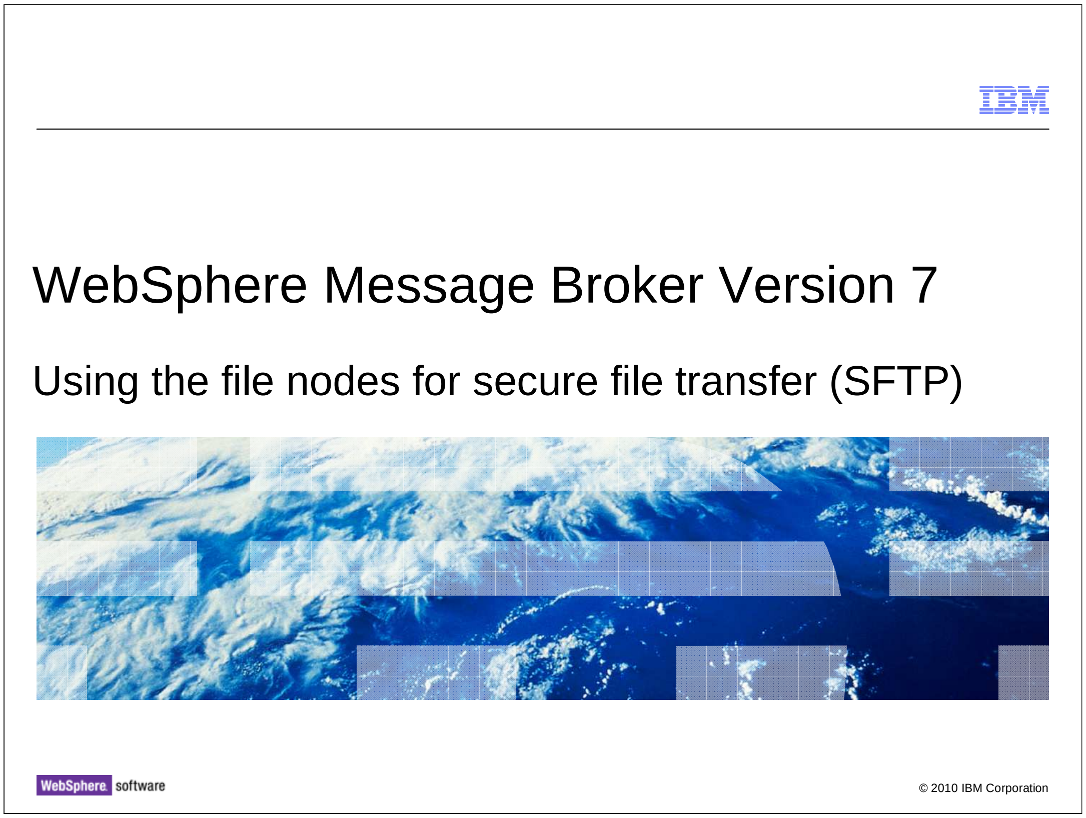
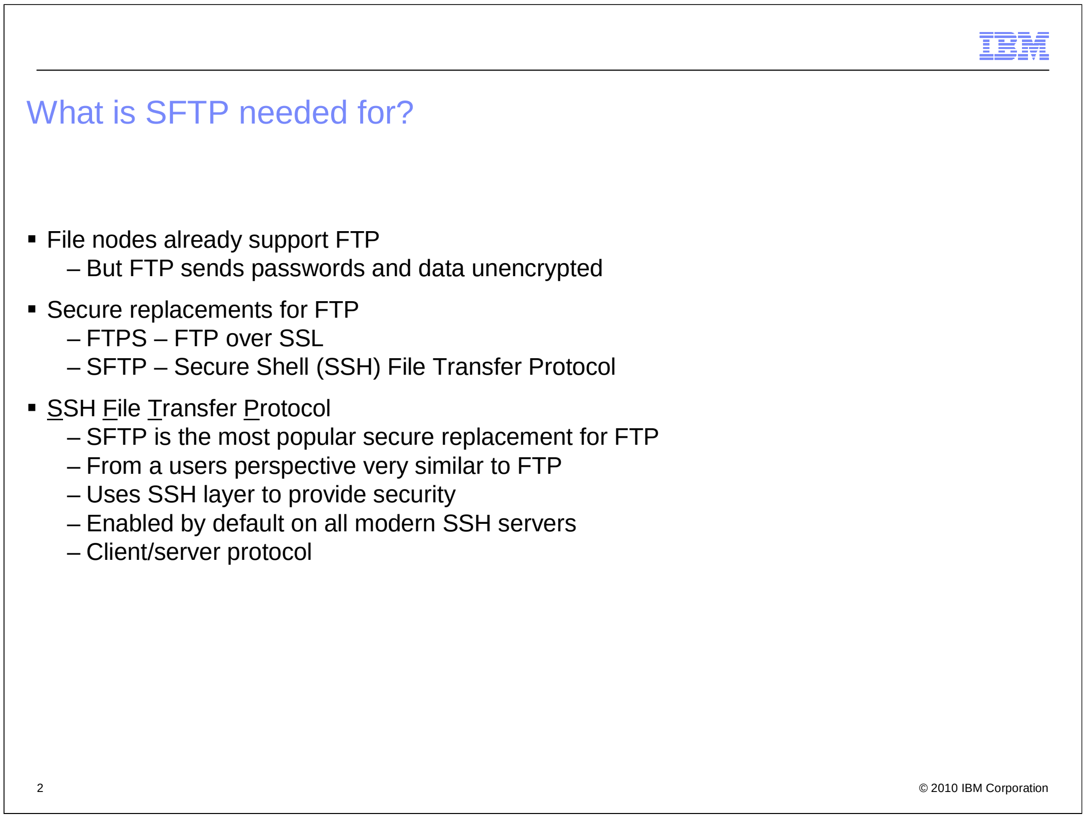


WebSphere Message Broker Version 7

Using the file nodes for secure file transfer (SFTP)

This session describes the new facilities in WebSphere® Message Broker version 7 to permit the secure transmission of file transfer data. This uses the SFTP function.



What is SFTP needed for?

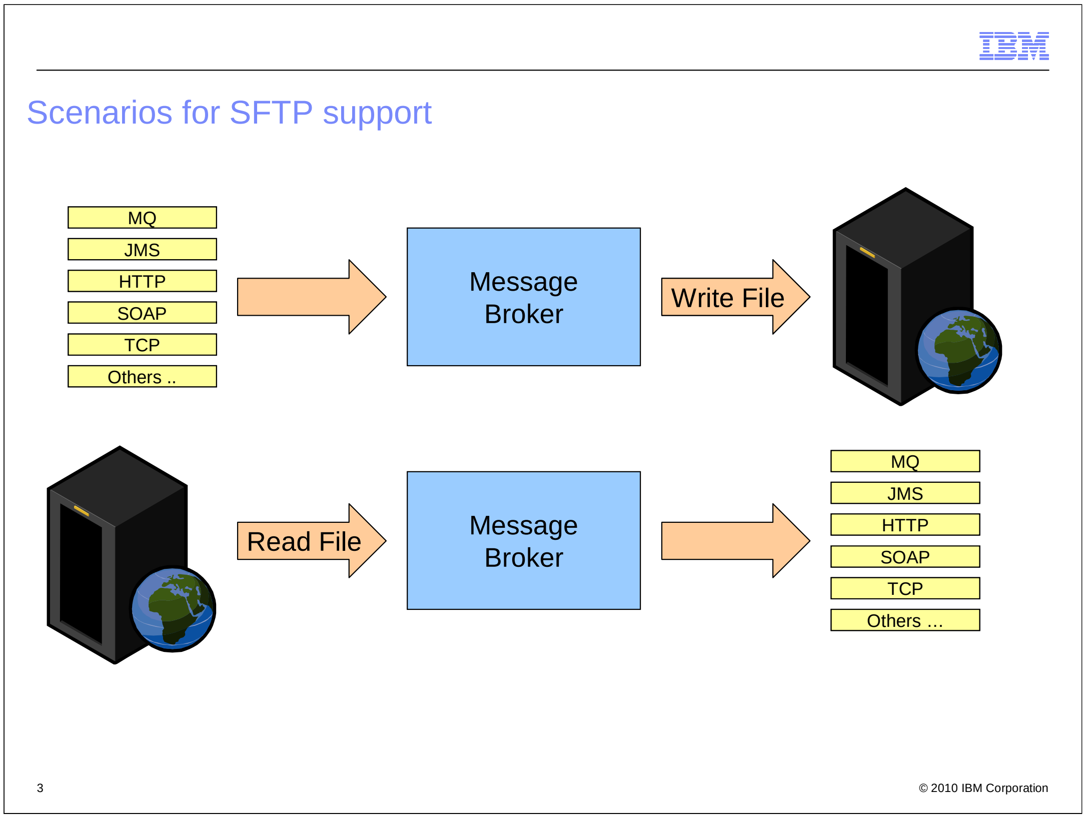
- File nodes already support FTP
  - But FTP sends passwords and data unencrypted
- Secure replacements for FTP
  - FTPS – FTP over SSL
  - SFTP – Secure Shell (SSH) File Transfer Protocol
- SSH File Transfer Protocol
  - SFTP is the most popular secure replacement for FTP
  - From a users perspective very similar to FTP
  - Uses SSH layer to provide security
  - Enabled by default on all modern SSH servers
  - Client/server protocol

2

Secure shell or SSH is a network protocol that allows data to be exchanged using a secure channel between two networked devices. Originating on UNIX®, SSH was designed as a replacement for Telnet and other insecure remote shells, which send information, including passwords, in plain text, leaving them open for interception. The encryption used by SSH provides confidentiality and integrity of data, by using public-key cryptography.

SFTP is not FTP run over SSH, but rather a new protocol designed from the ground up by an Internet Engineering Taskforce group. It is typically used with version two of the SSH protocol, using TCP port 22, to provide secure file transfer, but is intended to be usable with other protocols as well.

There are numerous SFTP server implementations both for UNIX and Windows®. The most widely known is OpenSSH, but there are also proprietary implementations.



Scenarios for SFTP support

MQ
JMS
HTTP
SOAP
TCP
Others ..

Message Broker

Write File

Read File

Message Broker

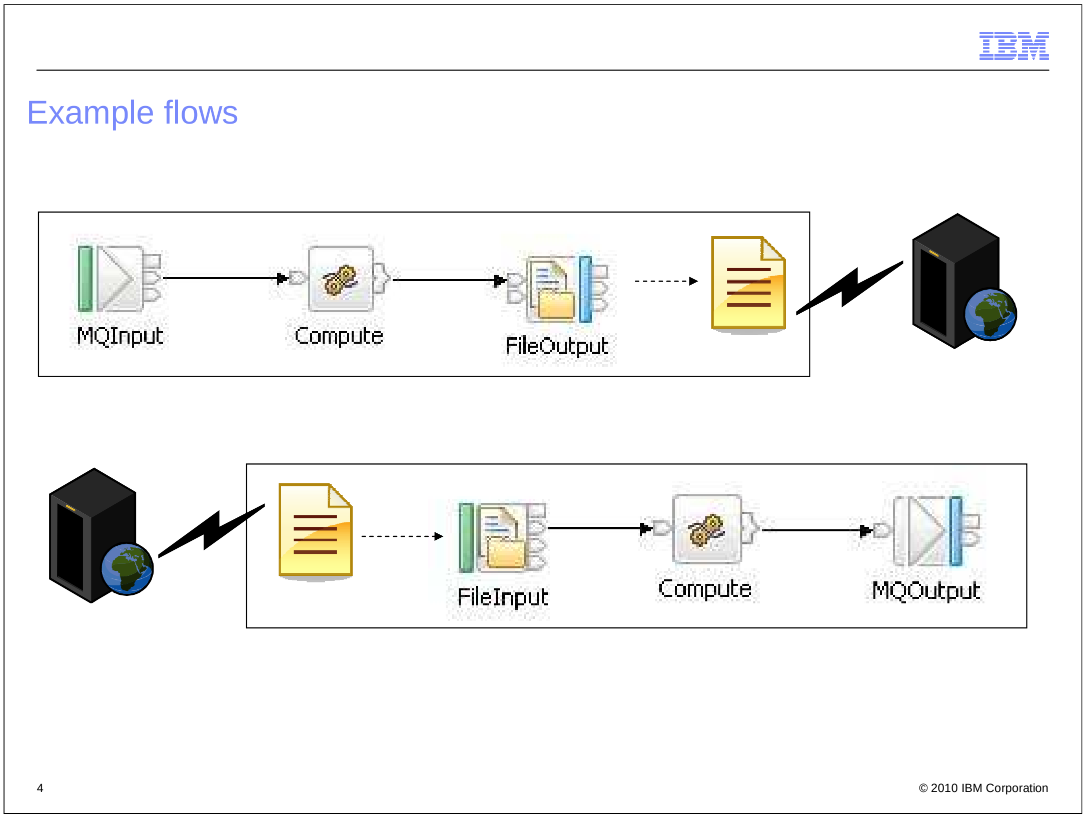
MQ
JMS
HTTP
SOAP
TCP
Others …

3

© 2010 IBM Corporation

Secure FTP extends the support that is already provided in Message Broker for standard FTP and file access. Message Broker can both read and write data using the built-in file nodes, and these also include an FTP capability. Both these two functions have now been extended to use the SFTP function.

As with all other nodes, the Message Broker flow can then connect this secure FTP input and output to any other form of input and output, such as MQ, JMS and HTTP.



Example flows

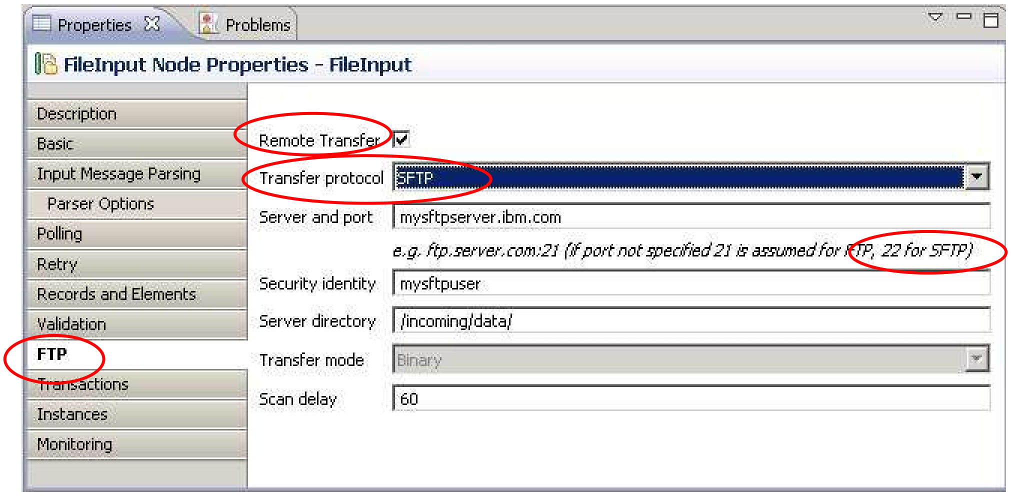
4

The scenarios supported take messages from an input destination and write them to a remote file using SFTP, and read records from a remote file using SFTP and propagate them to an output destination.

Notice in the example flows that no new nodes have been added to support this. The existing file Input and file output nodes have been updated to offer SFTP as an alternative protocol to FTP when handling remote files.

Behavior is very similar to FTP. When writing, the file is created locally on the broker’s file system then transferred to the remote destination using SFTP. When reading, the file is transferred from the remote destination using SFTP to the broker’s file system, then read locally.

SFTP Support in the File Nodes



FileInput Node Properties - FileInput

Remote Transfer ☑

Transfer protocol: SFTP

Server and port: mysftpserver.ibm.com

e.g. ftp.server.com:21 (if port not specified 21 is assumed for FTP, 22 for SFTP)

Security identity: mysftpuser

Server directory: /incoming/data/

Transfer mode: Binary

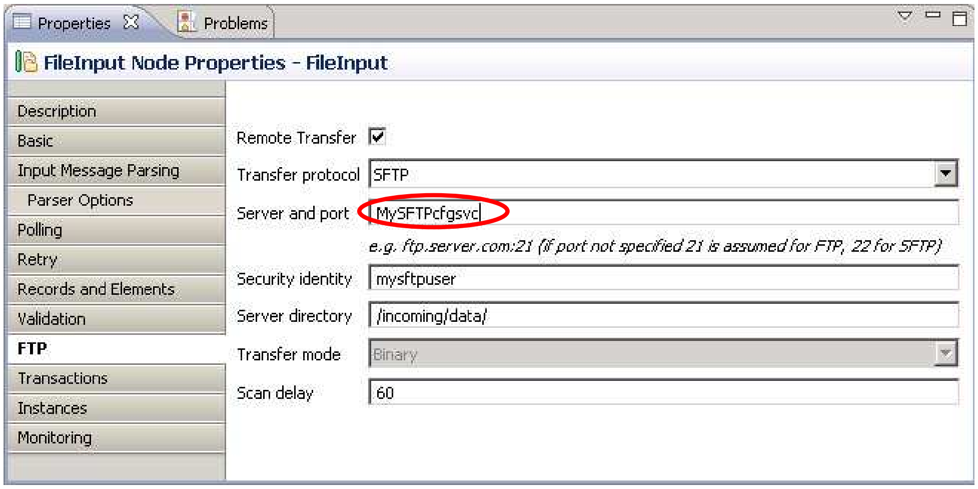
Scan delay: 60

The SFTP support in both file input and file output nodes is implemented by small changes to the node properties on the existing FTP tab.

The check box that was called FTP has been renamed ‘Remote transfer’, and a new drop-down property called ‘Transfer protocol’ has been added allowing you to select FTP or the new SFTP option.

If you do not specify a port when entering the server details, port 22 is assumed.

SFTP as a configurable service (1 of 2)



Remote Transfer ☑

Transfer protocol: SFTP

Server and port: MySFTPcfgsvc

e.g. ftp.server.com:21 (if port not specified 21 is assumed for FTP, 22 for SFTP)

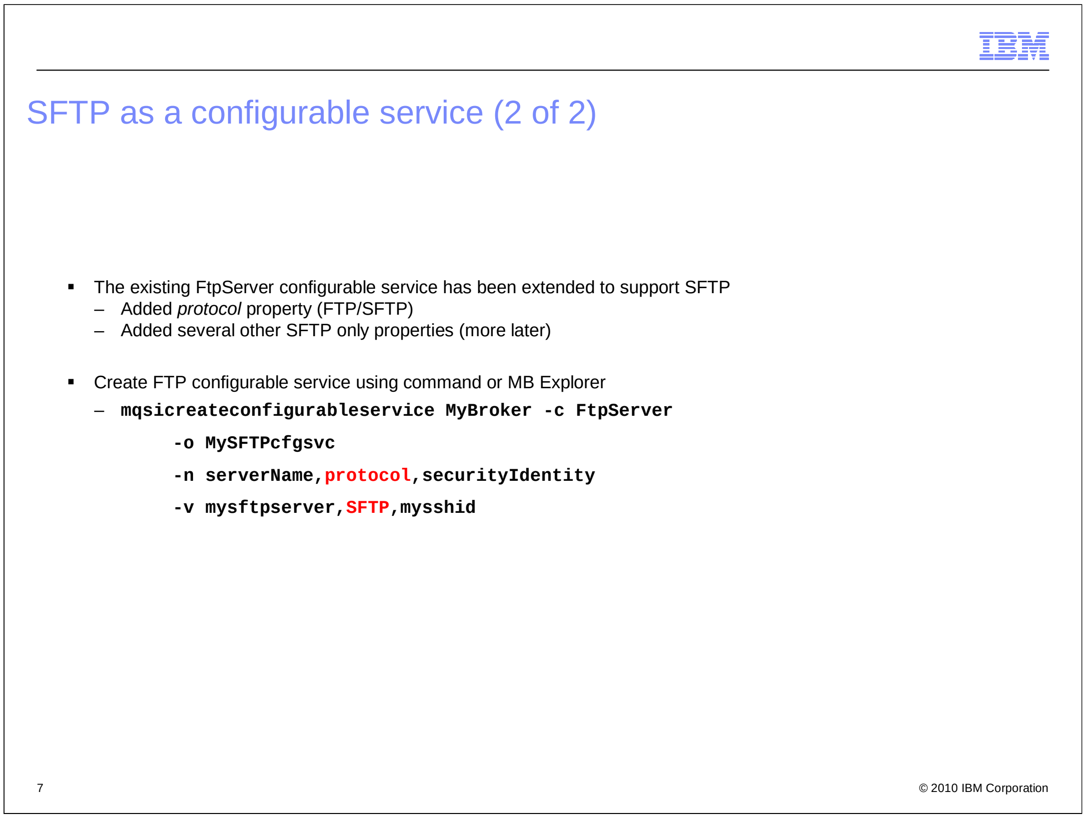
Security identity: mysftpuser

Server directory: /incoming/data/

Transfer mode: Binary

Scan delay: 60

As with the existing support for FTP access, the FTP server can be specified directly in the node properties. Alternatively, it can be specified using Message Broker configurable services, as shown in this screen capture. This is the recommended technique for defining and managing the connection properties to such systems.



SFTP as a configurable service (2 of 2)

- The existing FtpServer configurable service has been extended to support SFTP
  - Added *protocol* property (FTP/SFTP)
  - Added several other SFTP only properties (more later)

- Create FTP configurable service using command or MB Explorer
  - **mqsicreateconfigurableservice MyBroker -c FtpServer**

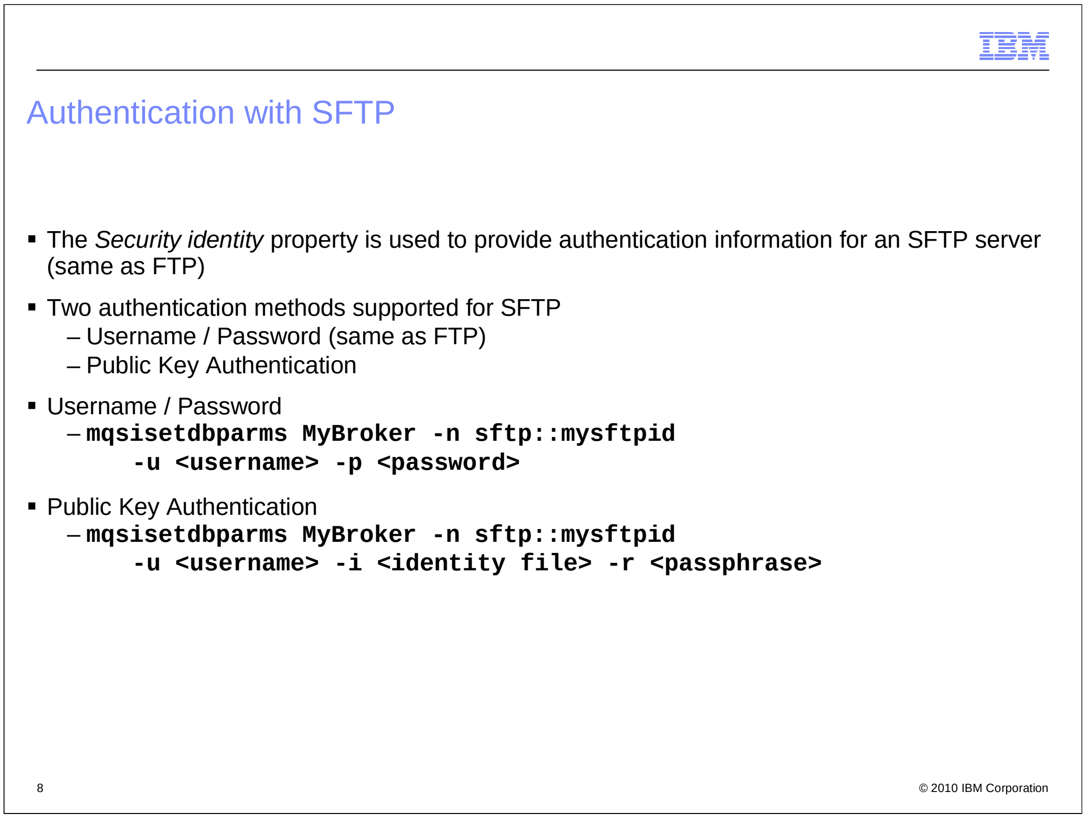
```
mqsicreateconfigurableservice MyBroker -c FtpServer
        -o MySFTPcfgsvc
        -n serverName,protocol,securityIdentity
        -v mysftpserver,SFTP,mysshid
```

The existing configurable service type for FTP, “FtpServer”, has been extended to support SFTP by adding a new protocol property. Other properties have also been added for configuring the security aspects of SFTP. The allowable values for the protocol property are FTP or SFTP.

The way of specifying a configurable service, instead of a server and port, remains the same. If the value of the Server and port property is not an IP address, it is assumed to be a configurable service name.

Any protocol property in the configurable service overrides that on the node.

Some of the existing FtpServer properties do not apply for SFTP. These include transferMode and connectionType.



Authentication with SFTP

- The *Security identity* property is used to provide authentication information for an SFTP server (same as FTP)
- Two authentication methods supported for SFTP
  - Username / Password (same as FTP)
  - Public Key Authentication
- Username / Password
  - **mqsisetdbparms MyBroker -n sftp::mysftpid -u <username> -p <password>**
- Public Key Authentication
  - **mqsisetdbparms MyBroker -n sftp::mysftpid -u <username> -i <identity file> -r <passphrase>**

8

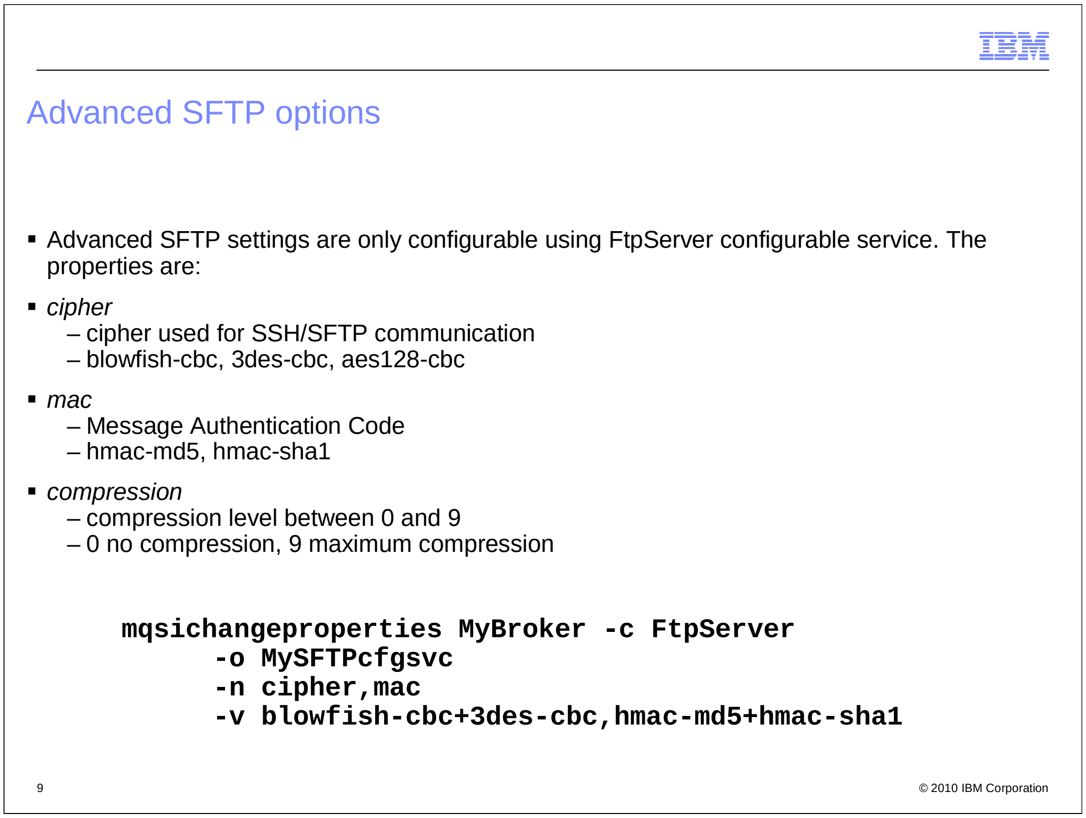
The Security identity property of the file node or configurable service is used to provide authentication information for an SFTP server, in the same manner as FTP.

As well as password, an alternative authentication method is supported using OpenSSH format identity files, sometimes called an SSH key file. Identity files only contain one identity, unlike x509 Key Stores.

The identity file can be protected by a pass phrase which is used when decrypting the identity.

The m-q-s-i-set-db-parms command has been extended to take an identity file and optional pass phrase, as an alternative to a password.

On z/OS systems, identity files are stored in EBCDIC format, and on other operating systems they are stored in ASCII format.



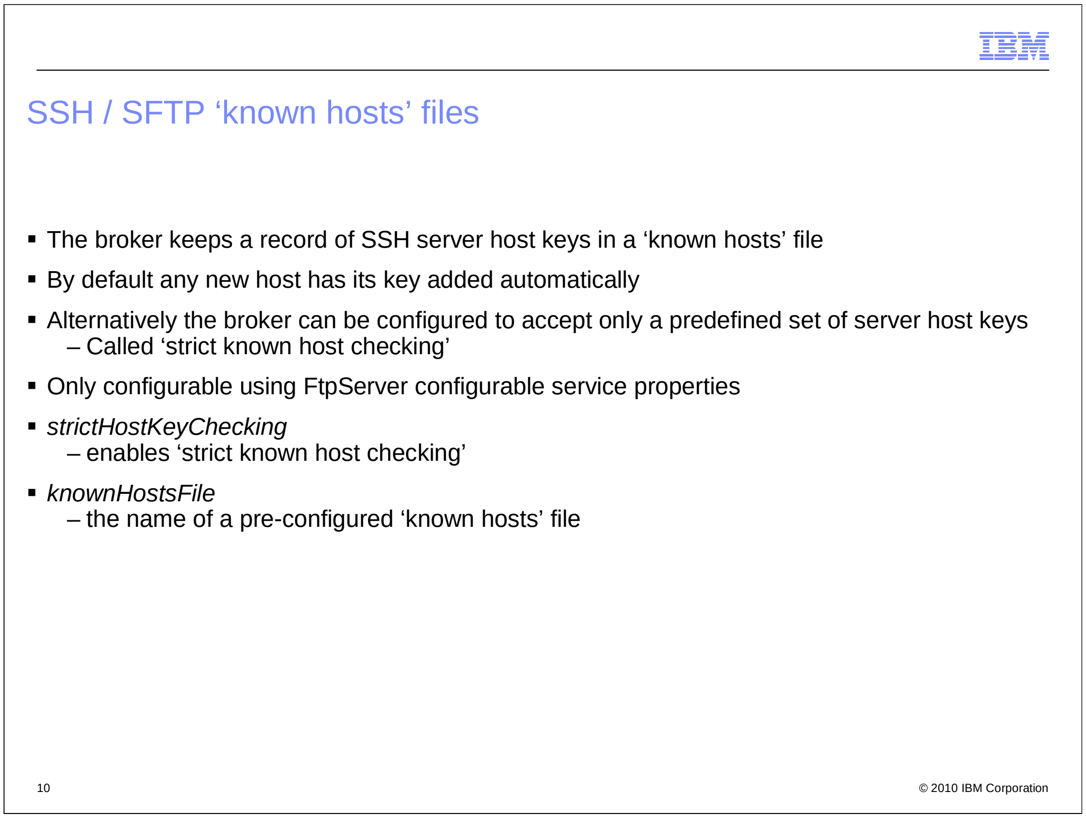
Advanced SFTP options

- Advanced SFTP settings are only configurable using FtpServer configurable service. The properties are:
- *cipher*
  - – cipher used for SSH/SFTP communication
  - – blowfish-cbc, 3des-cbc, aes128-cbc
- *mac*
  - – Message Authentication Code
  - – hmac-md5, hmac-sha1
- *compression*
  - – compression level between 0 and 9
  - – 0 no compression, 9 maximum compression

```
mqsichangeproperties MyBroker -c FtpServer
      -o MySFTPcfgsvc
      -n cipher,mac
      -v blowfish-cbc+3des-cbc,hmac-md5+hmac-sha1
```

The more advanced options of SFTP are not configurable on the file nodes, and must be specified using an FtpServer configurable service.

Note that the cipher and mac properties take the form of a list of one or more of the allowable values, separated by plus signs, as they are SSH implementation dependent. List the values in order of preference. An example of how to do this is shown on this slide.



SSH / SFTP ‘known hosts’ files

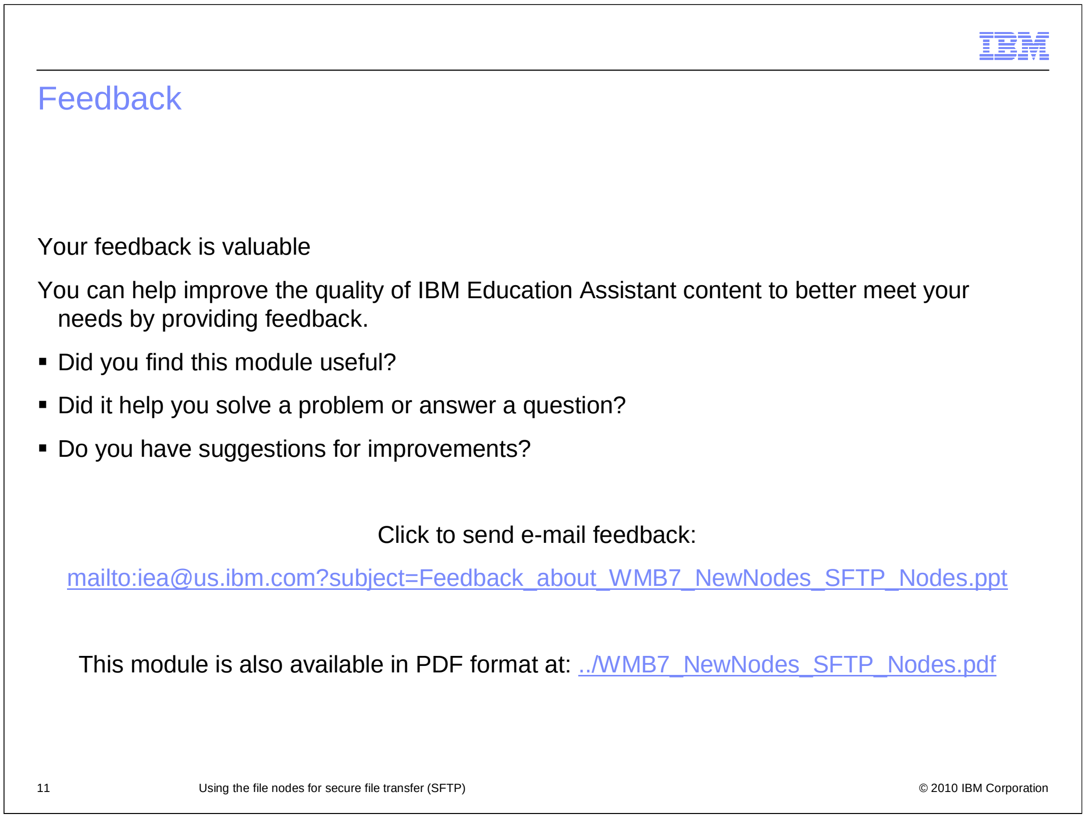
- The broker keeps a record of SSH server host keys in a ‘known hosts’ file
- By default any new host has its key added automatically
- Alternatively the broker can be configured to accept only a predefined set of server host keys
  - – Called ‘strict known host checking’
- Only configurable using FtpServer configurable service properties
- *strictHostKeyChecking*
  - – enables ‘strict known host checking’
- *knownHostsFile*
  - – the name of a pre-configured ‘known hosts’ file

10

The broker keeps a record of the host key of the SSH servers with which it communicates using a ‘known hosts’ file. By default, any new host has its key added automatically, and if a host’s key changes, an exception is thrown.

For added security, the broker can be configured to accept only a predefined set of host keys. This is called ‘strict known host checking’ and you provide a pre-configured ‘known hosts’ file. The broker only uses these servers and keys, and an unknown server causes an exception to be thrown. This is only configurable using an FtpServer configurable service.

No management capability is provided by the broker for ‘known hosts’ files.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

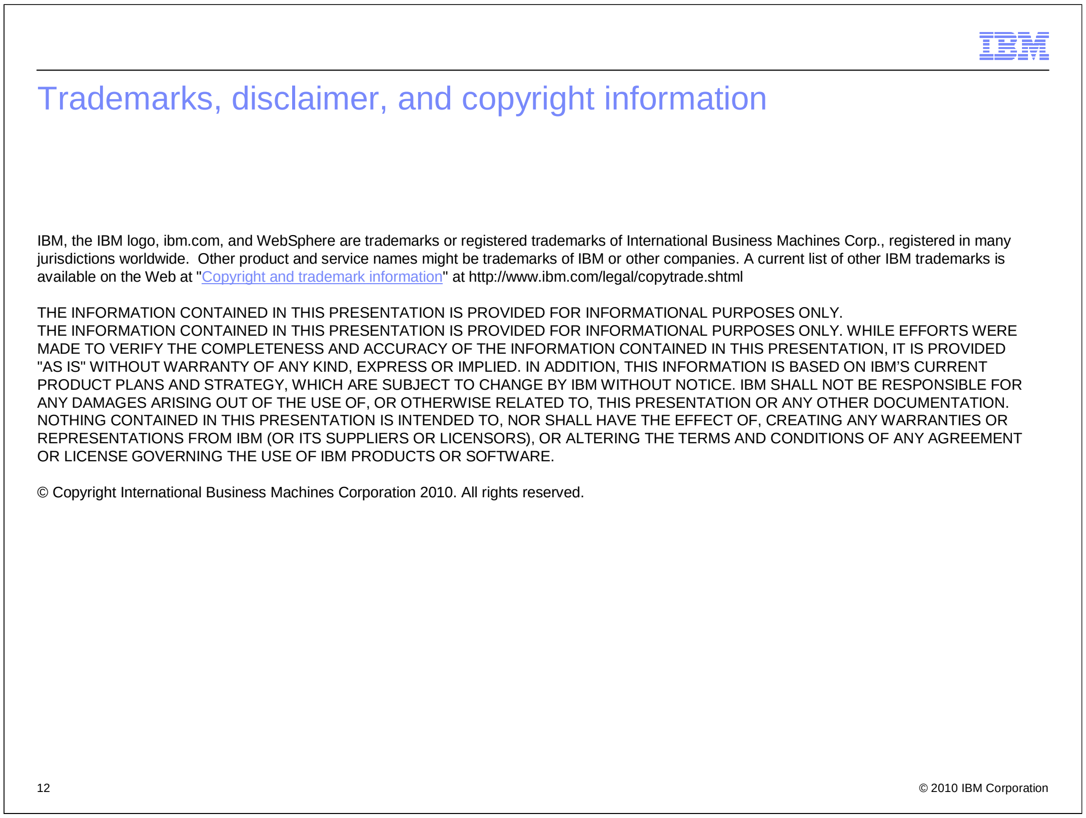
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WMB7_NewNodes_SFTP_Nodes.ppt

This module is also available in PDF format at: ../WMB7_NewNodes_SFTP_Nodes.pdf

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



Trademarks, disclaimer, and copyright information

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
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.