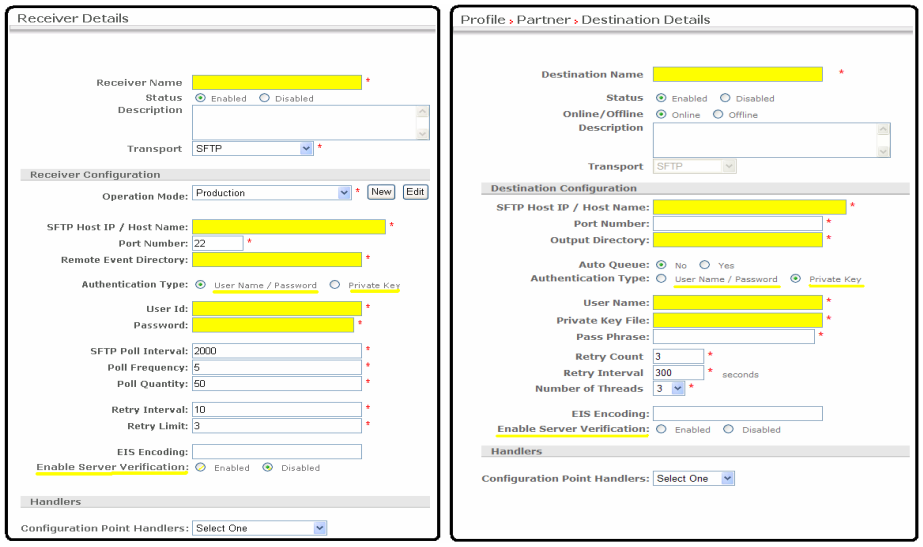
4 - Create and install the private and public key, when using this kind of client authentication, which includes the following configuration sub-tasks: :

4a - Create key pair with PuttyGen
4b - Install private key in WebSphere Partner Gateway

5 - Retrieve the server host key and update the host key file in WebSphere Partner Gateway, which includes the following configuration sub-task:

5a - Update the WebSphere Partner Gateway host key file

## SFTP Receiver and Destination configuration







Here are the two WebSphere Partner Gateway items to configure to be able to run SFTP transactions: The receiver and destination.

In yellow the fields are highlighted where you need to enter the configuration information. For example:

SSH server IP or host name

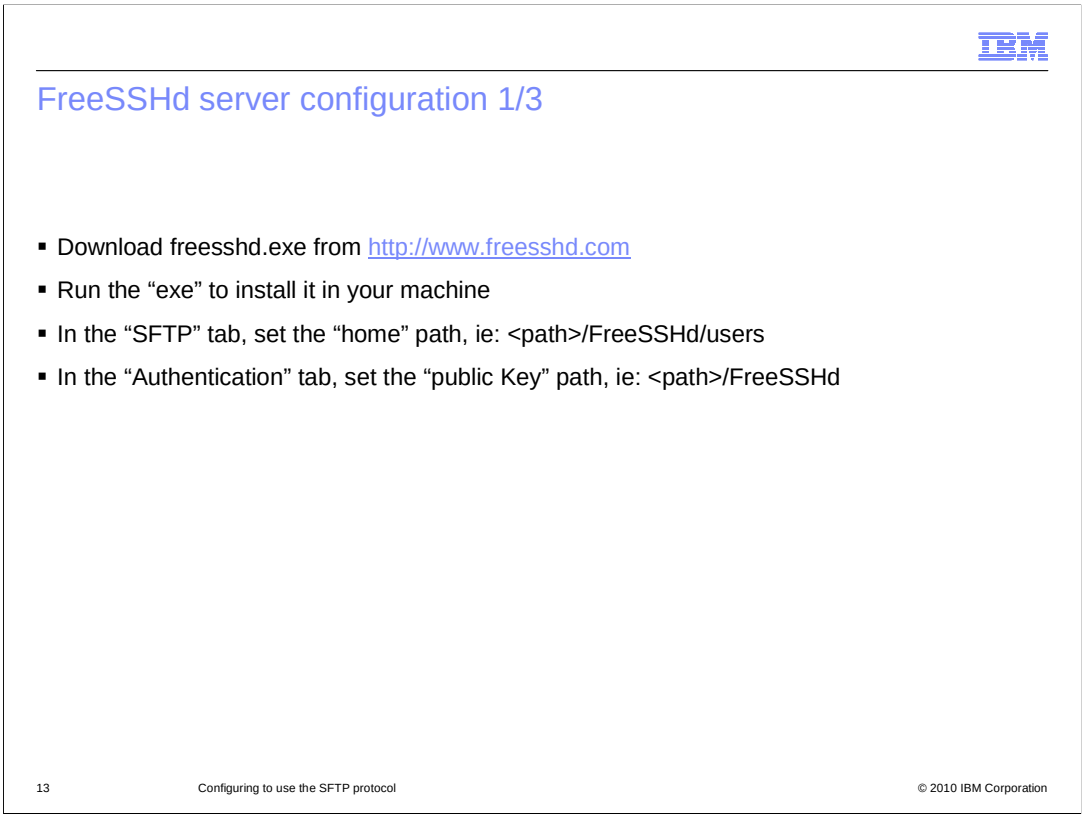
The port number

For the Receiver: The folder where the files to retrieve are. Which is indicated as "Remote Event Directory“

For the Destination: The folder where the files need to be delivered

The client authentication type: "private key" or "user/password", with the associated information concerning the user ID and password or, the private key location and pass-phrase if you choose to use the "private key“ form of authentication.

Server Verification: This is an option that can be turned off (which is the default), or on, in which case we'd need to configure a host key, file whose format and composition is shown in a few slides.




Another item in the configuration milestones is the SSH server configuration.

The first thing you need to do is to download the software package from the URL reported in the chart.

Then you have to install it, which is very simple, just run the exe and take all the defaults.

Next, you need to do a minimum configuration, strictly necessary to run your test:

> Configure the "home" path in the "SFTP" tab and the "public key" path in the "Authentication" tab.
>
> These steps are necessary so that the server knows where the home folders for the users are and where to go find the public keys.

## FreeSSHd server configuration 2/3

- Create Receiver/Destination users as follows:
  - For “user/pwd” authentication choose: “password stored as SHA1 hash”
  - For “Private key” authentication choose: “Public key (SSH Only)”
  - Check the “SFTP” box
  - Create the user home folders under /FreeSSHd/users (folder name = user name)




Continuing with the server configuration, you have to first create a user for the receiver and one for the destination.

Then Select what kind of authentication you want in place: "user/password" or "public key".

Then Check the "SFTP" box for the protocol being used

And then Create the user home folders which has to be named the same as the user name.

## FreeSSHd server configuration 3/3

- Copy the client public key in the location specified in the “Authentication” tab
  - Note: The Public key file name must be the same as the user name
- **Note: See next two charts for instruction on how to create key pairs**




The last thing you need to do on the server side, is to copy the client public key on the server folder specified in the "Authentication" tab.

The file containing the key must also be named the same as the user name.

## Use PuttyGen to create the key pair 1/2

- Download the puttygen.exe from http://www.putty.org/
- Run the “exe” and click the “Generate” button
- Copy/paste the “public key” text into a notepad file having the same file name as the user name (no extension). Do not save it using the “Save public key” button
- Save the “private key” from the "Conversion > Export OpenSSH Key“ pull-down




To create the key pair you need to download "PuttyGen" from the putty.org site.

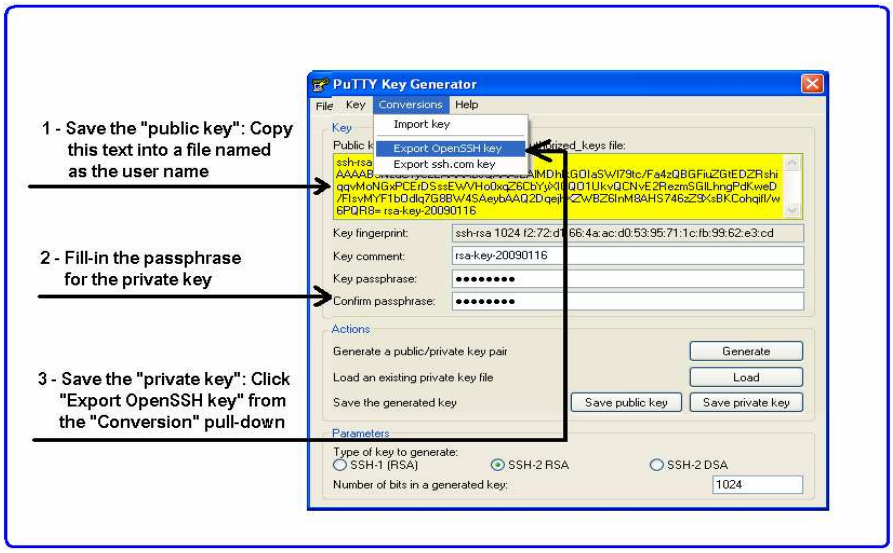
Running the executable file will prompt you a GUI panel and you can generate the keys clicking on the "Generate" button.

You are prompted to move your mouse around to generate the randomness that will be used to create the keys.

When this process is over you can copy the keys in your hard-drive.

There are some important tips on how to save the keys in the next slide.

## Use PuttyGen to create the key pair 2/2





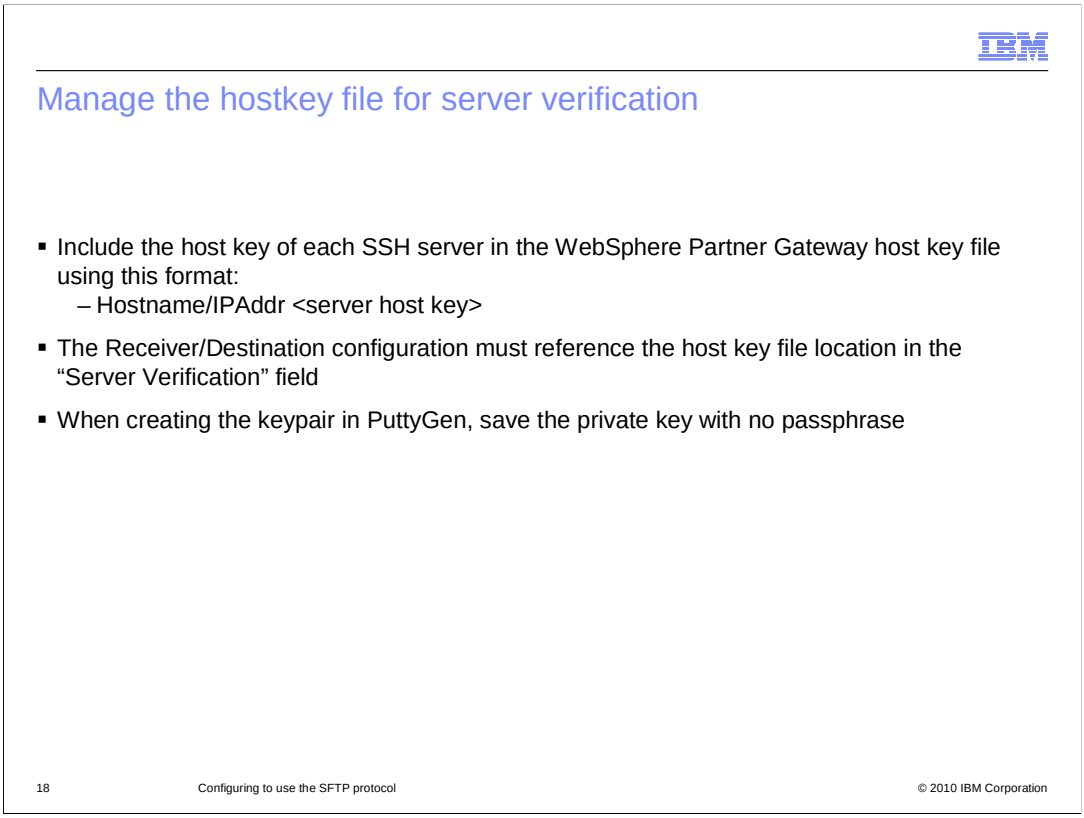


This is what the key generator panel looks like after the keys are created.

To save the public key, copy the text under the "Public key" heading (which is marked yellow in the slide), into a file which has to be named as the user name.

It's very important not to use the "Save public key" button in the panel because it will save the key in a different format than the needed OpenSSH format.

Next, you need to save the "private key", and again you do not want to use the "Save private key" button. What you need to do is to select the "Export OpenSSH key" off the "Conversions" pull-down, as shown in the chart.




This is the last of the five milestone configuration items and it’s dedicated to the handling of the server host key.

Now, each server has its own key which you need to copy in a file using a very simple format:

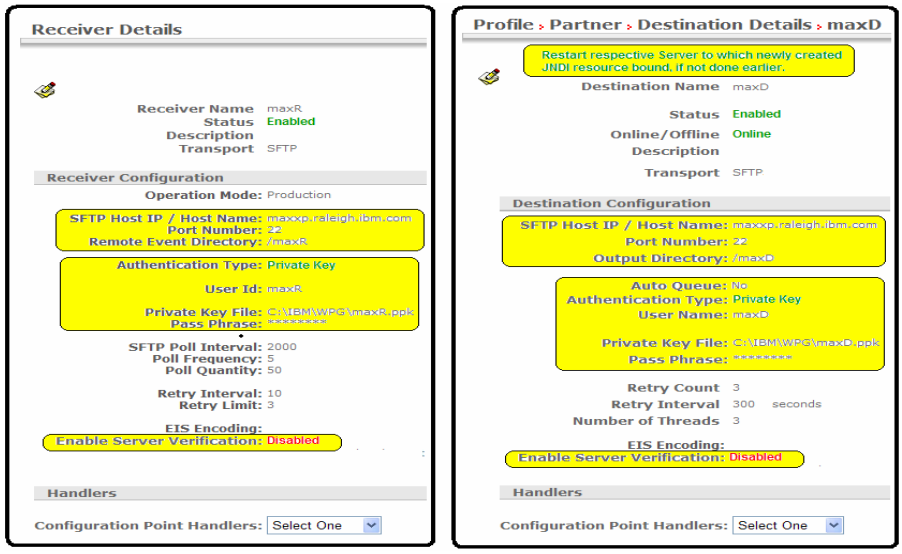
- <host name> <space> <server host key>

Then, you have to reference this file, using path and file name, in the Receiver or Destination configuration, when you enable "Server Verification".

One particular to remember is that when creating the FreeSSHd server key pair using PuttyGen, the private key must be saved without pass-phrase.

This concludes the configuration milestones. In the next 11 charts you will see the three labs mentioned earlier, in slide 10

## Lab 1: Receiver and destination configuration







The first lab is about configuring a transaction where a SFTP Receiver picks up a file from a folder in the FreeSSHd server, processes it and then handles it to the SFTP Destination to be dropped in another folder on the same server.

In this lab this setup is performed:

"private key" authentication for both Receiver and Destination.

The usernames are "maxR" for the Receiver and "maxD" for the Destination.

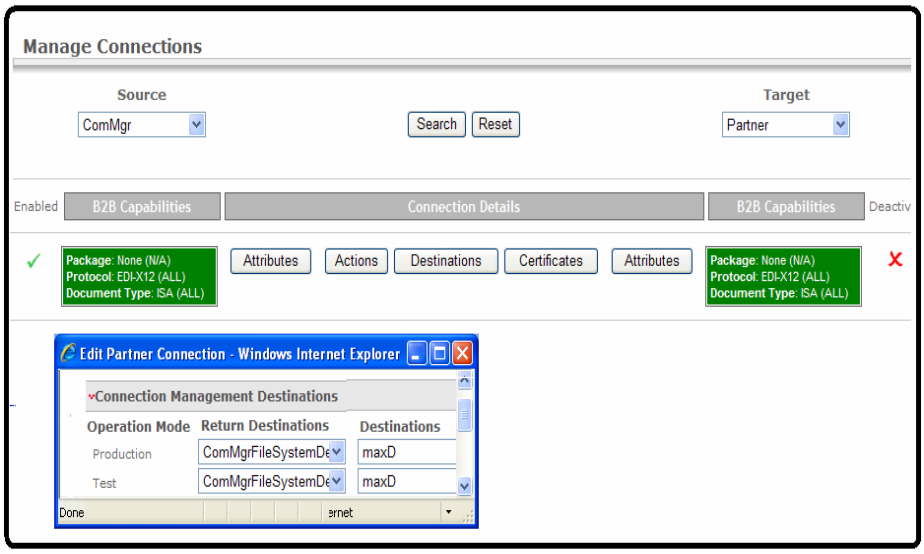
The location of the private key files is reported in the “Private Key File” field.

The "passphrase" value cannot be read, so make sure you remember it.

The "server verification" option is not being used in this lab

One more thing to notice is the message prompted in the Destination panel when you save the configuration. It warns you to restart the DocMgr cluster to activate the configuration change. Of course, if you are in simple or simple distributed mode then you need to restart server1 or the bcgserver cluster.

## Lab 1: Connection configuration

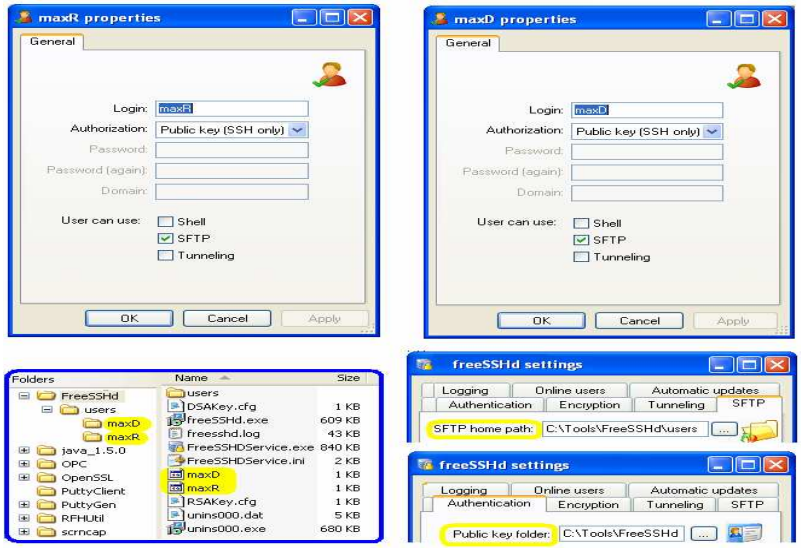






This slide shows the None,EDI-X12,ISA connection and the "maxD" SFTP Destination being used for this test

## Lab 1: FreeSSHd configuration







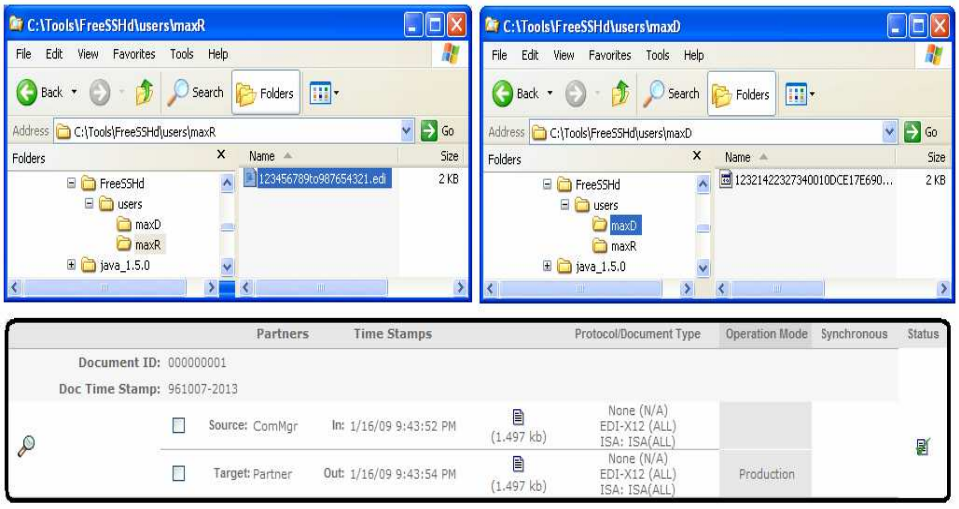
Here are the screen captures of the FreeSSHd configuration:

The users "maxR" and "maxD" are created and configured to use "public key" authorization and the SFTP protocol.

Two folders, named the same as the users, are created in the SFTP home path

The two public keys are placed in the "Public key folder" path

## Lab 1: Run the test



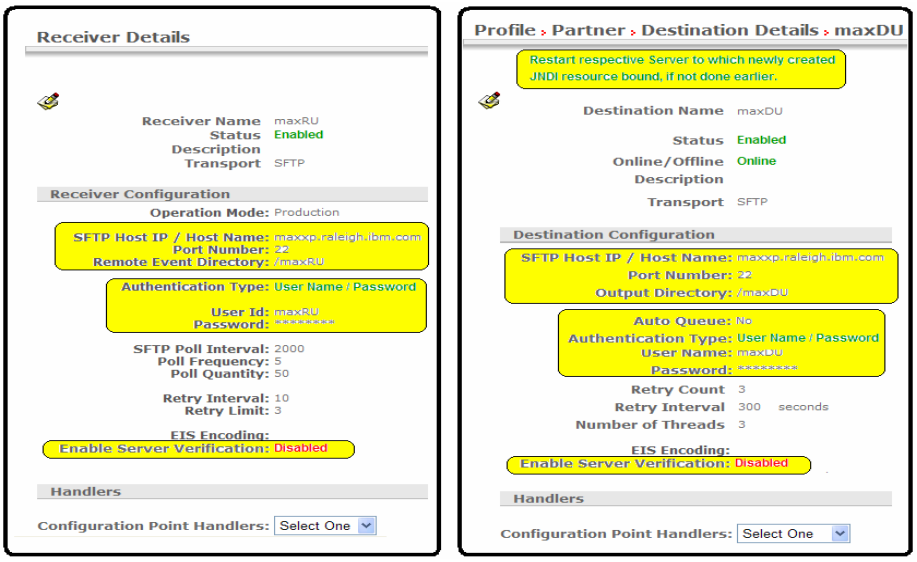




At this point, you are ready to run the test.

Place the EDI file in the "maxR" folder for the WebSphere Partner Gateway Receiver to pick up, and after being processed, the output file is dropped in the "maxD" folder as defined in the SFTP Destination configuration.

## Lab 2: Receiver and destination configuration



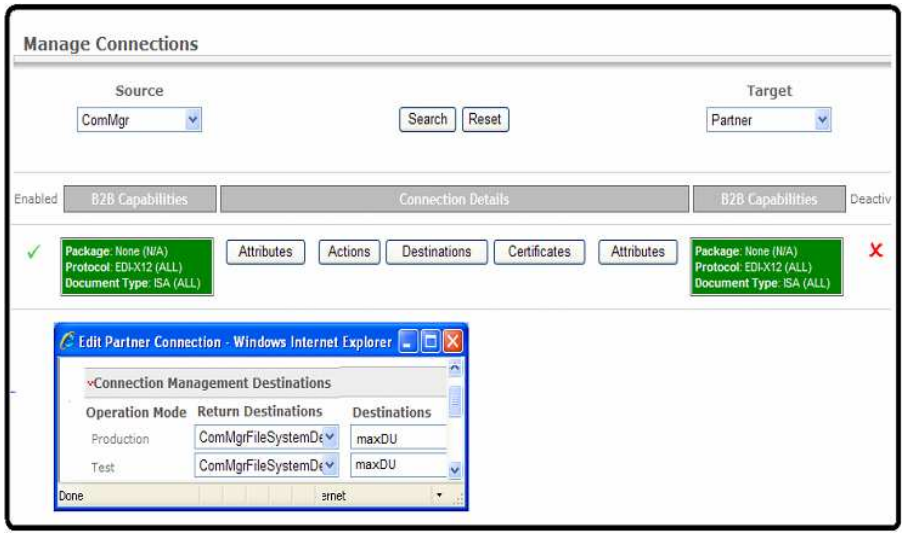




The second lab is similar to the first: The configuration flow is the same, but a new Receiver and a new Destination have been created to use "user/password" authentication (whereas "private key" authentication had been used in Lab1).

All the rest stays the same.

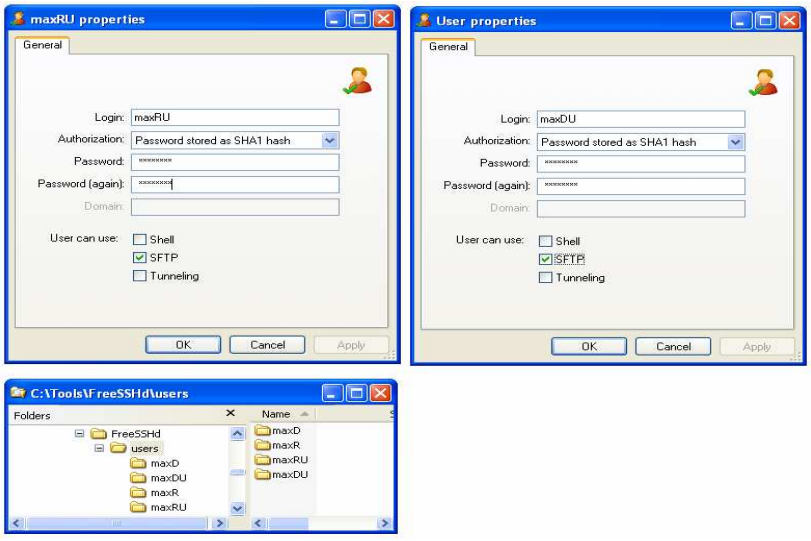
## Lab 2: Connection configuration




24 Configuring to use the SFTP protocol 

The connection also stays the same but the destination ID has to be changed to use "maxDU" which is the destination configured for "user/password" authentication.

## Lab 2: FreeSSHd configuration




25 Configuring to use the SFTP protocol 

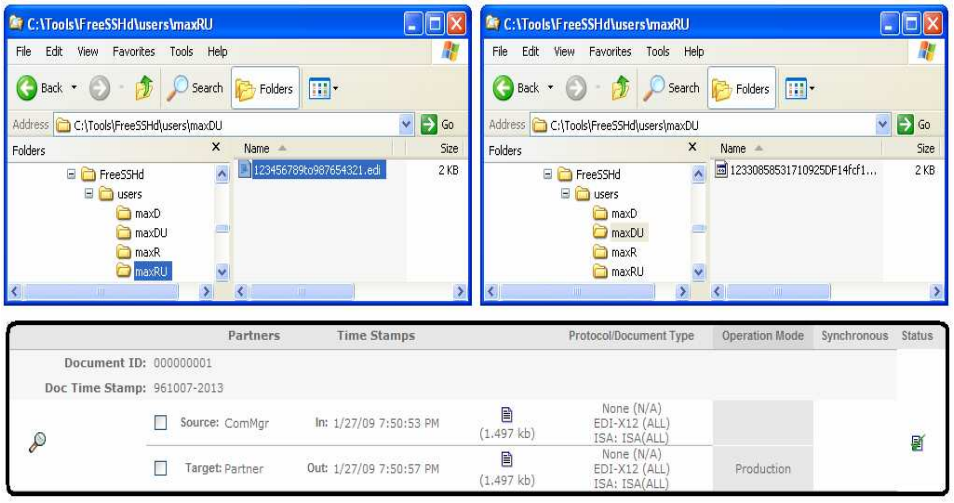
Now, this is the new FreeSSHd configuration that has to match the requirement to authenticate the client, using "user and password".

So, two new users "maxRU" and "maxDU" have been created, and this time you have to select the authorization to be "Password stored as SHA1 hash".

Then you have to enter the actual password and select to use the "SFTP" protocol

One last thing that needs to be done, is the creation of the two new users home folders as shown in the third screen capture.

## Lab 2: Run the test



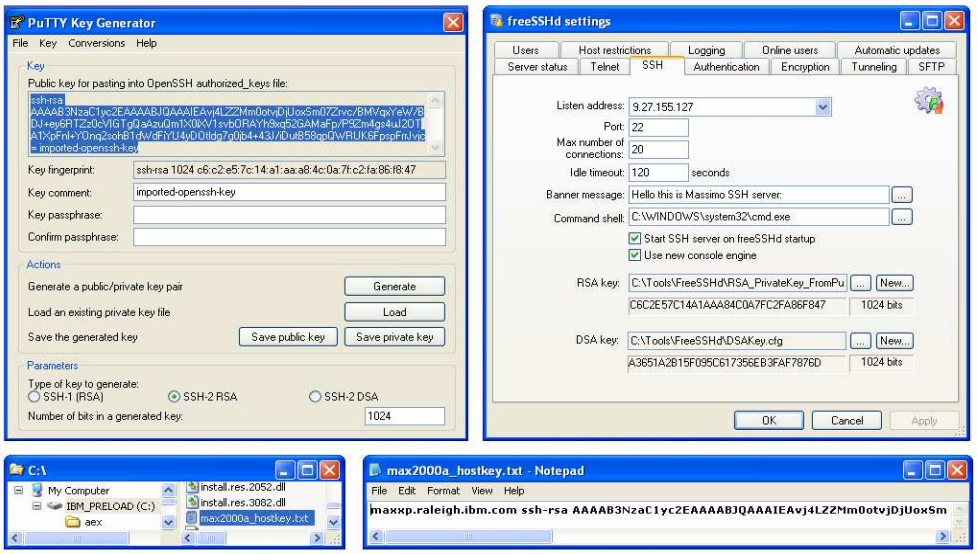




And finally, you can run the test just the same as you ran the one in the first example:

Drop the EDI file in the "maxRU" folder, which is picked up by the Receiver “maxRU”, processed and then delivered by the destination in the "maxDU" folder.

## Lab 3: Add SSH server public key in hostkey




27 Configuring to use the SFTP protocol 

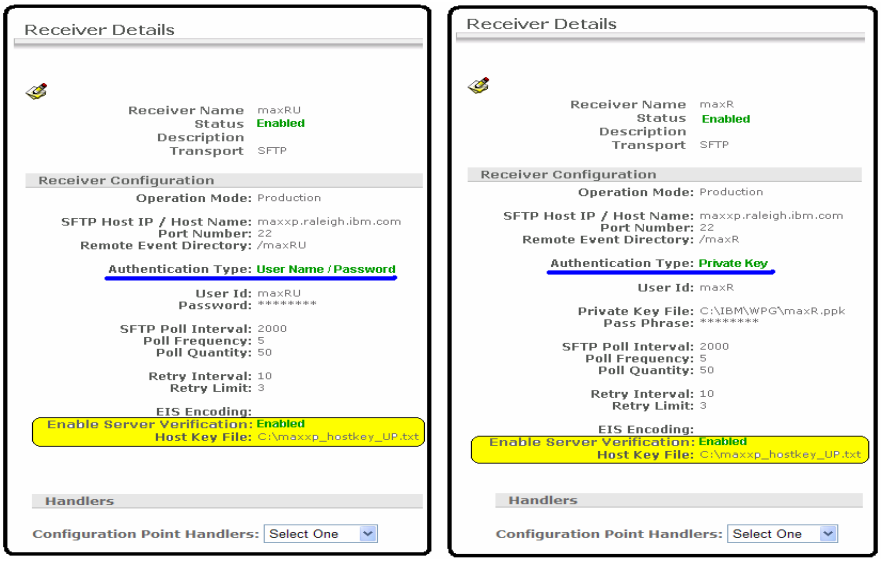
You are now to Lab 3, which adds "Server Verification" to the scenario.

In order to continue this test, you need to create a key pair for the server. This can be done with PuttyGen, the same way as you saw earlier when creating the key pair for the client. There is one difference though: the private key needs to be saved without the passphrase.

After saving the "private key" on disk, you can upload it in FreeSSHd as shown in this slide top-right screen capture.

The highlighted "public key" text however needs to be saved in the WebSphere Partner Gateway "host key", using the format shown in the slide bottom-right, that is: host name or IP Address, space and then the public key text.

## Lab 3: Receivers configuration





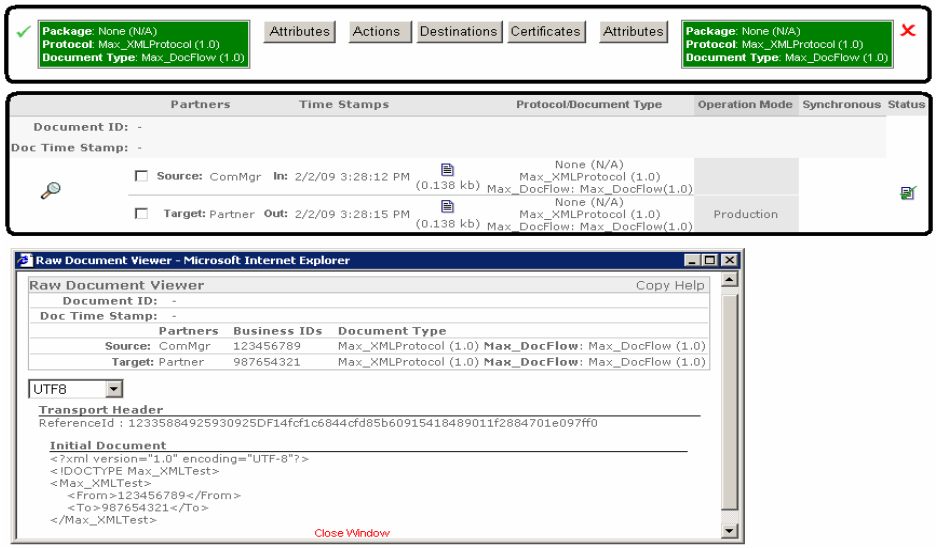


The Receiver configuration will have the "Server Verification" option enabled and, as consequence of that, you will have to enter the location of the host key file.

You can use the "Server Verification" option with either type of "Client Authentication": "user/password" as shown in the picture on the left or "Private Key", as shown in the picture on the right.

Of course the same concepts  apply if you want to use "Server Verification" for the Destination.

## Lab 3: Using custom XML protocol





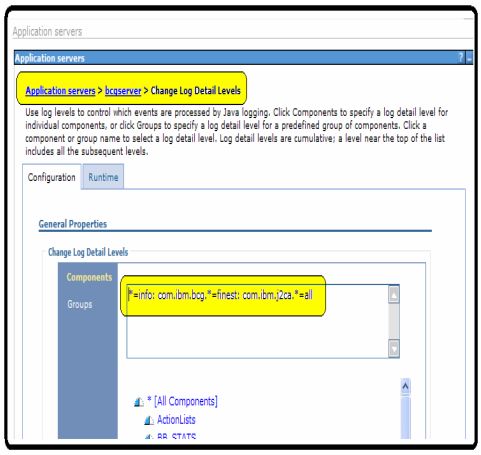


One more twist has been added in this lab, that is, we're using a custom XML instead of the EDI connection used in the previous labs.

The rest is just about the same: you drop the file in the server "maxRU" folder, the Receiver picks it up, and after processing it, is dropped by the Destination in the "maxD" folder.

## Logging and tracing

- Add this string to switch the adapter logging to “debug” level:
  - com.ibm.j2ca.*=all
- Affected application servers:
  - server1 (simple mode)
  - bcgserver (simple distributed)
  - BCGReceiver, BCGDocmgr (full distributed mode)







What about logging and tracing to debug SFTP problems?

You have a specific string:

"com.ibm.j2ca.*=all"

that needs to be used to switch the adapter logging to "debug" level.

This string can be added to the other logging levels already present. Just make sure to separate them using a colon character ":".

Of course depending on what mode is being used, the change applies to different application servers as indicated in the lower part of this slide under the “Affected application servers” heading.

## Troubleshooting: Tools

- WebSphere Partner Gateway console viewers
- WebSphere Partner Gateway component logs:
  - SystemOut.log
  - SystemErr.log
  - bcg_server.log
- SSH server log




What if something wrong happens? What troubleshooting tools can you use?

The debugging techniques still uses the same WebSphere Partner Gateway tools you are familiar with:

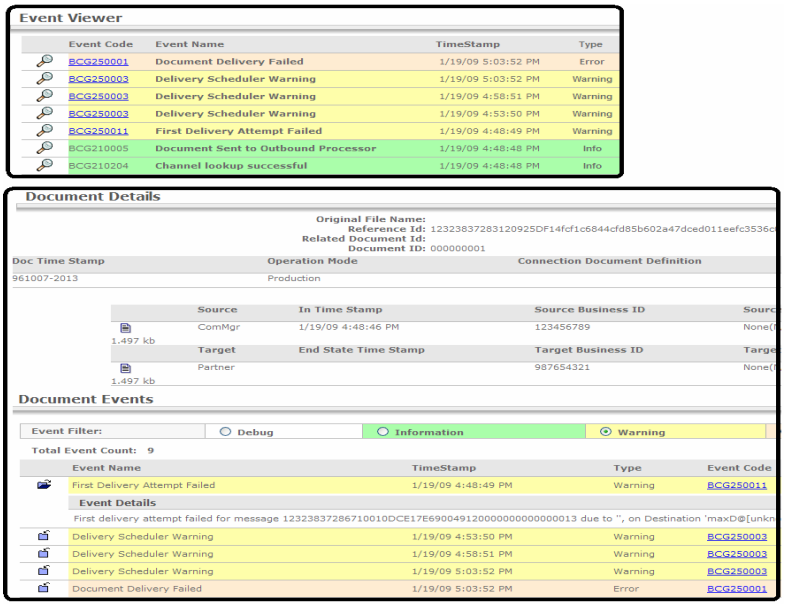
The console viewers and the component logs.

To these tools, you need to add:

The SSH server log, which can be very useful to understand some specific scenario.

Take a look at how these tools can be used.

## Troubleshooting: WebSphere Partner Gateway viewers







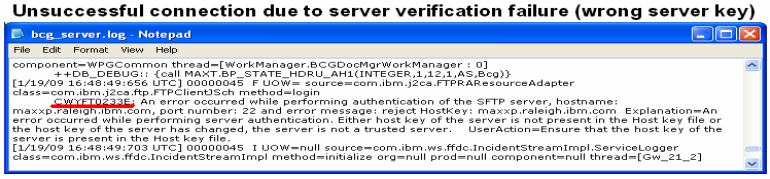
The "Event Viewer" and the "Document Details Viewer" are a good start to understand what went wrong.

In this particular case for example, you can see that the document was successfully retrieved by the Receiver, then was successfully processed and passed to the "Outbound Processor" to be delivered to the destination recipient. At this stage, the BCG250001 event error tells you that this last operation failed.

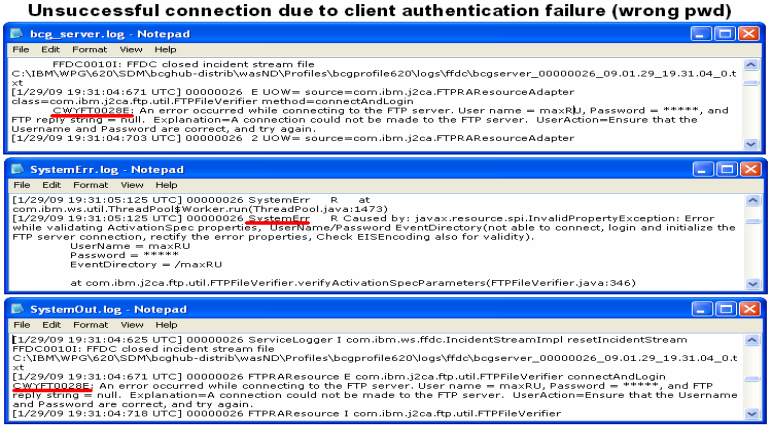
To find out more details on the specific reason for this failure, you need to debug the logs which will allow a more detailed analysis on the root cause of this error.

## Troubleshooting: WebSphere Partner Gateway logs

**Unsuccessful connection due to server verification failure (wrong server key)**




**Unsuccessful connection due to client authentication failure (wrong pwd)**



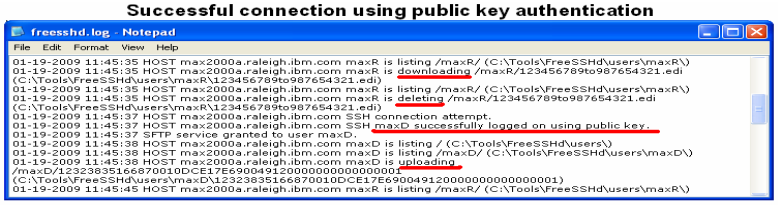
Without entering in the all the troubleshooting details, here are two error situations that can occur when the client authentication fails.

The first one refers to an error scenario caused by using a mismatched key pair, which is tracked in the log by the red mark, where the specific entry points to a problem with the keys being used.

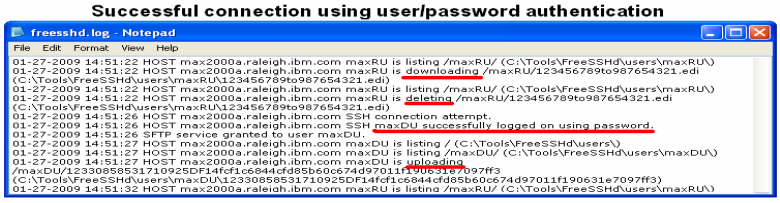
The second example shows the logs highlights for an unsuccessful connection due to a failure to authenticate the password for the user. In fact, the bcg_server, SystemError and SystemOut logs point to an error situation caused by incorrect credentials presented to the server.

## Troubleshooting: SSH server log

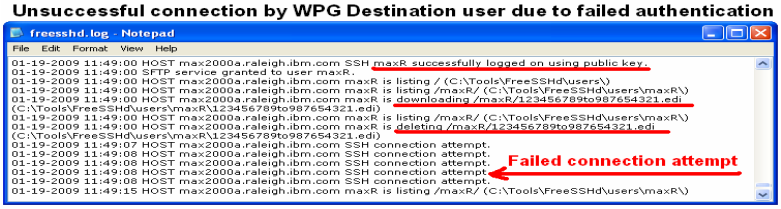
**Successful connection using public key authentication**




**Successful connection using user/password authentication**




**Unsuccessful connection by WPG Destination user due to failed authentication**



This is the last slide where some screen captures of the FreeSSHd server logs, can be very helpful to understand some error situations.

The pictures here show three different situations:

A successful connection using "public key" authentication:

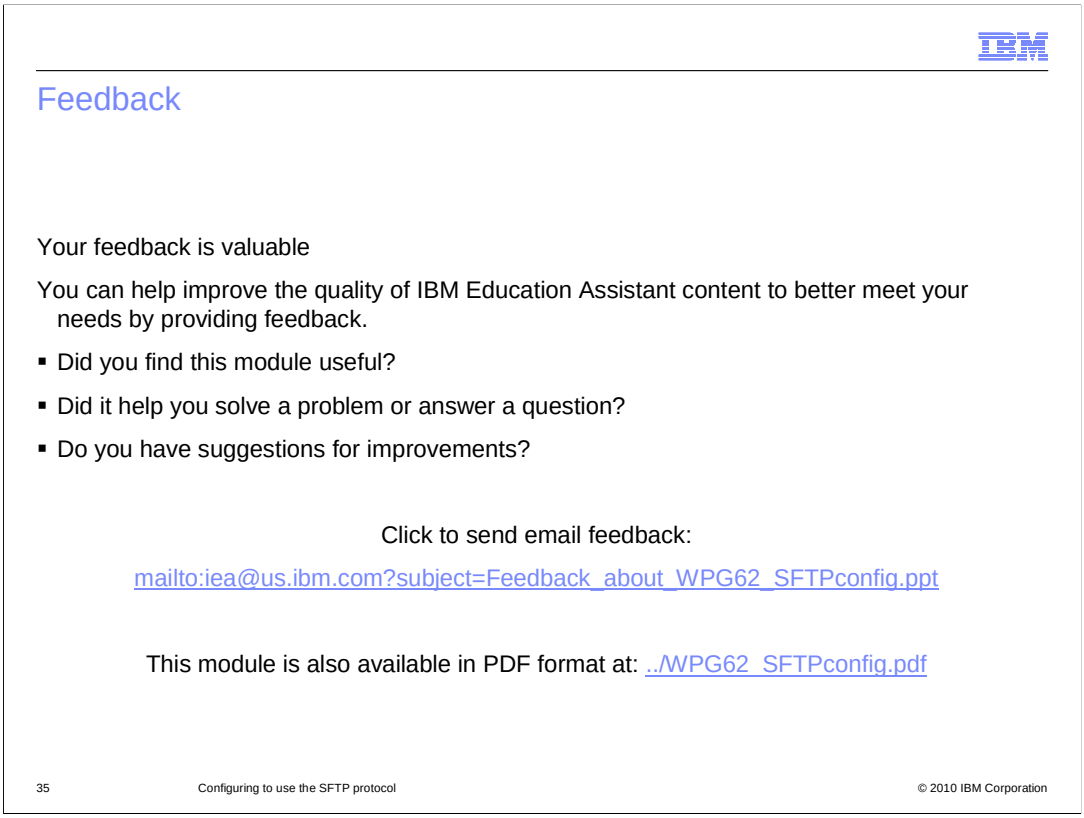
- you can see in there when the file is picked-up by the WebSphere Partner Gateway receiver (the line where the "downloading" word is underlined in red).
- The file is then removed after being picked-up (see the line where the "deleting" word is marked in red).
- Next, the Destination user "maxD" successfully connects using "public key" authorization.
- And then the last line marked indicates when the file is dropped in the "maxD" user folder by the WebSphere Partner Gateway Destination.

The second example shows a successful connection using "user/pasword" authentication:

- This is shown in the marked line saying "maxDU successfully logged on using password".
- The rest is pretty much the same sequence as you saw earlier.

The third picture shows an unsuccessful connection by the "maxD" WebSphere Partner Gateway Destination user which fails authentication:

- The first red mark shows a successful connection by the "maxR" WebSphere Partner Gateway Receiver user using "public key" authentication.
- The retrieval and deletion of the file from your folder
- And then follows the unsuccessful connection attempt by the WebSphere Partner Gateway Destination caused by the authentication failure.




You can help improve the quality of IBM Education Assistant content by providing feedback.

## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.