[](https://github.com/wso2/docs-ei/edit/7.1.0/en/micro-integrator/docs/references/connectors/file-connector/file-connector-config.md)

File Connector Reference[¶](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/file-connector-config/#file-connector-reference)

The following configurations allow you to work with the File Connector version 4.

Connection configurations[¶](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/file-connector-config/#connection-configurations)

The File connector can be used to deal with two types of file systems:

**Local File System**: A file system of the server where WSO2 EI is deployed.

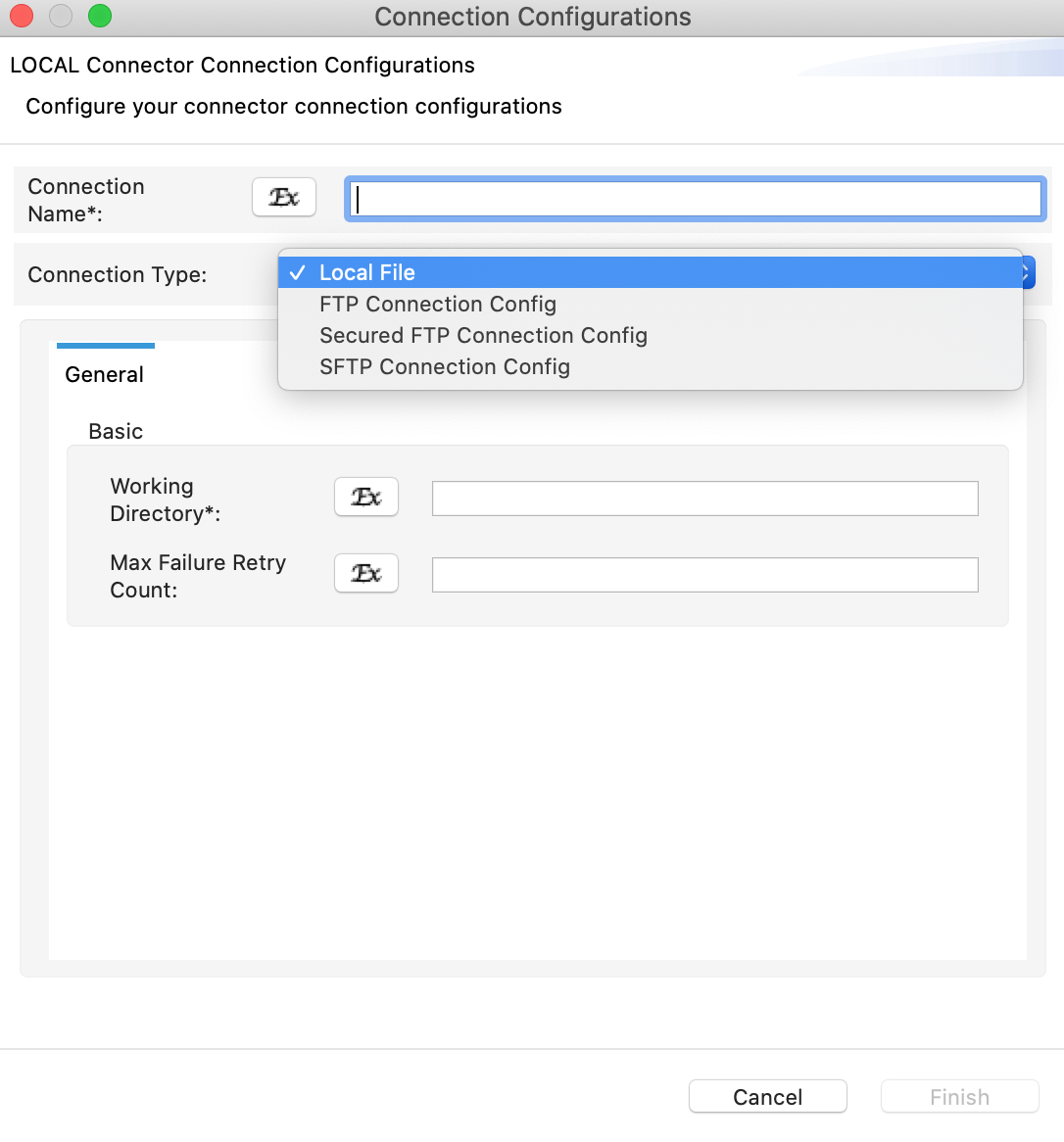
**Remote File System**: A file system outside the server where WSO2 EI is deployed. There are few industry standard protocols established to expose a file system over TCP. Following protocols are supported by the File connector.

FTP

FTPS

SFTP

There are different connection configurations that can be used for the above protocols. They contain a common set of configurations and some additional configurations specific to the protocol.



**Note**

The File connector internally uses the [Apache VFS Library](https://commons.apache.org/proper/commons-vfs/). According to the selected connection type, the following VFS connection urls will be generated.

Local File

[file://] absolute-path

file:///home/someuser/somedir

file:///C:/Documents and Settings

FTPFTPSSFTP

The supported file systems and types are listed in [the Apache Commons VFS supported file systems page](https://commons.apache.org/proper/commons-vfs/filesystems.html).

Common configs to all connection types[¶](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/file-connector-config/#common-configs-to-all-connection-types)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| Connection Name | String | A unique name to identify the connection. | - | Yes |
| Connection Type | String | The protocol used for communicating with the file system. **Possible values**:  **Local**: Provides access to the files on the local physical file system.  **FTP**: Provides access to the files on an FTP server.  **FTPS**: Provides access to the files on an FTP server over SSL.  **SFTP**: Provides access to the files on an SFTP server (that is, an SSH or SCP server). | - | Yes |
| Working Directory | String | This is the working directory. The file paths in operations, which are associated with the connection, should be provided relative to this folder. **Note**: As per [VFS documentation](https://commons.apache.org/proper/commons-vfs/filesystems.html#Local_Files), for windows, the working directory of local connections should be as follows: /C:/Documents. | Defaults to file system root. | No |
| File Locking Behaviour | String | Specify whether to acquire node-specific lock (Local) or cluster-wide lock (Cluster) when locks are acquired in read and write operations.  **Local** When a lock is acquired, it is acquired within the context of file operations performed by that EI node only. Local lock acquired by some file operation on a particular EI node is not visible to the other EI nodes that may access the same file system.  **Cluster** When multiple EI nodes access the same file system performing read and write operations, you may use this behaviour. Here, when a file lock is acquired, it is visible to all file connector operations across the nodes. This is acquired by creating a .lock file in the same file system (for the file that is being accessed). The behaviour depends on the OS and the file system. Therefore, this feature may not work as intended in high-concurrent scenarios.  **Note**: File locking is available for read and write operations. When enabled, a file specific lock is acquired before the operation and released after the operation. Parallel read/write operations are restricted when locking is enabled by a file operation. | Local | Yes |

Common remote connection configs[¶](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/file-connector-config/#common-remote-connection-configs)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| Host | String | Host name of the file server. | - | Yes |
| Port | Number | The port number of the file server | - | Yes |
| Username | String | User name used to connect with the file server. | - | No |
| Password | String | Password to connect with the file server. | - | No |
| User Directory Is Root | Boolean | If set to false (default), VFS will choose the file system's root as the VFS's root. If you want to have the user's home as the VFS root, then set this to 'true'. | false | No |

FTP/FTPS-specific configs[¶](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/file-connector-config/#ftpftps-specific-configs)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| Is Passive | Boolean | If passive mode is enabled, set this to 'true'.  **Note** the following about 'Active/Passive' mode:  **Active Mode**: The client starts listening on a random port for incoming data connections from the server (the client sends the FTP command PORT to inform the server on which port it is listening). Nowadays, the client is typically behind a firewall (e.g. built-in Windows firewall) or NAT router (e.g. ADSL modem), unable to accept incoming TCP connections. The passive mode was introduced and is heavily used for this reason.  **Passive Mode**: In the passive mode, the client uses the control connection to send a PASV command to the server and then receives a server IP address and server port number from the server, which the client then uses to open a data connection to the server IP address and server port number received. | true | No |
| FTP Connection Timeout | Number | Specify the timeout in milliseconds for the initial control connection. | 100000 | No |
| FTP Socket Timeout | Number | Specify the socket timeout in milliseconds for the FTP client. | 150000 | No |

FTPS-specific configs[¶](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/file-connector-config/#ftps-specific-configs)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| KeyStore Path | String | The keystore path. | - | No |
| KeyStore Password | String | The password to the keystore. | - | No |
| TrustStore Path | String | The truststore path. | - | Yes |
| TrustStore Password | String | The password to the truststore. | - | Yes |
| Implicit Mode Enabled | Boolean | Set this to 'true' if [implicit mode](https://en.wikipedia.org/wiki/FTPS#Implicit)is enabled.  **Implicit**: The TLS ClientHello message should be initiated by client.  **Explicit**: The client must "explicitly request" security from an FTPS server. | false | No |
| Channel Protection Level | String | The FTP Data Channel protection level. Possible values: C,S,E,P. **Example**: Sends a “PROT P” command when implicit SSL is enabled. |  |  |

SFTP connection configs[¶](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/file-connector-config/#sftp-connection-configs)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| SFTP Connection Timeout | Number | The **Jsch** connection timeout in milli seconds. | - | No |
| SFTP Session Timeout | Number | The **Jsch** session timeout in milli seconds. | 100000 | No |
| Strict Host Key Check | Boolean | Specifies whether the Host key should be checked. If set to 'true', the connector (JSch) will always verify the public key (fingerprint) of the SSH/SFTP server. | false | No |
| Private Key File | String | Path to the private key file.  **Note**: You can only use a key generated in a classic manner (*ssh-keygen -m PEM*). | false | No |
| Private Key Passphrase | String | The passphrase of the private key. The security of a key (even if encrypted) is retained because it is not available to anyone else. You can specify the passphrase when generating keys. | false | No |

Operations[¶](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/file-connector-config/#operations)

The following operations allow you to work with the File Connector version 4. Click an operation name to see parameter details and samples on how to use it.

createDirectory

Creates a new folder in a provided directory path.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| Directory Path | String | The new directory path. | - | Yes |

**Response**

<**createDirectoryResult**>

<**success**>true</**success**>

</**createDirectoryResult**>

checkExist

Check if a given file or folder exists.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| File/Folder Path | String | The new directory path that should be scanned. | - | Yes |

**Response**

<**checkExistResult**>

<**success**>true</**success**>

<**fileExists**>true</**fileExists**>

</**checkExistResult**>

compress

Archives a file or a directory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| Folder/File To Compress | String | The path to the folder that should be compressed. | - | Yes |
| Targer File Path | String | The path to the compressed file that will be created. If the file already exists, it is overwritten. | - | Yes |
| Include Sub Directories | Boolean | Specifies whether the sub folders in the original folder should be included in the compressed file. | true | No |

**Response**

<**compressResult**>

<**success**>true</**success**>

<**NumberOfFilesAdded**>16</**NumberOfFilesAdded**>

</**compressResult**>

**Error**

<**compressResult**>

<**success**>false</**success**>

<**code**>700102</**code**>

<**detail**>File or directory to compress does not exist</**detail**>

</**compressResult**>

copy

Copies the file or folder specified by a source path to a target path. The source can be a file or a folder. If it is a folder, the copying is recursive.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| Source Path | String | The path to the file that should be copied. | - | Yes |
| Targer Path | String | The location (folder) to which the file should be copied. If the target folder does not exist at the time of copy, a new folder is created. | - | Yes |
| Source File Pattern | String | The file name pattern of the source file. Example: *[a-zA-Z][a-zA-Z].(txt|xml|jar)* ***Important****: The file pattern is case sensitive in the file connector version 4.0.2 and in later versions.* | - | No |
| Copy Including Source Parent | Boolean | Specify whether the parent folder should be copied from the file source along with the content. By default, only the content inside the folder will get copied. | false | No |
| Overwrite Existing Files | Boolean | Specifies whether or not to overwrite the file if the same file already exists in the target destination. | false | No |
| Rename To | String | The new name of the copied file. | Original file name. | No |
| File Pattern | String | The pattern (regex) of the files to be moved. **Example**: [a-zA-Z][a-zA-Z].(txt|xml|jar). Available in file-connector **v4.0.5** and above | - | No |

**Response**

<**copyFilesResult**>

<**success**>true</**success**>

</**copyFilesResult**>

**Error**

<**copyFilesResult**>

<**success**>false</**success**>

<**code**>700103</**code**>

<**detail**>Destination file already exists and overwrite not allowed</**detail**>

</**copyFilesResult**>

move

Moves the file or folder specified by the source path to the target directory. The source can be a file or a folder. If it is a folder, the moving is recursive.

The move operation can only move a file/folder within the same server. For example, you can move a file/folder from one local location to another local location, or from one remote location to another remote location on the same server. You cannot use the move operation to move a file/folder between different servers. If you want to move a file/folder from a local location to a remote location or vice versa, use the **copy** operation followed by **delete** operation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| Source Path | String | The path to the file that should be copied. | - | Yes |
| Targer Path | String | The location to which the file should be copied. | - | Yes |
| Create Parent Directories | Boolean | Specifies whether the parent directory should be created if it doesn't already exist in the target folder. | - | No |
| Include Parent | Boolean | Specify whether the parent folder should be copied from the file source along with the content. | false | No |
| Overwrite Existing Files | Boolean | Specifies whether or not to overwrite the file if the same file already exists in the target destination. | false | No |
| Rename To | String | The new name of the moved files. | Original file name. | No |

**Response**

<**moveFilesResult**>

<**success**>true</**success**>

</**moveFilesResult**>

**Error**

<**moveFilesResult**>

<**success**>false</**success**>

<**code**>700103</**code**>

<**detail**>Destination file already exists and overwrite not allowed</**detail**>

</**moveFilesResult**>

read

Reads the content and metadata of a file at a given path. Metadata of the file is added as properties while content is set to the message body (or optionally to a message context property).

Known message properties representing file properties:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| FILE\_LAST\_MODIFIED\_TIME | DateTime | The time at which the file was last modified. | - | Yes |
| FILE\_SIZE | Number | The file size (in bytes). | - | Yes |
| FILE\_IS\_DIR | Boolean | Specifies whether a folder directory is represented as the file. | false | Yes |
| FILE\_PATH | String | The file path. | - | Yes |
| FILE\_URL | String | The VFS URL of the file. | - | Yes |
| FILE\_NAME | String | The file name or folder name. | - | Yes |
| FILE\_NAME\_WITHOUT\_EXTENSION | String | The file name without the extension. | - | Yes |

Important:

When reading a folder, the first file that matches the pattern will be read first. Note that sub directories are not scanned. If you need to move or delete the file before reading the folder again, use the FILE\_NAME context variable.

The MIME type (content-type) of the message is determined by the file extension (i.e an XML file will be read as a message with the application/xml MIME type). However, users can force the MIME type by the ContentType parameter. Similarly, the Encoding parameter can be used to force the encoding.

You can set EnableLock to true to enable file system lock until the reading is completed and the stream is closed.

When large files are read, use streaming=true. Note that you need to first make necessary changes in the deployment.toml. The ContentType parameter also needs to be application/binary. Note that file reading modes are not applicable when streaming is set to true. The complete file is always streamed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| File Path | String | The path to the file that should be read. | - | Yes |
| File Pattern | String Regex | The file name pattern that should be matched when reading the file. | All text files (.\*.txt) | No |
| Add Result To | String | Specify where to add the result of the file that is read.  **Message Body**  **Message Property**: The payload that was in the message body before applying the **file read** operation will remain intact. | - | No |
| Property Name | String | If Add Result To==Message Property, you need to specify this value. Result of the **file read** operation will be added as a default scope property by the specified name. This can now be accessed later in the flow. | - | No |
| Read Mode | String | Available file reading modes: Read complete file, between lines, from line, upto line, single line, metadata only. | Reads complete file. | Yes |
| Start Line Num | Number | Starts reading the file from the specified line. | 1 | No |
| End Line Num | Number | Reads the file upto the specified line. | Last line of file. | No |
| Specific Line number | Number | Specific line to read. | When the reading mode is SINGLE\_LINE. | No |
| MIMEType | String | Content type of the message set to the payload by this operation | Determined by the file extension. | No |
| Encoding | String | Encoding of the message set to the payload by this operation. | UTF-8 | No |
| Enable Streaming | Boolean | Specifies whether or not streaming is used to read the file without any interpretation of the content. | false | No |
| Enable Locking | Boolean | Specifies whether or not to lock the file. | false | No |

**Response**

This is line one.

This is line two.

This is line three.

This is line four.

This is line five.

This is line six.

This is line seven.

This is line eight.

This lis line nine.

This is line ten.

**Full Log**

[2020-10-06 06:01:44,083] INFO {LogMediator} - {api:TestAPI} To: /filetest, MessageID: urn:uuid:7ab557c0-f9cb-4cf6-9c7b-f06a4640522a, Direction: request, message = After Read, FILE\_LAST\_MODIFIED\_TIME = 10/06/2020 05:46:39, FILE\_SIZE = 30, FILE\_IS\_DIR = false, FILE\_NAME = test1.txt, FILE\_PATH = /wso2/test, FILE\_URL = file:///Users/hasitha/temp/file-connector-test/wso2/test/test1.txt, FILE\_NAME\_WITHOUT\_EXTENSION = test1, Envelope: <?xml version='1.0' encoding='utf-8'?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Body><text xmlns="http://ws.apache.org/commons/ns/payload">This is test1.txt file content</text></soapenv:Body></soapenv:Envelope>

**Error**

<**readResult**>

<**success**>false</**success**>

<**code**>700102</**code**>

<**detail**>File or folder not found: file:///Users/hasitha/temp/file-connector-test/wso2/test/abcd.txt</**detail**>

</**readResult**>

rename

Rename a file in a specified path. The new name cannot contain path separators.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| Path | String | The path to the file that should be renamed. | - | Yes |
| Rename To | String | The file's new name. | - | Yes |
| Overwrite Existing Files | Boolean | Specifies whether or not to overwrite the file in the target directory (if the same file exists). | false | No |

**Response**

<**renameFileResult**>

<**success**>true</**success**>

</**renameFileResult**>

**Error**

<**renameFileResult**>

<**success**>false</**success**>

<**code**>700103</**code**>

<**detail**>Destination file already exists and overwrite not allowed</**detail**>

</**renameFileResult**>

delete

Deletes the files matching in a given directory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| File/Directory Path | String | The path to the file/folder that should be deleted. | - | Yes |
| Pattern to Match Files | String | The pattern that should be matched when listing files. This does not operate recursively on sub folders. **Important**: The file pattern is case sensitive in the file connector version 4.0.2 and in later versions. | All files. | No |

**Response**

For a single file:

<**deleteFileResult**>

<**success**>true</**success**>

</**deleteFileResult**>

For a folder:

<**deleteFileResult**>

<**success**>true</**success**>

<**numOfDeletedFiles**>5</**numOfDeletedFiles**>

</**deleteFileResult**>

unzip

Unzip a specified file to a given location. If a folder with the same name exists, it is overwritten.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| Zip File Path | String | The path to the ZIP file that should be unzipped. | - | Yes |
| Target Directory | String | The location (folder) to which the ZIP file should be unzipped. | - | Yes |

NOTE: The latest File connector (v4.0.7 onwards) supports the decompressing the .gz files.

**Response**

<**unzipFileResult**>

<**success**>true</**success**>

<**zipFileContent**>

<**test1.txt**>extracted</**test1.txt**>

<**test2.txt**>extracted</**test2.txt**>

<**hasitha--a1.txt**>extracted</**hasitha--a1.txt**>

<**hasitha--a2.txt**>extracted</**hasitha--a2.txt**>

<**hasitha--b--b2.txt**>extracted</**hasitha--b--b2.txt**>

<**hasitha--b--b1.txt**>extracted</**hasitha--b--b1.txt**>

<**hasitha--b--c--test1.txt**>extracted</**hasitha--b--c--test1.txt**>

<**hasitha--b--c--c1.txt**>extracted</**hasitha--b--c--c1.txt**>

</**zipFileContent**>

</**unzipFileResult**>

**On Error**

<**unzipFileResult**>

<**success**>false</**success**>

<**code**>700102</**code**>

<**detail**>File not found: file:///Users/hasitha/temp/file-connector-test/wso2/archievenew.zip</**detail**>

</**unzipFileResult**>

JSON equivalent:

{

"unzipFileResult": {

"success": false,

"code": 700102,

"detail": "File not found: file:///Users/hasitha/temp/file-connector-test/wso2/archievenew.zip"

}

}

splitFile

Splits a file into multiple smaller files.

If the folder does not exist, it will be created.

If the folder has files, they will be overwritten.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| Path to the file to split | String | The path to the file that should be split. | - | Yes |
| Target Directory | String | The path to the target folder where the new files should be created. | - | Yes |
| Split Mode | String | The split mode to use. The available options are as follows:  ChunkSize  Linecount  XPATH Expression | - | Yes |
| Chunk Size | Number | If the **Split Mode** is 'Chunk Size', specify the chunk size (in bytes) into which the file should be split. | - | - |
| Line Count | Number | If the **Split Mode** is 'Line Count', specify the number of lines by which the original file should be split. | - | - |
| XPATH Expression | Number | If the **Split Mode** is 'XPATH Expression', specify the expression by which the file should be split. Only applies when splitting XML files. | Chunk Size | Yes |

**Response**

<**splitFileResult**>

<**success**>true</**success**>

<**numberOfSplits**>6</**numberOfSplits**>

</**splitFileResult**>

**On Error**

<**splitFileResult**>

<**success**>false</**success**>

<**code**>700107</**code**>

<**detail**>Parameter 'xpathExpression' is not provided</**detail**>

</**splitFileResult**>

listFiles

Lists all the files (that match the specified pattern) in the directory path.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| Directory Path | String | The path to the directory from which files should be listed. | - | Yes |
| Matching Pattern | String | The file pattern that should be used to select files for listing. **Important**: The file pattern is case sensitive in the file connector version 4.0.2 and in later versions. | - | No |
| List Files in Sub Directories | Boolean | List files from sub directories recursively. | false | No |
| File Sort Attribute | String | Files will get sorted and listed according to one of the follow: Name, Size, LastModifiedTime. | Name | No |
| Sort Order | String | The sorting order applicable to the **File Sort** attribute. **Possible Values**: Ascending, Descending. | Ascending | No |
| Response Format | String | Format to list the files in response. **Possible Values**: Hierarchical, Flat. | Hierarchical | No |

**Response**

<**listFilesResult**>

<**success**>true</**success**>

<**directory** name="test">

<**file**>.DS\_Store</**file**>

<**directory** name="aa"/>

<**file**>abc.txt</**file**>

<**directory** name="hasitha">

<**file**>a1.txt</**file**>

<**file**>a2.txt</**file**>

</**directory**>

<**file**>input.xml</**file**>

<**file**>output.csv</**file**>

</**directory**>

</**listFilesResult**>

exploreZipFile

Explore the contents of a ZIP file in a specific location.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| Zip File Path | String | The path to the ZIP file that should be explored. | - | Yes |

**Response**

<**exploreZipFileResult**>

<**success**>true</**success**>

<**zipFileContent**>

<**file**>test1.txt</**file**>

<**file**>test2.txt</**file**>

<**file**>hasitha/a1.txt</**file**>

<**file**>hasitha/a2.txt</**file**>

<**file**>hasitha/b/b2.txt</**file**>

<**file**>hasitha/b/b1.txt</**file**>

<**file**>hasitha/b/c/test1.txt</**file**>

<**file**>hasitha/b/c/c1.txt</**file**>

</**zipFileContent**>

</**exploreZipFileResult**>

**On Error**

<**exploreZipFileResult**>

<**success**>false</**success**>

<**code**>700102</**code**>

<**detail**>Zip file not found at path file:///Users/hasitha/temp/file-connector-test/wso2/test/archieve.zip</**detail**>

</**exploreZipFileResult**>

mergeFiles

Merge the contents of multiple files in a folder to a single file.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| Source Directory Path | String | The path to the source folder containing the files that should be merged. | - | Yes |
| Target File Path | String | Path to the folder that holds the merged file. | - | Yes |
| File Pattern | String | The pattern that should be used for selecting the source files that should be merged. **Example**: [a-zA-Z][a-zA-Z]\*.(txt|xml|jar). | - | No |
| Write Mode | String | If the file already exists, this parameter will determine whether the existing file should be overwritten or appended during the merge. Possible values are Ovewrite or Append. | Overwrite | Yes |

**Response**

<**mergeFilesResult**>

<**success**>true</**success**>

<**detail**>

<**numberOfMergedFiles**>5</**numberOfMergedFiles**>

<**totalWrittenBytes**>992</**totalWrittenBytes**>

</**detail**>

</**mergeFilesResult**>

**On Error**

<**mergeFilesResult**>

<**success**>false</**success**>

<**code**>700102</**code**>

<**detail**>Directory not found: file:///Users/hasitha/temp/file-connector-test/wso2/toMergesnsdfb</**detail**>

</**mergeFilesResult**>

write

Writes content to a specified file.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Type | Description | Default Value | Required |
| File Connection | String | The name of the file connection configuration to use. | - | Yes |
| File Path | String | The path to the file that should be written (include file name and extension). | - | Yes |
| Content/Expression | String | Static content or expression to evaluate content. | The content will be fetched from the body ("$Body") of the incoming message. | Yes |
| MIME Type | String | The MIME type that will be applied in order to format the outgoing message.  Possible values: "Automatic","text/plain", "application/xml", "application/binary", "application/json", "text/xml".  If you don't want to change the MIME type of the message that has been mediated before this operation, use the default "Automatic" value. If the value is set to "application/binary", a binary file will get created with base-64 decoded content. | Automatic | Yes |
| Write Mode | String | If the file already exists, this parameter will determine whether the existing file should be overwritten or appended. You can also specify if a new file should be created. Possible values: Ovewrite, Append, Create New. | Overwrite | Yes |
| Append New Line | Boolean | Specifies whether a new line should be added to the end of the file after the content is written. | false | Yes |
| Encoding | String | Applied only when some static content or evaluated content is written. **Possible Values**: US-ASCII, UTF-8, or UTF-16. | UTF-8 | No |
| Compress | Boolean | Specifies whether the content should be compressed after the content is written. Only available when the **Write Mode** is ‘Create New ‘or ‘OverWrite’. | false | No |
| Enable Streaming | Boolean | Write file using the stream set to the message context. | false | No |
| Enable Locking | Boolean | Specifies whether or not to lock the file during file write.  **Note**: If the connector is processing a file named 'xyz.xml', a file called 'xyz.xml.lock' is created to represent the lock (with the CREATE\_NEW mode). Once the file connector operation is completed, the file is deleted. When you create the lock, you can set an expiry time as well. If the connector operation fails to create the file because it already exists, that means that another process is working on it. Then connector operation will fail and the application will have to retry. Information such as the servername and PID is written to the lock file, which may be important for debugging. | false | No |
| Add Result To | String | Specify where to add the result after writing the file.  **Message Body**: The result will be added to the message property.  **Message Property**: The payload that was in the message body before applying the **file write** operation will remain intact. | Message Body | No |
| Property Name | String | If the **Add Result To** attribute is set to "Message Property", specify a property name. The result of the file write operation will be added as a default scope property by the specified name. This property can be accessed later in the message flow. | - | Yes (If **Add Restul To** is "Message Property") |

**Response**

<**writeResult**>

<**success**>true</**success**>

<**writtenBytes**>16</**writtenBytes**>

</**writeResult**>

**Error**

<**writeResult**>

<**success**>false</**success**>

<**code**>700108</**code**>

<**detail**>Target file already exists. Path = file:///Users/hasitha/temp/file-connector-test/copy/kandy/hasitha.txt</**detail**>

</**writeResult**>

[Top](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/file-connector-config/)

[PreviousFile Connector Example](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/file-connector-example/" \o "File Connector Example)

[NextFile Connector Example](https://ei.docs.wso2.com/en/latest/micro-integrator/references/connectors/file-connector/3.x/file-connector-3.x-example/" \o "File Connector Example)

WSO2 Enterprise Integrator - Documentation

Copyright © [WSO2](https://wso2.com/) Inc. 2015-2021

Content licensed under [CC By 4.0.](https://creativecommons.org/licenses/by/4.0) | Sample code licensed under [Apache 2.0](http://www.apache.org/licenses/).