

TIBCO iWay[®] Service Manager

FTP Solutions Development Guide

Version 8.0 and Higher

March 2021

DN3502290.0321



Contents

1. Introducing iWay FTP Services (Adapters)	7
iWay FTP Services Overview	7
Understanding Differences Between FTP and SFTP	7
What Is FTP?.....	7
Implementing Security.....	8
Secure FTP.....	8
What Is SFTP?.....	10
Capabilities of SFTP.....	10
iWay Service Manager Suite of FTP Tools	11
Client Suite.....	11
Sample Applications.....	12
Server Suite.....	13
Sample Applications.....	13
Common Listener Functionality.....	13
Available Listeners Reference	14
Available Emitters Reference	14
Available Services Reference	14
2. Configuring FTP and FTPS Components	17
FTP and FTPS Component Configuration Overview	17
Supported FTP Components.....	17
Configuring a FTP Listener	19
Troubleshooting the FTP Listener.....	24
Configuring a FTP Emitter	24
Configuring a FTP Read Document Service	27
Configuring a FTP File Emit Service	32
Configuring a FTP File Operations for Process Flows Service	38
Configuring a FTP Direct Transfer to Disk Service	45
Configuring a FTP Connection Cache Service	50
FTP Connection Cache Service Example.....	55
Configuring a FTP Directory Contents Service	56
Configuring a FTP File Read for Process Flows Service	60

3. Configuring Secure FTP (SFTP) Components 67

SFTP Component Configuration Overview	67
Password Authentication Versus Key Pair Based Authentication.	68
Supported Secure FTP Components.	68
Configuring a SFTP Listener	69
Configuring a SFTP Emitter	75
Configuring a SFTP Read Service	76
SFTP Read Service Output Example.	80
Configuring a SFTP Directory Contents Service	80
Configuring a SFTP Direct Transfer Service	83
Configuring a SFTP Connection Cache Service	87
Configuring a SFTP Emit Service	90
Configuring a SFTP File Ops Service	94
Configuring OpenSSH on Windows	115
The /home Directory.	118
Firewalls.	118
Converting a Private Key to the OpenSSH Key Format	118

4. Configuring FTP Server Components121

FTP Server Component Configuration Overview	121
Configuring a FTP Server Listener	122
Action on GET.	126
Action on PUT.	129
Process Flow Fails in FTP Server.	132
The Security File.	133
Permission Flags and What They Mean.	138
RDBMS User Configuration Tables.	140
iwftp_owners.	140
iwftp_permissions.	142
iwftp_user_hosts.	143
iwftp_templates.	143
Using FTP Permission Tables.	143
User Home Locations.	146

The SITE Command.....	146
FTP Server Log.....	146
FTP Commands	147
Security Considerations	156
Using the Trading Partner Management Facility	156
Configuring a FTP SREG Service	157
5. Configuring Secure FTP (SFTP) Server Components	163
SFTP Server Component Configuration Overview	163
Configuring a SFTP Listener	164
Action on GET.....	166
Action on PUT.....	167
Process Flow Fails in SFTP Server.....	167
Configuring the SSH Server Security Provider	168
A. Common Configuration Parameters	173
Listener Configuration Parameters	173
B. Configuring iWay Service Manager Components	177
Configuring Listeners	177
Configuring Services	181
Legal and Third-Party Notices	187

Introducing iWay FTP Services (Adapters)

This section provides an introduction to iWay Service Manager (iSM) FTP services. For more information on additional queuing protocol adapters that are supported by iSM, see the *iWay Service Manager Protocol Guide*.

In this chapter:

- ❑ [iWay FTP Services Overview](#)
 - ❑ [Understanding Differences Between FTP and SFTP](#)
 - ❑ [iWay Service Manager Suite of FTP Tools](#)
 - ❑ [Available Listeners Reference](#)
 - ❑ [Available Emitters Reference](#)
 - ❑ [Available Services Reference](#)
-

iWay FTP Services Overview

iWay FTP services provide tools that simplify the development and implementation of applications where file transport services are required. The simplification is accomplished by reducing the requirement for custom programming when implementing a range of file transport services and by providing access to packaged third-party adapter products. By minimizing the amount of code required for their implementation, these FTP services provide a solid foundation for flexible service-oriented architecture.

Understanding Differences Between FTP and SFTP

Before you begin using the FTP suite of tools that are available in iWay Service Manager (iSM), it is recommended that you have a good understanding of File Transfer Protocol (FTP), and which specific type of FTP you plan on supporting.

What Is FTP?

File Transfer Protocol (FTP) is a standard network protocol used to transfer files from one host to another host over TCP-based networks, such as the Internet.

FTP is built on a client-server architecture model and uses separate control and data connections between the client and the server. FTP users can authenticate themselves using a clear text sign-in protocol, normally in the form of a user name and password, but can also connect anonymously if the server is configured to allow this. To implement secure transmission that hides (encrypts) the user name and password, and encrypts the content, FTP is often secured with SSL/TLS (known as FTPS).

Implementing Security

FTP was not originally designed to be a secure protocol and has many security weaknesses. In May 1999, the authors of RFC 2577 listed a vulnerability to the following problems:

- ☐ Brute force attacks
- ☐ Bounce attacks
- ☐ Packet capture (sniffing)
- ☐ Port stealing
- ☐ Spoof attacks
- ☐ User name protection

FTP is not able to encrypt its traffic. All transmissions are in clear text. In addition, user names, passwords, commands, and data can be easily read by anyone who is able to perform packet capture (sniffing) on the network. This problem is common to many of the Internet Protocol specifications (such as SMTP, Telnet, POP, and IMAP) that were designed prior to the creation of encryption mechanisms such as Transport Layer Security (TLS) or Secure Sockets Layer (SSL). A common solution to this problem is to use the *secure*, TLS-protected versions of the insecure protocols (for example, FTPS for FTP, TelnetS for Telnet, and so on) or a different, more secure protocol that can handle the job, such as the SFTP/SCP tools included with most implementations of the Secure Shell protocol.

Secure FTP

There are several methods available for transferring files securely that have been called *Secure FTP* at one point or another:

- ☐ FTPS
- ☐ SFTP
- ☐ FTP Over SSH (Not SFTP)

FTPS

Explicit FTPS is an extension to the FTP standard that allows clients to request that the FTP session be encrypted. This is done by sending the *AUTH TLS* command. The server has the option of allowing or denying connections that do not request TLS. This protocol extension is defined in the proposed standard (RFC 4217). Implicit FTPS is a deprecated standard for FTP that required the use of a SSL or TLS connection. It was specified to use different ports other than plain FTP.

SFTP

SFTP, the *SSH File Transfer Protocol*, is not related to FTP except that it also transfers files and has a similar command set for users. SFTP, or secure FTP, is a program that uses Secure Shell (SSH) to transfer files. Unlike standard FTP, SFTP encrypts commands and data, which prevents passwords and sensitive information from being transmitted openly across a network. SFTP is functionally similar to FTP, but because it uses a different protocol, standard FTP clients cannot be used to talk to an SFTP server, nor can one connect to an FTP server with a client that supports only SFTP.

For more information about SFTP, see [What Is SFTP?](#) on page 10.

FTP Over SSH (Not SFTP)

FTP over SSH (not SFTP) refers to the practice of tunneling a normal FTP session over an SSH connection. Because FTP uses multiple TCP connections (unusual for a TCP/IP protocol that is still in use), it is particularly difficult to tunnel over SSH. With many SSH clients, attempting to set up a tunnel for the control channel (the initial client-to-server connection on port 21) will protect only that channel. When data is transferred, the FTP software at either end will set up new TCP connections (data channels), which bypass the SSH connection and thus have no confidentiality or integrity protection, and so on.

Otherwise, it is necessary for the SSH client software to have specific knowledge of the FTP protocol, to monitor and rewrite FTP control channel messages and autonomously open new packet forwarding for FTP data channels. Software packages that support this mode include, but are not limited to:

- ☐ Tectia ConnectSecure (Windows/Linux/Unix) of the SSH Communications Security software suite
- ☐ Tectia Server for IBM z/OS of the SSH Communications Security software suite
- ☐ FONC (the GPL licensed)
- ☐ Co:Z FTPSSH Proxy

FTP over SSH is sometimes referred to as secure FTP. This should not be confused with other methods of securing FTP, such as SSL/TLS (FTPS). Other methods of transferring files using SSH that are not related to FTP include SFTP and SCP. In each of these, the entire conversation (credentials and data) is always protected by the SSH protocol.

What Is SFTP?

SSH File Transfer Protocol (sometimes referred to as Secure File Transfer Protocol, or SFTP) is a network protocol that provides file access, file transfer, and file management functionalities over any reliable data stream. It was designed as an extension of the Secure Shell protocol (SSH) version 2.0 to provide secure file transfer capability, but is also intended to be used with other protocols.

This protocol assumes the following:

- ☐ It is being run over a secure channel, such as SSH.
- ☐ The server has already authenticated the client.
- ☐ The identity of the client user is available to the protocol.

Capabilities of SFTP

Compared to the earlier version of SCP protocol, which allowed only file transfers, the SFTP protocol allows for a range of operations on remote files, similar to a remote file system protocol. The extra capabilities of an SFTP client, compared to an SCP client, include resuming interrupted transfers, directory listings, and remote file removal.

SFTP attempts to be more platform-independent than SCP. For example, with SCP, the expansion of wildcards specified by the client depends with the server, whereas the design of SFTP avoids this problem. While SCP is most frequently implemented on UNIX platforms, SFTP servers are commonly available on most platforms.

SFTP is not FTP run over SSH, but rather a new protocol uniquely designed by the IETF SECSSH working group. It is sometimes confused with Simple File Transfer Protocol.

Note that the protocol itself does not provide authentication and security. It expects the underlying protocol to secure this. SFTP is most often used as a subsystem of SSH protocol version 2 implementations, having been designed by the same working group. However, it is possible to run it over SSH-1 (and some implementations support this) or other data streams. Running SFTP server over SSH-1 is not platform independent, as SSH-1 does not support the concept of subsystems. An SFTP client willing to connect to an SSH-1 server must know the path to the SFTP server binary on the server side.

For uploads, the transferred files may be associated with their basic attributes, such as timestamps. This is a unique feature over the common FTP protocol, which does not have the provision for uploads to include the original date and timestamp attribute.

iWay Service Manager Suite of FTP Tools

iWay Service Manager (iSM) offers a full complement of FTP services that are designed specifically to be used in transactional situations. These services include client and server-side, RFC-compliant capabilities for FTP, FTPS, and SFTP.

This section describes features in the iSM FTP suite of tools for the client side and the server side of the FTP/SFTP suite.

Client Suite

The FTP client suite allows iSM to connect to and interact with an FTP, FTPS, or SFTP server. The client is capable of detecting the existence of a file on the remote server and reacting to this existence (through an iSM FTP, FTPS, or SFTP listener).

The reaction of the client based on the existence of a file could be as simple as retrieving the file. It can also involve a complex iSM process flow that includes a series of steps that can manipulate that file or a group of files on the remote system or other systems that iSM is capable of accessing.

The following components are currently available in the FTP client suite of tools:

- ❑ Two listeners used to detect and react to the existence of a file on the server:
 - ❑ FTP/FTPS Client Listener
 - ❑ SFTP Client Listener
- ❑ Two services (agents) to read a file from the server:
 - ❑ FTP/FTPS Read Service (com.ibi.agents.XDNFTPReadAgent)
 - ❑ SFTP Read Service (com.ibi.agents.XDSFTPReadAgent)
- ❑ Two services (agents) to write (emit) a file to the server:
 - ❑ FTP/FTPS Emit Service (com.ibi.agents.XDNFTPEmitAgent)
 - ❑ SFTP Emit Service (com.ibi.agents.XDSFTPEmitAgent)

- ☐ Two services (agents) that perform various file-based operations (exists, delete, size, and so on) on a file(s) located on the server:
 - ☐ FTP/FTPS File Ops Service (com.ibi.agents.XDNFTPFileOpsAgent)
 - ☐ SFTP File Ops Service (com.ibi.agents.XDSFTPFileOpsAgent)
- ☐ Two services (agents) that list the contents of server directories:
 - ☐ FTP/FTPS Directory Listing Service (com.ibi.agents.XDNFTPDirListAgent)
 - ☐ SFTP Directory Listing Service (com.ibi.agents.XDSFTPDirListAgent)
- ☐ Two services (agents) to enable a process flow to share connections between the client services and the FTP server:
 - ☐ FTP/FTPS Connection Control Service (com.ibi.agents.XDNFTPConnectionCacheAgent)
 - ☐ SFTP Connection Control Service (com.ibi.agents.XDSFTPConnectionCacheAgent)
- ☐ Service (agent) to present context Special Registers (SREGs) to a remote iSM server (com.ibi.agents.XDNFTPSREGAgent).

Sample Applications

The FTP client suite of tools can be used to handle a variety of application requirements, such as:

- ☐ The FTP/SFTP listener can be configured to monitor a directory on the server.
 - ☐ When a specific file name (and/or extension) is written to that directory, that file will be retrieved and saved to a directory on the client.
 - ☐ Alternately, the file that is retrieved can be presented to an iSM process flow and used as the input for additional processing by iSM (for example, updating a database table and initiating an SAP transaction).
- ☐ When used in conjunction with other iSM listeners, the FTP/SFTP clients can be used within an iSM process flow to transfer data from the client system to the server.
- ☐ A messaging listener such as MSMQ or JMS can be used to take a message from a queue. The message itself or specific message contents can be transferred to the server.
- ☐ An SAP IDoc that is obtained using the iWay Application Adapter for mySAP ERP (SAP JCo 3.x) could be sent to the server.

- ❑ Sending an EDI file to a VAN.

Server Suite

The FTP server suite of tools provides two listeners:

- ❑ FTP/FTPS Server Listener
- ❑ SFTP Server Listener

Both listeners have the same characteristics in common. The listeners can be configured to interact with the client in the same way that any FTP/SFTP server would. For example, handling the request of the client (send, receive, rename, delete, and so on) as the server. The advantage of the FTP server suite of tools is the ability to configure the listeners to use the file(s) that are received as messages that can be used to initiate a complex iSM process flow. Corresponding results are then returned from those process flows.

Security

The FTP server suite of tools can be configured to handle login security using standard server authentication realms (LDAP, RDBMS, text based property file, Kerberos, and so on) as well as a complete directory authorization capability and user role tracking.

Sample Applications

The FTP server suite of tools can be used to handle a variety of application requirements, such as:

- ❑ Receiving transaction messages (including EDI) from partners who need to use a standard protocol, and then passing these messages to a process flow. This is a good solution for participating in standard EDI networks by EDI splitters and transformers.
- ❑ Receiving files and using a process flow to redistribute the incoming files to one or more internal recipients.
- ❑ Act as a relay for large files where not having the actual file materialized on a local disk is required. This can include the receipt of a file outside of a firewall and then relaying the file through the firewall.

Common Listener Functionality

All listeners offer the standard server startup failure flows to handle processing issues when the listener begins and when specific error conditions occur during the operation of the listener.

Available Listeners Reference

The following table provides a quick reference to the iSM listeners that are defined in this documentation for FTP services.

Listener Name
FTP Listener (See Configuring a FTP Listener on page 19)
SFTP Listener (See Configuring a SFTP Listener on page 69)
FTP Server Listener (See Configuring a FTP Server Listener on page 122)

Available Emitters Reference

The following table provides a quick reference to the iSM emitters that are defined in this documentation for FTP services.

Emitter Name
FTP Emitter (See Configuring a FTP Emitter on page 24)
SFTP Emitter (See Configuring a SFTP Emitter on page 75)

Available Services Reference

The following table provides a quick reference to the iSM services that are defined in this documentation for FTP services.

Service Name
FTP Read Document (com.ibi.agents.XDNFTPReadAgent) See Configuring a FTP Read Document Service on page 27.
FTP File Emit (com.ibi.agents.XDNFTPEmitAgent) See Configuring a FTP File Emit Service on page 32.

Service Name

FTP File Operations for Process Flows (com.ibi.agents.XDNPFFTPFileOpsAgent)

See [Configuring a FTP File Operations for Process Flows Service](#) on page 38.

FTP Direct Transfer to Disk (com.ibi.agents.XDNFTPDirectFileTransfer)

See [Configuring a FTP Direct Transfer to Disk Service](#) on page 45.

FTP Connection Cache (com.ibi.agents.XDNFTPConnectionCacheAgent)

See [Configuring a FTP Connection Cache Service](#) on page 50.

FTP Directory Contents (com.ibi.agents.XDNFTPDirListAgent)

See [Configuring a FTP Directory Contents Service](#) on page 56.

FTP File Read for Process Flows (com.ibi.agents.XDNPFFTPReadAgent)

See [Configuring a FTP File Read for Process Flows Service](#) on page 60.

FTP SREG (com.ibi.agents.XDNFTPSREGAgent)

See [Configuring a FTP SREG Service](#) on page 157.

SFTP Read (com.ibi.agents.XDSFTPReadAgent)

See [Configuring a SFTP Read Service](#) on page 76.

SFTP Directory Contents (com.ibi.agents.XDSFTPDirListAgent)

See [Configuring a SFTP Directory Contents Service](#) on page 80.

SFTP Direct Transfer (com.ibi.agents.XDSFTPDirectFileTransfer)

See [Configuring a SFTP Direct Transfer Service](#) on page 83.

SFTP Connection Cache (com.ibi.agents.XDSFTPConnectionCacheAgent)

See [Configuring a SFTP Connection Cache Service](#) on page 87.

SFTP Emit (com.ibi.agents.XDSFTPEmitAgent)

See [Configuring a SFTP Emit Service](#) on page 90.

SFTP File Ops (com.ibi.agents.XDSFTPFileOpsAgent)

See [Configuring a SFTP File Ops Service](#) on page 94.

Configuring FTP and FTPS Components

This section describes how to configure FTP and FTPS components using iWay Service Manager.

In this chapter:

- ☐ [FTP and FTPS Component Configuration Overview](#)
 - ☐ [Configuring a FTP Listener](#)
 - ☐ [Configuring a FTP Emitter](#)
 - ☐ [Configuring a FTP Read Document Service](#)
 - ☐ [Configuring a FTP File Emit Service](#)
 - ☐ [Configuring a FTP File Operations for Process Flows Service](#)
 - ☐ [Configuring a FTP Direct Transfer to Disk Service](#)
 - ☐ [Configuring a FTP Connection Cache Service](#)
 - ☐ [Configuring a FTP Directory Contents Service](#)
 - ☐ [Configuring a FTP File Read for Process Flows Service](#)
-

FTP and FTPS Component Configuration Overview

The iWay FTP/S Server extends the File Transfer Protocol (FTP) with the ability to treat FTP GET and PUT commands as iWay Service Manager transaction messages. These transaction messages can be then passed through the FTP channels and handled by any process flow configured to those channels. This enables the use of FTP as a transaction and messaging protocol.

Supported FTP Components

The supported FTP components available in iWay Service Manager are:

- ☐ **FTP Listener.** A listener that uses the FTP protocol component and is continuously polling the specified folder on the FTP server.
- ☐ **FTP Emitter.** The FTP Emitter will emit messages onto a FTP server. It requires the credentials on the server and the directory to emit as input.

- ❑ **Services.** The following are the types of FTP services available:
 - ❑ **FTP File Read for Process Flows (`com.ibi.agents.XDNPFFTPReadAgent`).** The FTP File Read for Process Flows service is used to read files from a FTP server (drive on UNIX or Windows). It can also be used in tandem with a file listener to embed file contents (the file picked up by the listener) into the XML file read from the FTP drive by specifying the tag. Generally used within process flows, this agent is also available to be configured as a standalone service.
 - ❑ **FTP Read Document (`come.ibi.agents.XDNFTPReadAgent`).** The FTP Read Document service is used to read files from a FTP server (drive on UNIX or Windows). It can also be used in tandem with a file listener to embed file contents (the file picked up by the listener) into the XML file read from the FTP drive by specifying the tag.
 - ❑ **FTP File Emit (`com.ibi.agents.XDNFTPEmitAgent`).** The FTP File Emit service is used to write files to an output directory through FTP (drive on UNIX or Windows). The output file name can be specified completely or using wildcard characters.
 - ❑ **FTP File Operations for Process Flows (`com.ibi.agents.XDNPFFTP0psAgent`).** The FTP Operations (Ops) for Process Flows service emits data to a given *host:port* using various common FTP commands. It can be used to perform operations, such as Copy, Prepend, Append, Size, and Move. Generally used within process flows, this agent is also available to be configured as a standalone service.
 - ❑ **FTP Directory Contents (`com.ibi.agents.XDNFTPDirListAgent`).** The FTP Directory Contents service gathers file information from the requested directory and returns a document listing the files and/or directories from an FTP server (drive on UNIX, or Windows).
 - ❑ **FTP Direct File Transfer to Disk (`com.ibi.agents.XDNFTPDirectFileTransfer`).** The FTP Direct File Transfer to Disk service transfers a file directly from the iSM to an FTP server (drive on UNIX, or Windows), or transfers a file directly from the FTP server to the iSM server, without moving the file through the iSM process flow.
 - ❑ **FTP Connection Cache (`com.ibi.agents.XDNFTPConnectionCacheAgent`).** The FTP Connection Cache service caches a single FTP connection (both control and data channels) to an FTP server. When the connection cache is started, an existing connection from the cache is utilized for an FTP agent within the iSM process flow that connects to an FTP server with the same address and authentication credentials (user ID and password).

Configuring a FTP Listener

To configure a FTP listener:

1. Perform the steps as described in [Configuring Listeners](#) on page 177.
2. Ensure that you select *FTP[S] Client* as the listener type to configure.

For a complete description of the configuration parameters that are available for the FTP listener, see [FTP Listener Configuration Parameters](#) on page 19.

For a complete description of the FTP listener Special Registers (SREGs), see [FTP Listener Special Registers](#) on page 23.

Reference: FTP Listener Configuration Parameters

The following table lists and describes parameters for the FTP listener.

Note: Parameters that are common to FTP listeners are described in [Listener Configuration Parameters](#) on page 173.

Property	Description
Host Name (required)	The FTP host to be accessed to search for the document.
User Name (required)	The user ID on the FTP host.
Password (required)	The user password on the FTP host.
Account Name	The name of the account on the FTP server. This is optional, depending on the FTP server.
Input Path	The directory on the FTP host from which to retrieve files. Specify the file name or use a DOS-style pattern. Do not use suffix <i>in</i> .
Destination Directory	The directory on FTP host to return response to.
Append	If the destination file exists, then append on PUT.
Pending Directory	The local directory where pending requests are held.

Property	Description
Suffix In	Limits input files to those with the same extension, for example, xml. Do not insert a period (.) before the suffix. A dash (-) indicates no extension, that is, the field is not used.
Duration	The maximum time that a document can remain in the retry pending queue.
Retry	The interval between retrying pending requests.
Quote Command	The entered command is sent as typed, BEFORE any data transfer begins. Following login to the FTP server, a command is sent to the server, if it is configured. This is often used to accomplish a secondary login.
Mode	In ASCII mode, conversions are done (EBCDIC to ASCII and vice versa). In BINARY mode, the data remains unchanged.
Payload Type (required)	<p>Determines what form of payload should be processed. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> Data. The file contents are passed. This is the default.</p> <p><input type="checkbox"/> Signal. A signal document containing the file name is passed. The file contents are not read into memory.</p> <p>Note: If Signal is set, then a value for the Local Store Directory parameter must be provided.</p>
Local Store Directory	The directory on the iWay server where files are saved. You must specify a directory path only if the Payload Type parameter is set to Signal.
Remove locally stored files	If set to <i>true</i> , the FTP listener deletes the file stored locally after it is processed. This parameter requires a directory path to be specified for the Local Store Directory parameter.
Do not unzip ZIP files	<p>Pass ZIP files as a single file for processing (requires ACCEPT FLAT turned on).</p> <p>The default is to unzip ZIP files and process the contents of the ZIP files individually.</p>

Property	Description
Socket Timeout	The value in seconds before the socket connection times out. With this option set to a non-zero timeout, a read() call on the Socket will block for only this amount of time. If the timeout expires, a java.net.SocketTimeoutException is raised.
Use Passive Command	If set to <i>true</i> , then use the PASV command. If set to <i>false</i> , then use the PORT command.
Bad File List	If set to <i>true</i> , then maintain a list of files with errors, preventing them from being re-accessed. If set to <i>false</i> , files will not be retried.
Delete After Read	Some FTP servers, such as VANS, automatically delete the file after read. In this case set this to false. If the file is not deleted, it will be reread at each FTP cycle.
File Protect	Emits a temporary name and then renames it to the desired name.
Security	
Secure Control Connection	<p>If set to <i>true</i>, then the user ID and password are transferred in a secure manner. If client authentication is required, you may be required to configure the keystore under the HTTPS section of the system properties.</p> <p>Note: If the keystore is configured in the system properties, ensure it has the CA certificate or the client certificate of the server to which you are connecting.</p> <p>If the keystore is not configured in the system properties, the default truststores under /lib/security/cacerts is used.</p>

Property	Description
SSL Security	<p>Select one of the following FTP server connection types from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> unknown. Defaults to explicit security then fails over to implicit security. <input type="checkbox"/> explicit. In order to establish the SSL link, explicit security requires that the FTP client issue a specific command to the FTP server after establishing a connection. The default FTP server port is used. <input type="checkbox"/> implicit. Implicit security automatically begins with an SSL connection as soon as the FTP client connects to a FTP server. In implicit security, the FTP server defines a specific port for the client (typically 990) to be used for secure connections.
Keystore Security Provider	Enter the iWay Keystore Security Provider name. If the component is secure and Keystore Security Provider is left blank, the default Keystore Security Provider is used.
Secure Data Connection	If set to <i>true</i> , transfers data in a secure manner. Used in conjunction with Secure Control Connection.
Use 128-bit Encryption	If set to <i>true</i> , enforces use of 128-bit encryption.
Security Protocol	<p>The protocol to enable security. Security protocol values include:</p> <ul style="list-style-type: none"> <input type="checkbox"/> SSL. Supports some version of SSL. May support other versions. <input type="checkbox"/> SSLv2. Supports SSL version 2 or higher. <input type="checkbox"/> SSLv3. Supports SSL version 3. May support other versions. <input type="checkbox"/> TLS. Supports some version of TLS. May support other versions. <input type="checkbox"/> TLSv1. Supports TLS version 1. May support other versions.
SITE	
Starting SITE command	The SITE command to issue before the transfer of data.

Property	Description
Successful SITE command	The SITE command to issue after the successful transfer of data.
Error SITE command	The SITE command to issue if the transfer of data fails.

Note: The FTP listener supports streaming. Streaming is used for large documents or for documents where the application needs to split the input into sections under the same transaction. For more information on streaming and configuring streaming preparers, see the *iWay Service Manager Component and Functional Language Reference Guide*.

Reference: FTP Listener Special Registers

The following table lists and describes the Special Registers (SREGs) available on the FTP listener.

Name	Level	Type	Description
iwayconfig	System	String	The current active configuration name.
iwayhome	System	String	The base at which the server is loaded.
iwayworkdir	System	String	The path to base of the current configuration.
msgsize	Document	Integer	The physical length of the message payload.
name	System	String	The assigned name of the master (listener).
protocol	System	String	The protocol on which the message was received.
source	Document	String	The full name of the input file.
basename	Document	String	The file name without an extension.
extension	Document	String	The extension to the file name (mime type).
filename	Document	String	The file name using the basename.extension format.
parent	Document	String	The path to the file name.

Name	Level	Type	Description
tid	Document	String	Unique transaction ID.

Troubleshooting the FTP Listener

The following table describes an error that you may encounter when using the FTP listener.

Error	Reason	Solution
Unable to successfully log in to FTP: [line 3]	System path was unable to log on to the FTP host with the supplied host, user ID, and password combination.	Verify that the host, user ID, and password combination is correct.

Configuring a FTP Emitter

Messages are sent to particular destinations at the completion of a workflow. The state of the document determines which destination is used. The order in which the destinations are used cannot be predicted.

Note: Configuring a FTP emitter is not required if the outlet (emitter) protocol is the same as the inlet (listener) protocol.

To route an output document or error message to a protocol other than that of the outlet (listener) used, you must configure an emitter. For example, if an application can send input over MQSeries, but you want to route the output to a FTP destination.

Reference: FTP Emitter Parameters

The following table lists and describes parameters for the FTP emitter.

Property	Description
Destination (required)	The file@host. If an asterisk (*) is entered, a time stamp is substituted for the file name.
User Name (required)	The user ID on the FTP host.
Password (required)	The user password on the FTP host.

Property	Description
Account Name	Some FTP servers require an ACCT command as part of their login credential exchange. If configured, this information is sent by the ACCT command when the login is attempted.
Mode	In ASCII mode, conversions are done (EBCDIC to ASCII and vice versa). In BINARY mode, the data remains unchanged.
Socket Timeout	The time, in seconds, in which a read() call on the socket blocks. If the timeout expires, a java.net.SocketTimeoutException is raised.
Rename To	The name to give the uploaded file after the upload is complete. Use an asterisk (*) in the name to be replaced by time stamp and # by a sequential counter, for example: <code>*.msg.someFolder/###.msg</code>
Append	If a destination file exists, the information is appended to that file. Otherwise, a new file is stored.
Quote Command	The entered command is sent as typed, BEFORE any data transfer begins. Following login to the FTP server, if this is configured it is a command to be sent to the server. Often this is used to accomplish a secondary login.
File Protect	Emits a temporary name and then renames it to the desired name.
Starting SITE command	The SITE command to issue before the transfer of data.
Successful SITE command	The SITE command to issue after the successful transfer of data.
Error SITE command	The SITE command to issue if the transfer of data fails.
Security	

Property	Description
Secure Control Connection	<p>Use a secure control connection, for example, transfer user ID and password securely. Data transfer will not be secured. You may need to configure the keystore under the HTTPS section of the system properties if client authentication is required.</p> <p>Note: If the keystore is configured in system properties, make sure it has the CA certificate or the client certificate of the server to which you are connecting. If the keystore is not configured in system properties, the default truststore located under <JRE_HOME>/lib/security/cacerts is used.</p>
SSL Security	<p>Select one of the following FTP server connection types from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> unknown. Defaults to explicit security then fails over to implicit security. <input type="checkbox"/> explicit. In order to establish the SSL link, explicit security requires that the FTP client issue a specific command to the FTP server after establishing a connection. The default FTP server port is used. <input type="checkbox"/> implicit. Implicit security automatically begins with an SSL connection as soon as the FTP client connects to a FTP server. In implicit security, the FTP server defines a specific port for the client (typically 990) to be used for secure connections.
Keystore Security Provider	Enter the iWay Keystore Security Provider name. If this component is secure and Keystore Security Provider is left blank, the default Keystore Security Provider is used.
Secure Data Connection	This determines whether to use a secure data connection (transferring data securely). This is used in conjunction with Secure Control Connection.
Use 128-Bit Encryption	This enforces the use of 128-bit encryption.

Property	Description
Security Protocol	<p>The security protocol. Available options are:</p> <ul style="list-style-type: none"> <input type="checkbox"/> SSL. Supports some version of SSL. May support other versions. <input type="checkbox"/> SSLv2. Supports SSL version 2 or higher. <input type="checkbox"/> SSLv3. Supports SSL version 3. May support other versions. <input type="checkbox"/> TLS. Supports some version of TLS. May support other versions. <input type="checkbox"/> TLSv1. Supports TLS version 1. May support other versions.

Configuring a FTP Read Document Service

The FTP Read Document service is used to read files from a FTP server.

To configure a FTP Read Document service:

1. Perform the steps described in [Configuring Services](#) on page 181.
2. Ensure that you select *FTP Read Document {com.ibi.agents.XDNFTPReadAgent}* as the service type you are configuring.

For a complete description of the configuration parameters that are available for the FTP Read Document service, see [FTP Read Document Service Parameters](#) on page 27.

For a complete description of the edges that are returned by the FTP Read Document service, see [FTP Read Document Service Edges](#) on page 31.

Reference: FTP Read Document Service Parameters

The following table lists and describes parameters for the FTP Read Document service.

Parameter	Description
Host Parameters	
Host Name	In this field, enter the DNS name (or IP address) of the FTP server that you wish to connect to. Use the host port if the standard port is not 21.

Parameter	Description
Remote Port	This is the port to connect to on the FTP site. Leave it blank for default port 21.
User Name	Name used as the valid user ID on the FTP server.
Password	The valid password for the FTP server.
Account Name	The valid account for the FTP server.
Use Passive Command	If set to <i>true</i> , the service uses a PASV command. Otherwise, it uses the PORT command.
Timeout	Timeout interval for socket in seconds.
Retry Interval	Retry interval in seconds (allows xxhxxmxxs format). You can omit or use 0 for no retry.
Connection Retry	This shows the number of attempted failed connections to the FTP server.
Service Parameters	
File Name Tag	Name of the tag from the input document in which to find the file name.
Enclose Tag	This parameter is the name of the tag that encloses data read. If omitted, no tagging of the data is done. If used, the output is an XML document. If the Transfer Mode is binary and the Enclose Tag is specified, the base64 Encoding should be selected or else the user risks getting an error when the resulting XML document is parsed.
Base Path	Optional directory to be used if the incoming name is not absolute. The user can use this parameter to specify a directory entry that will be combined with the File Name, obtained from the File Name Tag, to create a path to the file on the FTP server.

Parameter	Description
Input Data Format	<p>This parameter is the format of the input data. The default setting is flat.</p> <ul style="list-style-type: none"> <input type="checkbox"/> flat. The data is transferred from the FTP server as a flat unformatted document. <input type="checkbox"/> XML. The data is transferred from the FTP server as a text formatted document. The data that is transmitted is assumed to be a valid XML document and is parsed. If the document is not valid then an error is returned by the service.
Transfer Mode	<p>This is a form of FTP transmission. Select one of the following modes:</p> <ul style="list-style-type: none"> <input type="checkbox"/> ascii. The file is retrieved as text from the FTP server. The data received by iSM is translated into text, based on the code page configuration of iSM. <input type="checkbox"/> binary. The file is retrieved as a binary block. No text translation is performed by iSM.
Encoding	<p>This parameter is the character set encoding to be performed on input. Select one of the following options:</p> <ul style="list-style-type: none"> <input type="checkbox"/> asis. The data from the FTP server is not translated. <input type="checkbox"/> base64. The data from the FTP server is converted into a base 64 XML compatible string.
Delete After Read	<p>Use this parameter if you wish to delete the file after the read.</p>
SSL Parameters	
Use SSL	<p>If set to <i>true</i>, the connection is secured using Secure Sockets Layer (SSL).</p>

Parameter	Description
Security Protocol	<p>This shows the type of security protocol to be used. The following list describes the options of the security protocol.</p> <ul style="list-style-type: none"> <input type="checkbox"/> SSL. This protocol supports some versions of SSL, and may also support other versions. <input type="checkbox"/> SSLv2. This protocol supports SSL version 2 or higher. <input type="checkbox"/> SSLv3. This protocol supports SSL version 3, and may support other versions. <input type="checkbox"/> TLS. This protocol supports some versions of TLS, and may also support other versions. <input type="checkbox"/> TLSv1. This protocol supports TLS version 1, and may support other versions. <p>This field is not needed if Keystore is a SSL Provider.</p>
Secure Data Connection	<p>This is used to enable a secure data connection, for example. transfer data securely. It is used in conjunction with Secure Control Connection.</p>
Use 128-bit Encryption	<p>This parameter enforces the use of 128-bit encryption.</p>
SSL Security	<p>This parameter describes the FTP Server connection type. Select one of the following options:</p> <ul style="list-style-type: none"> <input type="checkbox"/> unknown. This setting defaults to Explicit Security then fails over to Implicit Security. <input type="checkbox"/> explicit. In order to establish the SSL link, explicit security requires that the FTP client issue a specific command to the FTP server after establishing a connection. The default FTP server port is used. <input type="checkbox"/> implicit. Implicit security automatically begins with an SSL connection as soon as the FTP client connects to an FTP server. In implicit security, the FTP server defines a specific port for the client (typically 990) to be used for secure connections.

Parameter	Description
Keystore File or Keystore Security Provider	<p>In this field, you can:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Enter the full path to the Keystore file, which provides certificate material to be used for SSL connection. <input type="checkbox"/> Name the Keystore Security Provider. <input type="checkbox"/> Use the configured default Keystore Security Provider by leaving it blank.
Keystore Password	This field is used to enter the password to access Keystore File. This is not required if Keystore File or Keystore Security Provider is the name of a Keystore Security Provider.
Keystore Type	This field shows the type of the Keystore. It is not needed if Keystore File or Keystore Security Provider is the name of a Keystore Security Provider.

Reference: FTP Read Document Service Edges

The following table lists and describes the edges that are returned by the FTP Read Document service.

Edge	Description
success	Operation completed successfully.
fail_parse	Failed to properly parse the input parameters of the service.
fail_connect	<p>Failed to connect to FTP host for any one of the following reasons:</p> <ul style="list-style-type: none"> <input type="checkbox"/> The host name (IP) is invalid. <input type="checkbox"/> The User ID is invalid. <input type="checkbox"/> The password of the user is invalid. <input type="checkbox"/> The connection failed.
fail_operation	Invalid parameters or other error.

Configuring a FTP File Emit Service

The FTP File Emit service transfers data from iWay Service Manager to a FTP server.

To configure a FTP Emit service:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select *FTP File Emit {com.ibi.agents.XDNFTPEmitAgent}* as the service type you are configuring.

For a complete description of the configuration parameters that are available for the FTP File Emit service, see [FTP File Emit Service Parameters](#) on page 32.

For a complete description of the edges that are returned by the FTP File Emit service, see [FTP File Emit Service Edges](#) on page 38.

Reference: FTP File Emit Service Parameters

The following table lists and describes parameters for the FTP File Emit service.

Parameter	Description
Host Parameters	
Host Name*	In this field, enter the DNS name (or IP address) of the FTP server that you wish to connect to. Use the host port if the standard port is not 21.
Remote Port	This is the port to connect to on the FTP site. Leave it blank for default port 21.
User Name*	Name used as the valid user ID on the FTP server.
Password*	The valid password for the FTP server.
Account Name	The valid account for the FTP server.
Use Passive Command	If set to <i>true</i> , the service uses a PASV command. Otherwise, it uses the PORT command.
Timeout	Timeout interval for socket in seconds.
Retry Interval	Retry interval in seconds (allows xxhxxmxxs format). You can omit or use 0 for no retry.

Parameter	Description
Connection Retry	This shows the number of attempted failed connections to the FTP server.
Agent Parameters	
Input Source	<p>Allows you to configure the source of the input. Select one of the following input source values from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Document. Input source is from the document, which is passed on the edge of the process flow. This input source (Document) is also the default value. <input type="checkbox"/> External File. Input source is from an external source (MFT use). If selected, then you must configure the Input Expression parameter to specify an external source. <input type="checkbox"/> Stream. Uses the stream from the configuration of the listener as the input source. If the listener can be configured to pass the document as a stream (or uses a preparer that results in a stream being passed), then this value allows that stream to be emitted.
Input Expression	If the input source is external, specify the file name here.
Remote Site Folder	Folder or directory on the FTP site that you want to use as a starting location when you connect. If you leave it blank, the login directory is used.
File Pattern	<p>This shows the output file pattern (* = timestamp). For example, (*.xml, *.txt, and so on).</p> <p>Note: *.* is unsupported.</p>

Parameter	Description
Behavior When Target File Exists	<p>The action to be performed if the file exists on the FTP server. Select one of the following options:</p> <ul style="list-style-type: none"><input type="checkbox"/> overwrite. The existing file on the server will be replaced with input.<input type="checkbox"/> append. Input will be added to the end of the existing file.<input type="checkbox"/> resume. If server supports REST and SIZE commands, iSM will determine the size of the file on the server using the SIZE command, and then send the REST command with the current file size. The input data is then sent starting at this position. The resume option is only supported for binary mode transfers.<input type="checkbox"/> fail. Generate a failure document and exit.
Quote Command	<p>The entered command is sent as typed, before any data transfer.</p>
Transfer Mode*	<p>This mode is a form of FTP Transmission. Choose one of the following options:</p> <ul style="list-style-type: none"><input type="checkbox"/> ascii. The input is sent as text to the FTP server. The data received by the server is translated into text, based on the code page configuration of the server.<input type="checkbox"/> binary. The input is sent as a binary block to the server. No text translation is performed on the server.

Parameter	Description
Put File Protection	<p>Determines whether the PUT parameter is protected by a rename of a temporary file name. Choose one of the following options:</p> <ul style="list-style-type: none"> <input type="checkbox"/> true. If set to <i>true</i>, the file is written to a temporary file name. When the transfer of data is complete, the temporary file on the server is renamed to the final file name <input type="checkbox"/> false. If set to <i>false</i>, the existing file contents (if applicable) are directly replaced with the input data.
Emit Zero Bytes	Allow the Emitter to transmit zero bytes to the server. This flag will be ignored if the document to be emitted is a stream.
Return	<p>Select what the service returns when execution completes:</p> <ul style="list-style-type: none"> <input type="checkbox"/> status. A status document will be the out document. This document reflects the status of the transfer to the FTP server. <input type="checkbox"/> input. The input document becomes the out document.
Proxy Settings	
Proxy Type	<p>Select one of the following proxy protocol values to use from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> NONE <input type="checkbox"/> HTTP <input type="checkbox"/> SOCKS4 <input type="checkbox"/> SOCKS5
Proxy Host Name	DNS name (or IP address) of the Proxy server that you want to connect to.
Proxy Port	Port to connect to on the Proxy site.

Parameter	Description
Proxy User ID	User ID on the Proxy server.
Password	The password of the proxy user on the Proxy server.
SSL Parameters	
Use SSL	If set to <i>true</i> , the connection is secured using Secure Sockets Layer (SSL).
Security Protocol	<p>This shows the type of security protocol to be used. The following list describes the options of the security protocol.</p> <ul style="list-style-type: none"> <input type="checkbox"/> SSL. This protocol supports some versions of SSL, and may also support other versions. <input type="checkbox"/> SSLv2. This protocol supports SSL version 2 or higher. <input type="checkbox"/> SSLv3. This protocol supports SSL version 3, and may support other versions. <input type="checkbox"/> TLS. This protocol supports some versions of TLS, and may also support other versions. <input type="checkbox"/> TLSv1. This protocol supports TLS version 1, and may support other versions. <p>This field is not needed if Keystore is a SSL Provider.</p>
Secure Data Connection	This is used to enable a secure data connection, such as transferring data securely. It is used in conjunction with Secure Control Connection.
Use 128-bit Encryption	This parameter enforces the use of 128-bit encryption.

Parameter	Description
SSL Security	<p>This parameter describes the FTP Server connection type. Select one of the following options:</p> <ul style="list-style-type: none"> <input type="checkbox"/> unknown. This setting defaults to Explicit Security then fails over to Implicit Security. <input type="checkbox"/> explicit. In order to establish the SSL link, explicit security requires that the FTP client issue a specific command to the FTP server after establishing a connection. The default FTP server port is used. <input type="checkbox"/> implicit. Implicit security automatically begins with an SSL connection as soon as the FTP client connects to an FTP server. In implicit security, the FTP server defines a specific port for the client (typically 990) to be used for secure connections.
Keystore File or Keystore Security Provider	<p>In this field, you can:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Enter the full path to the Keystore file, which provides certificate material to be used for SSL connection. <input type="checkbox"/> Name the Keystore Security Provider. <input type="checkbox"/> Use the configured default Keystore Security Provider by leaving it blank.
Keystore Password	This field is used to enter the password to access Keystore File. This is not required if Keystore File or Keystore Security Provider is the name of a Keystore Security Provider.
Keystore Type	This field shows the type of the Keystore. It is not needed if Keystore File or Keystore Security Provider is the name of a Keystore Security Provider.
SITE Parameters	
Starting SITE command	The SITE command to issue before the transfer of data.
Successful SITE command	The SITE command to issue after the successful transfer of data.

Parameter	Description
Error SITE Command	The SITE command to issue if the transfer of data fails.

Reference: FTP File Emit Service Edges

The following table lists and describes the edges that are returned by the FTP File Emit service.

Edge	Description
success	Operation completed successfully.
fail_parse	Failed to properly parse the input parameters of the service.
fail_connect	Failed to connect to FTP host for any one of the following reasons: <ul style="list-style-type: none"> <input type="checkbox"/> The host name (IP) is invalid. <input type="checkbox"/> The User ID is invalid. <input type="checkbox"/> The password of the user is invalid. <input type="checkbox"/> The connection failed.
fail_operation	Invalid parameters or other error.
fail_duplicate	Failed to duplicate the message.

Configuring a FTP File Operations for Process Flows Service

The FTP File Operations for Process Flows service is used to perform simple operations on the FTP server based on parameters provided by an XML input document.

To configure a FTP File Operations for Process Flows service:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select *FTP File Operations for Process Flows* {com.ibi.agents.XDNPFFTPOpsAgent} as the service type you are configuring.

For a complete description of the configuration parameters that are available for the FTP File Operations for Process Flows service, see [FTP File Operations for Process Flows Service Parameters](#) on page 39.

For a complete description of the edges that are returned by the FTP File Operations for Process Flows service, see [FTP File Operations for Process Flows Service Edges](#) on page 45.

Reference: FTP File Operations for Process Flows Service Parameters

The following table lists and describes parameters for the FTP File Operations for Process Flows service.

Parameter	Description
Host Parameters	
Host Name (required)	In this field, enter the DNS name (or IP address) of the FTP server that you wish to connect to. Use the host port if the standard port is not 21.
Remote Port	This is the port to connect to on the FTP site. Leave it blank for default port 21.
User Name (required)	Name used as the valid user ID on the FTP server.
Password (required)	The valid password for the FTP server.
Account Name	The valid account for the FTP server.
Use Passive Command	If set to <i>true</i> , the service uses a PASV command. Otherwise, it uses the PORT command.
Timeout	Timeout interval for socket in seconds.
Agent Parameters	

Parameter	Description
Operation (required)	<p>Operation to perform on the file hosted by the SFTP Server. Operations supported by this service are as follows:</p> <ul style="list-style-type: none"> <input type="checkbox"/> copy. Copies the data from the file addressed by the File (from) parameter to the file named in the File (to) parameter. <input type="checkbox"/> move. Moves the data from the file addressed by the File (from) parameter to the file named in the File (to) parameter. When successfully completed the file addressed by the File (from) parameter is deleted. <input type="checkbox"/> rename. Renames the file addressed by the File (from) parameter to the file named in the File (to) parameter. When successfully completed the file addressed by the File (from) no longer exists. <input type="checkbox"/> prepend. Copies the data from the file addressed by the File (from) parameter to the beginning of the file named in the File (to) parameter.
Operation (continued)	<ul style="list-style-type: none"> <input type="checkbox"/> append. Copies the data from the file addressed by the File (from) parameter to the end of the file named in the File (to) parameter. <input type="checkbox"/> delete. Deletes the file addressed by the File (from) parameter from the host. <input type="checkbox"/> size. Gets the size of the file addressed by the File (from) parameter from the host. The return is places in the Special Register named in the Remote Size parameter. <input type="checkbox"/> exist. Verifies that the file addressed by the File (from) parameter exists on the host.
File (from) (required)	Name of the source file. This field may be a relative or absolute file paths, a SREG or XPath expression. This is a required field.

Parameter	Description
File (to)	The name of the destination file. Wild cards are accepted. This is a required field except when operation is delete, size, or exist.
File (to) a directory name	References a directory. For more information on this parameter, see the description and example that follows this table.
File (to) Create Directories	Creates a directory if one does not exist. For more information on this parameter, see the description and example that follows this table.
Size	Name of the Special Register designated to hold size. This field is required when operation is size.
Out Document (required)	<p>Specify the document to be returned by the operation (bad input defaults to <i>result</i>).</p> <p>Selecting <i>result</i> returns the results of the requested operation. In the case of copy, move, rename, delete, size, and exist, the status document containing the status of the function is returned.</p> <p>The functions <i>prepend</i> and <i>append</i> result in the file data being returned. This data will be the same as the data found in the file addressed by the File (to) parameter.</p>
Action on Failure (required)	Determines whether the input document or status document is returned on failure.
Retry	If non-zero, the operation will be retried <i>n</i> times at one-second intervals.
SSL Parameters	
Use SSL	If set to <i>true</i> , the connection is secured using Secure Sockets Layer (SSL).

Parameter	Description
Security Protocol	<p>This shows the type of security protocol to be used. The following list describes the options of the security protocol.</p> <ul style="list-style-type: none"> <input type="checkbox"/> SSL. This protocol supports some versions of SSL, and may also support other versions. <input type="checkbox"/> SSLv2. This protocol supports SSL version 2 or higher. <input type="checkbox"/> SSLv3. This protocol supports SSL version 3, and may support other versions. <input type="checkbox"/> TLS. This protocol supports some versions of TLS, and may also support other versions. <input type="checkbox"/> TLSv1. This protocol supports TLS version 1, and may support other versions. <p>This field is not needed if Keystore is a SSL Provider.</p>
Secure Data Connection	<p>This is used to enable a secure data connection, such as transferring data securely. It is used in conjunction with Secure Control Connection.</p>
Use 128-bit Encryption	<p>This parameter enforces the use of 128-bit encryption.</p>
SSL Security (required)	<p>This parameter describes the FTP Server connection type. Select one of the following options:</p> <ul style="list-style-type: none"> <input type="checkbox"/> unknown. This setting defaults to Explicit Security then fails over to Implicit Security. <input type="checkbox"/> explicit. In order to establish the SSL link, explicit security requires that the FTP client issue a specific command to the FTP server after establishing a connection. The default FTP server port is used. <input type="checkbox"/> implicit. Implicit security automatically begins with an SSL connection as soon as the FTP client connects to an FTP server. In implicit security, the FTP server defines a specific port for the client (typically 990) to be used for secure connections.

Parameter	Description
Keystore File or Keystore Security Provider	<p>In this field, you can:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Enter the full path to the Keystore file, which provides certificate material to be used for SSL connection. <input type="checkbox"/> Name the Keystore Security Provider. <input type="checkbox"/> Use the configured default Keystore Security Provider by leaving it blank.
Keystore Password	This field is used to enter the password to access Keystore File. This is not required if Keystore File or Keystore Security Provider is the name of a Keystore Security Provider.
Keystore Type	This field shows the type of the Keystore. It is not needed if Keystore File or Keystore Security Provider is the name of a Keystore Security Provider.

File (to) a directory name Parameter

The *File (to) a directory name* parameter references a directory.

File (to) a directory name	<p>File (to) references a directory, if it is unclear whether path names a directory or a filename, the Service Manager will assume the path names a file.</p> <div> <input type="text" value="false"/> </div> <div> <input type="text" value="Pick one"/> </div>
----------------------------	---

The default value for this parameter is *false* and iSM will use the path specified in the File (to) parameter to reference a file path. If the File (to) a directory name parameter is set to *true*, then iSM will use the File (to) parameter to reference a directory path and the file created will have the same file name as the File (from) parameter.

Example:

The File (from) parameter has the file name `temp/output.xml`, the File (to) parameter has the name `prod/final`, and the File (to) a directory name parameter is set to *true*. The results can be found in the file `output.xml` in the directory `prod/final`. Otherwise, if the File (to) a directory name parameter is set to *false* (default), the results will be found in the file `final` in the directory `prod`.

Note: iSM supports the creation of dynamic File (to) file names using special iSM file name patterns using a combination of the following three characters (`#*^`). These characters are only allowed when the File (to) parameter is a file name (File (to) a directory name parameter is set to *false*). If the File (to) a directory name parameter is set to *true* and the parameter contains one of the iSM pattern characters (`#*^`), an error occurs.

Additionally, iSM's pattern control file is saved in the file directory:


```
[iwayworkdir]/ftpdata/File (to) parent directory
```

For example, if using the FTP Ops service and the File (to) parameter is set to `prod/ism####.xml`, then the pattern control file would be located as follows:

```
[iwayworkdir]/ftpdata/prod/.ism####.xml
```

File (to) Create Directories Parameter

The *File (to) Create Directories* parameter creates a directory if one does not exist.

File (to) Create Directories	Create if directory doesn't exist. Used only for Copy, Move, Rename and Append operations.
<input type="text" value="false"/>	
<input type="button" value="Pick one"/> 	

iSM now supports dynamic creation of the directory tree. The default value for the *File (to) Create Directories* parameter is *false*. If set to *false*, the directory structure is expected to already be in place, and if not, an error is returned. If set to *true* however, iSM will attempt to create the directory structure defined by the File (to) parameter. If successful, the full tree structure as defined in the File (to) parameter will be created *before* the function is performed.

Example:

The File (to) parameter is set to `prod/final/f0001.xml`. iSM checks for the existence of the directory *prod*. If *prod* does not exist, then *prod* is created. Next the directory *final* is checked. If *final* does not exist, then *final* is created and so on until the directory structure is complete. Once the directory structure is in place the service executes the configured function.

Note: If an attempt to create the tree structure fails (due to an error being returned from the remote system), some part of the tree structure may have been created. It is the responsibility of the user to determine the correct course of action to stabilize the directory structure of the remote system.

Reference: FTP File Operations for Process Flows Service Edges

The following table lists and describes the edges that are returned by the FTP File Operations for Process Flows service.

Edge	Description
success	Operation completed successfully.
fail_parse	Failed to properly parse the input parameters of the service.
fail_connect	Failed to connect to FTP host for any one of the following reasons: <ul style="list-style-type: none"> <input type="checkbox"/> The host name (IP) is invalid. <input type="checkbox"/> The User ID is invalid. <input type="checkbox"/> The password of the user is invalid. <input type="checkbox"/> The connection failed.
fail_operation	Invalid parameters or other error.

Configuring a FTP Direct Transfer to Disk Service

The FTP Direct Transfer to Disk service is used to read or write files directly from a FTP server to iWay Service Manager (iSM).

If you select *send* as a value for the Transfer Type parameter, then the file is moved from the directory specified in the Working Directory parameter to the FTP server. If you select *transfer* as a value for the Transfer Type parameter, then the file is moved from the FTP server directory specified in the Host Directory parameter, to the Working Directory in iSM.

To configure a FTP Direct Transfer to Disk service:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select *FTP Direct transfer to disk {com.ibi.agents.XDNFTPDirectFileTransfer}* as the service type you are configuring.

For a complete description of the configuration parameters that are available for the FTP Direct Transfer to Disk service, see [FTP Direct Transfer to Disk Service Parameters](#) on page 46.

For a complete description of the edges that are returned by the FTP Direct Transfer to Disk service, see [FTP Direct Transfer to Disk Service Edges](#) on page 50.

Reference: FTP Direct Transfer to Disk Service Parameters

The following table lists and describes parameters for the FTP Direct Transfer to Disk service.

Parameter	Description
Configuration parameters for FTP Direct transfer to disk service	
Host Name (required)	The DNS name (or IP address) of the FTP server that you want to connect to. Use the <i>host:port</i> format if the standard port is not 21.
Remote Port	The port to connect to on the FTP site. Leave it blank for default port 21.
User Name (required)	A valid user ID for the FTP server.
Password (required)	A valid password for the FTP server.
Account Name	A valid account for the FTP server.
Use Passive	<p>If set to <i>true</i>, a PASV command is used. Otherwise, the PORT command is used. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> true. Use PASV.</p> <p><input type="checkbox"/> false. Use Active.</p>
Timeout	The timeout interval (in seconds) for the socket.
Retry Interval	The retry interval (in seconds) for the socket. The <i>xxhxxmxxs</i> format can be used for this value. Specify a value of zero (0) for no retry interval.
Connection Retry	The number of retry attempts that were made after failed connections to the FTP server.
Agent Parameters	

Parameter	Description
Name of File (required)	This is the file to be read. A relative or absolute file path is supported explicitly or through a SREG or XPath expression that is evaluated using the incoming document.
Host Directory	An optional directory to be used if the name of the file is not absolute.
Transfer Mode (required)	<p>When files are transferred in ASCII mode, the transferred data is expected to contain only character-formatted text. Binary mode refers to transferring files as a binary stream of data. Where ASCII mode may use special control characters to format data, Binary mode transmits the raw bytes of the file being transferred. In this way, the file is transferred in its exact original form. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> ascii. The file is a text-based document.</p> <p><input type="checkbox"/> binary. Data is transferred as is and without textual evaluation.</p>
Working Directory (required)	The path to the iSM working directory.
Transfer Type (required)	<p>The direction of file transfer. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> retrieve. Retrieves the file from the FTP server.</p> <p><input type="checkbox"/> send. Sends the file from iSM to the FTP server.</p>
Return 'status'	<p>Determines the return document status. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> status. Creates a status document for the return.</p> <p><input type="checkbox"/> input. Returns the inbound document as the output</p>

Parameter	Description
Delete After Read	<p>Determines whether to delete the file after the read. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> true. Deletes the file.</p> <p><input type="checkbox"/> false. Leaves the file as is.</p>
Action on Failure	<p>Determines whether the input document or status document is returned on failure. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> status. Creates a status document for the return.</p> <p><input type="checkbox"/> input. Returns the inbound document.</p>
SSL Parameters	
Use SSL	<p>If set, the connection is secured through the Secure Sockets Layer (SSL) protocol. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> true. Use SSL.</p> <p><input type="checkbox"/> false. Do not use SSL.</p>
Security Protocol	<p>Determines the security protocol to be used. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> SSL. Supports some version of SSL.</p> <p><input type="checkbox"/> SSLv2. Supports SSL version 2.</p> <p><input type="checkbox"/> SSLv3. Supports SSL version 3.</p> <p><input type="checkbox"/> TLS. Supports some version of TLS.</p> <p><input type="checkbox"/> TLSv1. Supports TLS version 1.</p>

Parameter	Description
Secure Data Connection	<p>Use a secure data connection (for example, to transfer data securely). This parameter is used in conjunction with <i>Secure Control Connection</i>. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> true. Use SSL with the data connection. <input type="checkbox"/> false. Do not use SSL with the data connection.
Use 128-bit Encryption	<p>Enforces the use of 128-bit encryption. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> true. Use 128-bit encryption. <input type="checkbox"/> false. Do not use 128-bit encryption.
SSL Security (required)	<p>Determines the FTP server connection type. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Unknown. Initially attempts Explicit Security then fails over to Implicit Security. (default). <input type="checkbox"/> Explicit. In order to establish the SSL link, Explicit Security requires that the FTP client issue a specific command to the FTP server after establishing a connection. The default FTP server port is used. <input type="checkbox"/> Implicit. Implicit security automatically begins with an SSL connection as soon as the FTP client connects to an FTP server. In implicit security, the FTP server defines a specific port for the client (typically 990) to be used for secure connections.
Keystore File or Keystore Security Provider	<p>Keystore file or keystore security provider full path to the keystore file, which provides certificate material to be used for a SSL connection. Specify the name of a keystore security provider, or leave blank to use the configured default keystore security provider.</p>

Parameter	Description
Keystore Password	The password to access the Keystore file. This value is not required if the keystore file or keystore security provider is the name of a keystore security provider.
Keystore Type	The type of keystore. This value is not required if the keystore file or keystore security provider is the name of a keystore security provider.

Reference: FTP Direct Transfer to Disk Service Edges

The following table lists and describes the edges that are returned by the FTP Direct Transfer to Disk service.

Edge	Description
success	Operation completed successfully.
fail_parse	Failed to properly parse the input parameters of the service.
fail_connect	Failed to connect to FTP host for any one of the following reasons: <ul style="list-style-type: none"> <input type="checkbox"/> The host name (IP) is invalid. <input type="checkbox"/> The User ID is invalid. <input type="checkbox"/> The password of the user is invalid. <input type="checkbox"/> The connection failed.
fail_operation	Invalid parameters or other error.

Configuring a FTP Connection Cache Service

The FTP Connection Cache service caches a single FTP connection (both control and data channels) to an FTP server. When the connection cache is started, an existing connection from the cache is utilized for an FTP agent within the iSM process flow that connects to an FTP server with the same address and authentication credentials (user ID and password)

To configure a FTP Connection Cache service:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select *FTP Connection Cache* `{com.ibi.agents.XDNFTPConnectionCacheAgent}` as the service type you are configuring.

For a complete description of the configuration parameters that are available for the FTP Connection Cache service, see [FTP Connection Cache Service Parameters](#) on page 51.

For a complete description of the edges that are returned by the FTP Connection Cache service, see [FTP Connection Cache Service Edges](#) on page 54.

Reference: FTP Connection Cache Service Parameters

The following table lists and describes parameters for the FTP Connection Cache service.

Parameter	Description
Configuration parameters for FTP Direct transfer to disk service	
Host Name (required)	The DNS name (or IP address) of the FTP server that you want to connect to. Use the <i>host:port</i> format if the standard port is not 21.
Remote Port	The port to connect to on the FTP site. Leave it blank for default port 21.
User Name (required)	A valid user ID for the FTP server.
Password (required)	A valid password for the FTP server.
Account Name	A valid account for the FTP server.
Use Passive Command	<p>If set to <i>true</i>, a PASV command is used. Otherwise, the PORT command is used. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> true. Use PASV.</p> <p><input type="checkbox"/> false. Use Active.</p>
Timeout	The timeout interval (in seconds) for the socket.

Parameter	Description
Retry Interval	The retry interval (in seconds) for the socket. The <i>xxhxxmxxs</i> format can be used for this value. Specify a value of zero (0) for no retry interval.
Connection Retry	The number of retry attempts that were made after failed connections to the FTP server.
Agent Parameters	
Connection Caching (required)	<p>If set to <i>start</i>, the referenced connection is cached until stopped. If set to <i>stop</i>, the cached connection is closed. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> start. Starts the connection caching for this thread.</p> <p><input type="checkbox"/> stop. Stops the connection caching on this thread.</p>
SSL Parameters	
Use SSL	<p>If set, the connection is secured through the Secure Sockets Layer (SSL) protocol. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> true. Use SSL.</p> <p><input type="checkbox"/> false. Do not use SSL.</p>
Security Protocol	<p>Determines the security protocol to be used. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> SSL. Supports some version of SSL.</p> <p><input type="checkbox"/> SSLv2. Supports SSL version 2.</p> <p><input type="checkbox"/> SSLv3. Supports SSL version 3.</p> <p><input type="checkbox"/> TLS. Supports some version of TLS.</p> <p><input type="checkbox"/> TLSv1. Supports TLS version 1.</p>

Parameter	Description
Secure Data Connection	<p>Use a secure data connection (for example, to transfer data securely). This parameter is used in conjunction with <i>Secure Control Connection</i>. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> true. Use SSL with the data connection. <input type="checkbox"/> false. Do not use SSL with the data connection.
Use 128-bit Encryption	<p>Enforces the use of 128-bit encryption. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> true. Use 128-bit encryption. <input type="checkbox"/> false. Do not use 128-bit encryption.
SSL Security (required)	<p>Determines the FTP server connection type. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Unknown. Initially attempts Explicit Security then fails over to Implicit Security. (default). <input type="checkbox"/> Explicit. In order to establish the SSL link, Explicit Security requires that the FTP client issue a specific command to the FTP server after establishing a connection. The default FTP server port is used. <input type="checkbox"/> Implicit. Implicit security automatically begins with an SSL connection as soon as the FTP client connects to an FTP server. In implicit security, the FTP server defines a specific port for the client (typically 990) to be used for secure connections.
Keystore File or Keystore Security Provider	<p>Keystore file or keystore security provider full path to the keystore file, which provides certificate material to be used for a SSL connection. Specify the name of a keystore security provider, or leave blank to use the configured default keystore security provider.</p>

Parameter	Description
Keystore Password	The password to access the Keystore file. This value is not required if the keystore file or keystore security provider is the name of a keystore security provider.
Keystore Type	The type of keystore. This value is not required if the keystore file or keystore security provider is the name of a keystore security provider.

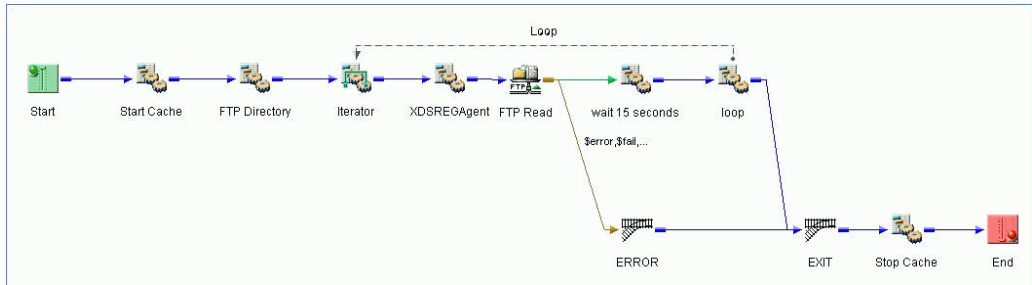
Reference: FTP Connection Cache Service Edges

The following table lists and describes the edges that are returned by the FTP Connection Cache service.

Edge	Description
success	Operation completed successfully.
fail_parse	Failed to properly parse the input parameters of the service.
fail_connect	<p>Failed to connect to FTP host for any one of the following reasons:</p> <ul style="list-style-type: none"> <input type="checkbox"/> The host name (IP) is invalid. <input type="checkbox"/> The User ID is invalid. <input type="checkbox"/> The password of the user is invalid. <input type="checkbox"/> The connection failed.
fail_operation	Invalid parameters or other error.

FTP Connection Cache Service Example

The following is an example Process flow for the FTP Connection Cache service.



ISM Object	Description
Start	Start of process flow.
Start Cache	Calls the XDNFTPConnectionCacheAgent for the server to connect to.
FTP Directory	XDNFTPDirListAgent uses the connection that was created in Start Cache to create a document containing the files within a directory.
Iterator	Loops through the document returned by FTP Directory using the expression /dir/item to retrieve each entry in the document.
XDSREGAgent	Creates Special Registers with the File name and Directory extracted via XPATH from the document segment created by the Iterator.
FTP Read	XDNPFFTPReadAgent uses the connection that was created in Start Cache to read the file from the directory.
wait 15 seconds	Uses the XDCopyAgent to delay processing for 15 seconds simulates processing that may occur.
loop	Loops back to the Iterator to get next file value.
ERROR	If XDNPFFTPReadAgent is unsuccessful, the FTP Read exits (\$error, \$fail, etc...)

ISM Object	Description
EXIT	Called after either the ERROR, or loop completes.
Stop Cache	Calls the XDnFTPConnectionCacheAgent to close the connection to the server.
End	Terminates the process flow.

Configuring a FTP Directory Contents Service

The FTP Directory Contents service is used to generate an XML document listing the contents of a FTP directory specified by the user in the Directory parameter. The XML document that is generated has the following pattern:

```
<dir base="Directory" count="number">
<item type="Item Type">Name</item>
</dir>
```

The content of the document differs based on the Include parameter configured by the user. If the user selects Files, then the item element will contain only file names from the directory. If the user selects Subdirectories, then only the subdirectory names contained in the Directory are included. If the user selects All, then file and subdirectory names are included.

Output

The XML document generated by the service contains only two elements (dir and item). The element dir is the root element and contains two attributes (name and count). The name attribute contains the Directory entry that was configured by the user. The count attribute contains the number of item elements to follow.

The item element is the only child of the dir element, but can have multiple item siblings. The item element has only one attribute (type). The type attribute identifies the type of name that this item element contains. If the type attribute is set to file, then this item element contains a name of a file. If the type attribute is set to directory, then this item element contains a name of a subdirectory.

To configure a FTP Directory Contents service:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select FTP Directory Contents {com.ibi.agents.XDnFTPDirListAgent} as the service type you are configuring.

For a complete description of the configuration parameters that are available for the FTP Directory Contents service, see [FTP Directory Contents Service Parameters](#) on page 57.

Reference: FTP Directory Contents Service Parameters

The following table lists and describes parameters for the FTP Directory Contents service.

Parameter	Description
Configuration parameters for FTP Directory Contents service	
Host Name (required)	The DNS name (or IP address) of the FTP server that you want to connect to. Use the host:port format if the standard port is not 21.
Remote Port	The port to connect to on the FTP site. Leave it blank for default port 21.
User Name (required)	A valid user ID for the FTP server.
Password (required)	A valid password for the FTP server.
Account Name	A valid account for the FTP server.
Use Passive Command	<p>If set to <i>true</i>, a PASV command is used. Otherwise, the PORT command is used. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> true. Use PASV.</p> <p><input type="checkbox"/> false. Use Active.</p>
Timeout	The timeout interval (in seconds) for the socket.
Retry Interval	The retry interval (in seconds) for the socket. The <i>xxhxxmxxs</i> format can be used for this value. Specify a value of zero (0) for no retry interval.
Connection Retry	The number of retry attempts that were made after failed connections to the FTP server.
Agent Parameters	

Parameter	Description
Directory	What directory should be listed. This is the name of the directory on the FTP server that is to be listed. If the directory does not exist on the server the agent will return on a failure edge.
Include	<p>What items should be included in the listing. Select from the dropdown list what items that should be included in the listing:</p> <ul style="list-style-type: none"> <input type="checkbox"/> File: File entries only. <input type="checkbox"/> Sub-directory: Directory entries only. <input type="checkbox"/> All: Both files and directory entries.
Selection Expression	Regular expression (or filter) used to select files and directories.
Pattern Type	<p>If a value for the Selection Expression parameter is supplied, then this parameter indicates whether it should be interpreted as a regular expression or a DOS style wildcard. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Regular Expression. Java-based Regular Expression syntax. <input type="checkbox"/> DOS-style Wildcard. DOS style wild card syntax. A question mark (?) indicates any single number or character. An asterisk (*) indicates any combination of numbers and characters.
Call EOD	<p>When using a streaming preparer in a channel, a last call is made after the last document. Determines whether this service exit should be called. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> true. Calls the service exit. <input type="checkbox"/> false. Does not call the service exit.

Parameter	Description
SSL Parameters	
Use SSL	<p>If set, the connection is secured through the Secure Sockets Layer (SSL) protocol. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> true. Use SSL.</p> <p><input type="checkbox"/> false. Do not use SSL.</p>
Security Protocol	<p>Determines the security protocol to be used. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> SSL. Supports some version of SSL.</p> <p><input type="checkbox"/> SSLv2. Supports SSL version 2.</p> <p><input type="checkbox"/> SSLv3. Supports SSL version 3.</p> <p><input type="checkbox"/> TLS. Supports some version of TLS.</p> <p><input type="checkbox"/> TLSv1. Supports TLS version 1.</p>
Secure Data Connection	<p>Use a secure data connection (for example, to transfer data securely). This parameter is used in conjunction with <i>Secure Control Connection</i>. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> true. Use SSL with the data connection.</p> <p><input type="checkbox"/> false. Do not use SSL with the data connection.</p>
Use 128-bit Encryption	<p>Enforces the use of 128-bit encryption. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> true. Use 128-bit encryption.</p> <p><input type="checkbox"/> false. Do not use 128-bit encryption.</p>

Parameter	Description
SSL Security (required)	<p>Determines the FTP server connection type. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Unknown. Initially attempts Explicit Security then fails over to Implicit Security. (default). <input type="checkbox"/> Explicit. In order to establish the SSL link, Explicit Security requires that the FTP client issue a specific command to the FTP server after establishing a connection. The default FTP server port is used. <input type="checkbox"/> Implicit. Implicit security automatically begins with an SSL connection as soon as the FTP client connects to an FTP server. In implicit security, the FTP server defines a specific port for the client (typically 990) to be used for secure connections.
Keystore File or Keystore Security Provider	Keystore file or keystore security provider full path to the keystore file, which provides certificate material to be used for a SSL connection. Specify the name of a keystore security provider, or leave blank to use the configured default keystore security provider.
Keystore Password	The password to access the Keystore file. This value is not required if the keystore file or keystore security provider is the name of a keystore security provider.
Keystore Type	The type of keystore. This value is not required if the keystore file or keystore security provider is the name of a keystore security provider.

Configuring a FTP File Read for Process Flows Service

The FTP File Read for Process Flows service is used to read files from a FTP server.

To configure a FTP File Read for Process Flows service:

1. Perform the steps describes in [Configuring Services](#) on page 181.
2. Ensure that you select FTP File Read for Process Flows {com.ibi.agents.XDNPFFTPReadAgent} as the service type you are configuring.

For a complete description of the configuration parameters that are available for the FTP File Read for Process Flows service, see [FTP File Read for Process Flows Service Parameters](#) on page 61.

For a complete description of the edges that are returned by the FTP File Read for Process Flows service, see [FTP File Read for Process Flows Edges Service](#) on page 65.

Reference: FTP File Read for Process Flows Service Parameters

The following table lists and describes parameters for the FTP File Read for Process Flows service.

Parameter	Description
Configuration parameters for FTP File Read for Process Flows service	
Host Name	In this field, enter the DNS name (or IP address) of the FTP server that you wish to connect to. Use the host port if the standard port is not 21.
Remote Port	The port to connect to on the FTP site. Leave it blank for default port 21.
User Name	Name used as the valid user ID on the FTP server.
Password	The valid password for the FTP server.
Account Name	The valid account for the FTP server.
Use Passive Command	Select from the following: <input type="checkbox"/> true: uses a PASV command. <input type="checkbox"/> false: uses the PORT command.
Timeout	Timeout interval for socket in seconds.
Agent Parameters	
Name of File	Name of the tag from the input document in which to find the file name.

Parameter	Description
Delete After Read	<p>Use this parameter if you wish to delete the file after the read. Select from the following:</p> <ul style="list-style-type: none"><input type="checkbox"/> true: delete the file when successfully read.<input type="checkbox"/> false: do not delete the file after reading.
Format	<p>This parameter is the format of the input data. The default setting is flat.</p> <ul style="list-style-type: none"><input type="checkbox"/> flat. The data is transferred from the FTP server as a flat unformatted document.<input type="checkbox"/> XML. The data is transferred from the FTP server as a text formatted document. The data that is transmitted is assumed to be a valid XML document and is parsed. If the document is not valid then an error is returned by the service.
Transfer Mode	<p>This is a form of FTP transmission. Select one of the following modes:</p> <ul style="list-style-type: none"><input type="checkbox"/> ascii. The file is retrieved as text from the FTP server. The data received by iSM is translated into text, based on the code page configuration of iSM.<input type="checkbox"/> binary. The file is retrieved as a binary block. No text translation is performed by iSM.
Tag	<p>This parameter is the name of the tag that encloses data read. If omitted, no tagging of the data is done. If used, the output is an XML document. If the Transfer Mode is binary and the Enclose Tag is specified, the base64 Encoding should be selected or else the user risks getting an error when the resulting XML document is parsed.</p>

Parameter	Description
Encoding	<p>This parameter is the character set encoding to be performed on input. Select one of the following options:</p> <ul style="list-style-type: none"> <input type="checkbox"/> asis. The data from the FTP server is not translated. <input type="checkbox"/> base64. The data from the FTP server is converted into a base 64 XML compatible string.
Embed	<p>Whether to embed the contents of the file that is read into the input document. Select from one of the following:</p> <ul style="list-style-type: none"> <input type="checkbox"/> true: the file will be embedded into the current input document within the parent tag specified. <input type="checkbox"/> false: the document that is read replaces the input document. The old input document is discarded.
Parent Tag	<p>Where within the input document the file data should be embedded. Needed when Embed is set to true.</p> <p>Note: When embedding a file that is read into a valid xml document the file that is read must either be a valid XML (structure sans the XML declaration statement; i.e. <?xml ...?>), or must be wrapped in a Tag.</p>
Action on Failure	<p>Determines whether the input document or status document is returned on failure. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> status. Creates a status document for the return. <input type="checkbox"/> input. Returns the inbound document.
Retry	<p>Retries the operation at the specified number of times (in seconds).</p>
SSL Parameters	
Use SSL	<p>If set to true, the connection is secured using Secure Sockets Layer (SSL).</p>

Parameter	Description
Security Protocol	<p>This shows the type of security protocol to be used. The following list describes the options of the security protocol.</p> <ul style="list-style-type: none"> <input type="checkbox"/> SSL. This protocol supports some versions of SSL, and may also support other versions. <input type="checkbox"/> SSLv2. This protocol supports SSL version 2 or higher. <input type="checkbox"/> SSLv3. This protocol supports SSL version 3, and may support other versions. <input type="checkbox"/> TLS. This protocol supports some versions of TLS, and may also support other versions. <input type="checkbox"/> TLSv1. This protocol supports TLS version 1, and may support other versions. <p>This field is not needed if Keystore is a SSL Provider.</p>
Secure Data Connection	<p>This is used to enable a secure data connection, for example. transfer data securely. It is used in conjunction with Secure Control Connection.</p>
Use 128-bit Encryption	<p>This parameter enforces the use of 128-bit encryption.</p>
SSL Security (required)	<p>This parameter describes the FTP Server connection type. Select one of the following options:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Unknown. This setting defaults to Explicit Security then fails over to Implicit Security. <input type="checkbox"/> Explicit. In order to establish the SSL link, explicit security requires that the FTP client issue a specific command to the FTP server after establishing a connection. The default FTP server port is used. <input type="checkbox"/> Implicit. Implicit security automatically begins with an SSL connection as soon as the FTP client connects to an FTP server. In implicit security, the FTP server defines a specific port for the client (typically 990) to be used for secure connections.

Parameter	Description
Keystore File or Keystore Security Provider	<p>In this field, you can:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Enter the full path to the Keystore file, which provides certificate material to be used for SSL connection. <input type="checkbox"/> Name the Keystore Security Provider. <input type="checkbox"/> Use the configured default Keystore Security Provider by leaving it blank.
Keystore Password	This field is used to enter the password to access Keystore File. This is not required if Keystore File or Keystore Security Provider is the name of a Keystore Security Provider.
Keystore Type	This field shows the type of the Keystore. It is not needed if Keystore File or Keystore Security Provider is the name of a Keystore Security Provider.

Reference: FTP File Read for Process Flows Edges Service

The following table lists and describes the edges that are returned by the FTP File Read for Process Flows service.

Edge	Description
success	Operation completed successfully.
fail_parse	Failed to properly parse the input parameters of the service.
fail_connect	<p>Failed to connect to FTP host for any one of the following reasons:</p> <ul style="list-style-type: none"> <input type="checkbox"/> The host name (IP) is invalid. <input type="checkbox"/> The User ID is invalid. <input type="checkbox"/> The password of the user is invalid. <input type="checkbox"/> The connection failed.

Edge	Description
fail_operation	Invalid parameters or other error.

Configuring Secure FTP (SFTP) Components

This section describes how to configure Secure FTP (SFTP) components using iWay Service Manager.

In this chapter:

- ☐ [SFTP Component Configuration Overview](#)
 - ☐ [Configuring a SFTP Listener](#)
 - ☐ [Configuring a SFTP Emitter](#)
 - ☐ [Configuring a SFTP Read Service](#)
 - ☐ [Configuring a SFTP Directory Contents Service](#)
 - ☐ [Configuring a SFTP Direct Transfer Service](#)
 - ☐ [Configuring a SFTP Connection Cache Service](#)
 - ☐ [Configuring a SFTP Emit Service](#)
 - ☐ [Configuring a SFTP File Ops Service](#)
 - ☐ [Configuring OpenSSH on Windows](#)
 - ☐ [Converting a Private Key to the OpenSSH Key Format](#)
-

SFTP Component Configuration Overview

When you connect to a server using Secure File Transfer Protocol (SFTP), SSH encryption is used to protect the connection between your client machine and the server. This protects your password and your data, preventing an eavesdropper from capturing or stealing them as they travel over the network.

Despite the similarity in name and operation, SFTP is a completely different protocol from FTP and does not support all the same features and commands as FTP. Also, while they are both secure file transfer protocols and have similar names, FTPS (FTP with TLS/SSL) should not be confused with SFTP.

To use SFTP for secure connections, the server you are connecting to must also support SFTP. If you try to connect with SFTP to a server that does not support it, you will receive an error. Your network administrator or service provider can tell you if your server supports SFTP, and what other information you might need to use SFTP.

Password Authentication Versus Key Pair Based Authentication

All the SFTP components support both password-based and key pair-based authentication without password.

In conventional password authentication, you prove who you are by entering the correct password. The only way to prove you know the password is to tell the server what you think the password is. This means that if the server has been hacked, or spoofed, an attacker can learn your password.

Key Pair authentication solves this problem. You generate a key pair, consisting of a public key (which everybody is allowed to know) and a private key (which you keep secret and do not give to anybody). The private key is able to generate signatures. A signature created using your private key cannot be forged by anybody who does not have that key, but anybody who has your public key can verify that a particular signature is genuine.

First, generate a key pair on your own computer and copy the public key to the server under a certain name. When the server asks you to prove who you are, WinSCP can generate a signature using your private key. The server can verify that signature (since it has your public key) and allow you to log in. Now if the server is hacked or spoofed, the attacker does not gain your private key or password. They only gain one signature. And signatures cannot be re-used, so they have gained nothing.

Note: While using key pair authentication, the private key file path has to be populated in the SFTP component that is invoked. If password based authentication is used, the password field has to be populated while the private key file is left blank. This applies for all SFTP components listed below.

Supported Secure FTP Components

This section lists the supported Secure FTP components that are available in iWay Service Manager.

- ☐ **SFTP Listener.** A listener that uses the SFTP protocol component and is continuously polling the specified folder on the SFTP server (machine that supports OpenSSH FTP).
- ☐ **SFTP Emitter.** The SFTP Emitter will emit messages onto a SFTP server. It requires the credentials on the server and the directory to emit as input.

- ❑ **Services.** The following are the types of SFTP services:
 - ❑ **SFTP Read (`com.ibi.agents.XDSFTPReadAgent`).** The SFTP Read service is used to read files from a SFTP server (drive on UNIX or Windows). It can also be used in tandem with a file listener to embed file contents (the file picked up by the listener) into the XML file read from the SFTP drive by specifying the tag.
 - ❑ **SFTP Emit (`com.ibi.agents.XDSFTPEmitAgent`).** The SFTP Emit service is used to write files to an output directory through SFTP (drive on UNIX or Windows). The output file name can be specified completely or using wildcard characters.
 - ❑ **SFTP File Ops (`com.ibi.agents.XDSFTPOpsAgent`).** The SFTP File Ops (Operations) service emits data using the SSH protocol to a given *host:port* using various common SFTP commands. It can be used to perform operations, such as Copy, Prepend, Append, Size, and Move.
 - ❑ **SFTP Directory Contents (`com.ibi.agents.XDSFTPDirListAgent`).** The SFTP Directory Contents service is used to generate an XML document listing the contents of a SFTP directory specified by the user in the Directory parameter.
 - ❑ **SFTP Direct Transfer (`com.ibi.agents.XDSFTPDirectFileTransfer`).** The SFTP Direct File Transfer service transfers a file directly from the iSM to an SFTP server (drive on UNIX, or Windows), or directly from the SFTP server to the iSM server, without out moving the file through the iSM process flow.
 - ❑ **SFTP Connection Cache (`com.ibi.agents.XDSFTPConnectionCacheAgent`).** The SFTP Connection Cache service caches a single SFTP connection (both the control and data channels) to an SFTP server. When the connection cache is started, an existing connection from the cache is utilized for any SFTP agent within the iSM process flow that connects to an SFTP server, with the same address and authentication credentials (user ID and password).

Configuring a SFTP Listener

To configure a Secure FTP (SFTP) listener:

1. Perform the steps as described in [Configuring Listeners](#) on page 177.
2. Ensure that you select *SFTP* as the listener type you are configuring.

For a complete description of the configuration parameters that are available for the SFTP listener, see [SFTP Listener Configuration Parameters](#) on page 70.

For a complete description of the SFTP listener Special Registers (SREGs), see [SFTP Listener Special Registers](#) on page 74.

Reference: SFTP Listener Configuration Parameters

The following table lists and describes parameters for the SFTP listener.

Note: Parameters that are common to SFTP listeners are described in [Listener Configuration Parameters](#) on page 173.

Property Name	Property Description
Host Name (required)	The name of host machine where the listener contacts the service to obtain requests from.
Remote Port (required)	The port to connect to on the SFTP host. If left blank, the default is port 22.
Input Path	The directory with optional pattern on SFTP host from which to retrieve files. A specific file name or DOS-style pattern) can be used. Do not use suffix in.
Include Symbolic Links (required)	If set to <i>true</i> , then the SFTP listener processes the symbolic links.
Include Hidden Files (required)	If set to <i>true</i> , then the SFTP listener processes the hidden files.
Destination Directory	The directory on SFTP host to return responses to.

Property Name	Property Description
Data, Signal or Streaming (required)	<p>Determines how the input will be processed by the SFTP listener. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Data. The data file will be retrieved from the SSH server and maintained in memory while processed by the listener. <input type="checkbox"/> Signal. The data file will be retrieved from the SSH server and stored locally. A signal document will be generated by the listener. Note: If Signal is set, then a value for the Local Store Directory parameter must be provided. <input type="checkbox"/> Streaming. A connection will be opened with the SSH server and data from the file will be retrieved and processed by the listener as required.
Local Store Directory	The directory on the iWay server where files are saved. You must specify a directory path only if the Payload Type parameter is set to Signal.
Remove locally stored files	If set to <i>true</i> , the FTP listener deletes the file stored locally after it is processed. This parameter requires a directory path to be specified for the Local Store Directory parameter.
Pending Queue	The directory that holds documents which are to be retried later.
Suffix In Filter	This limits input files to those with these extensions. For example, enter <i>XML,in</i> to accept files with extensions <i>xml</i> and <i>in</i> . Note that this is not case-sensitive. Do not use a period (.). Use a dash (-) to mean no extension, or an asterisk (*) to mean any extension.
Duration	The maximum time that a document can remain in the retry pending queue.
Retry	The interval between retrying pending requests

Property Name	Property Description
Do not unzip ZIP files	This passes ZIP files as a single file for processing (requires ACCEPT FLAT turned on).
Bad File List	This maintains a list of files with errors, preventing them from being re-accessed. If set to <i>true</i> , the files will not be retried.
Delete After Read	This determines whether to delete the file after it is read. If set to <i>true</i> , the file is deleted by the listener. If set to <i>false</i> , the file will not be deleted by the listener.
File Protect	This emits a temporary name and then renames it to the desired name.
Security	
User Name	The user ID on the SFTP server.
Password	The user password on the SFTP server.
Private Key	The path to the private key file for public-key authentication.
Passphrase	The passphrase used to protect the Private Key
Other	
Whitespace Normalization	Specifies how the parser treats whitespace in element objects. Select <i>preserve</i> (default) to turn off all normalization as prescribed by the XML Specification. Select <i>condense</i> to remove extra whitespaces in pretty printed documents and for compatibility with earlier versions.
Accepts non-XML (flat) only	If set to true, the listener expects flat (non-XML). Automatic parsing is not performed.
Optimize Favoring	Use this when the selection of memory is useful for large input document
Multithreading	The number of documents that can be processed in parallel

Property Name	Property Description
Execution Time Limit	The time limit for document execution in seconds, before cancellation is attempted. Also see the system property kill interval. This applies to agent stacks and sets a lower limit for process flows.
Polling Interval	The interval at which to check for new input.
Default Java File Encoding	The default encoding if incoming message is not self-declaring (that is, XML).
Agent Precedence	This changes the order by which the engine selects agents. Normally the document overrides the listener. This is used to manage iWay documents.
Always reply to listener default	If set to <i>true</i> , the default reply definition is used in addition to the defined replies.
Error Documents treated normally	If set to <i>true</i> , the error documents will get processed by any configured pre-emitters.
Listener is Transaction Manager	If set to <i>true</i> , the agents run within a local transaction managed by the listener.
Record in Activity Log(s)	If set to <i>true</i> , the activity on this channel will be recorded in the activity logs. If set to <i>false</i> , the activity will not be recorded.

Note: The SFTP listener supports streaming. Streaming is used for large documents or documents for which the application needs to split the input into sections under the same transaction. For more information on streaming and configuring streaming preparers, see the *iWay Service Manager Component and Functional Language Reference Guide*.

f

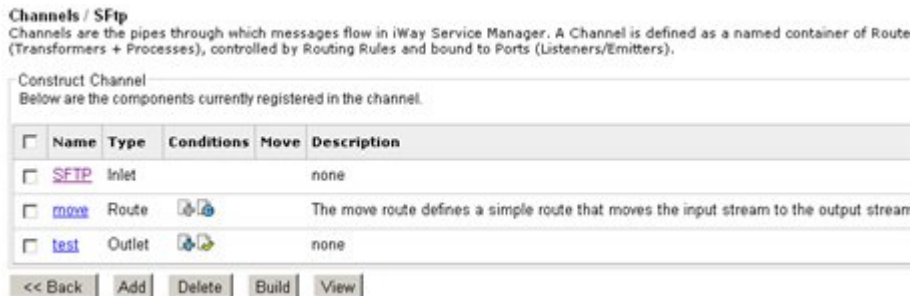
Reference: SFTP Listener Special Registers

The following table lists and describes the Special Registers (SREGs) available on the SFTP listener.

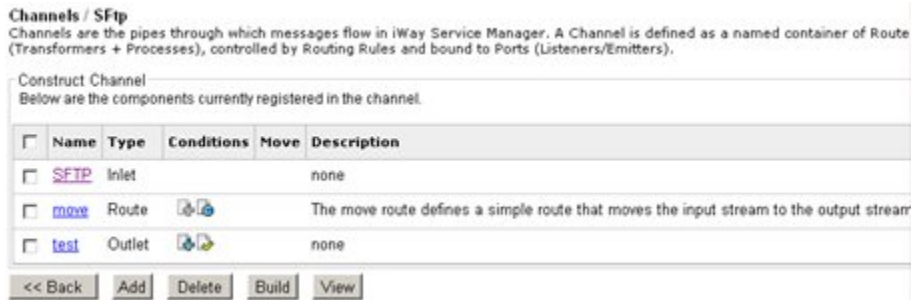
Name	Level	Type	Description
ipayconfig	System	String	The current active configuration name.
msgsize	Document	Integer	The physical length of the message payload.
name	System	String	The assigned name of the master (listener).
protocol	System	String	The protocol on which the message was received.
source	Document	String	The full name of the input file.
tid	Document	String	Unique transaction ID.

Procedure: How to Test the SFTP Listener Channel Using a Private Key File Without a Password

1. Construct an inlet consisting of a SFTP listener, as shown in the following image.



- Construct a channel (for example, mySFTP) consisting of the inlet, move Route and a File emitter which writes the output file to a test directory (for example, c:\test), which is shown in the following image.



- Build and deploy the channel.
- Start the channel.
- Place an XML file in the /home/org directory on the SFTP host using SFTP commands, or if the server is a Windows machine, by copying it to the folder.
- The SFTP channel processes this message and drops the file into the c:\test directory on the iway client machine.

Configuring a SFTP Emitter

This section describes how to configure a SFTP emitter.

The following table lists and describes the procedure for the SFTP Emitter.

Property Name	Property Description
Destination	The absolute path of the file which is being emitted at the name of the SFTP server. For instance, if the file needs to be emitted to a machine sftpsrv on the directory /home/org and the file is to be saved as out[1..9].xml then the value of this field would be /home/org/out*.xml@sftpsrv as shown above.
User Name	The user name on the SFTP server that has read and write access to the directory entered in the Input Path field.
Password	The password for the user account to use when connecting to protocol host.

Property Name	Property Description
Private Key	The SSH private Key file used for server authentication (required is password is omitted).
Pass Phrase	The SSH Passphrase used when Private Key was generated (optional).
Move To	The directory to which the file is to be moved after it is emitted.
File Protect	Emits a temporary name and then renames it to the desired name.

Procedure: How to Test a SFTP Emitter Test Channel

1. Construct an outlet consisting of a SFTP emitter (for example, mySftpEmit).
2. Construct a channel (for example, File1) consisting of a file inlet, a move Route and a File emitter which writes the output file to a test directory (for example, c:\test).
3. Build and deploy the channel.
4. Start the channel.
5. Copy an XML file to the directory where the File1 listener is listening.

The File1 channel processes this message and copies the out1.xml file onto the directory on the SFTP server machine, which is /home/org.

Configuring a SFTP Read Service

The SFTP Read service is used to read a file using SFTP and return the read result.

To configure a SFTP Read service:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select *SFTP Read Agent {com.ibi.agents.XDSFTPReadAgent}* as the service type you are configuring.

For a complete description of the configuration parameters that are available for the SFTP Read service, see [SFTP Read Service Parameters](#) on page 77.

For a complete description of the edges that are returned by the SFTP Read service, see [SFTP Read Service Edges](#) on page 78.

Reference: SFTP Read Service Parameters

The following table lists and describes parameters for the SFTP Read service.

Property Name	Property Description
File Name	The special register namespace in which protocol headers from the incoming request will be saved.
File Name not a Document Tag	The special register namespace from which protocol headers for the outbound response will be taken.
Enclose Tag	The name of the tag that encloses the data read. If omitted, no entagging is used. If used, the output is XML.
Base Path	The optional directory to be used if incoming name is not absolute.
Input Data Format	The format of the input data, default is flat.
Transfer Type	For non-XML, this parameter sets the transfer type.
Host Name	The name of the SFTP server to connect to.
Remote Port	The port to connect to on the SFTP site. If left blank, the default is port 22.
User Name	The username on the SFTP server through which files are emitted.
Password	The password for the user account to use when connecting to protocol host.
Private Key	The SSH Private Key file used for server authentication (required is password is omitted).
Pass Phrase	The SSH Passphrase used when Private Key was generated (optional).
Encoding	The character set encoding to be performed on the input.
Delete After Read	The flag to show whether to delete the file after the read.

Reference: SFTP Read Service Edges

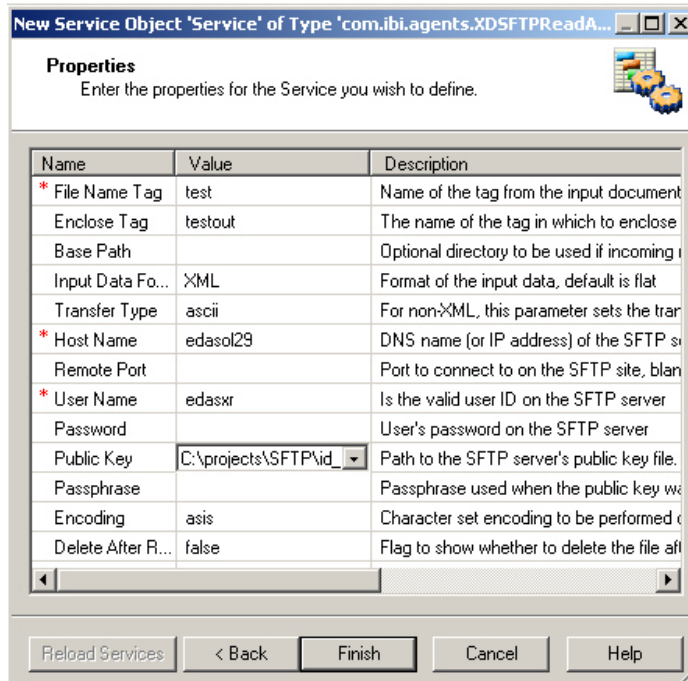
The following table lists and describes the edges that are returned by the SFTP Read service.

Edge	Description
success	Operation completed successfully.
fail_parse	Failed to properly parse the input parameters of the service.
fail_connect	Failed to connect to SFTP host for any one of the following reasons: <ul style="list-style-type: none"><input type="checkbox"/> The host name (IP) is invalid.<input type="checkbox"/> The User ID is invalid.<input type="checkbox"/> The password of the user is invalid.<input type="checkbox"/> The connection failed.
fail_operation	Invalid parms or other error.
fail_notfound	The file name in the File Name Tag parameter does not exist on the SFTP host.
fail_delete	Failed to delete the file either after the initial read, or when the transaction successfully completed.

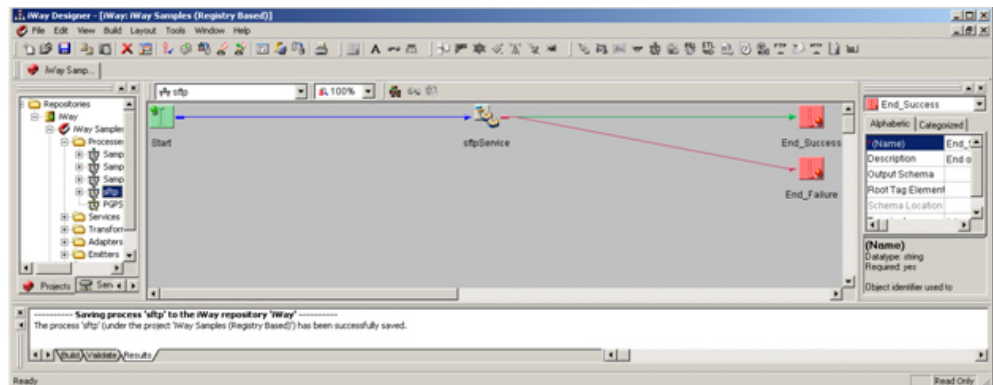
Procedure: How to Test a SFTP Read Service Using a Private Key File

The SFTP Read service is used to read files from a SFTP server (drive on UNIX or Windows). It can also be used in tandem with a file listener to embed file contents (the file picked up by the listener) into the xml file read from the SFTP drive by specifying the tag.

1. Construct a process flow called sftp, which consists of a Service object (for example, sftpService) that refers to the XDSFTPReadAgent class, as shown in the following image.



2. Construct the process flow as shown in the following image.



3. Add the sftp process to a route (for example, myRoute). Construct a channel (for example, File1) consisting of a file inlet, myRoute, and a default outlet.
4. Build and deploy the channel.
5. Start the channel.
6. Copy an XML file (for example, test.xml) to the directory where the File1 listener is listening. This XML file can have the following content:

```
<test>c:\test\a.txt</test>
```

7. Let the a.txt file consist of the following text:

```
This is a sftp readagent test
```

8. The File1 channel processes the XML file (test.xml), reading the file a.txt and generates the file testout.xml, which is copied to the destination directory specified for the file listener. The file testout.xml has the following content:

```
<testout> This is a sftp readagent test </testout>
```

SFTP Read Service Output Example

The following is an example of an SFTP Read Service output.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<dir base="/outbound" count="5">
  <item type="directory">Extra Directory</item>
  <item type="file">1.txt</item>
  <item type="file">2.txt</item>
  <item type="file">3.txt</item>
  <item type="file">ThisIsATest.txt</item>
</dir>
```

Configuring a SFTP Directory Contents Service

The SFTP Directory Contents service is used to generate an XML document listing the contents of a SFTP directory specified by the user in the Directory parameter. The XML document that is generated has the following pattern:

```
<dir base="Directory" count="number">
  <item type="Item Type">Name</item>
</dir>
```

The content of the document differs based on the Include parameter configured by the user. If the user selects *Files*, then the item element will contain only file names from the directory. If the user selects *Subdirectories*, then only the subdirectory names contained in the Directory are included. If the user selects *All*, then file and subdirectory names are included.

Output

The XML document generated by the service contains only two elements (*dir* and *item*).

The element *dir* is the root element and contains two attributes (*name* and *count*). The *name* attribute contains the Directory entry that was configured by the user. The *count* attribute contains the number of item elements to follow.

The *item* element is the only child of the *dir* element, but can have multiple *item* siblings. The *item* element has only one attribute (*type*). The *type* attribute identifies the type of name that this item element contains. If the *type* attribute is set to *file*, then this item element contains a name of a file. If the *type* attribute is set to *directory*, then this item element contains a name of a subdirectory.

To configure a SFTP Directory Contents service:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select *SFTP Directory Contents* (*com.ibi.agents.XDSFTPDirListAgent*) as the service type you are configuring.

For a complete description of the configuration parameters that are available for the SFTP Directory Contents service, see [SFTP Directory Contents Service Parameters](#) on page 81.

Reference: SFTP Directory Contents Service Parameters

The following table lists and describes parameters for the SFTP Directory Contents service.

Property Name	Property Description
Host Parameters	
Host Name (required)	DNS name (or IP address) of the SFTP server to which you want to connect. Use <i>host:port</i> format if the standard port is not 22.
Remote Port (required)	Port to connect to on the SFTP site. Leave blank for default port 22.
Buffer Size	Size of the SFTP buffer to be used when sending or retrieving data. The default value of 32768 is used if this parameter value is not specified. A larger buffer may improve performance but setting this field to a value greater than 65536 will default to 65536. The value must be entered as a whole number (for example, 32768, 65536). iWay recommends leaving the buffer size at 32768.

Property Name	Property Description
SSH Parameters	
User Name	Is the valid user ID on the SFTP server.
Password	Is the valid password for the SFTP server (optional and not used if Private Key is configured).
Private Key	Path to the private key file for public-key authentication.
Passphrase	Passphrase used to protect the Private Key (optional and required only if Private Key is passphrase protected).
Provider	Name of the SSH Client Security Provider to use. If no Provider name is specified, then enter the User Name and either a Password or a Private Key and Passphrase values. Passphrase is required only if the Private Key file is passphrase protected.
AGENT Parameters	
Directory (required)	Determines what directory should be listed. The directory must exist.
Include	<p>Determines what items should be included in the listing. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> Files. Include only file names.</p> <p><input type="checkbox"/> Subdirectories. Include only subdirectory names.</p> <p><input type="checkbox"/> All. Include files and subdirectories.</p>
Select Expression	Regular expression (or filter) used to select files and directories.

Property Name	Property Description
Pattern Type	<p>If a value for the Selection Expression parameter is supplied, then this parameter indicates whether it should be interpreted as a regular expression or a DOS style wildcard. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Regular Expression. Java-based Regular Expression syntax. <input type="checkbox"/> DOS-style Wildcard. DOS style wild card syntax. A question mark (?) indicates any single number or character. An asterisk (*) indicates any combination of numbers and characters.
Retry (required)	Retries the operation at the specified number of times (in seconds).
Call at EOS	<p>When using a streaming preparer in a channel, a last call is made <i>after</i> the last document. Determines whether this service exit should be called. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"> <input type="checkbox"/> true. Calls the service exit. <input type="checkbox"/> false. Does not call the service exit.

Configuring a SFTP Direct Transfer Service

The SFTP Direct File Transfer service is used to read or write files directly from a SFTP server to iWay Service Manager (iSM).

If you select *send* as a value for the Transfer Type parameter, then the file is moved from the directory specified in the Working Directory parameter to the SFTP server. If you select *transfer* as a value for the Transfer Type parameter, then the file is moved from the SFTP server directory specified in the Host Directory parameter, to the Working Directory in iSM.

To configure a SFTP Direct transfer:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select *SFTP Direct transfer {com.ibi.agents.XDSFTPDirectFileTransfer}* as the service type you are configuring.

For a complete description of the configuration parameters that are available for the SFTP Direct File Transfer service, see [SFTP Direct Transfer Service Parameters](#) on page 84.

Reference: SFTP Direct Transfer Service Parameters

The following table lists and describes parameters for the SFTP Direct File Transfer service.

Parameter	Description
Configuration parameters for SFTP Direct transfer service	
Host Name (required)	The DNS name (or IP address) of the SFTP server that you want to connect to.
Remote Port (required)	The port to connect to on the SFTP site. Leave it blank for default port 21.
Buffer Size	Size of the SFTP buffer to be used when sending or retrieving data. The default value of 32768 is used if this field is not set. A larger buffer may improve performance, but setting this field to a value greater than 65536 will default to 65536. The value must be entered as a whole number (for example, 32768 or 65536). iWay recommends leaving the buffer size set to 32768.
SSH Parameters	
User Name	A valid user ID for the SFTP server.
Password	A valid password for the SFTP server.
Private Key	Path to the private key file for public key authentication.
Passphrase	Passphrase used to protect the private key.
Provider	Name of the SSH Client Security Provider to use. If no provider name is specified, then enter the user ID and either a password or a private key and passphrase values. Passphrase is required only if the private key file is protected by a passphrase.
Agent Parameters	

Parameter	Description
Use SCP	<p>Use the Secure Copy protocol for file copy. Not all servers support the use of SCP. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> true. Use the Secure Copy protocol for file copy.</p> <p><input type="checkbox"/> false. Do not use the Secure Copy protocol for file copy (default).</p>
Name of File (required)	This is the file to be read. A relative or absolute file path is supported explicitly or through a SREG or XPath expression that is evaluated using the incoming document.
Host Directory	An optional directory to be used if the name of the file is not absolute.
Working Directory (required)	The path to the iSM working directory.
Transfer Type (required)	<p>The direction of file transfer. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> retrieve. Retrieves the file from the SFTP server.</p> <p><input type="checkbox"/> send. Sends the file from iSM to the SFTP server.</p>
Return 'status'	<p>Determines the return document status. Select one of the following options from the drop-down list:</p> <p><input type="checkbox"/> status. Creates a status document for the return.</p> <p><input type="checkbox"/> input. Returns the inbound document as the output</p>

Parameter	Description
Delete After Read	<p>Determines whether to delete the file after the read. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"><input type="checkbox"/> Keep. Do not delete the file (default).<input type="checkbox"/> Delete. Delete the file immediately after the read.<input type="checkbox"/> Delete on Success. This is a transactional delete if the listener supports transactions. Otherwise the file is deleted immediately after the file is read. If the listener supports transactions, then the file is not deleted unless the flow ends in success. <p>Note:The connection to the SFTP server is maintained until that time. Users are strongly cautioned that use of this option within an iteration can result in holding of connections to the server, which in turn can cause subsequent connection failures and resource buildup.</p>
Action on Failure	<p>Determines whether the input document or status document is returned on failure. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"><input type="checkbox"/> status. Creates a status document for the return.<input type="checkbox"/> input. Returns the inbound document.
Connection Retry	<p>The number of attempted failed connections to the SFTP server.</p>
Retry Interval	<p>The retry interval (in seconds) for the socket. The <i>xxhxxmxs</i> format can be used for this value. Specify a value of zero (0) for no retry interval.</p>

Configuring a SFTP Connection Cache Service

The SFTP Connection Cache service caches a single SFTP connection (both the control and data channels) to an SFTP server. When the connection cache is started, an existing connection from the cache is utilized for any SFTP agent, within the iSM process flow that connects to an SFTP server, with the same address and authentication credentials (user ID and password).

The SFTP connection cache that has been configured to start caching will create a cache if it does not already exist, and pre-populate it with a connection. Other SFTP agents will use this cache if it exists. Specifically, that means retries request new connections from the connection pool, and the connection is returned to the cache when the agent completes its execution.

To configure a SFTP Connection Cache service:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select *SFTP Connection Cache* `{com.ibi.agents.XDSFTPConnectionCacheAgent}` as the service type you are configuring.

For a complete description of the configuration parameters that are available for the SFTP Connection Cache service, see [SFTP Connection Cache Service Parameters](#) on page 87.

Reference: SFTP Connection Cache Service Parameters

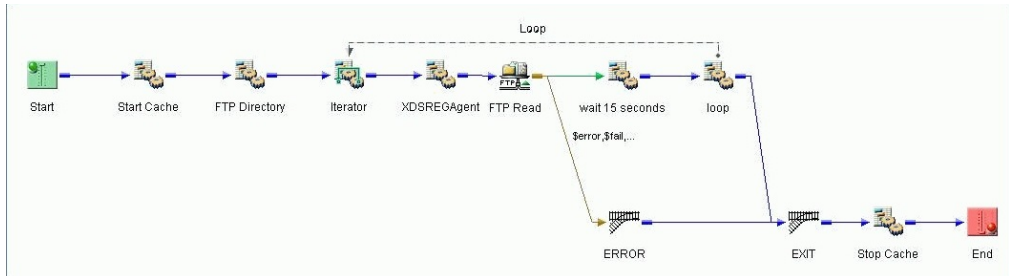
The following table lists and describes parameters for the SFTP Connection Cache service.

Parameter	Description
Configuration parameters for SFTP Connection Cache service	
Host Name (required)	The DNS name (or IP address) of the SFTP server that you want to connect to.
Remote Port (required)	The port to connect to on the SFTP site. Leave it blank for default port 22.
Buffer Size	Size of the SFTP buffer to be used when sending or retrieving data. The default value of 32768 is used if this field is not set. A larger buffer may improve performance, but setting this field to a value greater than 65536 will default to 65536. The value must be entered as a whole number (for example, 32768 or 65536). iWay recommends leaving the buffer size set to 32768.

Parameter	Description
SSH Parameters	
User Name	A valid user ID for the SFTP server.
Password	A valid password for the SFTP server.
Private Key	Path to the private key file for public key authentication.
Passphrase	Passphrase used to protect the private key.
Provider	Name of the SSH Client Security Provider to use. If no provider name is specified, then enter the user ID and either a password or a private key and passphrase values. Passphrase is required only if the private key file is protected by a passphrase.
Agent Parameters	
Connection Caching (required)	<p>Select start to cache the connection referenced by this service. Select stop to close the cached connection. Select one of the following options from the drop-down list:</p> <ul style="list-style-type: none"><input type="checkbox"/> start. Starts the caching connection on this thread.<input type="checkbox"/> stop. Stops the connection caching on this thread and closes the active connection.

Reference: SFTP Connection Cache Service Example

The following is a sample process flow for the SFTP Connection Cache service.



iSM Object	Description
Start	Start of process flow.
Start Cache	Calls the XDSFTPConnectionCacheAgent for the server to connect to.
FTP Directory	XDSFTPDirListAgent uses the connection that was created in Start Cache to create a document containing the files within a directory.
Iterator	Loops through the document returned by FTP Directory using the expression /dir/item to retrieve each entry in the document.
XDSREGAgent	Creates Special Registers with the File name and Directory extracted via XPATH from the document segment created by the Iterator.
FTP Read	XDSFTPReadAgent uses the connection that was created in Start Cache to read the file from the directory.
wait 15 seconds	Uses the XDCopyAgent to delay processing for 15 seconds simulates processing that may occur.

iSM Object	Description
Loop	Loops back to the Iterator to get next file value.
ERROR	If the XDSFTPReadAgent is unsuccessful, the FTP Read exits (\$error, \$fail, etc...).
EXIT	Called after either the ERROR, or loop completes.
Stop Cache	Calls the XDSFTPConnectionCacheAgent to close the connection to the server.
End	Terminates the process flow.

Configuring a SFTP Emit Service

The SFTP Emit service is used to emit a message using SFTP to a specific directory on a defined host.

To configure a SFTP Emit service:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select *SFTP Emit Agent* (*com.ibi.agents.XDSFTPEmitAgent*) as the service type you are configuring.

For a complete description of the configuration parameters that are available for the SFTP Emit service, see [SFTP Emit Service Parameters](#) on page 90.

For a complete description of the edges that are returned by the SFTP Emit service, see [SFTP Emit Service Edges](#) on page 91.

Reference: SFTP Emit Service Parameters

The following table lists and describes parameters for the SFTP Emit service.

Property Name	Property Description
Host Name	The DNS name (or IP address) of the SFTP server that you want to connect to.

Property Name	Property Description
Remote Port	The port number used to connect to the SFTP site, blank for default port 22.
User Name	The user ID on the SFTP server.
Password	The user password on the SFTP server.
Private Key	The path to the private key file used for public-key authentication.
Pass Phrase	The pass phrase used to protect the private key.
Remote Site Folder	The folder or directory on the SFTP site that you want to use as a starting location when you connect. A blank value defaults to the login directory.
File Protect	This emits a temporary name and then renames it to the desired name.
File Pattern	This specifies the output file pattern to be used.
Retry Interval	The retry interval in seconds (allows for xxhxxmxxs format). Omit or use 0 for no retry.
Return	<p>Select one of the following values:</p> <p><input type="checkbox"/> Status. The status document will be the output document.</p> <p><input type="checkbox"/> Input. The input document will be the output document.</p>

Reference: SFTP Emit Service Edges

The following table lists and describes the edges that are returned by the SFTP Emit service.

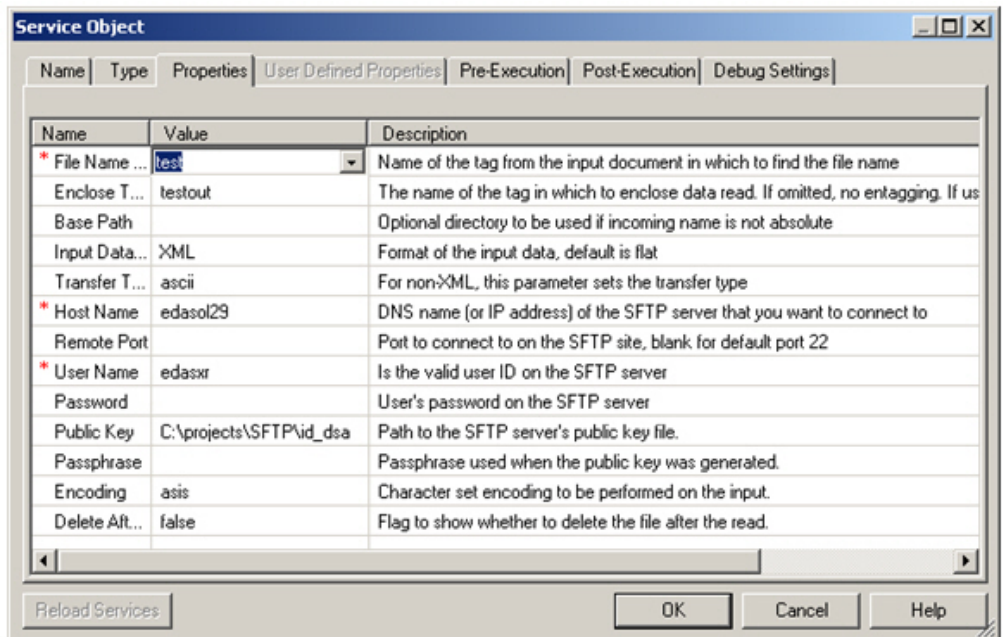
Edges	Description
success	Operation completed successfully.

Edges	Description
fail_connect	<p>Failed to connect to SFTP host for one of the following reasons:</p> <ul style="list-style-type: none"> <input type="checkbox"/> The host name (IP) is invalid. <input type="checkbox"/> The User ID is invalid. <input type="checkbox"/> The password of the user is invalid. <input type="checkbox"/> The connection failed.
fail_operation	Invalid parameters or other error.

Procedure: How to Test a SFTP Emit Service Using a Private Key File

The test case is similar to the SFTP emitter, as shown in [Configuring a SFTP Emitter](#) on page 75. However, in this case a service is used to emit the file to the SFTP server using private key file. A SFTP Emit service would be used instead of a SFTP Emitter when a process flow which performs a sequence of tasks (for a business process) needs to be used and statuses need to be evaluated.

1. Construct a process flow called sftp, which consists of a Service object (for example, sftpsservice) that refers to the XDSFTPEmitAgent class, as shown in the following image.



As shown in the above example, the value of the Host Name can be

`edasol29`

and the username can be

`edasxr`

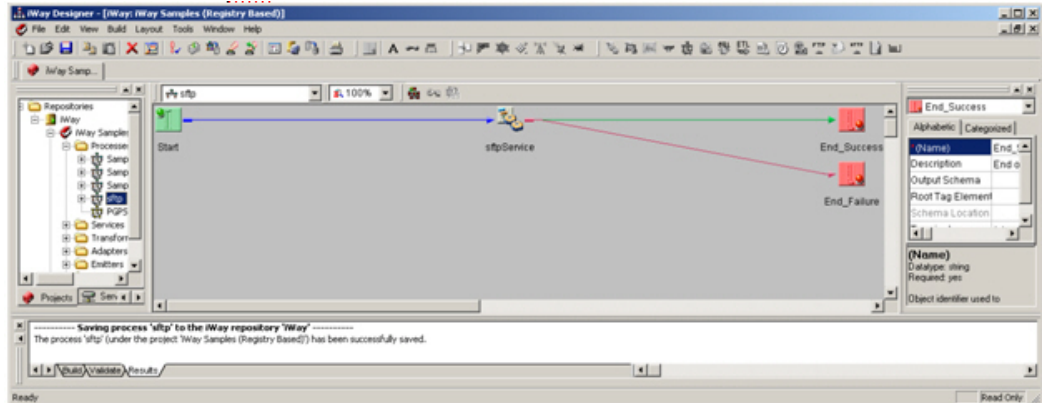
where:

`edasxr`

Has write access to the Remote Site Folder directory.

Also, the Private Key file can point to the id_dsa private key file on the iway server. Enter the values of the fields as shown above.

2. Construct the process flow as shown in the following image.



3. Add the sftp process to a route (for example, myRoute).
4. Construct a channel (for example, File1) consisting of a file inlet, myRoute, and a default outlet.
5. Build and deploy the channel.
6. Start the channel.
7. Copy an XML file to the directory where the File1 listener is listening.

The File1 channel processes this message and copies the out1.xml file to the directory on the SFTP server referred to by the attribute Remote Site Folder on the sftpservice object in the process flow.

Configuring a SFTP File Ops Service

The SFTP File Ops (Operations) service is used to perform simple SFTP file operations.

To configure a SFTP File Operations service:

1. Perform the steps as described in [Configuring Services](#) on page 181.
2. Ensure that you select *SFTP File Ops Agent {com.ibi.agents.XDSFTPFileOpsAgent}* as the service type you are configuring.

For a complete description of the configuration parameters that are available for the SFTP File Ops service, see [SFTP File Ops Service Parameters](#) on page 95.

For a complete description of the edges that are returned by the SFTP File Ops service, see [SFTP File Ops Service Edges](#) on page 99.

Reference: SFTP File Ops Service Parameters

The following table lists and describes parameters for the SFTP File Ops service.

Parameter	Description
Host Parameters	
Host Name (required)	DNS name (or IP address) of the SFTP server to which you want to connect. Use <i>host:port</i> format if the standard port is not 22.
Remote Port (required)	Port to connect to on the SFTP site. Leave blank for default port 22.
Buffer Size	Size of the SFTP buffer to be used when sending or retrieving data. The default value of 32768 is used if this parameter value is not specified. A larger buffer may improve performance but setting this field to a value greater than 65536 will default to 65536. The value must be entered as a whole number (for example, 32768, 65536). iWay recommends leaving the buffer size at 32768.
SSH Parameters	
User Name	Is the valid user ID on the SFTP server.
Password	Is the valid password for the SFTP server (optional and not used if Private Key is configured).
Private Key	Path to the private key file for public-key authentication.
Passphrase	Passphrase used to protect the Private Key (optional and required only if Private Key is passphrase protected).
Provider	Name of the SSH Client Security Provider to use. If no Provider name is specified, then enter the User Name and either a Password or a Private Key and Passphrase values. Passphrase is required only if the Private Key file is passphrase protected.

Parameter	Description
Agent Parameters	
Operation (required)	<p>Operation to perform on the file hosted by the SFTP Server. Operations supported by this service are as follows:</p> <ul style="list-style-type: none"> <input type="checkbox"/> copy. Copies the data from the file addressed by the File (from) parameter to the file named in the File (to) parameter. <input type="checkbox"/> move. Moves the data from the file addressed by the File (from) parameter to the file named in the File (to) parameter. When successfully completed the file addressed by the File (from) parameter is deleted. <input type="checkbox"/> rename. Renames the file addressed by the File (from) parameter to the file named in the File (to) parameter. When successfully completed the file addressed by the File (from) no longer exists. <input type="checkbox"/> prepend. Copies the data from the file addressed by the File (from) parameter to the beginning of the file named in the File (to) parameter.
Operation (continued)	<ul style="list-style-type: none"> <input type="checkbox"/> append. Copies the data from the file addressed by the File (from) parameter to the end of the file named in the File (to) parameter. <input type="checkbox"/> delete. Deletes the file addressed by the File (from) parameter from the host. <input type="checkbox"/> size. Gets the size of the file addressed by the File (from) parameter from the host. The return is places in the Special Register named in the Remote Size parameter. <input type="checkbox"/> exist. Verifies that the file addressed by the File (from) parameter exists on the host.
File (from) (required)	Name of the source file. This field may be a relative or absolute file paths, a SREG or XPath expression. This is a required field.

Parameter	Description
File (to) (required)	The name of the destination file. Wild cards are accepted. This is a required field except when operation is delete, size, or exist.
File (to) a directory name	References a directory. For more information on this parameter, see the description and example that follows this table.
File (to) Create Directories	Creates a directory if one does not exist. For more information on this parameter, see the description and example that follows this table.
Remote Size	Name of the Special Register designated to hold size. This field is required when operation is size.
Out Document (required)	<p>Specify the document to be returned by the operation (bad input defaults to <i>result</i>).</p> <p>Selecting <i>result</i> returns the results of the requested operation. In the case of copy, move, rename, delete, size, and exist, the status document containing the status of the function is returned.</p> <p>The functions <i>prepend</i> and <i>append</i> result in the file data being returned. This data will be the same as the data found in the file addressed by the File (to) parameter.</p>
Action on Failure (required)	Determines whether the input document or status document is returned on failure.
Retry (required)	If non-zero, the operation will be retried <i>n</i> times at one-second intervals.

File (to) a directory name Parameter

The *File (to) a directory name* parameter references a directory.

File (to) a directory name	File (to) references a directory, if it is unclear whether path names a directory or a filename, the Service Manager will assume the path names a file.
<input type="text" value="false"/>	
<input type="button" value="Pick one"/>	

The default value for this parameter is *false* and iSM will use the path specified in the File (to) parameter to reference a file path. If the File (to) a directory name parameter is set to *true*, then iSM will use the File (to) parameter to reference a directory path and the file created will have the same file name as the File (from) parameter.

Example:

The File (from) parameter has the file name `temp/output.xml`, the File (to) parameter has the name `prod/final`, and the File (to) a directory name parameter is set to *true*. The results can be found in the file `output.xml` in the directory `prod/final`. Otherwise, if the File (to) a directory name parameter is set to *false* (default), the results will be found in the file `final` in the directory `prod`.

Note: iSM supports the creation of dynamic File (to) file names using special iSM file name patterns using a combination of the following three characters (`#*^`). These characters are only allowed when the File (to) parameter is a file name (File (to) a directory name parameter is set to *false*). If the File (to) a directory name parameter is set to *true* and the parameter contains one of the iSM pattern characters (`#*^`), an error occurs.

Additionally, iSM's pattern control file is saved in the file directory:


```
[iwayworkdir]/sftpdata/File (to) parent directory
```

For example, if using the FTP Ops service and the File (to) parameter is set to `prod/ism####.xml`, then the pattern control file would be located as follows:

```
[iwayworkdir]/sftpdata/prod/.ism####.xml
```

File (to) Create Directories Parameter

The *File (to) Create Directories* parameter creates a directory if one does not exist.

File (to) Create Directories	Create if directory doesn't exist. Used only for Copy, Move, Rename and Append operations.
<input type="text" value="false"/>	
<input type="button" value="Pick one"/> 	

iSM now supports dynamic creation of the directory tree. The default value for the *File (to) Create Directories* parameter is *false*. If set to *false*, the directory structure is expected to already be in place, and if not, an error is returned. If set to *true* however, iSM will attempt to create the directory structure defined by the File (to) parameter. If successful, the full tree structure as defined in the File (to) parameter will be created *before* the function is performed.

Example:

The File (to) parameter is set to *prod/final/f0001.xml*. iSM checks for the existence of the directory *prod*. If *prod* does not exist, then *prod* is created. Next the directory *final* is checked. If *final* does not exist, then *final* is created and so on until the directory structure is complete. Once the directory structure is in place the service executes the configured function.

Note: If an attempt to create the tree structure fails (due to an error being returned from the remote system), some part of the tree structure may have been created. It is the responsibility of the user to determine the correct course of action to stabilize the directory structure of the remote system.

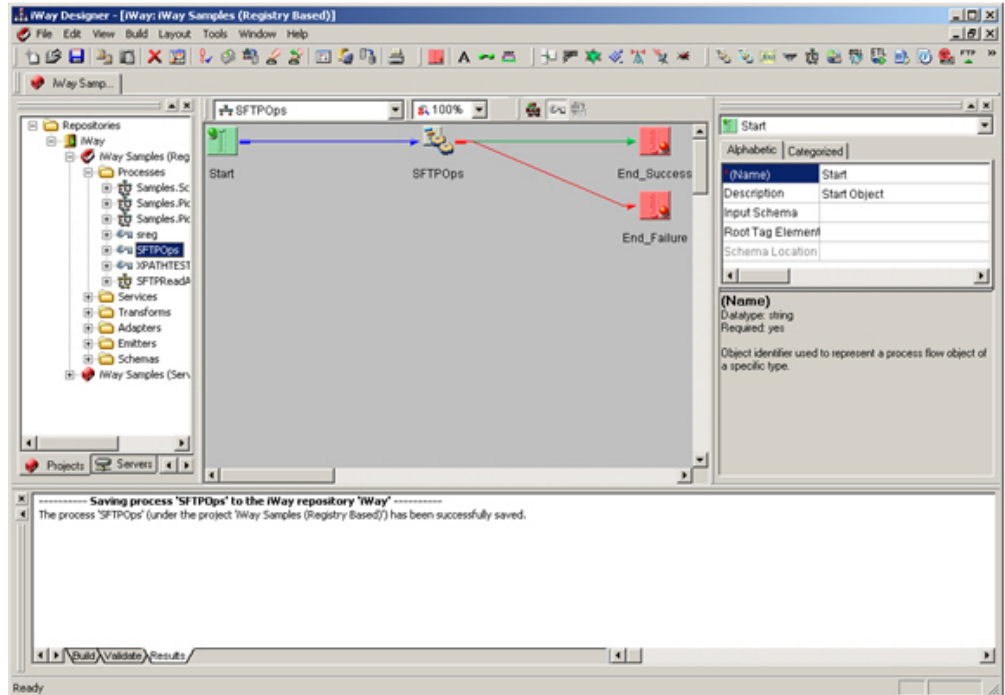
Reference: SFTP File Ops Service Edges

The following table lists and describes the edges that are returned by the SFTP File Ops service.

Edge	Description
success	Operation completed successfully.
fail_parse	Failed to properly parse the input parameters of the service.
fail_connect	Failed to connect to SFTP host for any one of the following reasons: <ul style="list-style-type: none"> <input type="checkbox"/> The host name (IP) is invalid. <input type="checkbox"/> The User ID is invalid. <input type="checkbox"/> The password of the user is invalid. <input type="checkbox"/> The connection failed.
fail_operation	Invalid parms or other error.
fail_notfound	The file name in the File Name Tag parameter does not exist on the SFTP host.

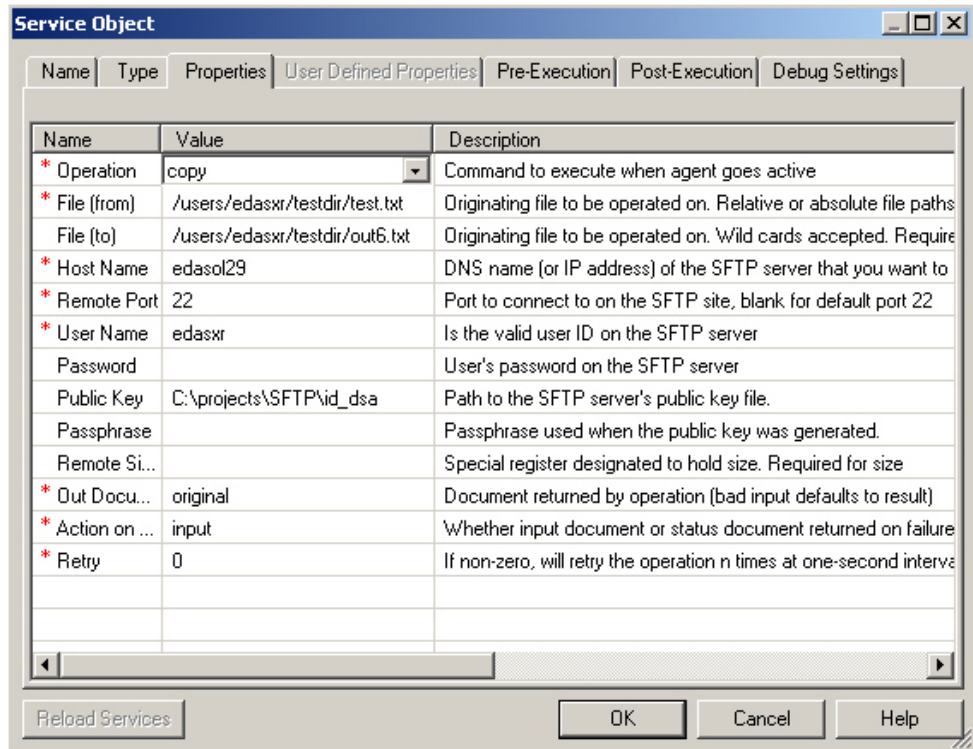
Procedure: How to Test the SFTP Ops Service for a Copy Operation

1. Create a process flow and add a service object for the SFTP File Operations service (com.ibi.agents.XDSFTPOpsAgent), as shown in the following image.



2. Set the Service object class name to com.ibi.agents.XDSFTPOpsAgent.

- Set the properties as shown in the following image.



The From file test.txt is:

```
This is an SFTPServer test
```

The to file out6.txt is:

```
<parent><test1>Soumya</test1></parent>
```

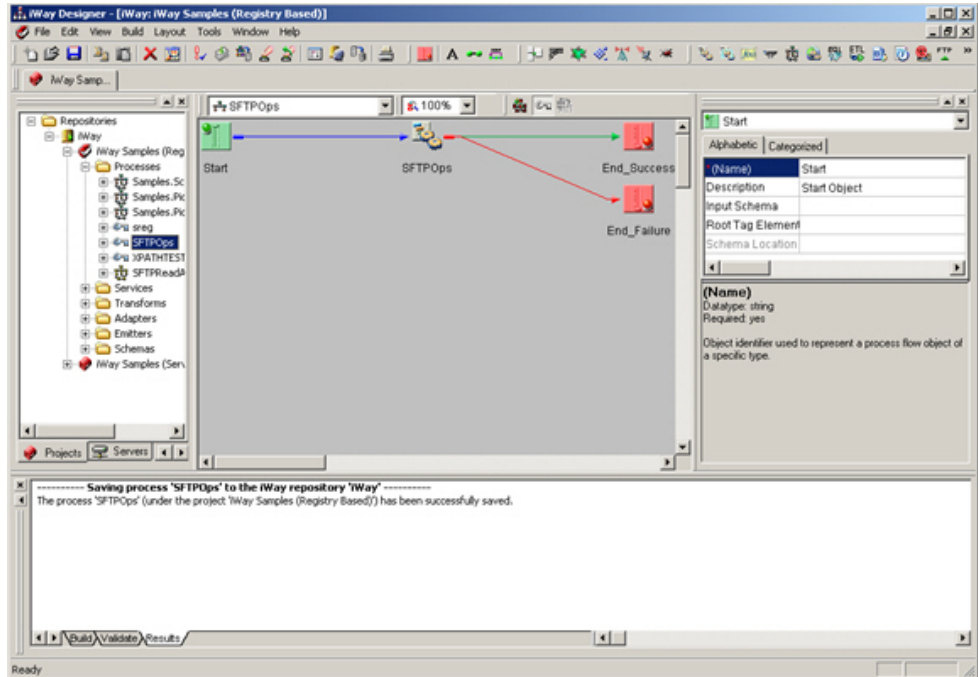
- Test run the process flow SFTPOps.

The to file out6.txt content is modified as:

```
This is an SFTPServer test
```

Procedure: How to Test the SFTP Ops Service for a Move Operation

1. Create a process flow and add a service object for the SFTP File Operations service (com.ibi.agents.XDSFTPOpsAgent), as shown in the following image.



2. Set the Service object class name to com.ibi.agents.XDSFTPOpsAgent.

Set the properties as shown in the following image.

Pre-Execution		Post-Execution	Debug Settings
Name	Type	Properties	User Defined Properties
* Operation	move	Command to execute when agent goes active	
* File (from)	/users/edasxr/testdir/test.txt	Originating file to be operated on. Relative or abs	
File (to)	/users/edasxr/testdir/testmove.txt	Originating file to be operated on. Wild cards acc	
* Host Name	edasol29	DNS name (or IP address) of the SFTP server tha	
* Remote Port	22	Port to connect to on the SFTP site, blank for def	
* User Name	edasxr	Is the valid user ID on the SFTP server	
Password		User's password on the SFTP server	
Public Key	C:\projects\SFTP\id_dsa	Path to the SFTP server's public key file.	
Passphrase		Passphrase used when the public key was gener	
Remote Si...		Special register designated to hold size. Requirec	
* Out Docu...	original	Document returned by operation (bad input defau	
* Action on ...	input	Whether input document or status document retu	
* Retry	0	If non-zero, will retry the operation n times at one-	

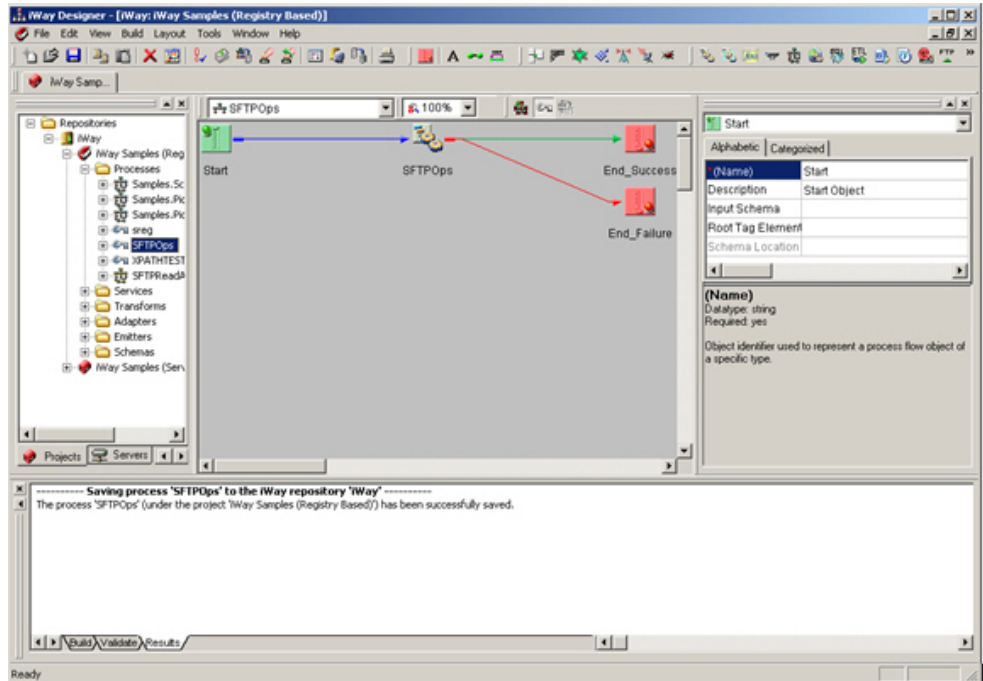
Reload Services OK Cancel Help

3. Test run the process flow SFTPOps.

The 'From' File test.txt is renamed to the 'To' file name i.e. testmove.txt.

Procedure: How to Test the SFTP Ops Service for a Prepend Operation

1. Create a process flow and add a service object for the SFTP File Operations service (com.ibi.agents.XDSFTPOpsAgent), as shown in the following image.



2. Set the Service object class name to com.ibi.agents.XDSFTPOpsAgent.

Set the properties as shown in the following image.

Name	Type	Properties	Debug Settings
Name	Value	Description	User Defined Properties
* Operation	prepend	Command to execute when agent goes active	
* File (from)	/users/edasxr/testdir/out6.txt	Originating file to be operated on. Relative or abs	
File (to)	/users/edasxr/testdir/testmove.txt	Originating file to be operated on. Wild cards acc	
* Host Name	edasol29	DNS name (or IP address) of the SFTP server tha	
* Remote Port	22	Port to connect to on the SFTP site, blank for del	
* User Name	edasxr	Is the valid user ID on the SFTP server	
Password		User's password on the SFTP server	
Public Key	C:\projects\SFTP\id_dsa	Path to the SFTP server's public key file.	
Passphrase		Passphrase used when the public key was gener	
Remote Si...		Special register designated to hold size. Required	
* Out Docu...	original	Document returned by operation (bad input defau	
* Action on ...	input	Whether input document or status document retu	
* Retry	0	If non-zero, will retry the operation n times at one-	

Reload Services OK Cancel Help

The From file /users/edasxr/testdir/out6.txt is:

```
this is a test
```

The to file /users/edasxr/testdir/testmove.txt is:

```
for sftpops agent
```

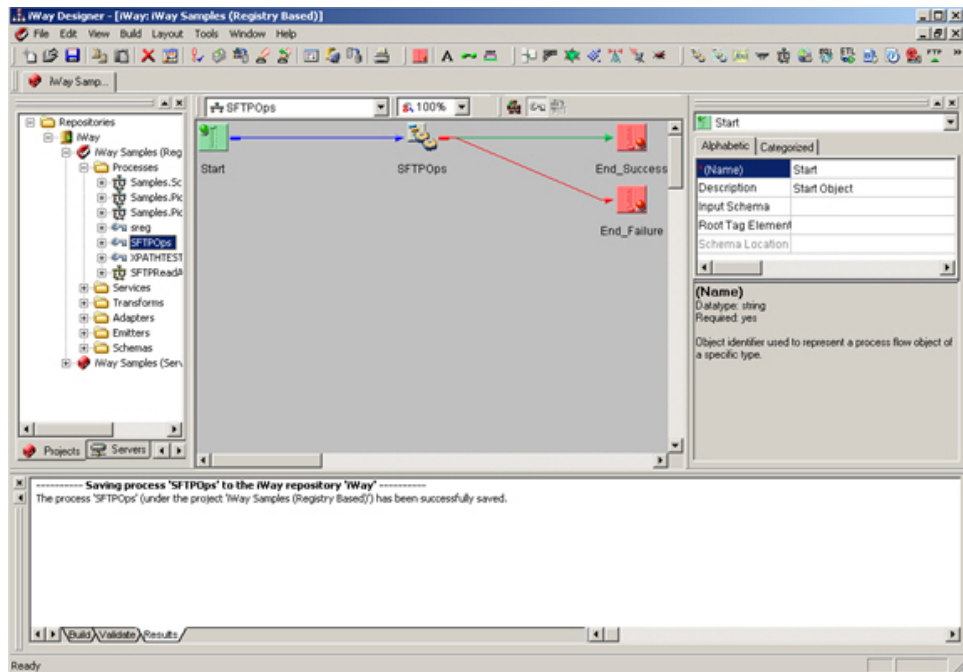
3. Test run the process flow SFTPOps.

The to file testmove.txt content is modified as:

```
this is a test for sftpops agent
```

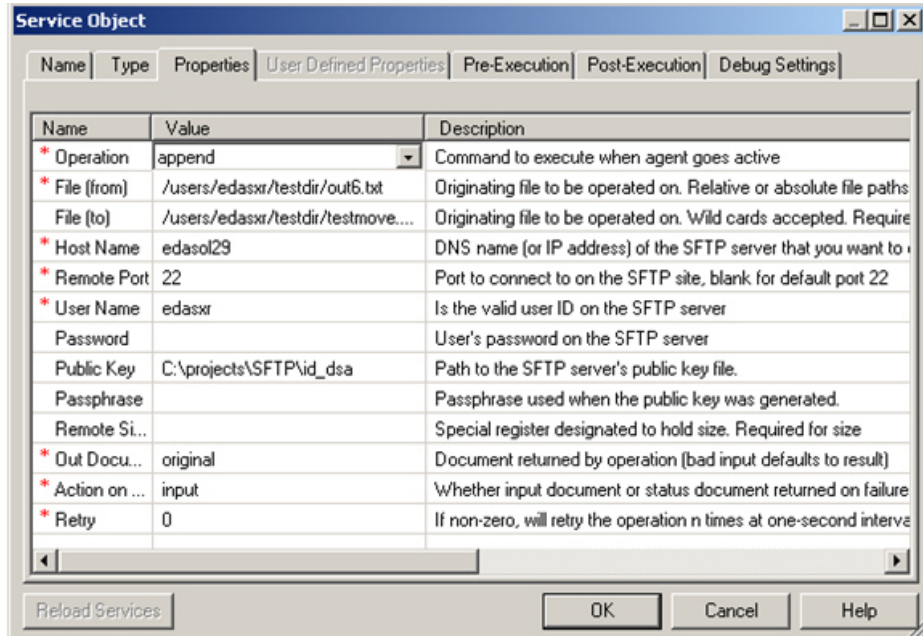
Procedure: How to Test the SFTP Ops Service for an Append Operation

1. Create a process flow and add a service object for the SFTP File Operations service (com.ibi.agents.XDSFTPOpsAgent), as shown in the following image.



2. Set the Service object class name to com.ibi.agents.XDSFTPOpsAgent.

Set the properties as shown in the following image.



The From file /users/edasxr/testdir/out6.txt is:

```
this is a test
```

The to file /users/edasxr/testdir/testmove.txt is:

```
for sftpops agent
```

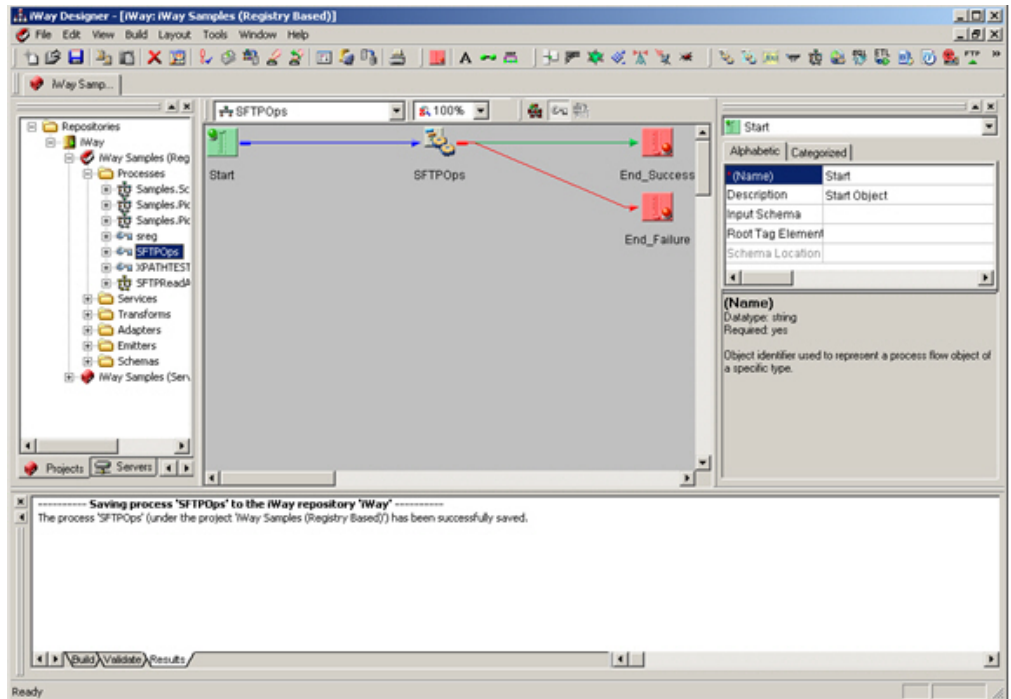
3. Test run the process flow SFTPOps.

The to file testmove.txt content is modified as:

```
this is a test for sftpops agent
```

Procedure: How to Test the SFTP Ops Service for a Delete Operation

1. Create a process flow and add a service object for the SFTP File Operations service (com.ibi.agents.XDSFTPOpsAgent), as shown in the following image.



2. Set the Service object class name to com.ibi.agents.XDSFTPOpsAgent.

Set the properties as shown in the following image.

Name	Type	Properties	User Defined Properties	Pre-Execution	Post-Execution	Debug Settings
Name						
* Operation	delete					
* File (from)	/users/edasxr/testdir/out6.txt					
File (to)						
* Host Name	edasol29					
* Remote Port	22					
* User Name	edasxr					
Password						
Public Key	C:\projects\SFTP\id_dsa					
Passphrase						
Remote Si...						
* Out Docu...	original					
* Action on ...	input					
* Retry	0					

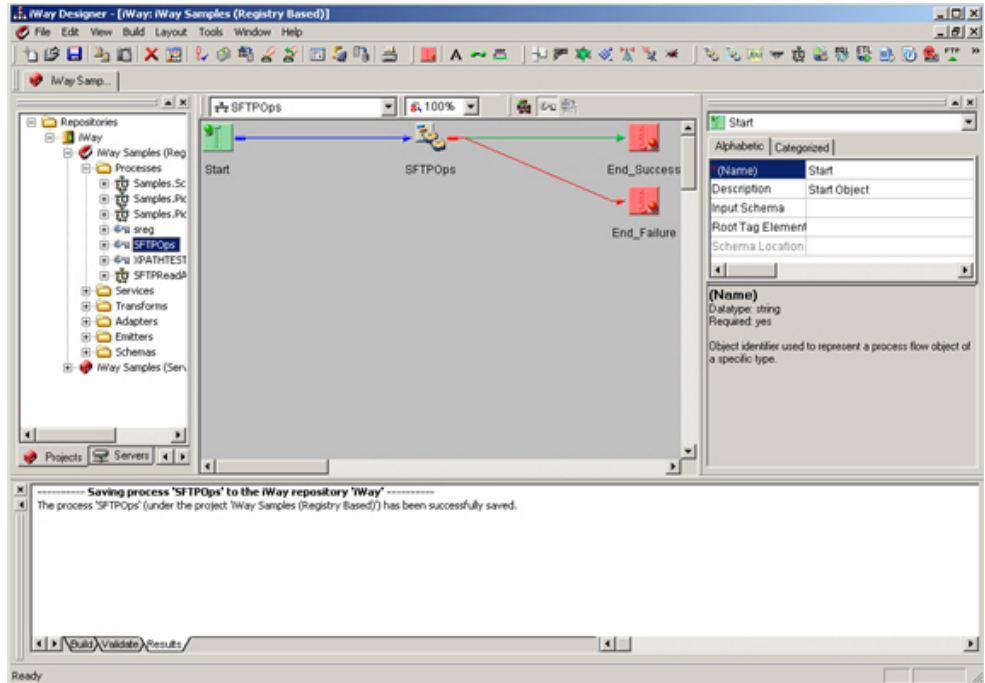
Reload Services OK Cancel Help

3. Test run the process flow SFTPOps.

The From file /users/edasxr/testdir/out6.txt is deleted.

Procedure: How to Test the SFTP Ops Service for a Rename Operation

1. Create a process flow and add a service object for the SFTP File Operations service (com.ibi.agents.XDSFTPOpsAgent), as shown in the following image.



2. Set the Service object class name to com.ibi.agents.XDSFTPOpsAgent.

Set the properties as shown in the following image.

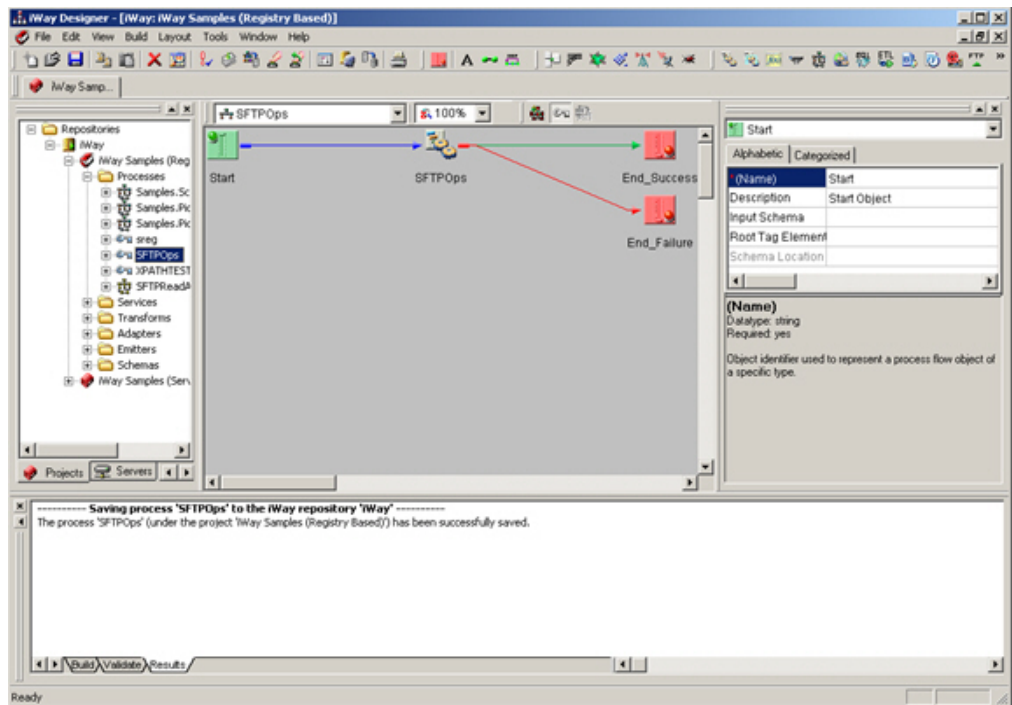
Name	Type	Properties	User Defined Properties	Pre-Execution	Post-Execution	Debug Settings
* Operation		rename				
* File (from)		/users/edasxr/testdir/testmove.txt				
File (to)		/users/edasxr/testdir/test.txt				
* Host Name		edasol29				
* Remote Port		22				
* User Name		edasxr				
Password						
Public Key		C:\projects\SFTP\id_dsa				
Passphrase						
Remote Si...						
* Out Docu...		original				
* Action on ...		input				
* Retry		0				

3. Test run the process flow SFTPOps.

The 'From' File test.txt is renamed to the 'To' file name i.e. testmove.txt.

Procedure: How to Test the SFTP Ops Service for a Size Operation

1. Create a process flow and add a service object for the SFTP File Operations service (com.ibi.agents.XDSFTPOpsAgent), as shown in the following image.



2. Set the Service object class name to com.ibi.agents.XDSFTPOpsAgent.

Set the properties as shown in the following image.

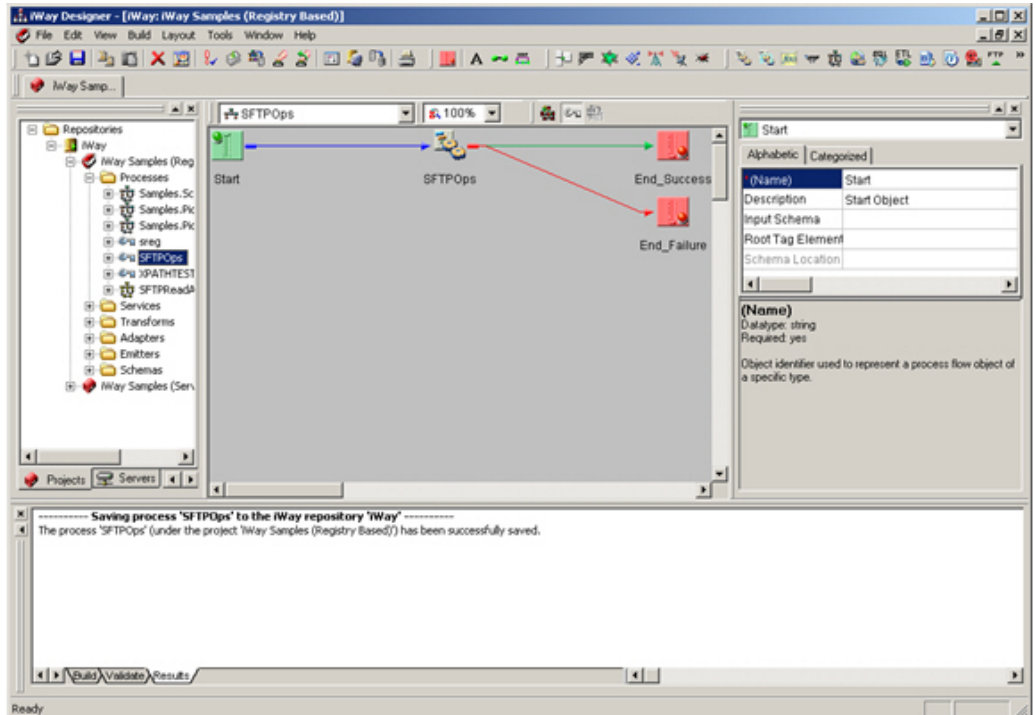
The screenshot shows the 'Service Object' configuration window with the 'User Defined Properties' tab selected. The window contains a table with columns 'Name', 'Value', and 'Description'. The properties are configured as follows:

Name	Value	Description
* Operation	copy	Command to execute when agent goes active
* File (from)	/users/edasxr/testdir/test.txt	Originating file to be operated on. Relative or absolute file paths
File (to)	/users/edasxr/testdir/out6.txt	Originating file to be operated on. Wild cards accepted. Required
* Host Name	edasol29	DNS name (or IP address) of the SFTP server that you want to
* Remote Port	22	Port to connect to on the SFTP site, blank for default port 22
* User Name	edasxr	Is the valid user ID on the SFTP server
Password		User's password on the SFTP server
Public Key	C:\projects\SFTP\id_dsa	Path to the SFTP server's public key file.
Passphrase		Passphrase used when the public key was generated.
Remote Si...		Special register designated to hold size. Required for size
* Out Docu...	original	Document returned by operation (bad input defaults to result)
* Action on ...	input	Whether input document or status document returned on failure
* Retry	0	If non-zero, will retry the operation n times at one-second intervals

At the bottom of the window, there are buttons for 'Reload Services', 'OK', 'Cancel', and 'Help'.

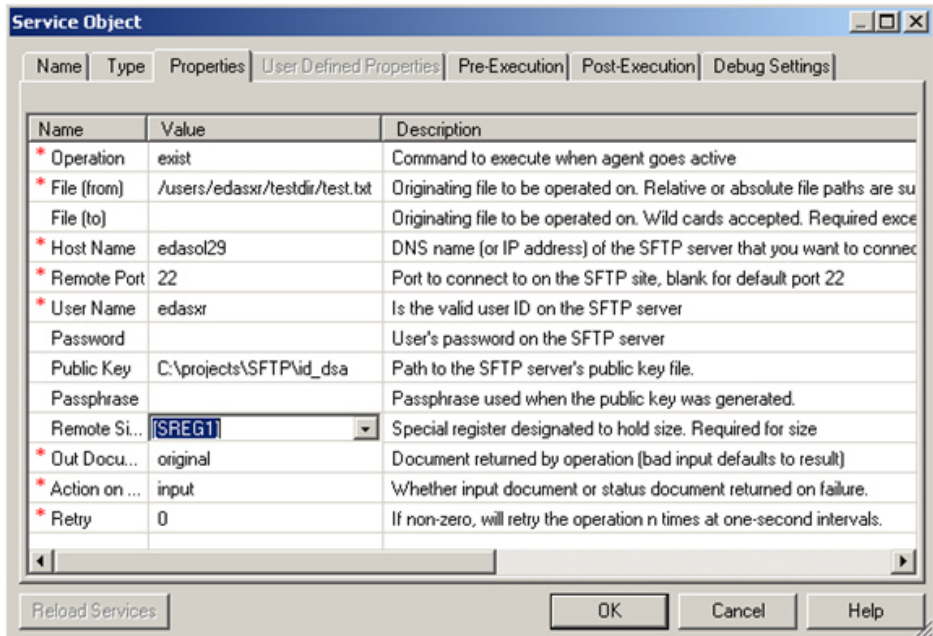
Procedure: How to Test the SFTP Ops Service for an Exist Operation

1. Create a process flow and add a service object for the SFTP File Operations service (com.ibi.agents.XDSFTPOpsAgent), as shown in the following image.



2. Set the Service object class name to com.ibi.agents.XDSFTPOpsAgent.

Set the properties as shown in the following image.



Configuring OpenSSH on Windows

OpenSSH is a set of applications providing encrypted communication sessions over a computer network using the SSH protocol.

Procedure: How to Install the OpenSSH Server on Windows

1. Install the typical version of the server, which can be downloaded from the following website:

<http://sourceforge.net/projects/sshwindows/>

2. Run the installer by selecting the default settings.

Note: No configuration is required during the installation.

The OpenSSH Server is installed under the following directory by default:

C:\Program Files\OpenSSH

Procedure: How to Configure an OpenSSH Server on Windows

1. Open a command prompt and navigate to the directory where OpenSSH Server is installed.
For example:

```
C:\Program Files\OpenSSH
```

2. Use the CD command navigate to the \bin subdirectory.
3. Use the mkgroup command to create a group permissions file.

For local groups, use the -l switch. For domain groups, use the -d switch.

For both domain and local, it is best to run the command twice (remember to use >>, not >). If you use both, make sure to edit the file to remove any duplicate entries.

```
mkgroup -l >> ..\etc\group      (local groups)
mkgroup -d >> ..\etc\group      (domain groups)
```

4. Use the mkpasswd command to add authorized users into the passwd file.

For local users, use the -l switch. For domain users, use the -d switch.

For both domain and local, it is best to run the command twice (remember to use >>, not >). If you use both, make sure to edit the file to remove any duplicate entries.

```
mkpasswd -l [-u <username>] >> ..\etc\passwd      (local users)
mkpasswd -d [-u <username>] >> ..\etc\passwd      (domain users)
```

Note:

- ☐ To add users from a domain that is not the primary domain of the machine, add the domain name after the user name.
 - ☐ Omitting the username switch adds ALL users from the machine or domain, including service accounts and the Guest account.
5. Enter the following command to start the OpenSSH server:

```
net start opensshd
```

6. Test the OpenSSH server.

Using a separate machine as the client is recommended. If you connect, but the connection is immediately dropped, then reboot the machine with the server and try to reconnect.

Note: The primary rule in using this utility is to only allow trusted users to have login permissions. The cygwin port of OpenSSH uses the full OpenSSH source code and the security of the program is not diluted.

Procedure: How to Setup an SSH Login Without a Password Using a Private Key

1. Connect to your SSH server (for example, edasol29), using your configured credentials.
2. Create an `.ssh` folder, under the default login directory. For example:

```
/users/[myusername]
```

Check permissions on your `~/ .ssh` folder and make sure to enter the following command if they are wrong:

```
chmod 700 .ssh
```

3. Generate the keys on the SSH server using the following command:

```
ssh-keygen -t dsa
```

or

```
ssh-keygen -t rsa
```

4. Accept the file names provided and enter a passphrase, if necessary.
5. Create an empty file `authorized_keys` under the `.ssh` folder and add public keys. For example:

```
mv id_dsa.pub authorized_keys
```

The `id_dsa` private key generated can be used to login without the password to the OpenSSH server.

6. Copy the private key (`id_dsa`) to your local Windows machine. You can use Winscp or SFTP.
7. Launch `puttygen.exe`. Under actions, select *load* and load the `id_dsa` file.
8. Enter the passphrase you set when you generated the key on the server. Puttygen will now convert the key to a format (`.ppk`), which is used for SFTP connections by most tools, such as putty, Winscp, and so on.

9. Save the file as

```
privatekey.ppk
```

10. Change your putty settings under *connection*, *SSH*, *auth* to use `privatekey.ppk`.
11. Try to connect. Enter the passphrase if you have one.

The /home Directory

In the `passwd` file, you will notice that the home directory of the user is set as `/home/username`, with `username` being the name of the account. In the default install, the `/home` directory is set to the default profile directory for all users. This is usually `C:\Documents and Settings` on Windows 2000 and XP, and `C:\WINNT\Profiles` on Windows NT 4.0. The location of `/home` can be edited to fit your special requirements by editing a registry key.

To change the Windows directory `/home` corresponds to, you will need to edit a registry entry under `HKEY_LOCAL_MACHINE\SOFTWARE\Cygnus Solutions\Cygwin\mounts v2\home`. The value of the key named `native` is the directory that `/home` is. If you want all your users to enter in a directory on your machine called `F:\Users`, change `native` to read `F:\Users`. By default, each user will then be placed in the directory `F:\Users\username`, where `username` is the name of the user account. To place the user directly under `F:\Users`, change the home directory password to `/home`.

Firewalls

The OpenSSH server listens for traffic on TCP port 22 by default. If your firewall setup does not allow connections on this port, it can be changed by editing the `etc/sshd_config` file.

Note: For additional troubleshooting and OpenSSH advanced configuration, refer to the `readme.txt` file, under `C:\Program files\OpenSSH\docs`.

Converting a Private Key to the OpenSSH Key Format

During implementations of the SFTP listener, you may be prompted to accept a public key from a SFTP server. In this scenario, you must ensure that the private key file being specified for the SFTP listener is generated using OpenSSH key format. Otherwise, the SFTP listener will be unable to open the file. Usually, private key files that are generated for PuTTY interfaces have a `.ppk` file extension. You must convert the private key file to OpenSSH key format by using the PuTTY Key Generator tool, which can be downloaded from the following website:

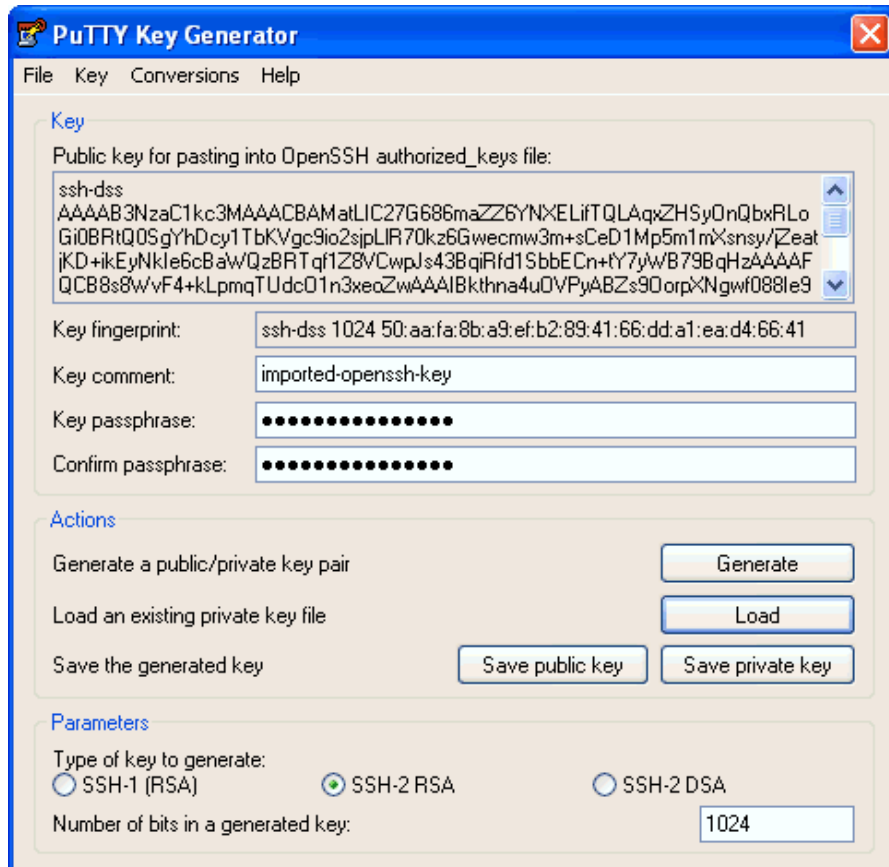
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Once you download the PuTTY Key Generator tool, perform the following steps.

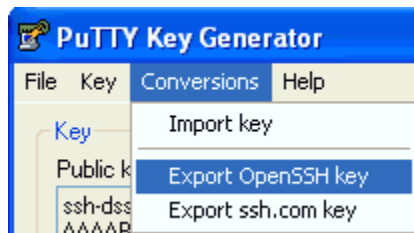
Procedure: How to Convert a Private Key to the OpenSSH Key Format

To convert a private key to the OpenSSH key format using the PuTTY Key Generator tool:

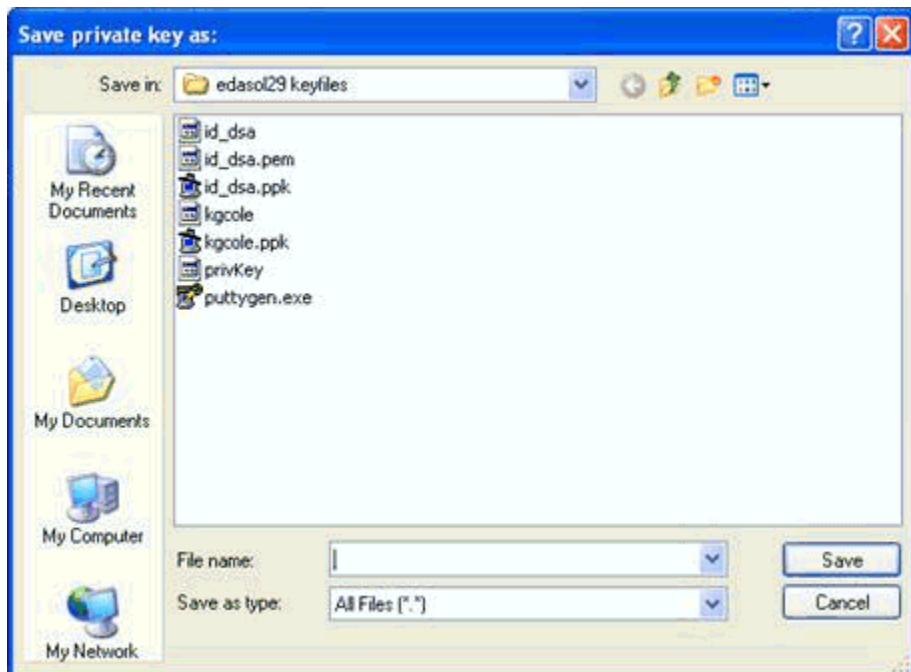
1. Open the PuTTY Key Generator tool, which is shown in the following image.



2. Enter a passphrase when prompted.
3. Click *Conversions* from the menu bar and select *Export OpenSSH key*, as shown in the following image.



The Save private key as dialog box opens, as shown in the following image.



4. Navigate to a directory on your file system that is accessible by iWay Service Manager.
5. Enter a file name for the private key followed by the .pem extension to distinguish it from .ppk PuTTY key files.
6. Click Save.

Configuring FTP Server Components

This section describes how to configure FTP Server components using iWay Service Manager.

In this chapter:

- ☐ [FTP Server Component Configuration Overview](#)
 - ☐ [Configuring a FTP Server Listener](#)
 - ☐ [FTP Commands](#)
 - ☐ [Security Considerations](#)
 - ☐ [Using the Trading Partner Management Facility](#)
 - ☐ [Configuring a FTP SREG Service](#)
-

FTP Server Component Configuration Overview

iWay offers an FTP server that is designed to extend the processing capabilities of iWay Service Manager (iSM). Although this service can be configured to operate as a standard FTP server, the purpose of this service is transaction receipt and mailboxing. The FTP Server listener listens on the port that is specified in the FTP server settings. The listener begins processing messages when a file is sent through FTP to iWay Service Manager. Therefore, it differs from general FTP servers in several ways:

- ☐ User login security is managed through the iSM security facilities. Full function evaluation of all parameters enables the storage of attributes, such as passwords, in a security directory, such as LDAP.
- ☐ User attributes can be stored in the Partner Management system of iWay Trading Manager, if installed.
- ☐ File actions for Get and Put can be treated as messages. In this case, they are immediately processed through the appropriate configuration.
- ☐ Messages received for execution can optionally be safe-stored with their context to prevent loss in the event of failure. Safe-stored messages are reloaded when the listener is started, and executed at that time. Only when all safe-stored messages have been processed does the listener open to receive new messages.

- ❑ Input-streaming is supported when the input is to be treated as a message. In input-stream mode, a large document can be broken into parts as it is received. Each part is extracted, based upon the message type configured using a standard streaming preparser, and processed as a separate message.
- ❑ Tracing and diagnostics are available using the server tracing system.
- ❑ Messages can be stored in the iWay Audit Manager logs for later analysis.

Configuring a FTP Server Listener

To configure a FTP Server listener:

1. Perform the steps as described in [Configuring Listeners](#) on page 177.
2. Ensure that you select *FTP[S] Server* as the listener type you are configuring.

For a complete description of the configuration parameters that are available for the FTP Server listener, see [FTP Server Listener Configuration Parameters](#) on page 122.

For a complete description of the FTP Server listener Special Registers (SREGs), see [FTP Server Listener Special Registers](#) on page 125.

Reference: FTP Server Listener Configuration Parameters

The following table lists and describes parameters for the FTP Server listener.

Note: Parameters that are common to FTP Server listeners are described in [Listener Configuration Parameters](#) on page 173.

Parameter	Definition
Port*	The TCP port for receipt of FTP requests. FTP standard is port 21.
Passive Port Range	Range of ports to which to limit the PASV/EPST command. This field may contain a range of ports from 0 through 65535 separated by a dash character (-). For example, 30000-40000 or 2200-3000. No entry provides unrestricted access, and any available port between 0 and 65535 may be used. Use this parameter if your server or firewall environment restricts the assignable ports controlled by the server for passive FTP requests.

Parameter	Definition
Local Bind Address	The local bind address for multi-homed hosts. Usually leave empty.
FTP Server Log	A log of server commands received, and the results of each command.
Server Root	The base directory for this FTP Server. When user mailbox paths are relative, they are below this directory.
Allow Anonymous	If set to <i>true</i> , then anonymous login is permitted. Anonymous users inherit the default read and write security.
Use Safestore	The Safestore preserves store requests while the incoming document is passing through execution. It is not meaningful for direct writes of messages to the file system. Using safestore can reduce system performance.
Users	
Authentication Realm	Enter the name of the configured Authentication Realm provider. If left blank, then the user and their authentication credentials must be maintained in the FTP Server user repository.
Repository Type	This determines how the user repository is stored. The repository can be stored either as an XML file or as a JDBC database.
Security File [JDBC URL] (required)	The security file location. The security file describes users permitted to exchange messages with this server. It also describes their mailbox and security characteristics. This is a JDBC or a file URL depending upon the repository type.
JDBC Provider	The name of the iWay JDBC provider.
JDBC Driver	The JDBC driver to use for accessing the repository. This is required if repository type is JDBC.
JDBC User Name	The password for accessing the JDBC repository. Required if repository type is JDBC.

Parameter	Definition
JDBC Password	<p>Select one of the following values from the drop-down list:</p> <p><input type="checkbox"/> column</p> <p><input type="checkbox"/> field</p> <p><input type="checkbox"/> row</p>
Default Permissions	
Default Can READ	If set to <i>true</i> , then users without specific security can read.
Default Can WRITE	If set to <i>true</i> , then users without specific security can write.
Action on GET	How the server should treat file retrieve type requests from the client.
Action on PUT	How the server should treat file store type requests from the client.
SITE EXEC	If set to <i>true</i> , clients can execute processes using the SITE EXEC command.
Security	
Session Timeout *	If a value greater than 0 is set, then that value is the maximum number of seconds between commands before the session automatically times out.
Require Secure Auth	The login must be through a secure authorization channel. You will need to configure the Keystore under HTTPS section of the system properties if client authentication is required. Note, if keystore is configured in system properties make sure it has the CA certificate or the client certificate of the server you are connecting to. If keystore is not configured in system properties default truststore located under /lib/security/cacerts will be used.
Security Provider	Enter the iWay keystore or SSL context security provider name. If this component is secure and no value is entered, then the default security provider is used.

Parameter	Definition
Require Secure Transfer	The data transfers must be through a secured channel. You must configure the keystore under HTTPS section of the system properties if client authentication is required. Note, if keystore is configured in system properties make sure it has the CA certificate or the client certificate of the server you are connecting to. If keystore is not configured in system properties default truststore located under /lib/security/cacerts will be used.
Use 128-bit Encryption	This enforces the use of 128-bit encryption for all TLS channels.
Security Protocol	The minimal security protocol supported for FTPS.

Note: The FTP Server listener supports streaming. Streaming is used for large documents or documents for which the application needs to split the input into sections under the same transaction. For more information on streaming and configuring streaming preparers, see the *iWay Service Manager Component and Functional Language Reference Guide*.

Reference: FTP Server Listener Special Registers

The following table lists and describes the Special Registers (SREGs) available on the FTP Server listener.

Name	Level	Type	Description
ftpd.command	Document	String	The name of the command.
ftpd.file	Document	String	The file name part of the command.
ftpd.mode	Document	String	The current transfer mode (ASCII or Binary).
ftpd.user	Document	String	The user ID of the sender.
ip	Document	String	The IP address of the sender.
iwayconfig	System	String	The current active configuration name.
msgcount	Document	String	The number of messages executed.

Name	Level	Type	Description
msgsize	Document	Integer	The physical length of the message payload.
name	System	String	The assigned name of the master (listener).
pdm	Document	String	If this is a possible duplicate message, 1 is displayed.
protocol	System	String	The protocol on which message was received.
source	Document	String	The host name of the sender.
tid	Document	String	Unique transaction ID.

Action on GET

Client GET (and MGET) requests the return of information to the client by the server. A GET client for a file can be handled differently, depending on the current settings for the action. The default setting is configured for the listener but it can be overridden for any specific user in the security file.

1. Use File System

This is a standard FTP retrieval. Data is returned from the named file in the file system.

For example:

```
ftp> get test.xml
200 PORT command successful.
150 ASCII data connection for test.xml (172.30.246.86,4071) (42 bytes).
226 ASCII transfer complete (42 bytes).
ftp: 42 bytes received in 0.00Seconds 42000.00Kbytes/sec.
```

Test.xml was retrieved from the FTPServer and sent to the FTP client.

2. Execute as a Message

A standardized XML ftpd request document is constructed containing the request. This document is passed to the system for execution. The result of the execution is returned to the client.

For example, create a process flow that contains a MQEmit service object. Add the process flow to the FTPServer channel.

```
ftp> get test.xml
200 PORT command successful.
150 ASCII data connection for /test.xml (172.30.246.86,4150) (0 bytes).
226 ASCII transfer complete
ftp: 1144 bytes received in 0.05Seconds 24.34Kbytes/sec.
```

The following document is constructed and passed to the process flow.

```
<ftpd user="user" type="get">
<command>retr</command>
<parm>/test.xml</parm>
</ftpd>
```

The process flow emits the above document to a MQ queue. The status document (shown below) is returned to the FTP client as test.xml.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<emitstatus status="0">
<protocol>MQ</protocol>
<parms>
<parm name="attrs">true</parm><parm name="ccsid"/>
<parm name="channel">SYSTEM.DEF.SVRCONN</parm>
<parm name="cipherspec">None</parm><parm name="correlid"/>
<parm name="correlidtag"/><parm name="expiry"/>
<parm name="format"/><parm name="host"/>
<parm name="localtrans">true</parm>
<parm name="manager">qm7</parm>
<parm name="neverUseMaster">true</parm>
<parm name="nopreemit">true</parm>
<parm name="persist">queue</parm>
<parm name="port">1414</parm>
<parm name="priority"/>
<parm name="queue">out</parm>
<parm name="reportq"/>
<parm name="reqcoa">>false</parm>
<parm name="reqcod">>false</parm>
<parm name="return">status</parm>
<parm name="type">datagram</parm>
<parm name="userregs">>false</parm>
<parm name="version">default</parm>
<parm name="wanteos">>false</parm>
</parms>
<timestamp>2010-11-29T21:25:51.800Z</timestamp>
<status>0</status>
<channel>ftpserver</channel>
<nodename>com.ibm.agents.XDMQEmitAgent</nodename>
<msgid>base64(AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA)</msgid>
<native>0</native>
<correlid>base64(QU1RIHFtNyAgICAgICAgIBIu7UwgPXwF)</correlid>
</emitstatus>
```

3. Deny Access to This Service

The user may not execute a retrieval.

For example:

```
ftp> get test.xml
200 PORT command successful.
553 'test.xml': cannot read.
Test.xml was not retrieved.
```

4. Use File System Then Execute a Message

The retrieval operates as a standard FTP retrieval. Data is returned from the named file in the file system. Then, a standardized XML ftpd request document is constructed containing the request. This document is passed to the system for execution. The result of the execution is sent to the configured emitter.

For example, create a process flow that contains a MQEmit service object. Add the process flow to the FTPServer channel.

```
ftp> get test.xml
200 PORT command successful.
150 ASCII data connection for /test.xml (172.30.246.86,4150) (0 bytes).
226 ASCII transfer complete
ftp: 1144 bytes received in 0.05Seconds 24.34Kbytes/sec.
```

Test.xml was retrieved from the FTPServer and returned to the FTP client. In addition, the following document is constructed:

```
<ftpd user="user" type="get">
<command>retr</command>
<parm>/test.xml</parm>
</ftpd>
```

The document is passed to the process flow. The process flow emits the above document to a MQ queue. The following status document is returned from the process flow and sent to any configured emitters:


```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<emitstatus status="0">
<protocol>MQ</protocol>
<parms>
<parm name="attrs">true</parm><parm name="ccsid"/>
<parm name="channel">SYSTEM.DEF.SVRCONN</parm>
<parm name="cipherspec">None</parm><parm name="correlid"/>
<parm name="correlidtag"/><parm name="expiry"/>
<parm name="format"/><parm name="host"/>
<parm name="localtrans">true</parm>
<parm name="manager">qm7</parm>
<parm name="neverUseMaster">true</parm>
<parm name="nopreemit">true</parm>
<parm name="persist">queue</parm>
<parm name="port">1414</parm>
<parm name="priority"/>
<parm name="queue">out</parm>
<parm name="reportq"/>
<parm name="reqcoa">false</parm>
<parm name="reqcod">false</parm>
<parm name="return">status</parm>
<parm name="type">datagram</parm>
<parm name="userregs">false</parm>
<parm name="version">default</parm>
<parm name="wanteos">false</parm>
</parms>
<timestamp>2010-11-29T21:25:51.800Z</timestamp>
<status>0</status>
<channel>ftpsvr</channel>
<nodename>com.ibm.agents.XMQEmitAgent</nodename>
<msgid>base64(AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA)</msgid>
<native>0</native>
<correlid>base64(QU1RIHFtNyAgICAgICAgIBIu7UwgPXwF)</correlid>
</emitstatus>

```

Action on PUT

Client PUT (including append) cause the transfer of information from the client to the server. A client PUT request can be handled differently, depending upon the current setting for this action. The default setting is configured for the listener but it can be overridden for any specific user.

1. Use File System

The request operates as a standard FTP store command. Data is stored from the named file in the file system.

For example:

```
ftp> put test.xml
200 PORT command successful.
150 ASCII data connection for /test.xml (172.30.246.86,4242).
226 ASCII /test.xml transfer complete (42 bytes)
ftp: 42 bytes sent in 0.00Seconds 42000.00Kbytes/sec.
```

Test.xml was sent to the FTPServer and saved in the server root directory.

2. Execute as a Message

The data is passed, as a document, into the system for execution. A standardized XML document is constructed containing the data. This document is passed to the system for execution.

For example, create a process flow that contains a MQEmit service object. Add the process flow to the FTPServer channel.

```
ftp> put test.xml
200 PORT command successful.
150 ASCII data connection for test.xml (172.30.246.86,4288).
226 'STOR EXEC entered for test.xml': was successful.
ftp: 42 bytes sent in 0.00Seconds 42000.00Kbytes/sec.
```

Test.xml was sent to the FTPServer. It is then passed to the configured process flow which emits the document to an MQ queue.

A 451 error code is returned to the FTP client when an error occurs in the process flow. For example:

```
ftp> put test.xml
200 PORT command successful.
150 ASCII data connection for test.xml (172.30.246.86,1060).
451 'STOR EXEC entered for test.xml': had an execution error!
ftp: 1144 bytes sent in 0.00Seconds 1144000.00Kbytes/sec.
```

An example of this type of error is when an MQEmit service object attempts to send a document to a queue that does not exist.

3. Deny Access to This Service

This user may not store a file.

For example:

```
ftp> put test.xml
200 PORT command successful.
553 'test.xml': cannot write. (DENY)
```

Test.xml was not sent to the FTPServer.

4. Use File System Then Execute a Message

The retrieval operates as a standard FTP store command. Data is stored from the named file in the file system. Then, a standardized XML ftpd request document is constructed containing the request. This document is passed to the system for execution. The result of the execution is sent to the configured emitter.

For example, create a process flow that contains a MQEmit service object. Add the process flow to the FTPServer channel.

Test.xml was sent to the FTPServer and saved in the server root directory. In addition, the following document is constructed:

```
<ftpd user="user" type="signal">  
<command>STOR</command>  
<parm>/test.xml</parm>  
<status>226 ASCII /test.xml transfer complete (42 bytes)</status>  
<path>c:\ftpserver\test.xml</path>  
<vpath>/test.xml</vpath>  
<transfersize>42</transfersize>  
</ftpd>
```

The document is passed to the process flow. The process flow emits the above document to a MQ queue. The status document (shown below) is returned from the process flow and sent to the emitter (if one is configured).

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<emitstatus status="0">
<protocol>MQ</protocol>
<parms>
<parm name="attrs">true</parm><parm name="ccsid"/>
<parm name="channel">SYSTEM.DEF.SVRCONN</parm>
<parm name="cipherspec">None</parm><parm name="correlid"/>
<parm name="correlidtag"/><parm name="expiry"/>
<parm name="format"/><parm name="host"/>
<parm name="localtrans">true</parm>
<parm name="manager">qm7</parm>
<parm name="neverUseMaster">true</parm>
<parm name="nopreemit">true</parm>
<parm name="persist">queue</parm>
<parm name="port">1414</parm>
<parm name="priority"/>
<parm name="queue">out</parm>
<parm name="reportq"/>
<parm name="reqcoa">false</parm>
<parm name="reqcod">false</parm>
<parm name="return">status</parm>
<parm name="type">datagram</parm>
<parm name="userregs">false</parm>
<parm name="version">default</parm>
<parm name="wanteos">false</parm>
</parms>
<timestamp>2010-11-29T21:25:51.800Z</timestamp>
<status>0</status>
<channel>ftpserver</channel>
<nodename>com.ibm.agents.XDMQEmitAgent</nodename>
<msgid>base64(AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA)</msgid>
<native>0</native>
<correlid>base64(QU1RIHFtNyAgICAgICAgIBIu7UwgPXwF)</correlid>
</emitstatus>
```

Process Flow Fails in FTP Server

Applications should, of course, always be designed to handle errors and manage the return from the listener to the client. Should a process flow report a failure, however, the FTP Server listener presents the error to the FTP client following standards of FTP, with a resultant error code of 451. Given an FTP Server listener configured to EXEC the message to a process flow which fails of the GET EXEC, the transient file on the server is not deleted, and the response seen by a line client (in this case Microsoft FTP client) is as follows:

```
ftp> get ftpserver036_trans_fail.txt
200 EPRT command successful.
150 ASCII data connection for /signal/ftpserver036_trans_fail.txt
(0:0:0:0:0:0:1,53940) (0 bytes).
451 ASCII Errors in process were reported.
ftp: 390 bytes received in 0.05Seconds 7.22Kbytes/sec.
ftp>
```

The Security File

The security file (in XML format) is loaded when the listener is initialized. This XML file describes the users that are allowed and their attributes. The XML file contains one `<user>` element for each authorized user, and attributes of the `<user>` element are used to describe the user. Each user can be a member of one group, which applies the access group criteria that can then be modified on an individual basis. The format for the `<access>` element allows the specification of the `use` attribute. The name of the directory to which the access applies is entered as the value for the `<access>` element.

Any user can have controlled access. Permissions are applied to the specified subdirectory, which is below the document root of a user, and to all remaining directories below this subdirectory. If a permission is not specified, it is taken from the specification of a parent. Permissions are specified by the `use` attribute of the `<access>` child of the user. Possible access types are read ('r'), and write ('w'). Read permission permits the client to list directories and access files. Write permission permits file storage and the ability to manipulate directories. Either type can be preceded by a minus sign (-), which removes the permission. A plus sign (+) is implied if the minus is omitted.

To add read and write permissions, specify:

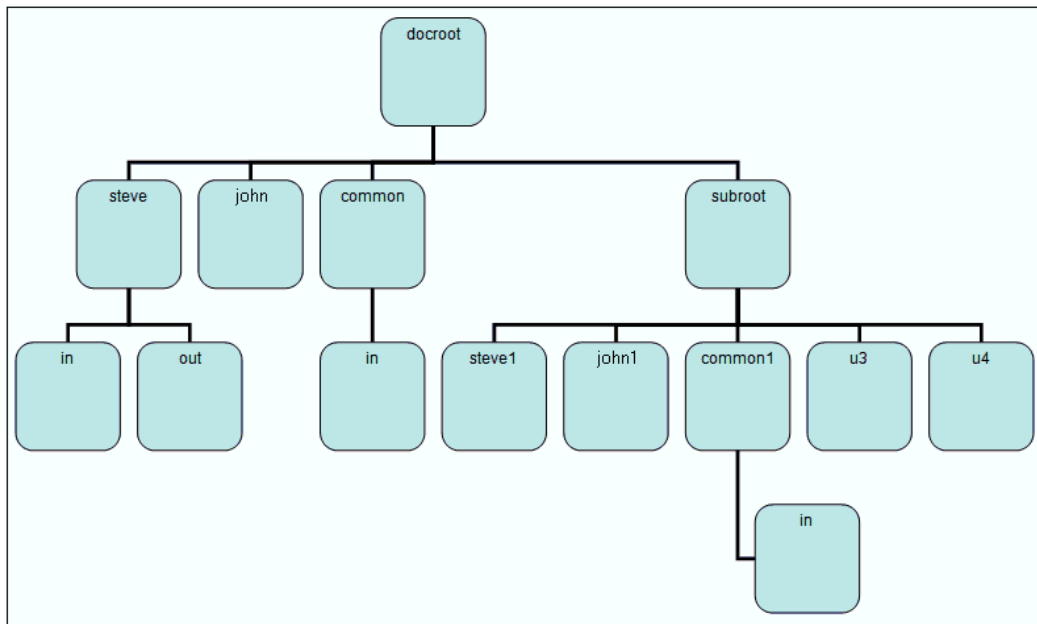
```
use= 'rw'
```

To accept the read permission of the parent and turn off write permission, specify:

```
use= '-w'
```

The order of the characters is not significant.

The following diagram illustrates a sample FTP directory structure that is referred to in the XML security file example.



The following is a sample XML file that represents the content of a security file:

```

<ftp>
  <users>
    <!-- ul user has same permissions as g1 group,
          but set at user level
          -->
    <user password="pw" init="/steve" use="rw">ul
      <access use="rw-du">/steve/in</access>
      <access use="r-w-d">/steve/out</access>
      <access use="v-r-w">/common</access>
      <access use="r">/common/in</access>
      <access use="-r-w-v">/*</access>
    </user>

    <!-- user john is exactly like ul (except that he starts in /john
directory)
          but permissions from group
          -->
    <user password="pw" group="g1">john</user>

    <!-- user steve is exactly like john, with two exceptions
(overriding group defaults)
          -->
    <user password="pw" group="g1">steve
      <access use="rwd">/steve/in</access>
      <access use="rw">/common/in</access>
    </user>

    <!-- user u2 is like group g2, with specified home directory below
docroot.
          Note that when home is used,
          init and permission paths are from the user (or group) home,
NOT from docroot,
          while home is from docroot.
          -->
    <user password="pw" init="/stevel" home="/subroot" use="v-r-w-d">u2
      <access use="rw-du">/stevel/in</access>
      <access use="r-w-d">/stevel/out</access>
      <access use="v-r-w">/commonl</access>
      <access use="r">/commonl/in</access>
      <access use="-v-r-w">/*</access>
    </user>

    <!-- john1 has g2 permissions
          -->
    <user password="pw" group="g2">john1</user>

```

```

    <!-- stevel has g2 permissions with override.  keep in mind that g2
           specifies non-docroot home, so paths
           are from home, not docroot.
    -->
    <user password="pw" group="g2">stevel
      <access use="rwd">/stevel/in</access>
      <access use="rw">/common1/in</access>
    </user>

    <!-- u3 is similar to u2, except starts in /subroot, no init pos. --
>
    <user password="pw" home="/subroot" use="vw-d-ru">u3
      <access use="rwduv">/u3</access>
      <access use="-r-w-v">/*</access>
    </user>

    <user password="pw" group="g3">u4</user>

  </users>

  <groups>
    <!-- g1 group has same permissions as u1,
           generalized with $user pattern
    -->
    <group init="/$user" use="v-r-w-d">g1
      <access use="rw-d">/$user/in</access>
      <access use="r-w-d">/$user/out</access>
      <access use="v-r-w">/common</access>
      <access use="r">/common1/in</access>
      <access use="-r-w-v">/*</access>
    </group>

    <!-- g2 group much like g1, but home dir is /subroot
           and all paths are beneath it
    -->
    <group init="/$user" home="/subroot" use="v-r-w-d">g2
      <access use="rw-d">/$user/in</access>
      <access use="r-w-d">/$user/out</access>
      <access use="v-r-w">/common1</access>
      <access use="r">/common1/in</access>
      <access use="-v-r-w">/*</access>
    </group>

    <!-- log into (virtual) root w/write, but not read permission.
    change to use subdirectory with full priv. except overwrite, but venture
    nowhere else -->
    <group home="/subroot" use="vw-d-ru">g3
      <access use="rwduv">/$user</access>
      <access use="-r-w-v">/*</access>
    </group>
  </groups>
</ftp>

```


Note: This sample XML security file describes several different types of users, illustrating some common usage patterns.

For the users u1, steve, and john in the sample security file:

- ☐ All have virtual file systems that begin in the docroot of the FTP server.
- ☐ All have initial positions other than the root of their virtual files systems. That is, after logging in, each of these users is positioned in the directory specified by their *init* attribute.
- ☐ All can change to and list the contents of, but cannot read or write files in the docroot and /common directories.
- ☐ All have some level of permission to access files in the /common/in directory.
- ☐ All have some level of permission in their init directories and below.
- ☐ None have any other access.

Users u1 and john differ only in that permissions of u1 are assigned to him directly, while john inherits them from the g1 group. User steve differs from john because he has access permissions for the /common/in and /steve/in directories that override the group g1 defaults.

For the users u2, steve1, and john1 in the sample security file:

- ☐ All have virtual file systems that begin below the FTP server's docroot, in /subroot. This means that they have no access whatsoever outside of /subroot. When logged in, /subroot appears to them as /.
- ☐ In other respects, these users have access permissions like u1, steve, and john. Users steve and john inherit access permissions from group g2.

For the users u3 and u4 in the sample security file:

- ☐ Like u2, steve1, and john1, these users have virtual file systems that begin in /subroot. However, since no initial position is specified, after login, these users are positioned in /subroot, which appears to them as /.
- ☐ These users can write files to /, but cannot read, overwrite, or delete.
- ☐ Each of these users has a private directory below /subroot, where they can read, write, and delete, but cannot overwrite existing files.
- ☐ Neither user has access to any other subdirectory of /subroot.
- ☐ User u4 differs from u3 in that their access permissions are inherited from group g3, while u3 has permissions assigned directly.

Permission Flags and What They Mean

The following table lists and describes the permission flags that are used in the sample security file:

Flag	Description
v	Signifies <i>visit and view</i> access. Without the v flag set on a directory, a user cannot change to that directory or list its contents. The v flag is set by default and must be revoked explicitly.
r	Signifies that a user can retrieve files from the specified directory.
w	Signifies that a user can store files in the specified directory.
d	When <i>d</i> is set on a directory, a user can delete files from that directory.
u	When a directory has the <i>u</i> flag set, a user cannot overwrite a file in that directory.
m	Signifies that the user can modify directories, by creating new directories and renaming existing ones. The -m flag is set by default and must be granted explicitly.

Note: The '-' character revokes the specific permission. For example, -d revokes the delete permission for that directory, -r revokes the read permission.

Access Tree Notes:

- ☐ Permissions of a user for the root of their virtual file system are specified by the *use* attribute on the user element or on the group to which the user belongs. Do not add an access element for the virtual root (/).
- ☐ Permissions of directory are inherited by its subdirectories unless they are explicitly overridden. For example, if the *use* attribute flags for /steve are *v-r-w-d* and use for /steve/in is set as *r*, then the permissions of the user in /steve/in are *vr-w-d*.
- ☐ When assigning directory permissions to a user, use paths corresponding to the virtual file system of the user. For example, if the home of the user is /subroot and you want to set permissions for /subroot/common1, the path to use is /common1.

Procedure: How to Migrate User Configurations From XML to RDBMS

User configuration information can be stored in any relational database that is accessible through JDBC. To configure the user database:

1. Create the database using any tool that can execute a DDL script. DDL scripts for MySQL, HSQLDB, and MSSQL are supplied and can be provided for other RDBMS as requested.
2. From the iSM command line, run the MigrateFTPUsers tool.
3. Type *MigrateFTPUsers* along with the command line arguments that are listed and described in the following table:

Command	Description
userfile	Path to the XML user configuration file you wish to migrate.
op: operation	Specify one of the following operations: <input type="checkbox"/> delete. Removes users, groups, and templates in the user file from the user database. <input type="checkbox"/> upsert. Inserts or update users, groups, and templates found in file.
driver	Name of JDBC driver class.
url	JDBC connection URL for the user database.
u	User ID that is used to connect to the user database. Must have insert/update/delete permission for user database tables.
pwd	Password that is used to connect to the user database.

4. Alternatively, you can use the p switch to point to a standard Java properties file that sets properties driver, url, user, and password to be used in the connection to the database.

For example:

```
Enter command:>tool MigrateFTPUsers -userfile c:/working/ftptest/
userfile.xml -driver com.mysql.jdbc.Driver -url
jdbc:mysql:///iwnftp -u root -pwd harrison -op upsert
```

or:

```
Enter command:>tool MigrateFTPUsers -p c:/working/mp.txt -userfile c:/
working/ftptest/userfile.xml -op
upsert
```

The mp.txt file that is referred to in this example can have the following structure and format:

```
driver=com.mysql.jdbc.Driver
url= jdbc:mysql:///itm_db
user=steve
password=secret
```

5. Perform the following steps to configure the FTPServer listener to use the user database for security information:
 - a. Set the Repository Type property to *JDBC*.
 - b. Set the Security File property to the JDBC connection URL for the user database.
 - c. Set driver, user, and password properties appropriately.

RDBMS User Configuration Tables

User information can be stored in a small relational database consisting of four tables:

- ☐ iwftp_owners
- ☐ iwftp_permissions
- ☐ iwftp_user_hosts
- ☐ iwftp_templates

iwftp_owners

The iwftp_owners table defines users and groups that can access the server. The fields are listed and described in the following table:

Field	Type	Description
iw_ownername	varchar(45)	The name (identifier) of the user or group.
iw_ownertype	char(1)	Specify G for group or U for user.
iw_password	varchar(45)	Password for FTP login, required for User (not for group).
iw_isanon	integer	Specify 1 if this user definition should be used for anonymous logins, or 0 otherwise.
iw_fullname	varchar(45)	The long (full) identifier for the user.

Field	Type	Description
iw_partner	varchar(45)	The reference to a partner agreement defined in iWay Trading Manager.
iw_groupname	varchar(45)	For users, the group to which they belong.
iw_comment	varchar(45)	The documentation for the owner.
iw_home	varchar(45)	The home directory is the root of the virtual file system for the owner.
iw_initialposition	varchar(45)	The directory where the owner should be positioned at login.
iw_getaction	integer	The action of the server on RETR command (0 is system default, 1 is exec, 2 is signal, and 3 is file).
iw_putaction	integer	The action of the server on STOR command (0 is system default, 1 is exec, 2 is signal, and 3 is file).
iw_canread	integer	Default read permission for this owner (0 for no, 1 for yes, and 2 for not set).
iw_canwrite	integer	Default write permission for this owner (0 for no, 1 for yes, and 2 for not set).
iw_candelete	integer	Default delete permission for this owner (0 for no, 1 for yes, and 2 for not set).
iw_canoverwrite	integer	Default overwrite permission for this owner (0 for no, 1 for yes, and 2 for not set).
iw_canvisit	integer	Default visit permission for this owner (0 for no, 1 for yes, and 2 for not set).
iw_canmod	integer	Default directory modify permission for this owner (0 for no, 1 for yes, and 2 for not set).

iwftp_permissions

The iwftp_permissions table defines resources to which an owner has access and the specific constraints on that access. The fields are listed and described in the following table:

Field	Type	Description
iw_owner	varchar(45)	User or group to which this permission entry applies. Foreign key for iwftp_owners.iw_ownershipname.
iw_ownershiptype	char(1)	Specify G for group or U for user. Foreign key for iwftp_owners.iw_ownershiptype.
iw_resource	varchar(45)	Name of the directory for which access should be defined. Enter with an absolute path using the virtual file system of the owner.
iw_canread	integer	Default read permission for this owner (0 for no, 1 for yes, and 2 for not set).
iw_canwrite	integer	Default write permission for this owner (0 for no, 1 for yes, and 2 for not set).
iw_candelete	integer	Default delete permission for this owner (0 for no, 1 for yes, and 2 for not set).
iw_canoverwrite	integer	Default overwrite permission for this owner (0 for no, 1 for yes, and 2 for not set).
iw_canvisit	integer	Default visit permission for this owner (0 for no, 1 for yes, and 2 for not set).
iw_canmod	integer	Default directory modify permission for this owner (0 for no, 1 for yes, and 2 for not set).

iwftp_user_hosts

The `iwftp_user_hosts` table defines host names or IP addresses from which a specific user can connect to the server. The fields are listed and described in the following table:

Field	Type	Description
<code>iw_username</code>	<code>varchar(45)</code>	User or group for which this host entry applies. Foreign key for <code>iwftp_owners.iw_ownername</code> .
<code>iw_hostname</code>	<code>varchar(45)</code>	Host name or IP address from which this user can connect to the server.

iwftp_templates

The `iwftp_templates` table defines patterns that can be used as shorthand when configuring users or groups. The fields are listed and described in the following table:

Field	Type	Description
<code>iw_templatenam e</code>	<code>varchar(45)</code>	The identifier for this template.
<code>iw_templatepatt ern</code>	<code>varchar(45)</code>	The substitution string for this template.

Using FTP Permission Tables

The **`iwftp_owners`** table defines an owner and the permissions of the owner regardless of resource. The type of the owner may be { 'U' | 'G' }. The primary key is: ``iw_ownername``, ``iw_ownertype``

The **`iwftp_permissions`** table defines the resource and permissions of an owner or group. The primary key has three fields: {`iw_owner='user or group name'`, `iw_ownertype='{ 'U' | 'G' }`, `iw_resource`}

The **`iwftp_user_hosts`** table defines a resource and a user name, of which the user name may be a group or a specific user. The primary key is: `iw_username`

If you are using only individual user names, then `'iw_groupname'` can be blank, and `"iwftp.iwftp_permissions"` and `"iwftp.iwftp_user_hosts"` should be defined using the same user name(s).

If you are using group names, then define a group name or names before defining users in “iwftp.iwftp_owners”.

When individual users are entered in “iwftp.iwftp_owners”, the owner name is defined in the “iw_ownername” field, and the group name must be entered in the “iw_groupname” field.

There is no explicit ‘groups’ table, so iw_groupname *behaves* as a foreign key. However, there is no explicit definition in SQL like “FOREIGN KEY (iw_groupname) REFERENCES “iwftp_permissions (iw_owner)””. It is a non-explicitly defined (no direct SQL) virtual foreign key relationship.

Example: Single User (No Groups)

iwftp_owners

iw_ownername	iw_ownertype	iw_groupname
'Smith'	'U'	null

iwftp_permissions

iw_ownername	iw_ownertype	iw_resource
'Smith'	'U'	'linux'

iwftp_user_hosts

iw_username	iw_hostname
'Smith'	'111.222.2342'

Example: Using Groups

iwftp_owners

iw_ownername	iw_ownertype	iw_groupname
--------------	--------------	--------------

'DocUser'	'G'	'DocUser'
iw_ownername	iw_ownertype	iw_groupname
'Smith'	'U'	'DocUser'

iwftp_permissions

iw_ownername	iw_ownertype	iw_groupname
'DocUser'	'G'	'DocUser'

iwftp_user_hosts

iw_username	iw_hostname
'DocUser'	'111.222.2342'

iwftp_permissions

iw_ownername	iw_ownertype	iw_resource
'DocUser'	'G'	'iwsys1'

Both:

iw_ownername	iw_ownertype	iw_resource
'DocUser'	'G'	'iwsys1'

Procedure: How to Configure the FTP Server for Anonymous Access

To configure for anonymous access:

1. Set the Allow Anonymous property for the listener to *true*.

- 2. Define a user with the permissions you wish to grant anonymous users.
It is customary to name this user *anonymous*. However, you may use any name you wish.
- 3. Set the anonxx attribute for this user to true, as shown in the following example:

```
<user anonxx="true" home="/common" use="v-r-w-d">anonymous</user>
```

When the anonymous user logs in, he is prompted for a password. Any string containing the @ character is accepted.

User Home Locations

If the path specified in the <home> element is absolute (for example, a physical location on a disk, such as d:/a/b), then the server root parameter on the listener is ignored and the user home is at and below the physical location specified. If the location is not absolute, then the home is below the server root (configured on the listener). In all cases, the user is restricted to the final home location. iWay recommends that only administrators receive physical home locations that permit access to data other than message data.

The SITE Command

SITE is a server-specific command that can be called by clients to execute some site-specific function. The FTPServer supports SITE for the following purposes. The SITE verb itself may or may not be inserted by the specific client in use. The means by which such site-specific commands are requested differs for individual clients. The following table lists available SITE commands.

Command	Description
SITE MIME <i>filename</i>	Returns the MIME-type of the named file.
SITE EXEC [parameters]	The password for the user. Can be a function reference, such as LDAP().
SITE REGISTER [type] <i>name value</i>	Creates or loads the special register. The type must be set to USR, HDR, or DOC.

A client needs specific permission to use the SITE EXEC command. Overall permission is set using the listener configuration, and can be overridden for a specific user through the user for or TPA.

FTP Server Log

Each command executed by the server can be logged. This log takes the following format:

```

2006-01-29T16:20:12Z 1
USER joe : 331 Password required for joe (667c768105d92c18).

2006-01-29T16:20:14Z 1
PASS ---- : 230 User joe logged in.

2006-01-29T16:20:16Z 1
PORT 127,0,0,1,15,10 : 200 PORT command successful.

2006-01-29T16:20:16Z 1
LIST : 226 ASCII transfer complete.

```

Each line in the log represents one command, using the following format:

- ☐ Time
- ☐ Session Handler Number
- ☐ Command
- ☐ Parameter
- ☐ Result (up to 40 characters)
- ☐ Comment (optional)

FTP Commands

The following table lists and describes the available FTP commands.

FTP Command	Description
ABOR	<p>The ABOR command aborts the previous FTP service command and any associated transfer of data.</p> <p>Server replies:</p> <ul style="list-style-type: none"> <input type="checkbox"/> 226 ABOR command successful.

FTP Command	Description
APPE	<p>The APPE command appends data to the end of a file on the remote host. If the file does not already exist, it is created.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 556 Permission denied.❑ 226 Transfer complete.
AUTH	<p>The AUTH command establishes an SSL encrypted session. Only the SSL type is supported.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 431 Service is unavailable.❑ 234 AUTH TLS request accepted.
CDUP	<p>The CDUP command changes the parent directory.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 250 CWD command successful.❑ 550 Invalid Access.
CWD	<p>The CWD command changes the working directory. If the directory name is not specified, the root directory (/) is assumed.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 250 CWD command successful.❑ 550 Invalid Access.
DELE	<p>The DELE command deletes the file specified by the provided path.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 250 Command okay.❑ 550 Not a valid file.❑ 550 User has insufficient rights.

FTP Command	Description
EPRT	<p>The EPRT command allows for the specification of an extended address for the data connection. The extended address must consist of the network protocol, as well as the network and transport addresses. The format of EPRT is:</p> <pre>EPRT <net-prt> <net-addr> <tcp-port> </pre> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 200 EPRT command successful.
EPSV	<p>The EPSV command requests that a server listen on a data port and wait for a connection.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 229 Entering passive mode (<message>).
FEAT	<p>The FEAT command displays the feature list.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 211 iWay extended features.
HELP	<p>The HELP command displays the HELP information.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 214 HELP information.
LIST	<p>The LIST command causes a list to be sent from the server. If the path name specifies a directory or other group of files, the server should transfer a list of files in the specified directory. If the path name specifies a file, then the server should send current information on the file. A null argument implies that the user is currently working in that directory or it is a default directory.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 226 Transfer complete.

FTP Command	Description
MDTM	<p>The MDTM command returns the date and time of when a file was modified.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 553 File not found.❑ 213 <timestamp>.
MFMT	<p>The MFMT command resets the remote files timestamp.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 213 ModifyTime=<timestamp>;/filename.
MKD	<p>The MKD command causes the directory specified in the path name to be created as a directory (if the path name is absolute) or as a subdirectory of the current working directory (if the path name is relative).</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 257 Directory Created.❑ 550 Already exists.❑ 550 No permission.
NLST	<p>The NLST command causes a directory listing to be sent from the server to the user site. The path name should specify a directory or other system-specific file group descriptor. A null argument implies the current directory. The server will return a stream of file names and no other information.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 550 Connection refused: connect.❑ 226 Closing data connection.

FTP Command	Description
NOOP	<p>The NOOP command means there is no operation.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 200 NOOP command successful.
PASS	<p>The PASS command is a Telnet string argument field specifying the user password. This command must be immediately preceded by the USER command.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 530 Login incorrect. ❑ 230 User <username> logged in.
PASV	<p>The PASV command requests the server to listen on a data port (which is not its default data port) and to wait for a connection rather than initiate one upon receipt of a transfer command. The response to this command includes the host and port address this server is listening on.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 227 Entering passive mode.
PBSZ	<p>The PBSZ command represents the protection buffer size.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 431 Service unavailable. ❑ 200 Command ok.

FTP Command	Description
PORT	<p>The PORT command is HOST-PORT specification argument for the data port to be used in data connection. There are defaults for both the user and server data ports, and under normal circumstances this command and its reply are not needed. If this command is used, the argument is the concatenation of a 32-bit internet host address and a 16-bit TCP port address. This address information is broken into 8-bit fields and the value of each field is transmitted as a decimal number (in character string representation). The fields are separated by commas. An example of a port command is:</p> <pre>PORT h1,h2,h3,h4,p1,p2</pre> <p>Server replies:</p> <ul style="list-style-type: none"><input type="checkbox"/> 200 PORT command successful.<input type="checkbox"/> 500 command not understood.
PROT	<p>The PROT command returns a data channel protection level. The supported level values are C and P. P protects the connection, while C clears the connection.</p> <p>Server replies:</p> <ul style="list-style-type: none"><input type="checkbox"/> 200 Command PROT ok.
PWD	<p>The PWD command displays the name of the current working directory.</p> <p>Server replies:</p> <ul style="list-style-type: none"><input type="checkbox"/> 257 "<current directory>."
QUIT	<p>The QUIT command closes the connection.</p> <p>Server replies:</p> <ul style="list-style-type: none"><input type="checkbox"/> 221 Goodbye.

FTP Command	Description
REST	<p>The RESET command is an argument field that represents a server marker at which a file transfer is to be restarted. This command does not cause the file transfer but rather skips over the file to the specified data checkpoint. This command must be immediately followed by the appropriate FTP service command, which resumes the file transfer.</p> <p>Server replies:</p> <ul style="list-style-type: none"> <input type="checkbox"/> 350 Restarting at <position>. Send STOR or RETR to initiate transfer. <input type="checkbox"/> 501 not a number.
RETR	<p>The RETR command causes the server to transfer a copy of the file, specified in the path name, to the server at the other end of the data connection. The status and contents of the file at the server site are unaffected.</p> <p>Server replies:</p> <ul style="list-style-type: none"> <input type="checkbox"/> 550 I/O Error: Connection refused: connect. <input type="checkbox"/> 226 Transfer complete.
RMD	<p>The RMD command causes the directory specified in the path name to be removed as a directory (if the path name is absolute) or as a subdirectory of the current working directory (if the path name is relative).</p> <p>Server replies:</p> <ul style="list-style-type: none"> <input type="checkbox"/> 550 No permission. <input type="checkbox"/> 550 Not a valid directory. <input type="checkbox"/> 250 Directory removed.

FTP Command	Description
RNFR	<p>The RNFR command specifies the old path name of the file which is to be renamed. This command must be immediately followed by a RNT0 command specifying the new file path name.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 350 File exists, ready for destination name.
RNT0	<p>The RNT0 command specifies the new path name of the designated file immediately preceding the RNFR command. Together the two commands cause a file to be renamed.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 250 RNT0 command successful.❑ 550 <filename>: cannot rename.
SITE	<p>The SITE command handles server-specific commands. For more information, see The SITE Command on page 146.</p>
SIZE	<p>The SIZE command returns the size of the file in bytes.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 553 <filename>: cannot read.❑ 213 <size>
STOR	<p>The STOR command causes the server to accept the data transferred through the data connection and to store the data as a file at the server site. If the file specified in the path name exists at the server site, then its contents are replaced by the data being transferred. A new file is created at the server site if the file specified in the path name does not already exist.</p> <p>Server replies:</p> <ul style="list-style-type: none">❑ 553 <filename>: cannot write.❑ 226 Transfer complete.

FTP Command	Description
STOU	<p>The STOU command behaves like the STOR command except that the resultant file is created in the current directory under a name unique to that directory.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 553 <filename>: cannot write. ❑ 226 Transfer complete.
SYST	<p>The SYST command is used to find out the type of operating system on the server.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 215 UNIX Type: Server Version <version>.
TYPE	<p>The TYPE command specifies the representation type. The allowed types are A and I.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 200 Type set to <value>.
USER	<p>The USER command is a Telnet string argument field identifying the user. The user identification is required by the server to access the file system.</p> <p>Server replies:</p> <ul style="list-style-type: none"> ❑ 230 Already logged-in. ❑ 530 Invalid user name.

Security Considerations

The FTP Server listener supports secure transfer using the FTP/S protocol as defined in RFC 2228 and RFC 4217. It supports the FEAT request from standard FTP clients and accepts requests to initiate operations on secure channels. Security control is done by the client. Configuration can be set to reject clients that do not request security operations. Secure control channels and data transfer channels are treated separately. For example, if the listener configuration does not require that secure data transfer be done, then the listener will accept client requests to transfer files securely if appropriate keystores are defined, but will accept transfers on non-secure channels. If the security configuration is not provided, the FEAT response to the client will not indicate support for TLS channels.

The server uses the keystore of the system, which is defined in the system properties. If a keystore is not defined, the server rejects the attempt of the client to establish a secure channel.

Minimum levels of security are configured for the listener, preventing socket negotiation below the specified levels.

The server implements data transfer port theft protection requiring that the data transfer socket be connected to the same IP address as the control socket.

Access permissions for directories can be configured on a per-user basis in the security file. Additionally, access from specific hosts can be specified on a per-user basis in the security file. This restricts the user to known client equipment. The client can be specified as a host name or as an IP address. The test is not case-sensitive. For more information, see [The Security File](#) on page 133.

Using the Trading Partner Management Facility

The Trading Partner Management facility can hold the identity of a user authorized to submit or receive messages through the FTPD listener. This user is identified in the Protocol/FTPD section of the Partner Editor. For more information, see *iWay Trading Partner Manager User's Guide*.

The following table lists and describes the fields you can specify in the agreement.

Field	Description
User Name	<p>This is the login name of the user associated with this partner. If a simple name is used, this must be unique across the system, as the username will be used to locate the appropriate trading partner. If the user logs in using a user ID of <partner>:user, then uniqueness is not required as the partner is identified by the value preceding the colon.</p> <p>The partner identifier is formed by TPN(SREG('ftpd-user')).</p>
Password	The password for the user. This can be a function reference, such as LDAP().
Mailbox	The mailbox assigned to this partner. This is a directory of the configured server root assigned to the listener. This is the users home location.
Require Secure Authentication	This partner requires that FTP control to and from this mailbox be done under secure authorization control as described in RFC 2228.
Require Secure Transfer	This partner requires that FTP transfers to/from this mailbox be done under secure channel conditions as described in RFC 2228.

The agreements are referenced at user login. Therefore, they can be changed while the system is operational. Note that when security requirements for a partner differ from that of the listener itself, the higher security takes precedence. Therefore, although a listener may not require that all transfers take place in a secure manner, an agreement with a specific partner may be required.

Configuring a FTP SREG Service

The FTP SREG service sets Special Registers (SREGs) on the FTP Server. This service is used only with the iWay FTP Server extension and FTP Server channel. The identified special registers are sent to the server using a SITE command. This facility is used when programming process flows for the Managed File Transfer capability of iWay Service Manager.

Note: This service has no meaning with non-iWay FTP servers and should not be used with non-iWay FTP servers.

To configure a FTP SREG service:

1. Perform the steps as described in [Configuring Services](#) on page 181.

- 2. Ensure that you select *FTP SREG Agent {com.ibi.agents.XDFTPSREGAgent}* as the service type you are configuring.

For a complete description of the configuration parameters that are available for the FTP SREG service, see [FTP SREG Service Parameters](#) on page 158.

For a complete description of the edges that are returned by the FTP SREG service, see [FTP SREG Service Edges](#) on page 161.

Reference: FTP SREG Service Parameters

The following table lists and describes parameters for the FTP SREG service.

Parameter	Description
Host Parameters	
Host Name	In this field, enter the DNS name (or IP address) of the FTP server that you wish to connect to. Use the host port if the standard port is not 21.
Remote Port	This is the port to connect to on the FTP site. Leave it blank for default port 21.
User Name	The name used as the valid user ID on the FTP server.
Password	The valid password for the FTP server.
Account Name	The valid account for the FTP server.
Use Passive Command	If set to <i>true</i> , the service uses a PASV command. Otherwise, it uses the PORT command.
Timeout	The timeout interval for socket in seconds.
Retry Interval	Retry interval in seconds (allows xxhxxmxxs format). You can omit or use 0 for no retry.
Connection Retry	This shows the number of attempted failed connections to the FTP server.
SREGs	

Parameter	Description
Type of variable	<p>This parameter shows the type of variable to be used (headers appear in emitted documents as header values).</p> <ul style="list-style-type: none"> <input type="checkbox"/> user. Temporary variable definition for general use. <input type="checkbox"/> doc. Temporary variable definition associated with a document. <input type="checkbox"/> hdr. Temporary variable to be included in message headers.
SSL Parameters	
Use SSL	If set to <i>true</i> , the connection is secured using Secure Sockets Layer (SSL).
Security Protocol	<p>This shows the type of security protocol to be used. The following list describes the options of the security protocol.</p> <ul style="list-style-type: none"> <input type="checkbox"/> SSL. This protocol supports some versions of SSL, and may also support other versions. <input type="checkbox"/> SSLv2. This protocol supports SSL version 2 or higher. <input type="checkbox"/> SSLv3. This protocol supports SSL version 3, and may support other versions. <input type="checkbox"/> TLS. This protocol supports some versions of TLS, and may also support other versions. <input type="checkbox"/> TLSv1. This protocol supports TLS version 1, and may support other versions. <p>This field is not needed if Keystore is a SSL Provider.</p>
Secure Data Connection	This is used to enable a secure data connection, for example, transfer data securely. It is used in conjunction with Secure Control Connection.
Use 128-bit Encryption	This parameter enforces the use of 128-bit encryption.

Parameter	Description
SSL Security	<p>This parameter describes the FTP Server connection type. Select one of the following options:</p> <ul style="list-style-type: none"><input type="checkbox"/> unknown. This setting defaults to Explicit Security then fails over to Implicit Security.<input type="checkbox"/> explicit. In order to establish the SSL link, explicit security requires that the FTP client issue a specific command to the FTP server after establishing a connection. The default FTP server port is used.<input type="checkbox"/> implicit. Implicit security automatically begins with an SSL connection as soon as the FTP client connects to an FTP server. In implicit security, the FTP server defines a specific port for the client (typically 990) to be used for secure connections.
Keystore File or Keystore Security Provider	<p>In this field, you can:</p> <ul style="list-style-type: none"><input type="checkbox"/> Enter the full path to the Keystore file, which provides certificate material to be used for SSL connection.<input type="checkbox"/> Name the Keystore Security Provider.<input type="checkbox"/> Use the configured default Keystore Security Provider by leaving it blank.
Keystore Password	<p>This field is used to enter the password to access Keystore file. This is not required if Keystore File or Keystore Security Provider is the name of a Keystore Security Provider.</p>
Keystore Type	<p>This field shows the type of the Keystore. It is not needed if Keystore File or Keystore Security Provider is the name of a Keystore Security Provider.</p>

Reference: FTP SREG Service Edges

The following table lists and describes the edges that are returned by the FTP SREG service.

Edge	Description
success	Operation completed successfully.
fail_parse	Failed to properly parse the input parameters of the service.
fail_connect	Failed to connect to FTP host for any one of the following reasons: <ul style="list-style-type: none"><input type="checkbox"/> The host name (IP) is invalid.<input type="checkbox"/> The User ID is invalid.<input type="checkbox"/> The password of the user is invalid.<input type="checkbox"/> The connection failed.
fail_operation	Invalid parameters or other error.

This section describes how to configure the SSH version of the FTP Server components using iWay Service Manager.

In this chapter:

- ❑ [SFTP Server Component Configuration Overview](#)
 - ❑ [Configuring a SFTP Listener](#)
 - ❑ [Configuring the SSH Server Security Provider](#)
-

SFTP Server Component Configuration Overview

iWay offers an SFTP server that is designed to extend the processing capabilities of iWay Service Manager (ISM). Although this service can be configured to operate as a standard SFTP server, the purpose of this service is transaction receipt and mailboxing. The SFTP Server listener listens on the port that is specified in the SFTP server settings. The listener begins processing messages when a file is sent through SFTP to iWay Service Manager. Therefore, it differs from general SFTP servers in several ways:

- ❑ User login security is managed through the iSM security facilities. Full function evaluation of all parameters enables the storage of attributes, such as passwords, in a security directory, such as LDAP.
- ❑ User attributes can be stored in the Partner Management system of iWay Trading Manager, if installed.
- ❑ File actions for Get and Put can be treated as messages. In this case, they are immediately processed through the appropriate configuration.
- ❑ Messages received for execution can optionally be safe-stored with their context to prevent loss in the event of failure. Safe-stored messages are reloaded when the listener is started, and executed at that time. Only when all safe-stored messages have been processed does the listener open to receive new messages.
- ❑ Input-streaming is supported when the input is to be treated as a message. In input-stream mode, a large document can be broken into parts as it is received. Each part is extracted, based upon the message type configured using a standard streaming preparer, and processed as a separate message.

- ❑ Tracing and diagnostics are available using the server tracing system.
- ❑ Messages can be stored in the iWay Audit Manager logs for later analysis.

Configuring a SFTP Listener

To configure a SFTP listener:

1. Perform the steps as described in [Configuring Listeners](#) on page 177.
2. Ensure that you select *SFTP Server* as the listener type you are configuring.

For a complete description of the configuration parameters that are available for the SFTP listener, see [SFTP Listener Configuration Parameters](#) on page 70.

For a complete description of the SFTP listener Special Registers (SREGs), see [SFTP Listener Special Registers](#) on page 74.

Reference: SFTP Server Listener Configuration Parameters

The following table lists and describes parameters for the SFTP Server listener.

Note: Parameters that are common to SFTP listeners are described in [Listener Configuration Parameters](#) on page 173.

Parameter	Definition
Port	TCP port for receipt of SFTP requests. SFTP standard is port 22.
Local Bind Address	Local bind address for multi-homed hosts: usually leave empty.
SFTP Server Log	If entered, full path to SFTP Server log file. Name can be an iWay unique file pattern such as log####.txt.
Server Root	Base directory for this SFTP Server– when user mailbox paths are relative, they are below this directory.
Use Safestore	Safestore preserves store requests while the incoming document is passing through execution. It is not meaningful for direct writes of messages to the file system. Using safestore can reduce system performance.

Parameter	Definition
Default Permissions	
Default Can READ	If true, users without specific security can read
Default Can WRITE	If true, users without specific security can write
Action on GET	How should the server treat file retrieve type requests from the client
Action on PUT	How should the server treat file store type requests from the client
SITE EXEC	If true, clients can execute processes via the SITE EXEC command
Security	
Session Timeout	If > 0, maximum seconds between commands before automatic session timeout
Allowable Access	Attempts Number of access attempts that will be allowed before invoking the Access Denied Flow.
Access Denied Flow	Optional iSM process flow to call when user fails to login within the Allowable Access Attempts.
Secure Shell Provider	Name of the Secure Shell provider. If missing the default secure shell provider will be used

Note: The SFTP listener supports streaming. Streaming is used for large documents or documents for which the application needs to split the input into sections under the same transaction. For more information on streaming and configuring streaming preparers, see the *iWay Service Manager Component and Functional Language Reference Guide*.

Reference: SFTP Server Listener Special Registers

The following table lists and describes the Special Registers (SREGs) available on the SFTP Server listener.

Name	Level	Type	Description
sftpd.file	System	String	The current active configuration name.
sftpd.user	Document	Integer	The physical length of the message payload.
sftpd.frompart y	System	String	The assigned name of the master (listener).
sftpd.comman d	System	String	The protocol on which the message was received.
sftpd.mode	Document	String	The full name of the input file.
iwayconfig	System	String	The current active configuration name.
msgsize	Document	Integer	The physical length of the message payload.
name	System	String	The assigned name of the master (listener).
protocol	System	String	The protocol on which the message was received.
source	Document	String	The full name of the input file.
tid	Document	String	Unique transaction ID.

Action on GET

Client GET (and MGET) requests the return of information to the client by the server. A GET client for a file can be handled differently, depending on the current settings for the action. The default setting is configured for the listener but it can be overridden for any specific user in the security file.

- ☐ **Deny Access.** User cannot read files from the server
- ☐ **Execute as a Message.** A standardized XML sftpd request document is constructed containing the request. This document is passed to the system for execution. The result of the execution is returned to the client.

- ❑ **Use file system.** This is a standard FTP retrieval. Data is returned from the named file in the file system.
- ❑ **Use file system then execute message.** The retrieval operates as a standard SFTP retrieval. Data is returned from the named file in the file system. Then, a standardized XML `sftpd` request document is constructed containing the request. This document is passed to the system for execution. The result of the execution is sent to the configured emitter.

Action on PUT

Client PUT (including append) cause the transfer of information from the client to the server. A client PUT request can be handled differently, depending upon the current setting for this action. The default setting is configured for the listener but it can be overridden for any specific user.

1. **Deny Access.** User cannot write files to the server
2. **Execute as a Message.** The data is passed as a document into the system for execution. A standardized XML document is constructed containing the data. This document is passed to the system for execution.
3. **Use file system.** The request operates as a standard SFTP store command. Data is stored from the named file in the file system.
4. **Use file system then execute message.** The retrieval operates as a standard *sftpd* store command. Data is stored from the named file in the file system. Then, a standardized XML SFTP request document is constructed containing the request. This document is passed to the system for execution. The result of the execution is sent to the configured emitter.

Process Flow Fails in SFTP Server

Applications should, of course, always be designed to handle errors and manage the return from the listener to the client. Should a process flow report a failure, however, the SFTP Server listener presents the error to the SFTP client following standards of SFTP. Given an SFTP Server listener configured to EXEC the message to a process flow which fails of the GET EXEC, the transient file on the server is not deleted.

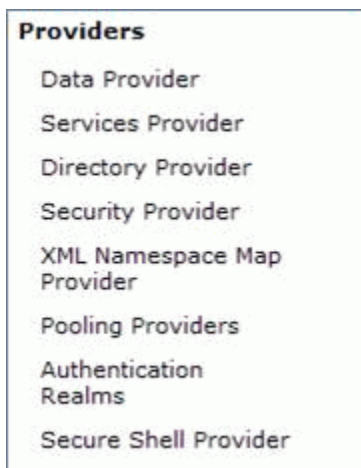
The file is empty since the error is reported during the open and no data transfer took place.

The response seen by a line client (in this case Putty SFTP client) is as follows:

```
sftp> get ftpserver036_trans_fail.txt
Fetching /ftpserver036_trans_fail.txt to ftpserver036_trans_fail.txt
Unexpected reply 21
/qa/iwayqa> debug1: client_input_channel_req: channel 0 rtype exit-status
reply 0
debug1: channel 0: free: client-session, nchannels 1
debug1: fd 0 clearing O_NONBLOCK
Transferred: sent 3432, received 9568 bytes, in 38.7 seconds
Bytes per second: sent 88.8, received 247.4
debug1: Exit status 0
```

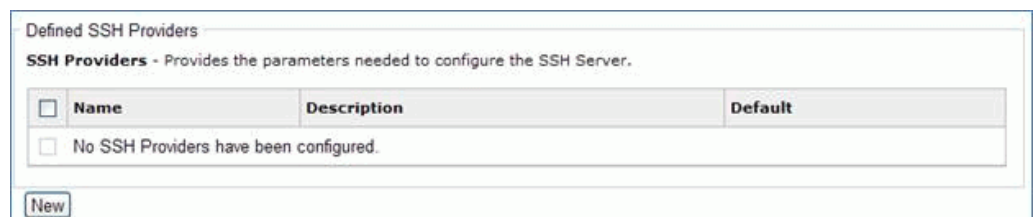
Configuring the SSH Server Security Provider

Installation of the SFTPserver installs a new Secure Shell Provider within the iSM listing of providers as shown in the example below.



1. Click *Secure Shell Provider*.

The Defined SSH Providers dialog box is displayed.



2. Click *New* to add a new provider.

Reference: SFTP SSH Shell Provider

The following table lists and describes parameters for the SFTP SSH Shell Provider.

Parameter	Description
SSH Providers	
Key Exchange Factories*	<p>Select classes used to exchange keys between the SSH client and the this SSH server.</p> <p>Currently the SSH Server listener supports the Diffie-Hellman (DH) key exchange. DH is commonly used when you encrypt data on the Web using either SSL or TLS (Secure Socket Layer and Transport Layer Security Respectively).</p> <p>diffie-hellman-group1-sha. This group provides basic security (768-bit key) and good performance.</p> <p>diffie-hellman-group14-sha1. This group provides a stronger security (2048 bits key vs. 768 bits) than the diffie-hellman-group1-sha1. Please note that this group is not supported by the default JCE provider.</p> <p>You can choose to support one or the other or both.</p>
Random Factory	<p>Pseudo random number generator.</p> <p>Only Bouncy Castle version of the Random Number generator is supported by the SSH Server listener.</p>
Cipher Factories*	Classes for a cryptographic cipher, used either for encryption or decryption.
Compression Factories*	Classes used to compress the stream of data between the server and SSH clients.
MAC Factories*	Classes used for Message Authentication Code for use in SSH.
Signature Factory*	Classes used by the server to sign and verify packets sent between the server and client.
Key Pair File	

Parameter	Description
Provider*	Provider for key pairs. The provider is used to create the SSH Key Pair repository when it doesn't exist. When the repository exists the Provider returns the Key Pair generated by the Signature Factory that was used to create the repository.
Key Pair File Signature*	Used by the SFTPServer to sign the Key Pair File that is generated if the Key Pair File does not exist.
Key Pair File Path*	<p>Fully qualified path to the Key Pair File. If the path points to a file that does not exist the Provider will create a Key Pair file at this location using the Key Pair File Signature that was selected.</p> <p>Enter any directory accessible to iSM and use the name <code>key.ser</code>. The name doesn't matter at this point the file will be generated (or regenerated) if it doesn't exist.</p>
Key Pair File Password	Password for the SSH Key Pair file
Authentication	
Password Authenticator*	<p>The class used by the server to authenticate the SSH client's password.</p> <p>Select "File Based Authenticator" unless you have created a JDBC based authentication for the iSM SFTPServer. Both the SFTPServer and FTPServer share the same authentication algorithms.</p> <p>Note: Both the SFTPServer and FTPServer share the same authentication algorithms; and can share the same files/RDBMS tables.</p>
Public Key Authenticator*	The class to authenticate the SSH client's public keys.
User Repository	

Parameter	Description
Repository Type	How the user repository is stored. The repository can be stored either as an XML file or as a JDBC database. This repository defines the users permitted to exchange messages with this server along with their mailbox and security characteristics.
Security File	Security file location. This field is required either when the Repository Type is set to XML.
JDBC Provider Name	Name of the JDBC Data Provider. If repository is set to JDBC.
Basic	
Reuse Address.	If true, when the connection is closed, immediately make the address available, bypassing TCP's defaults.
Connections Backlog*	Number of connections allowed to queue up before a failure

Common Configuration Parameters

This section provides a reference for common configuration parameters used by iWay Service Manager (ISM) components.

In this appendix:

❏ [Listener Configuration Parameters](#)

Listener Configuration Parameters

The following table lists and describes common parameters used by the FTP, Secure FTP (SFTP), and FTP Server listeners.

Parameter	Description
Whitespace Normalization	Specifies how the parser treats whitespace in Element content. Select <i>preserve</i> to turn off all normalization as prescribed by the XML Specification. Select <i>condense</i> to remove extra whitespace in pretty printed documents and for compatibility with earlier versions.
Accepts non-XML (flat) only	If set to <i>true</i> , the input data is sent directly to the business logic step. The data is not preparsed, parsed, or validated. This flag is used primarily to send non-XML to the business logic or replyTo without processing it.
Optimize Favoring	Use this option to customize how the listener performs. For smaller transactions, select <i>performance</i> . For large input documents that could monopolize the amount of memory used by iWay Service Manager, select <i>memory</i> .

Parameter	Description
Multithreading	<p>Indicates the number of worker threads (documents or requests) that iWay Service Manager can handle in parallel. Setting this to a value of greater than 1 enables the listener to handle a second request while an earlier request is still being processed. The total throughput of a system can be affected by the number of threads operating. Increasing the number of parallel operations may not necessarily improve throughput.</p> <p>The default is 1.</p> <p>The max value is 99.</p>
Maximum Threads	<p>The parallel threads can grow to this count automatically on demand. Over time, the worker count will decrease back to the multithreading level. Use this parameter to respond to bursts of activity.</p>
Execution Time Limit	<p>The maximum time that a request may take to complete. Used to prevent runaway requests. Any request that takes longer to complete than this value will be attempted to be terminated.</p>
Polling Interval	<p>The maximum wait interval (in seconds) between checks for new requests or commands. The higher this value, the longer the interval, and the fewer system resources that are used. The side effect of a high value is that the worker thread will not be able to respond to a stop command.</p> <p>The default is 2.0 seconds.</p>
Default Java File Encoding	<p>The default encoding if the incoming message is not self-declaring (that is, XML).</p>

Parameter	Description
Agent Precedence	<p>Sets the order by which iWay Service Manager selects agents. iWay Service Manager selects the agent or agents to process the document by searching through the configuration dictionary. Usually, it looks for a document entry in the configuration and when a match is found, the agent specified in that document entry is selected. If a matching document entry is not found, or no agent is specified, the engine looks in the input protocol configuration (listener). To have the processing agent taken directly from the listener (thus ignoring the document entry), use <listener> overrides <document>.</p> <p>Possible values are <document> overrides <listener> and <listener> overrides <document>.</p> <p>The default value is <document> overrides <listener>.</p>
Always reply to listener default	If set to <i>true</i> , the default reply definition is used in addition to defined reply-to and error-to destinations.
Error Documents treated normally	If set to <i>true</i> , error documents are processed by any configured preemitters.
Listener is Transaction Manager	If set to <i>true</i> , agents run within a local transaction.
Record in Activity Log(s)	If set to <i>true</i> , activity on this channel will be recorded in the activity logs, otherwise the activity will not be recorded.
AES Key	If the channel will receive encrypted AFTI messages, set the AES key (maximum 16 characters) to be used for decrypting.
Failed ReplyTo Flow	Name of a published process flow to run if a message cannot be emitted on an address in its reply address list.
Dead Letter Flow	Name of a published process flow to run if an error cannot be emitted on an address in its error address list.

Parameter	Description
Channel Failure Flow	Name of a published process flow to run if this channel cannot start or fails during message handling. iWay Service Manager will attempt to call this process flow during channel shut down due to the error.
Startup Dependencies	A comma-separated list of channel names that must be started before this one is called.

Configuring iWay Service Manager Components

During the FTP solutions development process, you are required to configure listeners and services using iWay Service Manager (iSM). This section provides the steps that are needed to access and configure these iSM components. Descriptions of the parameters for each component are provided within the corresponding sections.

In this appendix:

- ❑ [Configuring Listeners](#)
 - ❑ [Configuring Services](#)
-

Configuring Listeners

This section describes how to configure listeners using iSM.

Procedure: How to Configure a Listener

1. Ensure that iSM is running.

On Windows, you can start iSM by clicking *Start*, selecting *Programs*, *iWay 7.0 Service Manager*, and then *Start Service Manager* for the configuration you are currently using.

For more information on starting and stopping iSM, see the *iWay Service Manager User Guide*.

2. Open a browser window and point to the following URL:

<http://host:port/ism>

where:

[host](#)

Is the host machine on which iSM is installed.

[port](#)

Is the port on which iSM is listening. The default port is 9999.

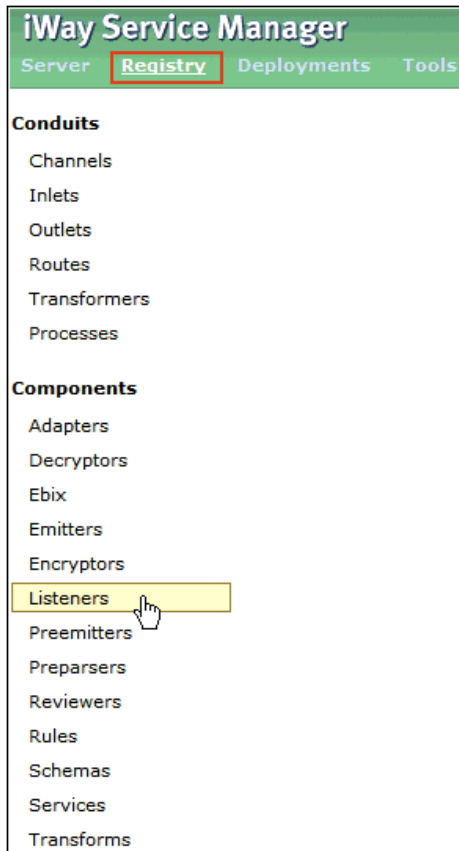
On Windows, alternatively, you can click *Start*, select *Programs*, *iWay 7.0 Service Manager*, and then click *Console*.

A logon dialog box opens.

3. Type a user name and password for the configuration you are using, and click *OK*.

The iWay Service Manager Administration Console opens.

4. Click *Registry* in the top pane, and then click *Listeners* in the left pane, as shown in the following image.









The Listeners pane opens, as shown in the following image.

Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Listeners

☐ Filter By Name Where Name Equals

<input type="checkbox"/>	Name	Type	References	Description
<input type="checkbox"/>	file1	File		A default/sample file listener.
<input type="checkbox"/>	javadoc	HTTP 1.0 [deprecated]		The javadoc listener is used to make the iWay Service Manager API available to a remote browser.
<input type="checkbox"/>	pictures_loader	File		The pictures listener locates files with a variety of common image file extensions (img, gif, jpg, ...).
<input type="checkbox"/>	pictures_viewer	HTTP 1.0 [deprecated]		The pictures.viewer is used to kickoff the image retrieval process as defined by the pictures sample.
<input type="checkbox"/>	scifibooks	Schedule Recurring Execution		This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.
<input type="checkbox"/>	SOAP2	SOAP		This listener is used by the stock SOAP channel.

The table provided lists all the previously configured listeners and a brief description for each.

- Click *Add*.

The Select listener type pane opens, as shown in the following image.

Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Select listener type

Type * Type of the new listener

Select a type 

- Select the type of listener that you want to configure (for example, *FTP[S] Client*, *FTP[S] Server*, or *SFTP*) from the Type drop-down list and click *Next*.

A configuration parameters pane for the selected listener opens. For example, the following image shows a portion of the Secure FTP (SFTP) listener configuration pane.

Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Configuration parameters for new listener of type SFTP	
Host Name *	DNS name (or IP address) of the SFTP server that you want to connect to <input type="text"/>
Remote Port *	Port to connect to on the SFTP site, blank for default port 22 <input type="text" value="22"/>
Input Path	Directory with optional pattern on SFTP host from which to retrieve files. A specific file name or DOS-style pattern) can be used. Do not use <i>suffix in</i> . <input type="text"/>
Include Symbolic Links *	Set to true if you want the sftp listener to process the symbolic links <input type="text" value="false"/> Pick one <input type="button" value="v"/>
Include Hidden Files *	Set to true if you want the sftp listener to process the hidden files <input type="text" value="false"/> Pick one <input type="button" value="v"/>
Destination Directory	Directory on SFTP host to return responses to <input type="text"/>
Data, Signal or Streaming *	How the input will be processed by the SFTP listener Data - Data file will be retrieved from the SSH server and maintained in memory while processed by the listener. Signal - Data file will be retrieved from the SSH server and stored locally (requires that the <i>Local Store Directory</i> be filled in); a signal document will be generated by the listener. Stream - A connection will be opened with the SSH server and data from the file will be retrieved and processed by the listener as needed. <input type="text" value="Data"/>

Note: The parameters prefixed with an asterisk (*) in the listener configuration pane are required.

7. Provide the appropriate values for the listener parameters. In most cases, you will need to scroll down the page to view all of the available parameters for the selected listener.
8. Click *Next* at the bottom of the page to continue.

A listener name and description pane opens, as shown in the following image.

Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Select listener type	
Name *	Name of the new listener <input type="text"/>
Description	Description for the new listener <input type="text"/>
<input type="button" value=" << Back"/> <input type="button" value=" Finish"/>	

9. Enter a name for the selected listener and a brief description (optional).
10. Click *Finish*.

You return to the Listeners pane, where the new listener that has been configured is added to the table of available listeners.

You can use this listener as part of your channel configuration where the business logic will be applied to the received messages.

Configuring Services

This section describes how to configure services using iSM.

Procedure: How to Configure a Service

1. Ensure that iSM is running.

On Windows, you can start iSM by clicking *Start*, selecting *Programs*, *iWay 7.0 Service Manager*, and then *Start Service Manager* for the configuration you are currently using.

For more information on starting and stopping iSM, see the *iWay Service Manager User Guide*.

2. Open a browser window and point to the following URL:

<http://host:port/ism>

where:

[host](#)

Is the host machine on which iSM is installed.

[port](#)

Is the port on which iSM is listening. The default port is 9999.

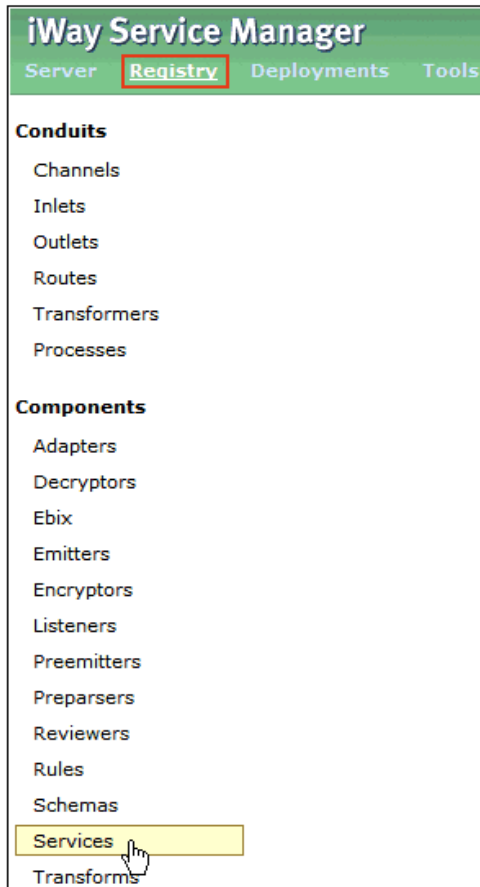
On Windows, alternatively, you can click *Start*, select *Programs*, *iWay 7.0 Service Manager*, and then click *Console*.

A logon dialog box opens.

3. Type a user name and password for the configuration you are using, and click *OK*.

The iWay Service Manager Administration Console opens.

4. Click *Registry* in the top pane, and then click *Services* in the left pane, as shown in the following image.



The Services pane opens, as shown in the following image.

Services

Services are executed java procedures that handle the business logic of a message.

Services

☐ Filter

<input type="checkbox"/>	Name	Type	References	Parms	Description
<input type="checkbox"/>	DeleteAllSciFiBooks1	Constant Agent			Sets a call to the RDBMS Adapter to delete all records from the SciFiBooks Database.
<input type="checkbox"/>	move1	Move Agent transfers input to output			The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
<input type="checkbox"/>	pictures_img2xml	Entag Agent			converts the image to base64 and wraps it in a <picture> tag
<input type="checkbox"/>	pictures_iterator	XML Iterator			Iterate a loop for each portion of an XML document
<input type="checkbox"/>	RSSRead1	HTTP Read Agent		0	Reads an RSS Document from url that is specified in the original incoming document.
<input type="checkbox"/>	Snip1	Snip Agent			Copies a subtree of the input document as defined by the PFIVP schema to the root of the output document as defined by PFIVPResponse schema.

The table provided lists all the previously configured services and a brief description for each.

- Click **Add**.

A select service type pane opens, as shown in the following image.

Services

Services are executed java procedures that handle the business logic of a message.

Select the type for the new Service object definition

Type * Available Service types

- Select the type of service that you want to configure (for example, *SFTP Emit Agent* {*com.ibi.agents.XDSFTPEmitAgent*} from the Type drop-down list and click **Next**.

A configuration parameters pane for the selected service opens. For example, the following image shows a portion of the SFTP Emit service configuration pane.

Services

Services are executed java procedures that handle the business logic of a message.

Configuration parameters for SFTP Emit Agent service	
Host Name *	DNS name (or IP address) of the SFTP server that you want to connect to <input type="text"/>
Remote Port	Port to connect to on the SFTP site, blank for default port 22 <input type="text"/>
User Name *	User ID on the SFTP server <input type="text"/>
Password	User's password on the SFTP server <input type="password"/>
Private Key	Path to the private key file for public-key authentication. <input type="text"/> <input type="button" value="Browse"/>
Passphrase	Passphrase used to protect the Private Key <input type="password"/>
Remote Site Folder	Folder or directory on the SFTP site that you want to use as a starting location when you connect, blank means login directory <input type="text"/>
File Protect	Emit temporary name and then rename to the desired name <input type="text" value="false"/> <input type="button" value="Pick one"/> <input type="button" value="v"/>

Note: The parameters prefixed with an asterisk (*) in the service configuration pane are required.

- Provide the appropriate values for the service parameters. In most cases, you will need to scroll down the page to view all of the available parameters for the selected service.
- Click *Next* at the bottom of the page to continue.

A service name and description pane opens, as shown in the following image.

Services

Services are executed java procedures that handle the business logic of a message.

Provide a name and description for the new Service object definition	
Name *	Name of the new Service object definition <input type="text"/>
Description	Description for the new Service object definition <input type="text"/>

9. Enter a name for the selected service and a brief description (optional).
10. Click *Finish*.

You return to the Services pane, where the new service that has been configured is added to the table of available services.

Legal and Third-Party Notices

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, FOCUS, iWay, Omni-Gen, Omni-HealthData, and WebFOCUS are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2021. TIBCO Software Inc. All Rights Reserved.