



IBM Software Group

# Using WebSphere DataPower SOA Appliance with the FTP Transport Protocol

David Shute ([dshute@us.ibm.com](mailto:dshute@us.ibm.com))

DataPower Enablement Program Manager

1 February 2011



WebSphere® Support Technical Exchange



# Agenda

- Overview
- Scenario 1: FTP Poller
- Scenario 2: SFTP Poller
- Scenario 3: FTP Server FSH
- Scenario 4: SFTP Server FSH
- Summary

# Overview

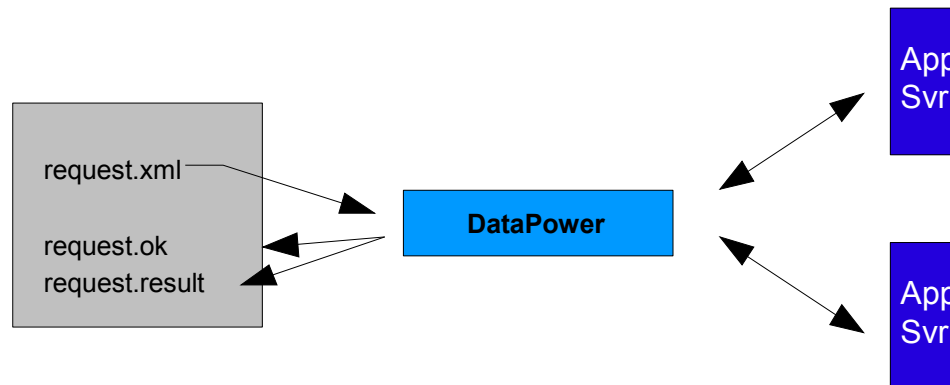
- Enterprise customers want/need to bridge legacy S/FTP-based messaging flows to newer HTTP-based Web Services
- Enterprise customers want/need to bridge S/FTP-based messaging flows to other protocols (i.e. MQ)
- Enterprise customers want/need to transparently proxy S/FTP servers
- This presentation examines configuration patterns for each business need



# Scenario 1: FTP Poller

The DataPower device bridges the legacy FTP-based transport used by business partners to newer HTTP-based services within the enterprise. The DataPower device performs the following tasks:

- Retrieve message files by polling a remote FTP server
- Route messages to an enterprise HTTP application server for processing
- Capture the response provided by the HTTP-based application and place the response file on the remote FTP server, using a predetermined naming pattern.



# Scenario 1: FTP Poller

## Configure Multi-Protocol Gateway

General | Advanced | Stylesheet Params | Headers | Monitors | WS-Addressing | WS-ReliableMessaging

Apply | Cancel | Delete

Export | View Log | View Status | Show Probe | Validate Conformance | Help

Multi-Protocol Gateway status: [up]

### General Configuration

Multi-Protocol Gateway Name  
FTPPoller \*

Summary

Type  
 dynamic-backend  
 static-backend \*

XML Manager  
default + ... \*

Multi-Protocol Gateway Policy  
FTP-Poll-Route + ... \*

URL Rewrite Policy  
(none) + ...

### Back side settings

With a dynamic proxy back end type, the back end server address and port are determined by a stylesheet in a policy action.

### Front side settings

Front Side Protocol  
FTPPoller (FTP Poller Front Side Handler)

### User Agent settings

Match	Property
Note: To edit the User Agent, please access via the XML Manager above.	

FTP Poller Front Side Handler: FTPPoller [down - Object is disabled]

Apply | Cancel | Undo

Administrative State  enabled  disabled

Comments

Target Directory  \*

Delay Between Polls  milliseconds \*

Input File Match Pattern  \*

Processing File Renaming Pattern

Delete Input File on Success  on  off

Success File Renaming Pattern

Delete file on processing error  on  off

Error File Renaming Pattern

Generate Result File  on  off

Result File Name Pattern  \*

Processing Seize Timeout  \*

XML Manager  + ... \*

Maximum File Transfers Per Poll Cycle

# Scenario 1: FTP Poller

## Additional Considerations

This configuration renames rather than deletes files as they are processed. Some systems may work better if processed files are deleted. See the FTP Poller FSH configuration to delete files automatically.

By default, the FTP Poller retrieves only one file at a time. This may not provide the throughput needed for the client systems. Use the **Maximum File Transfers Per Poll Cycle** input of the Front Side Handler to increase the number of matching files retrieved per connection. Note that the optimal setting will depend upon the size of the files retrieved and the message processing latency for each message. This setting may also be affected by the **Delay Between Polls** setting of the Handler. Note that the next cycle of retrieval will not start until the current cycle completes.

The frequency of polling can be controlled using the **Delay Between Polls** setting of the Handler. The optimal value for this setting will be determined by the processing latency of each message and the ability of the back end services to accept traffic. This may also be affected by the **Maximum File Transfers Per Poll Cycle** setting.

The device opens a new connection for each file retrieved, which results in two connections to the front side FTP server for each file obtained. This behavior can be modified by setting the Gateway **Persistent Connections** property to *On* (on the Advanced tab). This will cause the poller to reuse connections to the server as much as possible.

When more than one service polls the same remote directory, consider setting the **Processing Seize Timeout** input to a non-zero value. This can potentially prevent an inbound message from failing to reach the intended destination. Without using this mechanism, a file can be marked as “in process” forever.

# Scenario 1: FTP Poller

XML Manager: ftpmgr [up]

Apply Cancel Undo

Administrative State  enabled  disabled

Comments

URL Refresh Policy  + ...

Compile Options Policy  + ...

XSL Cache Size  stylesheets

SHA1 Caching  on  off

Static Document Call  on  off

XSLT Expression Optimization  on  off

Load Balance Groups  Add + ...

User Agent Configuration  + ... \*

## Configure User Agent

Main Proxy Policy SSL Proxy Profile Policy **Basic-Auth Policy**

User Agent: FTPER [up]

Apply Cancel Undo [Export](#) | [View Lo](#)

**Basic-Auth Policy**

URL Matching Expression	User name	Password	
ftp*	sqa	*****	↑ ↓ ✎ ✕
			Add

The Basic-Auth Policy of the User Agent attached to the XML Manager provides credentials for connecting to the remote FTP Server.

# Scenario 1: FTP Poller



Configure Multi-Protocol Gateway Style Policy

**Policy:**  
 Policy Name:  \*  
  [Export](#)

**Rule:**  
 Rule Name:  Rule Direction:

Create rule: Click New, drag action icons onto line. Edit rule: Click on rule, double-click on action

Filter Sign Verify Validate Encrypt Decrypt Transform Route AAA Results Advanced

XPath Routing Map: XML-Poller-Map [up]

[Export](#) | [View Log](#) | [View Status](#) | [Help](#)

**Rules**

XPath Expression	Remote Host	Remote Port	SSL	
<code>/*[namespace-uri()='urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-1.0' and local-name()='QuoteRequest']</code>	127.0.0.1	3888	off	↑ ↓ ✎ ✕
<code>/*[namespace-uri()='urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-1.0' and local-name()='Order']</code>	127.0.0.1	3888	off	↑ ↓ ✎ ✕
<input type="button" value="Add"/>				

**Input**  
 Input:  INPUT \*

**Options**  
 Route (Using Stylesheet or XPath Expression)

Selection Method  
 Use Stylesheet to Select Destination  
 Use Variable to Select Destination  
 Use XPath to Select Destination

XPath Routing Map:  + ...

Asynchronous:  on  off

**Output**  
 Output:  OUTPUT \*

The Route Action uses an Xpath Routing Map keyed on the root node of the request to determine destination.



# Scenario 1: FTP Poller

## Additional Considerations

No response rule was configured. All responses will pass through the gateway unaltered.

Request files that do not meet either of the routing criteria will result in an error. This is, in effect, a firewall protecting the enterprise from processing requests placed in the remote FTP location that do not meet the criteria of the back-end application. This same benefit can be achieved by using an Xpath expression in the Match action for matching requests to processing rules. Two rules would be needed for the two Xpath expression cases configured in the Route action. A third rule, matching all other requests, could then be configured to send non-conformant requests to a different processing system rather than triggering an error.

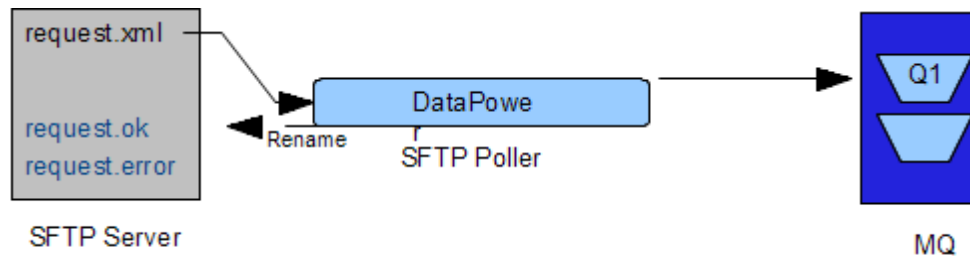
No error rule was configured. This is done to illustrate what will happen by default if the device encounters an error in processing requests. This behavior can be controlled by creating an error rule that takes the desired action.



## Scenario 2: SFTP Poller

In this scenario, the DataPower device extends the ESB to include FTP-based clients requiring additional security. The message exchange pattern established by the enterprise does not require a response back to the SFTP server (such information is provided in a different manner), which makes this use case a one-way transaction through the DataPower device. The DataPower device performs the following tasks:

- Retrieve message files by polling a remote SFTP server
- Pass possibly binary message files through to an enterprise MQ queue.
- Capture the response from the MQ system and place the response file on the remote SFTP server, using a predetermined naming pattern.



# Scenario 2: SFTP Poller

## Configure Multi-Protocol Gateway

General | **Advanced** | Stylesheet Params | Headers | Monitors | WS-Addressing | WS-ReliableMessaging | XML Threa

Apply | Cancel | Delete

Export | View Log | View Status | Show Probe | Validate Conformance | Help

Multi-Protocol Gateway status: [up]

### General Configuration

**Multi-Protocol Gateway Name**  
FTPPoller \*

**Summary**  
[ ]

**Type**  
 dynamic-backend  
 static-backend \*

**XML Manager**  
default + ... \*

**Multi-Protocol Gateway Policy**  
default + ... \*

**URL Rewrite Policy**  
(none) + ...

<p><b>Back side settings</b></p> <p><b>Backend URL</b> dpmq://Transactor//transact?Requ *</p> <p>MQHelper   TibcoEMSHelper WebSphereJMSHelper IMSConnectHelper</p> <p><b>Response Type</b>  <input type="radio"/> JSON  <input type="radio"/> Non-XML  <input checked="" type="radio"/> Pass-Thru  <input type="radio"/> SOAP  <input type="radio"/> XML</p> <p><b>Propagate URI</b>  <input type="radio"/> on  <input checked="" type="radio"/> off</p>	<p><b>Front side settings</b></p> <div style="border: 1px solid red; padding: 5px;"> <p><b>Front Side Protocol</b> SFTPPoller (SFTP Poller Front Side Handler) X</p> <p>[ ] Add + ...</p> </div> <p><b>Request Type</b>  <input type="radio"/> JSON  <input type="radio"/> Non-XML  <input checked="" type="radio"/> Pass-Thru  <input type="radio"/> SOAP  <input type="radio"/> XML</p>
--	---

## Configure Multi-Protocol Gateway

General | **Advanced** | Stylesheet Params | He

Apply | Cancel | Delete

Multi-Protocol Gateway status: [up]

### Advanced settings

**Persistent Connections**  
 on  off

**Loop Detection**  
 on  off

**Follow Redirects**  
 on  off

**Allow Chunked Uploads**  
 on  off

**Process Backend Errors**  
 on  off

**Front Persistent Timeout**  
 180 seconds \*

**Back Persistent Timeout**  
 180 seconds \*

Gateway Configuration settings reside on both the General and the Advanced Tabs.

# Scenario 2: SFTP Poller

## Configure SFTP Poller Front Side Handler

This configuration has been modified, but not yet saved.

### Main

SFTP Poller Front Side Handler: SFTPPoller [up]

Apply Cancel Undo

Administrative State	<input checked="" type="radio"/> enabled <input type="radio"/> disabled
Comments	<input type="text"/>
Target Directory	<input type="text" value="sftp://9.70.155.68/%2Fhome/sqa/"/> *
Delay Between Polls	<input type="text" value="3000"/> milliseconds *
Input File Match Pattern	<input type="text" value="([0-9]{6})\.xml\$"/> *
Processing File Renaming Pattern	<input type="text"/>
Delete Input File on Success	<input type="radio"/> on <input checked="" type="radio"/> off
Success File Renaming Pattern	<input type="text" value="\$1.processed.ok"/>
Delete file on processing error	<input type="radio"/> on <input checked="" type="radio"/> off
Error File Renaming Pattern	<input type="text" value="respond/\$0.processed.error"/>
Generate Result File	<input checked="" type="radio"/> on <input type="radio"/> off
Result File Name Pattern	<input type="text" value="respond/\$1.result.sft"/> *
Processing Seize Timeout	<input type="text" value="0"/> *
XML Manager	<input type="text" value="default"/> [v] [ + ] [ ... ] *
Maximum File Transfers Per Poll Cycle	<input type="text" value="5"/>
SSH Client Connection	<input type="text" value="Squa"/> [v] [ + ] [ ... ] *

## Configure SSH Client Profile

### Main

SSH Client Profile: Squa [up]

Apply Cancel Undo

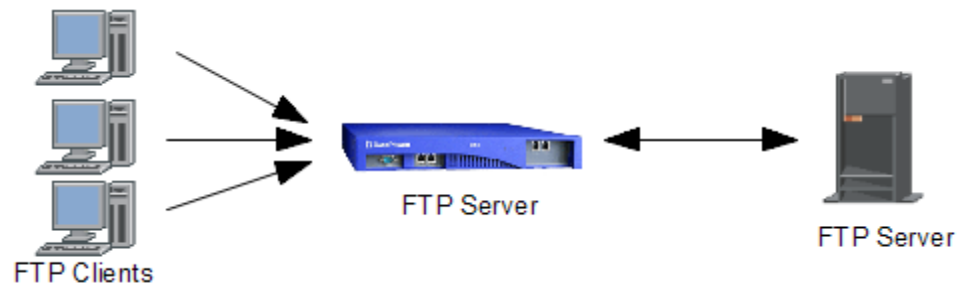
Administrative State	<input checked="" type="radio"/> enabled <input type="radio"/> disabled
Comments	<input type="text"/>
User name	<input type="text" value="sqa"/> *
Profile Usage	<input type="text" value="SFTP"/> *
User Authentication	<input checked="" type="checkbox"/> Public Key <input checked="" type="checkbox"/> Password *
User Private Key	<input type="text" value="idrsa"/> [v] [ + ] [ ... ] *
Password	<input type="password" value="....."/> <input type="password" value="....."/> *
Persistent Connections	<input checked="" type="radio"/> on <input type="radio"/> off *
Persistent Connection Idle Timeout	<input type="text" value="120"/> seconds *
Strict Host Key Checking	<input type="radio"/> on <input checked="" type="radio"/> off *

Note the use of File Renaming conventions to signal success or failure, and also the ability to rename files in process.

## Scenario 3: FTP Server FSH

The DataPower device protects existing enterprise FTP-based assets. The device acts as a transparent FTP server proxy, accepting inbound FTP connections from remote FTP clients and allows the remote clients to transparently access the FTP files and directories offered by the back end server.

- Accepts inbound FTP connections from remote FTP clients and allows the remote clients to transparently access the FTP files and directories offered by the back end server.
- Authenticates the username and password of inbound connections before allowing the remote client to gain access to the server system.
- Supports the streaming of large files through the device.



# Scenario 3: FTP Server FSH

## Configure Multi-Protocol Gateway

General | Advanced | Stylesheet.Params | Headers | Monitors | WS-Addressing | WS-ReliableMessaging | XML Threat

Apply | Cancel | Delete

Export | View Log | View Status | Show Probe | Validate Conformance | Help

Multi-Protocol Gateway status: [up]

### General Configuration

**Multi-Protocol Gateway Name**  
 \*

**Summary**

**Type**  
 dynamic-backend  
 static-backend \*

**XML Manager**  
 + ... \*

**Multi-Protocol Gateway Policy**  
 + ... \*

**URL Rewrite Policy**  
 + ...

---

**Back side settings**      **Front side settings**

**Backend URL**  
 \*

**Front Side Protocol**  
 + ... \*

**MQHelper**   **TibcoEMSHelper**  
**WebSphereIMSHelper**  
**IMSCoconnectHelper**

**User Agent settings**

Match	Property
ftp*	BasicAuthPolicies

### Response Type

- JSON
- Non-XML
- Pass-Thru
- SOAP
- XML

### Request Type

- JSON
- Non-XML
- Pass-Thru
- SOAP
- XML

### Flow Control

- on
- off

### Back attachment processing format

- Dynamic
- MIME
- DIME
- Detect

### Front attachment processing format

- Dynamic
- MIME
- DIME
- Detect

### Back Side Timeout

 \*

### Front Side Timeout

 \*

### Stream Output to Back

- Buffer Messages
- Stream Messages

### Stream Output to Front

- Buffer Messages
- Stream Messages

### HTTP Version to Server

- HTTP 1.0
- HTTP 1.1

### Propagate URI

- on
- off

### Compression

- on
- off

# Scenario 3: FTP Server FSH

## Configure FTP Server Front Side Handler

Main Virtual Directories

FTP Server Front Side Handler: FTPtransparent [up]

Apply Cancel Undo

Administrative State  enabled  disabled

Comments

Local IP Address  Select Alias \*

Port Number  \*

Filesystem Type

Default Directory

Maximum Filename Length

Access Control List  + ...

Require TLS  on  off

SSL Proxy  + ...

Password AAA Policy  + ...

Certificate AAA Policy  + ...

Allow CCC Command  on  off

Passive (PASV) Command

Enable LIST command support  on  off

## Configure AAA Policy

Main Identity Authenticate MapCredentials Resource MapResource Authorize

AAA Policy: FTPProxyPassword [up]

Apply Cancel Undo

Methods

- HTTP Authentication Header
- Password-carrying UsernameToken Element from WS-Security Header
- Derived-key UsernameToken Element from WS-Security Header
- BinarySecurityToken Element from WS-Security Header
- WS-SecureConversation Identifier
- WS-Trust Base or Supporting Token
- Kerberos AP-REQ from WS-Security Header
- Kerberos AP-REQ from SPNEGO Token
- Subject DN of the SSL Certificate from the Connection Peer
- Name from SAML Attribute Assertion
- Name from SAML Authentication Assertion
- SAML Artifact
- Client IP Address
- Subject DN from Certificate in the Message's signature
- Token Extracted from the Message
- Token Extracted as Cookie Value
- LTPA Token
- Processing Metadata
- Custom Template
- HTML Forms-based Authentication \*

Processing Metadata Items  + ...

The FSH Password AAA Policy relies on the use of Metadata to capture the ID/PWD of the remote client, which is used to authenticate that client.

# Scenario 3: FTP Server FSH

AAA Policy: FTPProxyPassword [up]

Apply Cancel Undo

Resource Information

- URL Sent to Back End
- URL Sent by Client
- URI of Toplevel Element in the Message
- Local Name of Request Element
- HTTP Operation (GET/POST)
- XPath Expression
- Processing Metadata

\*

Processing Metadata Items

(none) [v] [ + ] [ ... ]

## Configure AAA Policy

Main Identity Authenticate MapCredentials Resource MapResource Authorize

AAA Policy: FTPProxyPassword [up]

Apply Cancel Undo

[Export](#) | [View Log](#) | [View Status](#) | [Help](#)  
[Flush Cache](#)

Method

Allow Any Authenticated Client [v] \*

Cache authorization results

Disabled [v] \*

The FSH Password AAA Policy relies on the use of Metadata to configure the AAA Policy. No other value will work.



# Scenario 3: FTP Server FSH

XML Manager: ftpmgr [up]

Apply Cancel Undo

Administrative State  enabled  disabled

Comments

URL Refresh Policy

Compile Options Policy

XSL Cache Size  stylesheets

SHA1 Caching  on  off

Static Document Call  on  off

XSLT Expression Optimization  on  off

Load Balance Groups

User Agent Configuration    \*

## Configure User Agent

Main Proxy Policy SSL Proxy Profile Policy **Basic-Auth Policy**

User Agent: FTPER [up]

Apply Cancel Undo

[Export](#) | [View Lo](#)

### Basic-Auth Policy

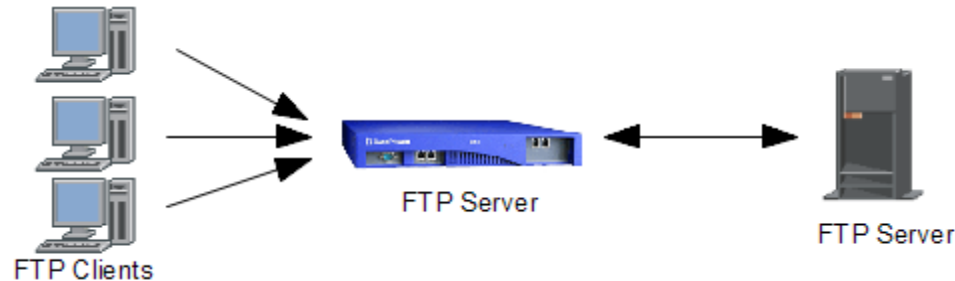
URL Matching Expression	User name	Password	
ftp*	sqa	*****	↑ ↓ ✎ ✕
			<input type="button" value="Add"/>

The Basic-Auth Policy of the User Agent associated with the XML Manager in use provides the credentials necessary to connect to the destination (back end) FTP Server.

## Scenario 4: SFTP Server FSH

In this scenario, the DataPower device provides additional security for SFTP-based message exchange patterns. The device performs the following tasks:

- acts as an SFTP server to which remote clients connect
- presents a virtual filesystem that exists only on the device, masking the actual back end server.
- dynamically routes files to the desired location on the back end SFTP server.
- performs public key authentication on the inbound connections



# Scenario 4: SFTP Server FSH

## Configure Multi-Protocol Gateway

General | Advanced | Stylesheet Params | Headers | Monitors | WS-Addressing | WS-ReliableMessaging

Apply | Cancel | Delete

Export | View Log | View Status | Show Probe | Validate Conformance | Help

Multi-Protocol Gateway status: [up]

### General Configuration

**Multi-Protocol Gateway Name**  
 \*

**Summary**

**Type**  
 dynamic-backend  
 static-backend  
 \*

**XML Manager**  
 + ... \*

**Multi-Protocol Gateway Policy**  
 + ... \*

**URL Rewrite Policy**  
 + ...

### Back side settings

### Front side settings

With a dynamic proxy back end Multi-Protocol Gateway type, the back end server address and port are determined by a stylesheet in a policy action.

**Front Side Protocol**  
 ✕

Add + ...

\*

### User Agent settings

Match	Property
sftp://*	BasicAuthPolicies
sftp://*	SFTPPolicies

**Note:**To edit the User Agent, please access via the XML Manager above.

### Response Type

- JSON
- Non-XML
- Pass-Thru
- SOAP
- XML

### Request Type

- JSON
- Non-XML
- Pass-Thru
- SOAP
- XML



# Scenario 4: SFTP Server FSH

## Configure XML Manager

Main XML Parser Document Cache Extension Functions Document Cache

XML Manager: sftp\_xml\_manager [up]

Apply Cancel Undo

Administrative State  enabled  disabled

Comments

URL Refresh Policy (none) + ...

Compile Options Policy (none) + ...

XSL Cache Size 256 stylesheets

SHA1 Caching  on  off

Static Document Call  on  off

XSLT Expression Optimization  on  off

Load Balance Groups (empty) Add + ...

User Agent Configuration sftp\_policy + ... \*

## Configure User Agent

Compression Policy Header-Retention Policy Restrict to HTTP 1.0 Policy Inject Header Policy Chunked Uploads Policy FTP Client Policies SFTP Client Policies

User Agent: sftp\_policy [up]

Apply Cancel Undo

Export View Log View Status Help

### SFTP Client Policies

URL Matching Expression	SSH client profile	Use unique file names
sftp://*	SSH_Client_VDF	on

Add

Administrative State  enabled  disabled

Comments

User name sqa \*

User Authentication  Public Key  Password \*

User Private Key idrsa + ... \*

Password ..... \*

Persistent Connections  on  off

Persistent Connection Idle Timeout 120 seconds \*

Strict Host Key Checking  on  off

The User Agent used by the XML Manager provides the necessary credentials to connect to the destination SFTP server.



# Scenario 4: SFTP Server FSH

## Configure Multi-Protocol Gateway Style Policy

**Policy:**  
 Policy Name:  \*  
 Apply Policy Cancel Export View Log

**Rule:**  
 Rule Name:  Rule Direction: Client to Server  
 New Rule Delete Rule

Create rule: Click New, drag action icons onto line. Edit rule: Click on rule, double-click on action

Filter Sign Verify Validate Encrypt Decrypt Transform Route AAA Results Advanced

Create Reusable Rule

Order	Rule Name	Direction	Actions
↑↓	sftp-virtu-route_rule_0	Client to Server	Filter, Sign, Verify, Validate, Encrypt, Decrypt, Transform, Route, AAA, Results, Advanced
↑↓	sftp-virtu-route_return_rule_1	Server to Client	Filter, Sign, Verify, Validate, Encrypt, Decrypt, Transform, Route, AAA, Results, Advanced

The Gateway dynamically routes messages through the use of a stylesheet executed by the Route action.

**Input**  
 Input:  INPUT \*

**Options**  
**Route (Using Stylesheet or XPath Expression)**

Selection Method  
 Use Stylesheet to Select Destination  
 Use Variable to Select Destination  
 Use XPath to Select Destination  
 \*

Processing Control  
 File:  Upload... Fetch... Edit... View...

Asynchronous  on  off

**Output**  
 Output:  OUTPUT

Delete Done Cancel



## Scenario 4: SFTP Server FSH

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fn="http://www.w3.org/2005/xpath-functions" xmlns:xdt="http://www.w3.org/2005/xpath-datatypes"
xmlns:dp="http://www.datapower.com/extensions" xmlns:dpconfig="http://www.datapower.com/param/config"
extension-element-prefixes="dp" exclude-result-prefixes="dp dpconfig">
  <xsl:output method="xml"/>
  <xsl:template match="/">
    <xsl:variable name="inboundurl">
      <xsl:value-of select="dp:variable('var://service/routing-url')" />
      <!-- http://9.30.12.17/inbound/987654.xml -->
    </xsl:variable>
    <xsl:variable name="filen" select="substring-after($inboundurl,'inbound/')" />
    <xsl:variable name="fulluri">sftp://9.70.155.68/home/sqa/poller/<xsl:value-of select="$filen" />
    </xsl:variable>
    <dp:set-variable name="'var://service/routing-url'" value="$fulluri" />

    <dp:set-variable name="'var://context/uservars/resultURL'" value="$fullurl"/>
    <xsl:message dp:priority="alert" dp:type="mpgw">
      <xsl:value-of select="$fullurl"/>
    </xsl:message>

  </xsl:template>
</xsl:stylesheet>
```

This stylesheet captures the filename sent by the client and uses it to place the file on the back end server. The remote client never sees the files on the destination server.

# Scenario 4: SFTP Server FSH

## Configure SFTP Server Front Side Handler

Main Allowed Commands Virtual Directories

SFTP Server Front Side Handler: sftp-server-wizard [up]

Apply Cancel Undo

Administrative State  enabled  disabled

Comments

Local IP Address   \*

Port Number  \*

Access Control List

Host Private Keys

User Authentication  Public Key  Password \*

AAA Policy

Filesystem Type

Default Directory  \*

Idle Timeout  seconds

## Configure SFTP Server Front Side Handler

Main Allowed Commands Virtual Directories

SFTP Server Front Side Handler: sftp-server-wizard [up]

Apply Cancel Undo

Virtual Directories

Virtual Directory	
/inbound	<input type="button" value="✎"/> <input type="button" value="✕"/>
	<input type="button" value="Add"/>

This Server FSH uses a virtual ephemeral filesystem, obscuring the file/directory structure of the destination server. This server also authenticates clients by using a public key exchange.

# Scenario 4: SFTP Server FSH

## Configure AAA Policy

Main Identity Authenticate MapCredentials Resource MapResource Authorize

AAA Policy: PubKey [up]

Apply Cancel Undo

Methods

- HTTP Authentication Header
- Password-carrying UsernameToken Element from WS-Security Header
- Derived-key UsernameToken Element from WS-Security Header
- BinarySecurityToken Element from WS-Security Header
- WS-SecureConversation Identifier
- WS-Trust Base or Supporting Token
- Kerberos AP-REQ from WS-Security Header
- Kerberos AP-REQ from SPNEGO Token
- Subject DN of the SSL Certificate from the Connection Peer
- Name from SAML Attribute Assertion
- Name from SAML Authentication Assertion
- SAML Artifact
- Client IP Address
- Subject DN from Certificate in the Message's signature
- Token Extracted from the Message
- Token Extracted as Cookie Value
- LTPA Token
- Processing Metadata
- Custom Template
- HTML Forms-based Authentication

Processing Metadata Items

ssh-password-metadata + ...

## Configure AAA Policy

Main Identity Authenticate MapCredentials Resource MapResource Authorize

AAA Policy: PubKey [up]

Apply Cancel Undo

Method Custom Template \*

Custom URL local:///sftp\_aaa\_validate2.xml \*

Cache authentication results Absolute \*

Cache Lifetime 3 Seconds

The AAA Policy used by the FSH identifies clients by examining the Processing Metadata, which contains the public key provided by the remote client.

To Authenticate the client, the AAA Policy uses a custom stylesheet.





## Scenario 4: SFTP Server FSH

```
<!-- Get EI metadata information from datapower, store it to variables, log it,
      and authenticate user/pass or user/pubkey combinations -->
<xsl:variable name="ssh_user">
  <xsl:value-of select="dp:variable('var://context/INPUT/ssh/username')"/>
</xsl:variable>
<xsl:variable name="ssh_cert">
  <xsl:value-of select="dp:variable('var://context/INPUT/ssh/publickey')"/>
</xsl:variable>
<xsl:variable name="ssh_pass">
  <xsl:value-of select="dp:variable('var://context/INPUT/ssh/password')"/>
</xsl:variable>
<xsl:message>The SFTP User is: <xsl:value-of select="$ssh_user"/>
</xsl:message>
```

The custom stylesheet begins by retrieving the identification information provided by the Metadata.

## Scenario 4: SFTP Server FSH

```
<!-- This function performs validation the public key. -->
<xsl:template name="validatePublicKey">
  <xsl:param name="user"/>
  <xsl:param name="pubkey"/>
  <xsl:choose>
    <xsl:when test="($pubkey = 'ssh-rsa AAAAB...xnaKQ==')">
      <publickey_authenticator user="FITS">Passed</publickey_authenticator>
    </xsl:when>
    <xsl:when test="($pubkey = 'ssh-dss
AAAAB3NzaC1kc3MAAAEBAKtnTISZHRc7....zyR5w=')">
      <publickey_authenticator user="SABRE">Passed</publickey_authenticator>
    </xsl:when>
    <xsl:when test="($pubkey = 'ssh-rsa AAAA...Z22c0=')">
      <publickey_authenticator user="NEWTESTER">Passed</publickey_authenticator>
    </xsl:when>
    <xsl:otherwise/>
  </xsl:choose>
</xsl:template>
```

The public key provided by the remote client is then compared the the public key held by the device. (Note that the key material is truncated here.) Rather than embed this key material in the stylesheet, the stylesheet could open an independent file to get the key materials.

# Scenario 4: SFTP Server FSH

## Additional Considerations

In some cases, the back end application returns a result (as would be the case for an HTTP transaction, for example) which must be returned to the front side client. In such a case, the device automatically places the result file in the same virtual directory as the request file. Front side clients using FTP can then retrieve the result from the virtual directory system.

The example stylesheet shown here only authenticates by public key. The stylesheet could also authenticate by username/password, requiring all information from the remote client.



# Summary

- DataPower offers both Poller and Server methods for obtaining FTP-based messages
- DataPower can easily connect FTP-based traffic to other protocols
- The full power of message handling is available
- DataPower can proxy back end S/FTP servers with a high degree of security
- FTP support continues to evolve and improve



# Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:  
[http://www.ibm.com/software/websphere/support/supp\\_tech.html](http://www.ibm.com/software/websphere/support/supp_tech.html)
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:  
<http://www.ibm.com/developerworks/websphere/community/>
- Join the Global WebSphere Community:  
<http://www.websphereusergroup.org>
- Access key product show-me demos and tutorials by visiting IBM® Education Assistant:  
<http://www.ibm.com/software/info/education/assistant>
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:  
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:  
<http://www.ibm.com/software/support/einfo.html>

# We Want to Hear From You!

## Tell us about what you want to learn

Suggestions for future topics  
Improvements and comments about our webcasts  
We want to hear everything you have to say!

Please send your suggestions and comments to:  
[wsehelp@us.ibm.com](mailto:wsehelp@us.ibm.com)

# Questions and Answers

