software AG

webMethods ActiveTransfer Built-In Services Reference

WEBMETHODS

Table of Contents

About this Guide

webMethods ActiveTransfer Built-In Services Reference is for developers who want to programmatically access the functions of webMethods ActiveTransfer. It describes the built-in services provided with ActiveTransfer.

ActiveTransfer includes several built-in services that enable you to import, export, and manage ActiveTransfer assets, execute configured events, generate PGP certificates, and view and modify configuration parameters.

You can execute built-in services using Software AG Designer. For information about Software AG Designer refer, *Software AG Designer Online Help*.

Important:
If you have a lower fix level of webMethods ActiveTransfer installed, some of the features described in *webMethods ActiveTransfer Built-In Services Reference* might not be available to you. For a cumulative list of fixes and features, see the latest fix readme on the Empower website at https://empower.softwareag.com.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	Identifies:
	Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies:
	Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at https://documentation.softwareag.com.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at https://empower.softwareag.com/.

You can find product information on the Software AG Empower Product Support website at https://empower.softwareag.com.

To submit feature/enhancement requests, get information about product availability, and download products, go to Products.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the Knowledge Center.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.aspx and give us a call.

Software AG Tech Community

You can find documentation and other technical information on the Software AG Tech Community website at https://techcommunity.softwareag.com. You can:

■ Access product documentation, if you have Tech Community credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.

■ Access articles, code samples, demos, and tutorials.

■ Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.

■ Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Admin Folder

Overview

Use the administrative services (services in the wm.mft.admin folder) to:

■ Export information from ActiveTransfer database to an XML file.

■ Import information into the ActiveTransfer database.

■ Retrieve and modify the ActiveTransfer configuration properties.

Summary of Elements in the Admin Folder

The following elements are available in this folder:

Element	Description
wm.mft.admin:archiveData	Archives transaction data available in the ActiveTransfer production schema or database.
wm.mft.admin:exportData	Exports ActiveTransfer assets to an XML file.
wm.mft.admin:importData	Imports ActiveTransfer assets from an XML export file.
wm.mft.admin:property	Retrieves and modifies ActiveTransfer configuration properties.
wm.mft.admin:manageProperties	Retrieves and modifies ActiveTransfer configuration properties.

wm.mft.admin:archiveData

Archives transaction data available in the ActiveTransfer production schema (Oracle) or database (SQL Server) to an archive schema or database.

Input Parameters

schema   **String** Specifies the name of the ActiveTransfer archive schema (Oracle) or database (SQL Server).

retentionPeriodInDays   **String** Specifies the number of days that the data should be retained in the production schema or database after the archival process.

The value cannot be less than 1 or greater than 9999.

batchSize   **String** Specifies the batch size for data archival.

The value cannot be less than 1. If an invalid value is set, then the default value is set to 5000.

Output Parameters

status	**String** Displays one of the following status values of the service execution:

- `true` - When the execution is successful.

- `false` - When the execution fails.

message	**String** Displays the message of the service execution status, whether `success` or `failure`.
validationErrors	**String** Displays the errors that occur during the service execution.

wm.mft.admin:exportData

Exports ActiveTransfer assets to an XML file.

Input Parameters

all	**String** (Optional) Specifies whether to export all ActiveTransfer assets or only selected assets. The following values apply:

- `true` - Export all assets. This is the default. When this value is specified, *exportData* is ignored.

- `false` - Export only the assets identified in *exportData*.

exportData	**Document List** (Optional) Specifies the assets, and their dependent assets where applicable, to export when *all* is set to `false`. Variables are:

Variable	Description
assetType	**String** (Optional) Specifies the type of ActiveTransfer asset to export. Valid values are `port`, `serverInstance`, `serverPreference`, `user`, `userTemplate`, `postProcessEvent`, `scheduleAction`, `VFS`, and `partnerMapping`.
all	**String** (Optional) Specifies whether to export all assets of the type specified in *assetType*. The following values apply:

- `true` - Export all assets of the specified asset type.

- `false` - Export only assets of a specified asset type that match the internal IDs specified in *ids*.

Variable	Description
ids	**String List** (Optional) Specifies the internal IDs associated with the assets to export. For the user asset types, the internal ID is considered the name of the asset; for all other asset types, the internal ID is the individual asset ID.

	Administrators can obtain internal IDs from the ActiveTransfer database.
names	**String List** (Optional) Specifies the name or path (for VFS hierarchy) of the assets to export. For the user asset type, the authentication ID (user ID) is considered the name of the asset; for all other asset types, the individual asset name is used.
	When there are multiple VFSes with the same name, it is useful to specify the exact path, so that ActiveTransfer exports only the specific VFS you require. Provide the complete VFS path in the asset hierarchy. For example: `Number/Jim/jimL/day/Trigger`. If you do not provide the complete hierarchical path, ActiveTransfer exports the first VFS that has the specified name, and ignores all others.
exportFileName	**String** Specifies the path and name of the XML file in which to save the exported assets.

Output Parameters

output	**Document** Contains the results of the export. The document contains the following keys:

Key	Description
exportFilePath	**String** The path and name of the XML file that contains the exported assets.
errorMessages	**String List** Any errors encountered during the export process.

Usage Notes

■ Use the wm.mft.admin:importData service to import the assets you exported using wm.mft.admin:exportData.

■ For more information about ActiveTransfer assets you can export, see the "Migrating Assets" chapter in *Managing File Transfers with webMethods ActiveTransfer*.

wm.mft.admin:importData

Imports ActiveTransfer assets from an XML export file.

Input Parameters

force	**String** If the imported assets match the assets present on the target system, this variable indicates whether the rows in the database should be overwritten. The following values apply:

- `true` - Overwrite rows in the database if the imported assets match the assets present on the target system. This is the default.

- `false` - Do not overwrite rows in the database if the imported assets match the assets present on the target system.

The *force* parameter does not apply when mapping partner assets; if the partner information already exists on the target server, the imported partner asset is ignored.

For more information about asset matching, see the "Migrating Assets" chapter in *Managing File Transfers with webMethods ActiveTransfer*.

all	**String** (Optional) Specifies whether to import all of the ActiveTransfer assets contained in the file created by the wm.mft.admin:exportData service or only selected assets. The following values apply:

- `true` - Import all assets.

- `false` - Import only the assets identified in *importData*. This is the default.

importDataFileName	**String** Specifies the name of the file from which data is to be imported.
importData	**Document List** (Optional) Specifies the assets to import when *all* is set to `false`. Variables are:

Variable	Description
assetType	**String** (Optional) Specifies the type of ActiveTransfer asset to import.
all	**String** (Optional) Specifies whether to import all assets of the type specified in *assetType*. The following values apply:

- `true` - Import all assets of the specified asset type.

- `false` - Import only assets of a specified asset type that match the internal IDs specified in *ids*.

Variable	Description
ids	**String List** (Optional) Specifies the internal IDs associated with the assets to import. For the user asset types, the internal ID is considered the name of the asset; for all other asset types, the internal ID is the individual asset ID.

names	**String List** (Optional) Specifies the name of the assets to import. For the user asset type, the authentication ID (user ID) is considered the name of the asset; for all other asset types, the individual asset name is used.

Output Parameters

output   **Document** Contains the results of the import. The document contains the following keys:

Key	Description
assets	**Document List** The data that was imported into the ActiveTransfer database. The document contains the following keys:

- *assetType* - **String** Indicates which asset type was imported.

- *id* - **String** Indicates the internal IDs associated with the asset.

- *name* - **String** Indicates the name of the asset.

- *status* - **String** Indicates the outcome of the import of the asset (`success` or `fail`).

- *message* - **String** Contains the status message from the last attempt to import the asset.

- *stacktrace* - **String** Indicates the stack trace of the exception if the import failed.

- *validationMessages* - **String** Contains any validation errors that occurred during import of each asset.

errorMessages	**String List** Any errors encountered during the import process.

Usage Notes

- Use the wm.mft.admin:exportData service to export the assets you import using wm.mft.admin:importData.

- You can use the wm.mft.admin:importData service to import an asset into ActiveTransfer without specifying the asset ID in the input parameter. ActiveTransfer generates a new identifier for the asset and returns the value using the output parameter, *id*.

- For more information about the ActiveTransfer assets that this service can import, see the "Migrating Assets" chapter in *Managing File Transfers with webMethods ActiveTransfer*.

wm.mft.admin:property

Retrieves and modifies ActiveTransfer configuration properties.

Input Parameters

action   **String** (Optional) Specifies whether to view (get) or update (set) the values of configuration properties. Specify `get` to retrieve the current value of the configuration properties. Specify `set` to update the configuration properties, and then provide values for the remaining input parameters in the service signature.

defaultHTTPPort   **String** (Optional) Specifies the HTTP port that ActiveTransfer Server uses to collect data for the Logs page. The default is 2080.

gatewayRegistrationPort   **String** (Optional) Specifies the registration port on which the ActiveTransfer Gateway runs. The default is 8500.

> **Note:**
> When ActiveTransfer Gateway runs on a port (for example, 8500), the Gateway uses the next numbered port (for example, 8501) as the data port. Therefore, when you configure the registration port, make sure the next port on the machine is also free.

> **Important:**
> If you supply this parameter, you must restart ActiveTransfer Gateway.

gatewayAcceptIPList   **String** (Optional) Specifies a list of IP addresses, separated by commas, representing the ActiveTransfer Servers that can connect to this ActiveTransfer Gateway. When this Gateway is started with this property set, only the ActiveTransfer Servers running on these IP addresses can connect to this Gateway.

> **Important:**
> If you supply this parameter, you must restart ActiveTransfer Gateway.

fromEmailID   **String** (Optional) Specifies the email address of the ActiveTransfer administrator who will send messages to ActiveTransfer users when adding or editing the user's profile. If no value is specified, the message is sent without any "from" email address.

serverPublicIP   **String** (Optional) Specifies the ActiveTransfer Server host name to use for the external server URL that is emailed to users for logging in to the server.

                            For example, suppose the host for ActiveTransfer Server port 8080 is defined on the Server Management page as localhost or 127.9.1.10. If

you do not set *serverPublicIP*, the server URL that is emailed to users will contain the internal IP address of the server (in this example, http://localhost:8080 or http://127.9.1.10:8080, respectively). If you set this parameter to the external host or domain name that your organization uses to represent the server's internal IP address, the server URL will reflect the external host name (for example, http://xyz.com:8080).

emailSubjectLine **String** (Optional) Specifies the subject line of the email message that is sent to the user. If this property is not set, messages are sent without any subject.

Output Parameters

defaultHTTPPort **String** The HTTP port that ActiveTransfer Server uses to collect data for the Logs page.

gateway RegistrationPort **String** The registration port on which the ActiveTransfer Gateway runs.

gatewayAcceptIPList **String** The list of IP addresses, separated by commas, representing the ActiveTransfer Servers that can connect to this ActiveTransfer Gateway.

fromEMailID **String** The email address of the ActiveTransfer administrator who will send messages to ActiveTransfer users when adding or editing the user's profile.

serverPublicIP **String** The ActiveTransfer Server host name to use for the external server URL that is emailed to users for logging in to the server.

emailSubject Line **String** The subject line of the email message that is sent to the user. If this property is not set, messages are sent without any subject.

Usage Notes

Restart ActiveTransfer Server after running this service.

wm.mft.admin:manageProperties

Retrieves and modifies ActiveTransfer configuration properties.

Input Parameters

action	**String** Specifies whether to view (get) or update (set) the values of configuration properties.

Specify get to retrieve the current value of the configuration properties. You can specify a property name to view its current value as part of the remaining input parameters in the service signature. If you do not specify a property name, then all the properties are listed.

Specify set to update the configuration properties, and then provide values for the remaining input parameters in the service signature.

props **Document List** If you select the get option in the *action* parameter, then specify the property names for which you want to view the values. If you select the set option in the *action* parameter, then specify the property names and the values that you want to update.

- *name* - **String** Specify the name of the property to retrieve or update.

- *value* - **String** Specify a value to update the property.

Output Parameters

message **String** (Optional) The message related to the result of the service execution, if any.

output **Document** Displays the result of the service execution.

props **Document List**

- get: Displays the name and value pair of properties in ActiveTransfer Server.

- set: Displays the name and updated value of properties in ActiveTransfer Server.

Specify the following fields to update a property:

- *name* - **String** Indicates the name of the property that is retrieved.

- *value* - **String** Indicates the value of the property that is viewed or updated.

Usage Notes

Restart ActiveTransfer Server or reload WmMFT package after running this service.

2 Asset Folder

Overview

You can use the built-in services provided in the Integration Server WmMFT package to add, update, deleted, and view ActiveTransfer Server assets. The services are provided in the wm.mft.assets folder and are organized based on the asset types:

■ Partner

■ Port

■ Post-processing events

■ Scheduled events

■ Monitor folder action

■ Template

■ User

■ VFS

Note:
For descriptions of document types used in the asset services, see "Asset Document Types" on page 69.

Summary of Elements in the Asset Folder

The following elements are available in this folder:

Asset	Service	Description
Partner		
	wm.mft.assets.partner:add	Creates a new partner in ActiveTransfer Server.
	wm.mft.assets.partner:delete	Deletes an existing partner from ActiveTransfer Server.
	wm.mft.assets.partner:list	Retrieves basic information for all the partners in ActiveTransfer Server.
	wm.mft.assets.partner.syncPartnerProfiles	Triggers synchronization of Trading Networks partner information with ActiveTransfer Server.
	wm.mft.assets.partner:update	Sets an existing partner in ActiveTransfer Server to the enterprise.

Asset	Service	Description
	wm.mft.assets.partner:view	Retrieves complete information from ActiveTransfer Server for a specified partner.
Port		
	wm.mft.assets.port:add	Creates a new port in ActiveTransfer Server or ActiveTransfer Gateway.
	wm.mft.assets.port:delete	Deletes an existing port in the ActiveTransfer Server or ActiveTransfer Gateway.
	wm.mft.assets.port:list	Retrieves basic information for all the ports configured in ActiveTransfer Server or ActiveTransfer Gateway.
	wm.mft.assets.port:start	Starts the specified ActiveTransfer Server or ActiveTransfer Gateway port.
	wm.mft.assets.port:stop	Stops or restarts the specified ActiveTransfer Server or ActiveTransfer Gateway port.
	wm.mft.assets.port:update	Updates the ActiveTransfer Server or ActiveTransfer Gateway port information.
	wm.mft.assets.port:view	Retrieves the complete information for a specified ActiveTransfer Server or ActiveTransfer Gateway port.
Post-processing Event		
	wm.mft.assets.postProcessEvent:add	Creates a new post-processing event in ActiveTransfer Server.
	wm.mft.assets.postProcessEvent:delete	Deletes an existing post-processing event in ActiveTransfer Server.
	wm.mft.assets.postProcessEvent:list	Retrieves basic information for all post-processing events configured in ActiveTransfer Server.

Asset	Service	Description
	wm.mft.assets.postProcessEvent:update	Updates a post-processing event in ActiveTransfer Server.
	wm.mft.assets.postProcessEvent:view	Retrieves complete information of a specified post-processing event in ActiveTransfer Server.
Scheduled Event		
	wm.mft.assets.scheduledEvent:add	Creates a new scheduled event in ActiveTransfer Server.
	wm.mft.assets.scheduledEvent:delete	Deletes an existing scheduled event in ActiveTransfer Server.
	wm.mft.assets.scheduledEvent:list	Retrieves basic information for all the scheduled events in ActiveTransfer Server.
	wm.mft.assets.scheduledEvent:update	Updates an existing scheduled event in ActiveTransfer Server.
	wm.mft.assets.scheduledEvent:view	Retrieves complete information for a specified scheduled event in ActiveTransfer Server.
Monitor folder action		
	wm.mft.assets.monitorFolderAction:add	Creates a new monitor folder action in ActiveTransfer Server.
	wm.mft.assets.monitorFolderAction:delete	Deletes an existing monitor folder action in ActiveTransfer Server.
	wm.mft.assets.monitorFolderAction:list	Retrieves basic information for all the monitor folder actions in ActiveTransfer Server.
	wm.mft.assets.monitorFolderAction:update	Updates an existing monitor folder action in ActiveTransfer Server.
	wm.mft.assets.monitorFolderAction:view	Retrieves complete information for a specified monitor folder action in ActiveTransfer Server.
Template		

Asset	Service	Description
	wm.mft.assets.template:add	Creates a new user template in ActiveTransfer Server.
	wm.mft.assets.template:delete	Deletes an existing ActiveTransfer Server user template.
	wm.mft.assets.template:list	Retrieves basic information on all the ActiveTransfer Server user templates.
	wm.mft.assets.template:update	Updates an existing ActiveTransfer Server user template.
	wm.mft.assets.template:view	Retrieves complete information on the specified ActiveTransfer Server user template.
User		
	wm.mft.assets.user:add	Creates a new user in ActiveTransfer Server.
	wm.mft.assets.user:delete	Deletes an existing ActiveTransfer Server user.
	wm.mft.assets.user:list	Retrieves basic information for all the users in ActiveTransfer Server.
	wm.mft.assets.user:update	Updates the ActiveTransfer Server user information.
	wm.mft.assets.user:view	Retrieves complete information for a specified ActiveTransfer Server user.
VFS		
	wm.mft.assets.vfs:add	Creates a new virtual folder in the ActiveTransfer Server Virtual File System (VFS).
	wm.mft.assets.vfs:changeParent	Use this service to change the parent virtual folder of a folder and link it to another parent folder, or to change a subfolder to a root folder in the

Asset	Service	Description
		ActiveTransfer Server Virtual File System (VFS).
	wm.mft.assets.vfs:delete	Deletes an existing virtual folder from the ActiveTransfer Server VFS.
	wm.mft.assets.vfs:list	Retrieves basic information for all virtual folders in the ActiveTransfer Server VFS
	wm.mft.assets.vfs:update	Updates the details of the ActiveTransfer Server VFS.
	wm.mft.assets.vfs:view	Retrieves complete information for a specified virtual folder in ActiveTransfer Server.

wm.mft.assets.partner:add

Creates a new partner in ActiveTransfer Server.

Input Parameters

partnerInfo   **Document Reference** A document reference to the wm.mft.assets.rec.partner:Partner document type, which specifies the details of the partner's information.

Output Parameters

success   **String** If the partner was successfully created in ActiveTransfer Server.

- `true` The new partner was successfully created.

- `false` The new partner was not created.

message   **String** A message that is logged after the service execution is complete.

validationErrors   **String List** An array of all the error messages logged during validation of the values assigned to the wm.mft.assets.rec.partner:Partner document type.

Usage Notes

- ActiveTransfer Server generates a unique ID for the new partner. Hence, it overrides the *partnerId* input parameter specified in *partnerInfo* for this service.

wm.mft.assets.partner:delete

Deletes an existing partner from ActiveTransfer Server.

Input Parameters

partnerId   **String** The internal identifier for the partner in ActiveTransfer Server.

Output Parameters

success   **String** If the partner was successfully deleted from ActiveTransfer Server.

- `true` The partner was successfully deleted.

- `false` The partner was not deleted.

Usage Notes

- The wm.mft.assets.partner.delete service does not delete a partner who is assigned to a virtual folder or an ActiveTransfer Server user. Hence, use this service to delete only partners who are not linked to a virtual folder or an ActiveTransfer Server user.

wm.mft.assets.partner:list

Retrieves basic information for all the partners in ActiveTransfer Server.

Input Parameters

None.

Output Parameters

success   **String** If the service was successful in retrieving information for all partners in ActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to retrieve partner information from ActiveTransfer.

partners   **Document List** Specifies the partner details. Variables are:

- *id* **String** The internal identifier for the partners in ActiveTransfer Server.

■ *name* **String** The partner name in ActiveTransfer Server.

■ *isTNPartner* **String** If the partner is a Trading Networks partner, the value `true` is returned.

■ *isEnterprise* **String** If the partner is the enterprise, the value `true` is returned.

Usage Notes

None.

wm.mft.assets.partner.syncPartnerProfiles

Triggers synchronization of Trading Networks partner information with ActiveTransfer Server.

Input Parameters

None.

Output Parameters

None.

Usage Notes

Although Trading Networks partner information refreshes in ActiveTransfer Server periodically, the synchronization process could fail for some reason. In such cases, use this service to manually trigger synchronization.

wm.mft.assets.partner:update

Sets an existing partner in ActiveTransfer Server to the enterprise.

Input Parameters

partnerId **String** The internal identifier for the partner in ActiveTransfer.

partnerName **String** (Optional) The partner name in ActiveTransfer.

setEnterprise **String** (Optional) If the partner profile should be changed to an enterprise. If the value of this parameter is set to `true`, ActiveTransfer sets the partner as the enterprise.

Output Parameters

success	**String** If the service was successful in changing the partner to an enterprise in ActiveTransfer.

-     ■ `true` The service was successful.

-     ■ `false` The service was not able to set the partner as the enterprise in ActiveTransfer.

message	**String** A message that is logged after the service execution is complete.
validationErrors	**String List** An array of all the error messages logged during validation of the *partnerId* and *partnerName* values.

Usage Notes

■ If you have installed Trading Networks along with ActiveTransfer Server and have configured a partner as an enterprise, the same will be retained as the enterprise in ActiveTransfer Server.

wm.mft.assets.partner:view

Retrieves complete information from ActiveTransfer Server for a specified partner.

Input Parameters

partnerId	**String** (Optional) The internal identifier for the partner in ActiveTransfer Server.

> **Note:**
> Specify either the *partnerId* or *partnerName*. If you specify the *partnerId*, ActiveTransfer Server ignores the value specified in the *partnerName* parameter.

partnerName	**String** (Optional) The partner name in ActiveTransfer Server.

Output Parameters

success	**String** If the service was successful in retrieving the partner in ActiveTransfer.

-     ■ `true` The service was successful.

-     ■ `false` The service was not able to retrieve the partner information in ActiveTransfer.

partnerInfo	**Document Reference** A document reference to the wm.mft.assets.rec.partner:Partner document type, which specifies the details of the partner's information.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.port:add

Creates a new port in ActiveTransfer Server or ActiveTransfer Gateway.

Input Parameters

portInfo   **Document Reference** A document reference to the wm.mft.assets.rec.port:PortInfo document type, which specifies the details of the ActiveTransfer Server or ActiveTransfer Gateway port.

Output Parameters

success   **String** If the service was successful in creating a new ActiveTransfer Server or ActiveTransfer Gateway port.

- `true` The service was successful.

- `false` The service was not able to create a new ActiveTransfer Server or ActiveTransfer Gateway port.

portId   **String** The internal identifier for the ActiveTransfer Server or ActiveTransfer Gateway port.

message   **String** A message that is logged after the service execution is complete.

validationErrors   **String List** An array of all the error messages logged during validation of *portInfo*.

Usage Notes

- ActiveTransfer Server generates a unique ID for the new port. Hence, it overrides the *portId* input parameter specified in *portInfo* for this service.

- The *instanceId* and *instanceName* in the wm.mft.assets.rec.port:PortInfo document type in the input parameter, *portInfo* determines if the port is a server port or a gateway port.

- The port is active as soon as it is created.

wm.mft.assets.port:delete

Deletes an existing port in the ActiveTransfer Server or ActiveTransfer Gateway.

Input Parameters

portId   **String** The internal identifier for the port in ActiveTransfer Server or ActiveTransfer Gateway.

Output Parameters

success   **String** If the service was successful in deleting an ActiveTransfer Server or ActiveTransfer Gateway port.

- `true` The service was successful.

- `false` The service was not able to delete the ActiveTransfer Server or ActiveTransfer Gateway port.

portId   **String** The internal identifier for the ActiveTransfer Server or ActiveTransfer Gateway port.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

- The port that is deleted is stopped in the ActiveTransfer Server or ActiveTransfer Gateway and will not be accessible.

wm.mft.assets.port:list

Retrieves basic information for all the ports configured in ActiveTransfer Server or ActiveTransfer Gateway.

Input Parameters

None.

Output Parameters

ports   **Document List** Specifies the ActiveTransfer Server or ActiveTransfer Gateway port details. Variables are:

- *portId* **String** The internal identifier for the ActiveTransfer Server or ActiveTransfer Gateway port.

- *portName* **String** The ActiveTransfer Server or ActiveTransfer Gateway port name.

- *host* **String** Host name or IP address of the Integration Server that hosts the ActiveTransfer Server or ActiveTransfer Gateway instance.

- *port* **String** Port number of the ActiveTransfer Server or ActiveTransfer Gateway instance.

- *protocol* **String** The transfer protocol to be used on this ActiveTransfer Server or ActiveTransfer Gateway port.

- *instanceName* **String** Name of the ActiveTransfer Server or ActiveTransfer Gateway instance.

  Note:
  You can have multiple instances of ActiveTransfer Server and ActiveTransfer Gateway in a My webMethods installation.

- *active* **String** Indicates if the port is active.

success   **String** If the service was successful in retrieving the ActiveTransfer Server or ActiveTransfer Gateway port details.

- `true` The service was successful.

- `false` The service was not able to retrieve the ActiveTransfer Server or ActiveTransfer Gateway port details.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.port:start

Starts the specified ActiveTransfer Server or ActiveTransfer Gateway port.

Input Parameters

portId   **String** The internal identifier for the ActiveTransfer Server or ActiveTransfer Gateway port.

instanceId   **String** (Optional) The internal identifier for the ActiveTransfer Server or ActiveTransfer Gateway instance.

instanceName   **String** (Optional) Name of the ActiveTransfer Server or ActiveTransfer Gateway instance.

> **Note:**
> You can have multiple instances of ActiveTransfer Server and ActiveTransfer Gateway in a My webMethods installation.

Output Parameters

success   **String** If the service was successful in starting the ActiveTransfer Server or ActiveTransfer Gateway port.

- `true` The service was successful.

- `false` The service was not able to start the ActiveTransfer Server or ActiveTransfer Gateway port.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

- If you do not specify a *instanceId* or *instanceName*, the service uses the default ActiveTransfer Server or ActiveTransfer Gateway instance.

wm.mft.assets.port:stop

Stops or restarts the specified ActiveTransfer Server or ActiveTransfer Gateway port.

Input Parameters

portId   **String** The internal identifier for the ActiveTransfer Server or ActiveTransfer Gateway port.

instanceId   **String** (Optional) The internal identifier for the ActiveTransfer Server or ActiveTransfer Gateway instance.

instanceName   **String** (Optional) Name of the ActiveTransfer Server or ActiveTransfer Gateway instance.

action   **String** The action to be performed on the ActiveTransfer Server or ActiveTransfer Gateway port.

- `stop` (default) Stops the port.

- `restart` Restarts the port

Output Parameters

success	**String** If the service was successful in performing the action on the ActiveTransfer Server or ActiveTransfer Gateway port.

- `true` The service was successful.

- `false` The service was not able to perform the action on the ActiveTransfer Server or ActiveTransfer Gateway port.

message	**String** A message that is logged after the service execution is complete.

Usage Notes

- If you do not specify an *instanceId* or *instanceName*, the service uses the default ActiveTransfer Server instance.

wm.mft.assets.port:update

Updates the ActiveTransfer Server or ActiveTransfer Gateway port information.

Input Parameters

portInfo	**Document Reference** A document reference to the wm.mft.assets.rec.port:PortInfo document type, which specifies the details of the ActiveTransfer Server or ActiveTransfer Gateway port.

Output Parameters

success	**String** If the service was successful in updating the ActiveTransfer Server or ActiveTransfer Gateway port information.

- `true` The service was successful.

- `false` The service was not able to update the ActiveTransfer Server or ActiveTransfer Gateway port information.

portId	**String** The internal identifier for the ActiveTransfer Server or ActiveTransfer Gateway port.
message	**String** A message that is logged after the service execution is complete.
validationErrors	**String List** An array of all the error messages logged during validation of *portInfo*.

Usage Notes

None.

wm.mft.assets.port:view

Retrieves the complete information for a specified ActiveTransfer Server or ActiveTransfer Gateway port.

Input Parameters

portId   **String** (Optional) The internal identifier for the ActiveTransfer Server or ActiveTransfer Gateway port.

> **Note:**
> Specify either the *portId* or *portName*. If you specify the *portId*, ActiveTransfer Server ignores the value specified in the *portName* parameter.

portName   **String** (Optional) The ActiveTransfer Server or ActiveTransfer Gateway port name.

Output Parameters

success   **String** If the service was successful in retrieving the ActiveTransfer Server or ActiveTransfer Gateway port information.

-     true The service was successful.

-     false The service was not able to retrieve the ActiveTransfer Server or ActiveTransfer Gateway port information.

portInfo   **Document Reference** A document reference to the wm.mft.assets.rec.port:PortInfo document type, which specifies the details of the ActiveTransfer Server or ActiveTransfer Gateway port.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.postProcessEvent:add

Creates a new post-processing event in ActiveTransfer Server.

Input Parameters

eventInfo	**Document Reference** A document of type wm.mft.assets.rec.event:PostProcessEvent, which specifies the details of the post-processing event in ActiveTransfer Server.

Output Parameters

success	**String** If the service was successful in creating a new post-processing event in ActiveTransfer Server.

- ◾ true The service was successful.

- ◾ false The service was not able to create the post-processing event in ActiveTransfer Server.

eventId	**String** The internal identifier for the post-processing event in ActiveTransfer Server.
message	**String** A message that is logged after the service execution is complete.
validationErrors	**String List** An array of all the error messages logged during validation of *eventInfo*.

Usage Notes

- ◾ ActiveTransfer Server generates a unique ID for a new post-processing event. Hence, it overrides the *eventId* input parameter specified in *eventInfo* for this service.

wm.mft.assets.postProcessEvent:delete

Deletes an existing post-processing event in ActiveTransfer Server.

Input Parameters

eventId	**String** The internal identifier for the post-processing event in ActiveTransfer Server.

Output Parameters

success	**String** If the service was successful in deleting the post-processing event in ActiveTransfer Server.

- ◾ true The service was successful.

■ `false` The service was not able to delete the post-processing event in ActiveTransfer Server.

Usage Notes

None.

wm.mft.assets.postProcessEvent:list

Retrieves basic information for all post-processing events configured in ActiveTransfer Server.

Input Parameters

None.

Output Parameters

success **String** If the service was successful in retrieving the list of all post-processing events inActiveTransfer Server.

    ■ `true` The service was successful.

    ■ `false` The service was not able to retrieve the list of the post-processing events in ActiveTransfer Server.

events **Document List** Specifies the details of the post-processing events inActiveTransfer Server. Variables are:

    ■ *id* **String** The internal identifier for the post-processing event in ActiveTransfer Server.

    ■ *name* **String** Name of the post-processing event in ActiveTransfer Server.

    ■ *description* **String** The description of the post-processing event in ActiveTransfer Server.

    ■ *active* **String** The status of the post-processing event in ActiveTransfer Server.

        ■ `true` The post-processing event is active.

        ■ `false` The post-processing event is not active.

message **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.postProcessEvent:update

Updates a post-processing event in ActiveTransfer Server.

Input Parameters

eventInfo   **Document Reference** A document of type wm.mft.assets.rec.event:PostProcessEvent, which specifies the details of the post-processing event in ActiveTransfer Server.

Output Parameters

success   **String** If the service was successful in updating the post-processing event in ActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to update the post-processing event in ActiveTransfer Server.

message   **String** A message that is logged after the service execution is complete.

validationErrors   **String List** An array of all the error messages logged during validation of *eventInfo*.

Usage Notes

None.

wm.mft.assets.postProcessEvent:view

Retrieves complete information of a specified post-processing event in ActiveTransfer Server.

Input Parameters

eventId   **String** (Optional) The internal identifier for the post-processing event in ActiveTransfer Server.

> **Note:**
> Specify either the *eventId* or *eventName*. If you specify the *eventId*, ActiveTransfer Server ignores the value specified in the *eventName* parameter.

eventName   **String** (Optional) The name of the post-processing event in ActiveTransfer Server.

Output Parameters

success   **String** If the service was successful in retrieving the post-processing event in ActiveTransfer Server.

- ■ `true` The service was successful.

- ■ `false` The service was not able to retrieve the post-processing event in ActiveTransfer Server.

eventInfo   **Document Reference** A document of type wm.mft.assets.rec.event:PostProcessEvent, which specifies the details of the post-processing event in ActiveTransfer Server.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.scheduledEvent:add

Creates a new scheduled event in ActiveTransfer Server.

Input Parameters

scheduleInfo   **Document Reference** A document of type wm.mft.assets.rec.event:ScheduledEvent, which specifies the details of the scheduled event in ActiveTransfer Server.

Output Parameters

success   **String** If the service was successful in creating a new scheduled event in ActiveTransfer Server.

- ■ `true` The service was successful.

- ■ `false` The service was not able to create the scheduled event in ActiveTransfer Server.

scheduleId   **String** The internal identifier for the new scheduled event in ActiveTransfer Server.

message   **String** A message that is logged after the service execution is complete.

validationErrors	**String List** An array of all the error messages logged during validation of *scheduleInfo*.

Usage Notes

■ ActiveTransfer Server generates a unique identifier for the new scheduled event. Hence, it overrides the *scheduleId* input parameter specified in *scheduleInfo* for this service.

wm.mft.assets.scheduledEvent:delete

Deletes an existing scheduled event in ActiveTransfer Server.

Input Parameters

scheduleId	**String** The internal identifier for the scheduled event in ActiveTransfer Server.

Output Parameters

success	**String** If the service was successful in creating a new scheduled event in ActiveTransfer Server.

　　　■ `true` The service was successful.

　　　■ `false` The service was not able to create the scheduled event in ActiveTransfer Server.

Usage Notes

None.

wm.mft.assets.scheduledEvent:list

Retrieves basic information for all the scheduled events in ActiveTransfer Server.

Input Parameters

None.

Output Parameters

success	**String** If the service was successful in retrieving basic information for all the scheduled events inActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to retrieve information for the scheduled events in ActiveTransfer Server.

schedules   **Document List** Contains basic details for all the scheduled events in ActiveTransfer Server. Variables are:

- *id* **String** The internal identifier for the scheduled event in ActiveTransfer Server.

- *name* **String** Name of the scheduled event in ActiveTransfer Server.

- *description* **String** The description of the scheduled event in ActiveTransfer Server.

- *isActive* **String** The status of the scheduled event in ActiveTransfer Server.

  - `true` The scheduled event is active.

  - `false` The scheduled event is not active.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.scheduledEvent:update

Updates an existing scheduled event in ActiveTransfer Server.

Input Parameters

scheduleInfo   **Document Reference** A document of type wm.mft.assets.rec.event:ScheduledEvent, which specifies the details of the scheduled event in ActiveTransfer Server.

Output Parameters

success   **String** If the service was successful in updating the scheduled event in ActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to update the scheduled event in ActiveTransfer Server.

scheduleId   **String** The internal identifier for the scheduled event in ActiveTransfer Server.

message   **String** A message that is logged after the service execution is complete.

validationErrors   **String List** An array of all the error messages logged during validation of *scheduleInfo*.

Usage Notes

None.

wm.mft.assets.scheduledEvent:view

Retrieves complete information for a specified scheduled event in ActiveTransfer Server.

Input Parameters

scheduleId   **String** (Optional) The internal identifier for the scheduled event in ActiveTransfer Server.

> **Note:**
> Specify either the *ScheduleId* or *ScheduleName*. If you specify the *ScheduleId*, ActiveTransfer Server ignores the value specified in the *ScheduleName* parameter.

scheduleName   **String** (Optional) The name of the scheduled event in ActiveTransfer Server.

Output Parameters

success   **String** If the service was successful in retrieving information on the specified scheduled event in ActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to retrieve the scheduled event in ActiveTransfer Server.

scheduleInfo   **Document Reference** A document of type wm.mft.assets.rec.event:ScheduledEvent, which specifies the details of the scheduled event in ActiveTransfer Server.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.monitorFolderAction:add

Creates a new monitor folder action in ActiveTransfer Server.

Input Parameters

eventInfo	**Document Reference** A document of type wm.mft.assets.rec.event:MonitorFolderAction, which specifies the details of the monitor folder action in ActiveTransfer Server.

Output Parameters

success	**String** If the service was successful in creating a new monitor folder action in ActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to create the monitor folder action in ActiveTransfer Server.

eventId	**String** The internal identifier for the monitor folder action in ActiveTransfer Server.
message	**String** A message that is logged after the service execution is complete.
validationErrors	**String List** An array of all the error messages logged during validation of *eventInfo*.

Usage Notes

- ActiveTransfer Server generates a unique ID for a new monitor folder action. Hence, it overrides the *eventId* input parameter specified in *eventInfo* for this service.

wm.mft.assets.monitorFolderAction:delete

Deletes an existing monitor folder action in ActiveTransfer Server.

Input Parameters

actionId	**String** The internal identifier for the monitor folder action in ActiveTransfer Server.

Output Parameters

success	**String** If the service was successful in deleting the monitor folder action in ActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to delete the monitor folder in ActiveTransfer Server.

Usage Notes

None.

wm.mft.assets.monitorFolderAction:list

Retrieves basic information for all monitor folder action configured in ActiveTransfer Server.

Input Parameters

None.

Output Parameters

success	**String** If the service was successful in retrieving the list of all monitor folder action inActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to retrieve the list of the monitor folder action in ActiveTransfer Server.

actions	**Document List** Specifies the details of the monitor folder action inActiveTransfer Server. Variables are:

- *id* **String** The internal identifier for the monitor folder action in ActiveTransfer Server.

- *name* **String** Name of the monitor folder action in ActiveTransfer Server.

- *description* **String** The description of the monitor folder action in ActiveTransfer Server.

- *active* **String** The status of the monitor folder action in ActiveTransfer Server.

  - true The monitor folder action is active.

  - false The monitor folder action is not active.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.monitorFolderAction:update

Updates a monitor folder action in ActiveTransfer Server.

Input Parameters

eventInfo   **Document Reference** A document of type wm.mft.assets.rec.event:MonitorFolderAction, which specifies the details of the monitor folder action in ActiveTransfer Server.

Output Parameters

success   **String** If the service was successful in updating the monitor folder action in ActiveTransfer Server.

  - true The service was successful.

  - false The service was not able to update the monitor folder action in ActiveTransfer Server.

message   **String** A message that is logged after the service execution is complete.

validationErrors   **String List** An array of all the error messages logged during validation of *eventInfo*.

Usage Notes

None.

wm.mft.assets.monitorFolderAction:view

Retrieves complete information of a specified monitor folder action in ActiveTransfer Server.

Input Parameters

actionId	**String** (Optional) The internal identifier for the monitor folder action in ActiveTransfer Server.

> **Note:**
> Specify either the *actionId* or *actionName*. If you specify the *actionId*, ActiveTransfer Server ignores the value specified in the *actionName* parameter.

actionName	**String** (Optional) The name of the monitor folder action in ActiveTransfer Server.

Output Parameters

success   **String** If the service was successful in retrieving the monitor folder action in ActiveTransfer Server.

- ■ `true` The service was successful.

- ■ `false` The service was not able to retrieve the monitor folder action in ActiveTransfer Server.

eventInfo   **Document Reference** A document of type wm.mft.assets.rec.event:MonitorFolderAction, which specifies the details of the monitor folder action in ActiveTransfer Server.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.template:add

Creates a new user template in ActiveTransfer Server.

Input Parameters

templateInfo   **Document Reference** A document reference to the wm.mft.assets.rec.user:UserTemplate document type, which specifies the details of the ActiveTransfer Server user template.

Output Parameters

templateId   **String** An internal identifier for the ActiveTransfer Server user template.

success   **String** If the service was successful in creating the ActiveTransfer Server user template.

- ▪ `true` The service was successful.

- ▪ `false` The service was not able to create the ActiveTransfer Server user template.

message   **String** A message that is logged after the service execution is complete.

validationErrors **String List** An array of all the error messages logged during validation of *templateInfo*.

Usage Notes

- ▪ ActiveTransfer Server generates a unique ID for the new user template. Hence, it overrides the *templateId* input parameter specified in *templateInfo* for this service.

wm.mft.assets.template:delete

Deletes an existing ActiveTransfer Server user template.

Input Parameters

templateId   **String** An internal identifier for the ActiveTransfer Server user template.

Output Parameters

success   **String** If the service was successful in deleting the ActiveTransfer Server user template.

- ▪ `true` The service was successful.

- ▪ `false` The service was not able to delete the ActiveTransfer Server user template.

Usage Notes

None.

Type	Value	Description

wm.mft.assets.template:list

Retrieves basic information on all the ActiveTransfer Server user templates.

Input Parameters

None.

Output Parameters

success   **String** If the service was successful in retrieving the ActiveTransfer Server user template.

■ `true` The service was successful.

■ `false` The service was not able to retrieve the ActiveTransfer Server user template.

templates   **Document List** Specifies the ActiveTransfer Server user template details. Variables are:

■ *id* String The internal identifier of the user template in ActiveTransfer Server.

■ *name* String The name of the user template.

■ *description* The description of the user template

■ *defaultTemplate* String If the user template is the default template.

■ `true` The template is the default.

■ `false` The template is not the default.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.template:update

Updates an existing ActiveTransfer Server user template.

Input Parameters

templateInfo **Document Reference** A document reference to the wm.mft.assets.rec.user:UserTemplate document type, which specifies the details of the ActiveTransfer Server user template.

Output Parameters

success **String** If the service was successful in updating the ActiveTransfer Server user template.

- ■ true The service was successful.

- ■ false The service was not able to update the ActiveTransfer Server user template.

message **String** A message that is logged after the service execution is complete.

validationErrors **String List** An array of all the error messages logged during validation of *templateInfo*.

Usage Notes

None.

wm.mft.assets.template:view

Retrieves complete information on the specified ActiveTransfer Server user template.

Input Parameters

templateId **String** (Optional) An internal identifier for the ActiveTransfer Server user template.

> **Note:**
> Specify either the *templateId* or *templateName*. If you specify the *templateId*, ActiveTransfer Server ignores the value specified in the *templateName* parameter.

templateName **String** (Optional) The name of the ActiveTransfer Server user template.

Output Parameters

success **String** If the service was successful in retrieving the specified ActiveTransfer Server user template.

- ■ `true` The service was successful.

- ■ `false` The service was not able to retrieve the specified ActiveTransfer Server user template.

templateInfo **Document Reference** A document reference to the wm.mft.assets.rec.user:UserTemplate document type, which specifies the details of the ActiveTransfer Server user template.

message **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.user:add

Creates a new user in ActiveTransfer Server.

Input Parameters

addToSystemDirectory **String** (Optional) Specifies if the new user should be added to the My webMethods Server system directory.

- ■ `true` Add the user to the system directory.

- ■ `false` Create the user only in ActiveTransfer Server.

userInfo **Document Reference** A document reference to the wm.mft.assets.rec.user:User document type, which specifies the details of the ActiveTransfer Server user.

Output Parameters

success **String** If the service was successful in creating a new ActiveTransfer Server user.

- ■ `true` The service was successful.

- ■ `false` The service was not able to create a new ActiveTransfer Server user.

userId	**String** The internal identifier for the ActiveTransfer Server user.
message	**String** A message that is logged after the service execution is complete.
validationErrors	**String List** An array of all the error messages logged during validation of *userInfo*.

Usage Notes

■ ActiveTransfer Server generates a unique ID for the new user. Hence, it overrides the *userId* input parameter specified in *userInfo* for this service.

wm.mft.assets.user:delete

Deletes an existing ActiveTransfer Server user.

Input Parameters

| *userId* | **String** The internal identifier for the ActiveTransfer Server user. |

Output Parameters

| *success* | **String** If the service was successful in deleting the ActiveTransfer Server user. |

■ `true` The service was successful.

■ `false` The service was not able to delete the ActiveTransfer Server user.

Usage Notes

None.

wm.mft.assets.user:list

Retrieves basic information for all users in ActiveTransfer Server.

Input Parameters

None.

Output Parameters

success   **String** If the service was successful in retrieving the ActiveTransfer Server users.

- `true` The service was successful.

- `false` The service was not able to retrieve a list of the ActiveTransfer Server users.

users   **Document List** Specifies the ActiveTransfer Server user details. Variables are:

- *userId* **String** The internal identifier for the ActiveTransfer Server user.

- *disableLogin* **String** If you want to disable the *userId* and prevent the user from logging in to the ActiveTransfer Server.

- *templateId* **String** The internal identifier of the template used to create the user.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.user:update

Updates the ActiveTransfer Server user information.

Input Parameters

userInfo   **Document Reference** A document reference to the document type, wm.mft.assets.rec.user:UserInfo which specifies the details of the ActiveTransfer Server user.

Output Parameters

success   **String** If the service was successful in updating the ActiveTransfer Server user information.

- `true` The service was successful.

- `false` The service was not able to update the ActiveTransfer Server user information.

message	**String** A message that is logged after the service execution is complete.
validationErrors	**String List** An array of all the error messages logged during validation of *userInfo*.

Usage Notes

None.

wm.mft.assets.user:view

Retrieves complete information for a specified ActiveTransfer Server user.

Input Parameters

userId	**String** The internal identifier for the ActiveTransfer Server user.

Output Parameters

success	**String** If the service was successful in updating the ActiveTransfer Server user information.

- `true` The service was successful.

- `false` The service was not able to retrieve the specified ActiveTransfer Server user information.

userInfo	**Document Reference** A document reference to the wm.mft.assets.rec.user:User document type, which specifies the details of the ActiveTransfer Server user.
message	**String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.vfs:add

Creates a new virtual folder in the ActiveTransfer Server Virtual File System (VFS).

Input Parameters

parentVfsId	**String** (Optional) The internal identifier of the parent virtual folder under which you want to create this virtual folder if the folder is not the root folder in a ActiveTransfer Server VFS.
parentVfsPath	**String** (Optional) Specify the path of the parent virtual folder for the new virtual folder, if you want to create the virtual folder under the parent virtual folder. For example, the value data/inbound/po indicates that data is the root folder with the subfolder inbound, and po is a subfolder in the inbound folder.
vfsDetails	**Document Reference** A document reference to the wm.mft.assets.rec.vfs:VfsDetails document type, which specifies the details of the virtual folder in ActiveTransfer Server.

Output Parameters

success	**String** If the service was successful in creating a new virtual folder in ActiveTransfer Server.
	■ true The service was successful.
	■ false The service was not able to create a new virtual folder in ActiveTransfer Server.
vfsId	**String** The internal identifier for the virtual folder in ActiveTransfer Server.
message	**String** A message that is logged after the service execution is complete.
validationErrors	**String List** An array of all the error messages logged during validation of *vfsDetails*.

Usage Notes

■ ActiveTransfer Server generates a unique ID for the new virtual folder. Hence, it overrides the *vfsId* input parameter specified in *vfsDetails* for this service.

■ If *parentVfsId* is specified, ActiveTransfer Server ignores the value provided in *parentVfsPath*.

wm.mft.assets.vfs:changeParent

Use this service to change the parent virtual folder of a folder and link it to another parent folder, or to change a subfolder to a root folder in the ActiveTransfer Server Virtual File System (VFS).

Input Parameters

vfsId	**String** The internal identifier of the virtual folder in ActiveTransfer Server for which you want to change the parent folder.
parentVfsId	**String** The internal identifier of the new parent virtual folder to which you want to link the virtual folder identified by *vfsId* above. Specify a null value in this parameter to set the virtual folder as the root folder in the VFS.

Output Parameters

success	**String** If the service was successful in assigning a new parent to the specified virtual folder in ActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to change the parent of the virtual folder in ActiveTransfer Server.

message	**String** A message that is logged after the service execution is complete.
validationErrors	**String List** An array of all the error messages logged during validation of input parameters.

Usage Notes

- After this service is executed, the folder identified by *VfsId* will be the child of the folder identified by *parentVfsId*.

- If you specify a null value in *parentVfsId* for the folder identified by *vfsId*, the folder will not be assigned a parent after you run this service. The folder will be moved to the root level.

- If the child folder already has a parent and you specify a new parent in *parentVfsId*, running this service will remove the existing link between the parent and child folder. The child folder will be assigned the new parent.

- If the new parent has a partner, the child folder will be assigned the partner of the new parent. If *parentVfsId* is null, the partner associated with the child folder will not be changed.

- If the *parentVfsId* is not null, after you run this service, the user access permissions of the users associated with the child folder will change as follows:

  - All user access permissions inherited from the old parent folder are removed.

  - The child folder will inherit all the user permissions other than the folder traversal permission of the new parent.

  - If you have explicitly assigned user permissions to the child folder, these permissions will be retained.

■ Circular parent-child relationships in the VFS will not be allowed.

wm.mft.assets.vfs:delete

Deletes an existing virtual folder from the ActiveTransfer Server VFS.

Input Parameters

vfsId   **String** The internal identifier for the virtual folder in ActiveTransfer Server.

Output Parameters

success   **String** If the service was successful in deleting the virtual folder in
                 ActiveTransfer Server.

     ■ `true` The service was successful.

     ■ `false` The service was not able to delete the virtual folder in
                 ActiveTransfer Server.

vfsId   **String** The internal identifier for the virtual folder in ActiveTransfer Server.

message   **String** A message that is logged after the service execution is complete.

validationErrors **String List** An array of all the error messages logged during validation
                 of the input parameter.

Usage Notes

■ Deleting a virtual folder does not delete its contents in the local or remote location.

■ You can delete a folder at the root level or its subfolders. When you delete a folder, all its sub
   folders will also be deleted.

wm.mft.assets.vfs:list

Retrieves basic information for all virtual folders in the ActiveTransfer Server VFS. You can use
wild card characters in a search string for the input parameter, *vfsName* to retrieve information
only on the virtual folders whose name match the search string.

Input Parameters

vfsName   **String** The name of the virtual folder(s) in ActiveTransfer Server. You can use a
                 wild card in the parameter to retrieve only those folder(s) that match the search
                 string. This parameter is not case sensitive.

partnerId **String** The internal identifier of the partner associated with the VFS.

Output Parameters

vfsList **Document Reference List** A document reference to the wm.mft.assets.rec.vfs:VfsInfo document type, which specifies the details of the virtual folders in the VFS.

success **String** If the service was successful in retrieving the virtual folder details from ActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to retrieve the virtual folder details from ActiveTransfer Server.

message **String** A message that is logged after the service execution is complete.

Usage Notes

None.

wm.mft.assets.vfs:update

Updates the details of the virtual folder in ActiveTransfer Server VFS.

Input Parameters

vfsDetails **Document Reference** A document reference to the wm.mft.assets.rec.vfs:VfsDetails document type, which specifies the details of the virtual folder in ActiveTransfer Server.

Output Parameters

success **String** If the service was successful in updating the virtual folder details in ActiveTransfer Server.

- `true` The service was successful.

- `false` The service was not able to update the virtual folder details in ActiveTransfer Server.

vfsId **String** The internal identifier for the virtual folder in ActiveTransfer Server.

message **String** A message that is logged after the service execution is complete.

2 Asset Folder

validationErrors **String** An array of all the error messages logged during validation of *vfsDetails*.

Usage Notes

■ The *vfsId* variable in the document type referred in *vfsDetails* identifies the virtual folder to be updated.

wm.mft.assets.vfs:view

Retrieves complete information for a specified virtual folder in the ActiveTransfer Server VFS.

Input Parameters

vfsId   **String** (Optional) The internal identifier for the virtual folder in ActiveTransfer Server on which you want to retrieve information.

vfsPath   **String** Path of the virtual folder in ActiveTransfer Server on which you want to retrieve information. For example, if you want to retrieve information on the folder po which is a subfolder under inbound, and inbound is a subfolder under the root folder data, the *vfsPath* value to be specified is data/inbound/po.

Output Parameters

success   **String** If the service was successful in retrieving information on the specified virtual folder in ActiveTransfer Server.

    ■ true The service was successful.

    ■ false The service was not able to retrieve information on the specified virtual folder in ActiveTransfer Server.

vfsDetails   **Document Reference** A document reference to the wm.mft.assets.rec.vfs:VfsDetails document type, which specifies the details of the virtual folder in ActiveTransfer Server.

childVfsList **Document List** A list of all the subfolders under the virtual folder. Variables are:

    ■ *vfsId* **String** Internal identifier of the subfolder under the virtual folder.

    ■ *vfsName* **String** Name of the subfolder under the virtual folder.

message   **String** A message that is logged after the service execution is complete.

Usage Notes

- If you provide a value in the input parameter *vfsId*, the value provided in the *vfsPath* input parameter will be ignored.

3 Schedule Folder

Overview

Use the schedule services (services in the wm.mft.schedule folder) to create a remote URL in the ActiveTransfer format and to test a scheduled event configured in ActiveTransfer.

Summary of Elements in the Schedule Folder

The following elements are available in this folder:

Element	Description
wm.mft.schedule:createRemoteURL	Creates a remote file URL in a format that is understood by ActiveTransfer Server.
wm.mft.schedule:executeEvent	Executes an event configured on the Event Management page's **Scheduled Events** tab.

wm.mft.schedule:createRemoteURL

Creates a remote file URL in a format that is understood by ActiveTransfer Server.

Input Parameters

remotePath　　**String** The folder path on a remote server. The format is *protocol://host:port/relative path* (for example, ftp://ftpmc:56/projectfolder/download/).

userName　　**String** The name of the user who has access to the *remotePath*.

password　　**String** The password for the user.

Output Parameters

remoteURL　　**String** The remote path URL in the format *<protocol>://<username>:<password>@<host>:<port>/<path>/* (for example, `FTP://user:password@ftp.softwareag.com/outbound/`).

success　　**String** If the service was successful in creating the remote path URL.

- ■ *true* - The service was successful.

- ■ *false* - The service was not able to create the ActiveTransfer Server scheduled event.

Usage Notes

None.

wm.mft.schedule:executeEvent

Executes scheduled events. This service is useful to test the execution of an action defined for a scheduled event. You can also parameterize the settings of an ActiveTransfer Server scheduled event action and use this service to provide the values of the parameters at runtime. Parameterizing the event action settings at runtime helps in reducing the number of events that you configure in ActiveTransfer Server especially when you transfer files across several source and destination file systems.

Input Parameters

scheduleName   **String** Specifies the name of the scheduled event.

allowEmptyValueForParams   **String** Specifies if the parameterized scheduled event action settings can be empty.

- `true` - Allows null values. If you do not provide a value in the parameterized action setting, the service will assign the value ("") to the setting and the event will execute successfully.

- `false` - Do not allow null values. The service will fail and a message is displayed indicating that the value for the parameterized setting is not provided.

eventParams   **Document List** The details of the parameterized scheduled event action settings. Variables are:

- *name* **String** The name assigned to the parameterized configuration setting of the scheduled event action.

- *value* **String** The value you want to assign to the parameterized field.

Output Parameters

output   **Document** Specifies if the scheduled event was successfully executed or not. Variables are:

- *success* **String** Takes the value `true` or `false`.

  - `true` - The scheduled event was successfully executed and all files were processed.

  - `false` - The scheduled event failed with one or more errors.

- *message* **String** A message that indicates the status of the scheduled event.

Usage Notes

■ For any remote file path you can parameterize only the URL and not the username and password. The runtime value for the URL should contain the username and password to be used. The URL information should be provided in format *<protocol>://<username>:<password>@<host>:<port>/<path>/*. For example, `FTP://user:password@ftp.softwareag.com/outbound/`

> **Note:**
> If you parameterize the file path URL and provide a value for the URL in the format *<protocol>://<username>:<password>@<host>:<port>/<path>/*, the values specified for the username and password parameters at runtime are ignored. This is applicable for the remote file URLs that you configure in the following actions:
>
> ■ Find action : Find URL
> ■ Copy action : Destination
> ■ Move action : Destination URL
> ■ Unzip action : Destination URL
> ■ Zip action : Zip File Path

■ You can use the wm.mft.schedule:createRemoteURL service to create the URL in the ActiveTransfer Server format.

■ You can parameterize only the following event action settings:

Action	Action Settings
Find	File URL, File Name
Copy	Destination URL, Rename file to, Source Filter, File Name
Decrypt	Decryption Key File, Source Filter
Send Email	From, To, CC, BCC, Subject, Body, Source Filter
Encrypt	Decryption Key File, Source Filter
Execute Script	Command, Arguments, Separator, Working Directory, Source Filter
Jump	variable, variable2, Source Filter
Move	Destination URL, Rename file to, Source Filter, File Name
Rename	New File Name, Source Filter
Unzip	Destination URL, Source Filter
Write Content	File Path, Source Filter
Zip	Zip File Path, File Name, Source Filter, Zip File Name

4 Security Folder

Overview

Use the security service (service in the wm.mft.security folder to generate PGP keys for encryption.

wm.mft.security.pgp:generatePGPKeyFiles

Generates an OpenPGP key pair (public and private key) that can be used to configure PGP in ActiveTransfer Server. The key pair is generated with the following properties:

Property	Description
keySize	2048
Supported Compression Algorithms	ZIP, ZLIP, Uncompressed
Supported Hashing Algorithms	SHA1, SHA256, SHA384, SHA512, MDS
Supported Symmetric Algorithms	CAST5, AES_128, AES_192, AES_256, TWOFISH
Key Common Name	commonName

Input Parameters

privateKey Path **String** Specifies the directory in the server where the key files are generated. The files are generated with the names "private.key" and "private.pub" in the directory.

validityDays **String** Optional. Specifies the number of days for which the keys will be valid. The default value is 365.

Password **String** Optional. Specifies the password for the private key.

Output Parameters

Document type **String** Contains "success = true" if the service runs successfully. Contains an error message if the service fails.

5 Server Services

Overview

Use these services in the wm.mft.server folder to list banned IP addresses and to unban IPs that are banned. Based on the hammering settings configured in the Server Management page, these services are applicable to either ActiveTransfer Server or ActiveTransfer Gateway.

Summary of Elements in Server Folder

The following elements are available in this folder:

Element	Description
wm.mft.server:listBannedIPs	Based on the host server (ActiveTransfer Server or ActiveTransfer Gateway), Integration Server lists all the IP addresses that are banned in ActiveTransfer Server or ActiveTransfer Gateway.
wm.mft.server:unbanIPs	Unbans the specified IP addresses in ActiveTransfer Server or ActiveTransfer Gateway.

wm.mft.server:listBannedIPs

Based on the host server (ActiveTransfer Server or ActiveTransfer Gateway), Integration Server lists all the IP addresses that are banned in the ActiveTransfer Server or ActiveTransfer Gateway.

Input Parameters

None.

Output Parameters

output   **Document** A document type that consists of the following:

- *ips* List of all the banned IP addresses.

- *error* The error message.

wm.mft.server:unbanIPs

Use this service to unban the IP addresses that are banned in ActiveTransfer Server or ActiveTransfer Gateway.

Input Parameters

ips   **String List** List of the IP addresses that you want to unban.

Output Parameters

output   **Document** A document type that consists of the following:

- *status* The status of the IP addresses that you provided to the service.

- *error* Error message.

6 Asset Document Types

Overview

The asset document types are available in the wm.mft.assets.rec folder. The document types are organized as follows:

- "Event Document Types" on page 70

- "Partner Document Type" on page 99

- "Port Document Types" on page 99

- "User Document Types" on page 104

- "VFS Document Types" on page 106

- "Service Response Document Type" on page 112

Event Document Types

ActiveTransfer Server provides document types which are referred to in the post-processing and scheduled event services. These document types are organized in the wm.mft.assets.rec.event folder as follows:

- Action

- Action Properties

- Post-processing Event

- Schedule Criteria

- Scheduled Event

- Schedule Data

- Monitor Folder Action

The following table lists the event document types in ActiveTransfer:

Document Type	Description
wm.mft.assets.rec.event.action:Action	Document type that specifies the details of an action defined in an ActiveTransfer Server post-processing or scheduled event.
wm.mft.assets.rec.event.action.properties:CopyFile	Document type that contains the details of the ActiveTransfer Server copy file action.

Document Type	Description
wm.mft.assets.rec.event.action.properties:DecryptFile	Document type that contains the details of the ActiveTransfer Server decrypt file action.
wm.mft.assets.rec.event.action.properties:DeleteFile	Document type that contains the details of the ActiveTransfer Server delete file action.
wm.mft.assets.rec.event.action.properties:EncryptFile	Document type that contains the details of the ActiveTransfer Server encrypt file action.
wm.mft.assets.rec.event.action.properties:ExcludeFile	Document type that contains the details of the ActiveTransfer Server exclude file action.
wm.mft.assets.rec.event.action.properties:ExecuteISService	Document type that contains the details of the execute Integration Server service action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:ExecuteScript	Document type that contains the details of the execute script action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:ExecuteTNService	Document type that contains the details of the execute Trading Networks service action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:FindFile	Document type that contains the details of the find file action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:Jump	Document type that contains the details of a jump action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:MoveFile	Document type that contains the details of the move file action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:RenameFile	Document type that contains the details of the rename action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:SendBrokerNotification (Deprecated)	Document type that contains the details of the send Universal Messaging or Broker notification action in ActiveTransfer Server.

Document Type	Description
wm.mft.assets.rec.event.action.properties:SendEmail	Document type that contains the details of the send email action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:UnzipFile	Document type that contains the details of the unzip file action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:WriteFileToDB	Document type that contains the write file content to database action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:WriteToFile	Document type that contains the details of the write content to file action in ActiveTransfer Server.
wm.mft.assets.rec.event.action.properties:ZipFile	Document type that contains the details of the zip file action in ActiveTransfer Server.
wm.mft.assets.rec.event:PostProcessEvent	Document type that contains the details of a ActiveTransfer Server post-processing event.
wm.mft.assets.rec.event:ScheduleCriteria	Document type that contains the schedule frequency and details for an ActiveTransfer Server scheduled event.
wm.mft.assets.rec.event:ScheduledEvent	Document type that contains the details of an ActiveTransfer Server scheduled event.
wm.mft.assets.rec.event.scheduleData	Document type that contains the details of the schedule for an ActiveTransfer Server scheduled event. This document type contains information on how often to execute an action.
"wm.mft.assets.rec.event:MonitorFolderAction" on page 97	Document type that contains the details of a ActiveTransfer Server monitor folder action.

wm.mft.assets.rec.event.action:Action

Document type that specifies the details of an action defined in an ActiveTransfer Server post-processing event, scheduled event or monitor folder action.

Variables

actionName	**String** The name of an action defined in the ActiveTransfer Server event.
sourceFilter	**String** (Optional) The name of the file whose transfer will trigger this event. By default, ActiveTransfer Server considers all files.

> **Note:**
> You can use wildcard characters to filter the file names. For example, enter `*.zip` to trigger the event only when zip files are uploaded or downloaded. To trigger an event based on a name string in the zip files, use the name string in the **Source Filter** box, preceded and followed by wildcard characters. For example, enter `*invoice*.zip` to trigger the event based on the file URLs, when zip files containing the character string `invoice` in their file names are uploaded or downloaded.

actionType	**String** The action type of the actions configured in the ActiveTransfer Server event. The actions could be one of the following:

- `FindFile` (default)

- `CopyFile`

- `DeleteFile`

- `DecryptFile`

- `EncryptFile`

- `MoveFile`

- `RenameFile`

- `UnzipFile`

- `WriteToFile`

- `ZipFile`

- `ExecuteISService`

- `ExecuteScript`

- `ExecuteTNService`

- `SendBrokerNotification`

- `SendEmail`

- `WriteFileToDB`

- `Jump`

■   `Exclude`

> **Note:**
> If you are defining a scheduled event, make sure the "Find File" action is the first action you define. Otherwise, the scheduled event will fail.

executeErrorAction   **String** (Optional) If you have defined an error action in the current action, specifies whether the error action should be executed.

■   `true` - (default) Execute the error action if an error occurs in the current action.

> **Note:**
> You can use any of the event actions that ActiveTransfer Server offers as the defined error action. ActiveTransfer Server executes an error action if any of the configured actions for a event fail.

■   `false` - Do not execute the error action.

executeAsynchronously   **String** (Optional) Specifies if the actions should be run in different threads for all the files. This is to ensure that the actions do not interfere with each other.

■   `true` - (default) Run the action in a different thread.

■   `false` - Run the action in parallel with the action instance working on another file.

isErrorAction   **String** Specifies if the action is an error action.

■   `false` - (default) The action is a regular action.

■   `true` - The action is an error action.

isDisabled   **String** Specifies if the action is disabled or not.

■   `false` - (default) The action is enabled.

■   `true` - The action is disabled.

properties   **Object** Specifies the properties of the action. The *properties* can take anyone of the following document types for action properties:

■   wm.mft.assets.rec.event.action.properties:FindFile

■   wm.mft.assets.rec.event.action.properties:CopyFile

■   wm.mft.assets.rec.event.action.properties:MoveFile

■   wm.mft.assets.rec.event.action.properties:DeleteFile

■   wm.mft.assets.rec.event.action.properties:RenameFile

- wm.mft.assets.rec.event.action.properties:EncryptFile

- wm.mft.assets.rec.event.action.properties:DecryptFile

- wm.mft.assets.rec.event.action.properties:Jump

- wm.mft.assets.rec.event.action.properties:ExcludeFile

- wm.mft.assets.rec.event.action.properties:ExecuteISService

- wm.mft.assets.rec.event.action.properties:ExecuteScript

- wm.mft.assets.rec.event.action.properties:ExecuteTNService

- wm.mft.assets.rec.event.action.properties:SendBrokerNotification

- wm.mft.assets.rec.event.action.properties:SendEmail

- wm.mft.assets.rec.event.action.properties:ZipFile

- wm.mft.assets.rec.event.action.properties:UnzipFile

- wm.mft.assets.rec.event.action.properties:WriteFileToDB

- wm.mft.assets.rec.event.action.properties:WriteToFile

wm.mft.assets.rec.event.action.properties:CopyFile

Document type that contains the details of the ActiveTransfer Server copy file action.

Variables

destination **Document** Specifies the location to copy the file to. Variable are:

- *destinationUrl* **String** Specifies the complete path of the destination folder to which the file is copied. The path could be a on the local machine or network, or could be on a remote machine. Example: FILE://c:/ProjectFolder/download/ or FILE:////<host>/SharedFolder/

  Note:
  Make sure that the operating system user running the ActiveTransfer Server instance has full access to the shared location. When you enter file path locations, be sure to end the path with a slash character ("/") to identify the location as a folder and not a file.

- *createDirectory* **Boolean** (Optional) Specifies if, when the folder path given in **Destination URL** does not exist, ActiveTransfer must create the directory.

- false - (default) ActiveTransfer does not create the missing directory.

        - true - ActiveTransfer creates the missing directory.

    - *isRemote* **String** Specifies if the destination folder is a remote path.

        - false - (default) The destination folder is in the same network.

        - true - The destination folder is a remote path.

    - *isVFS* **String** Specifies if the destination folder is a VFS.

        - false - (default) The destination folder is not a VFS.

        - true - The destination folder is a VFS.

    - *username* **String** (Optional) Specifies the username to access the remote system.

    - *password* **String** (Optional) Specifies the password to the remote system.

    Note:
    If both *isRemote* and *isVFS* are set to true, then ActiveTransfer Server does not create the copy file action.

renameFileTo	**String** (Optional) Specifies the file name to rename the file to.
waitTime	**String** (Optional) Specifies the number of seconds to wait before starting the copy operation. This is to ensure that an outside process is not writing to the file.
giveUpAfter	**String** (Optional) Stops the copy operation if it does not complete within the number of seconds specified in this parameter.
retryFactors	**Document** (Optional) Specifies the when ActiveTransfer Server should retry the copy file action.

- *retryWhenInterrupted* **String** Retry a failed copy operation for the specified number of times.

    - false (default)

    - true

- *retryAttempts* **String** Specifies the number of retries.

- *resumeFromPointofInterruption* **String** Specifies that an interrupted file copy action should resume from the point of interruption.

    - false (default)

    - true

- *retryInterval* **String** Specifies the time interval between retries.

ftp **Document** Specifies additional details for FTP, FTPS, FTPES protocols. Variables are:

- *asciiTransfer* **String** (Optional) Set to `true`, to change the file transfer mode to ASCII.

> **Note:**
> The default transfer mode is *binary*.

- *simpleMode* **String** (Optional) Set to `true`, to change the file transfer mode to simple mode. Use this option if you are transferring files to AS/400 systems.

- *convertLineEnding* **String** (Optional) Set to `true` for ActiveTransfer Server to change the line endings of the file.

fileExistsAtDestination **Document**(Optional) Checks if the file already exists at the destination location. Variables are:

- *skipTransferIfFileExists* **String** Set to `true`, to skip the file transfer if the file already exists at the destination location.

- *failActionIfFileExists* **String** Set to `true`, to fail the action if the file already exists at the destination location.

> **Note:**
> If both *skipTransferIfFileExists* and *failActionIfFileExists* variables are set to `true`, then *skipTransferIfFileExists* will be considered.

wm.mft.assets.rec.event.action.properties:DecryptFile

Document type that contains the details of the ActiveTransfer Server decrypt file action.

Variables

encryptionKeyFile **String** Specifies the name of the private key file (for example, xyz.pgp).

> **Note:**ActiveTransfer Server can decrypt the file only if the file was encrypted with the corresponding public key.

password **String** (Optional) Specifies the password for the encryption file.

isAsciiArmour **String** (Optional) Set this variable to `true`, if you want to Wrap PGP files in BASE64-encoded format to make them more secure when emailing them.

wm.mft.assets.rec.event.action.properties:DeleteFile

Document type that contains the details of the ActiveTransfer Server delete file action.

Variables

timeout   **String** (Optional) Specifies the time out details for the delete file action.

retryFactors   **Document** (Optional) Specifies the when ActiveTransfer Server should retry the delete file action.

- *retryWhenInterrupted* **String** Retry a failed delete operation for the specified number of times.

  - false (default)

  - true

- *retryAttempts* **String** Specifies the number of retries.

- *resumeFromPointofInterruption* **String** Specifies that an interrupted file delete action should resume from the point of interruption.

  - false (default)

  - true

- *retryInterval* **String** Specifies the time interval between retries.

wm.mft.assets.rec.event.action.properties:EncryptFile

Document type that contains the details of the ActiveTransfer Server encrypt file action.

Variables

encryptionKeyFile **String** Specifies the name of the public key file (for example, xyz.pgp).

isAsciiArmour   **String** (Optional) Set this variable to true, if you want to wrap the PGP files in BASE64-encoded format to make them more secure when emailing them.

wm.mft.assets.rec.event.action.properties:ExcludeFile

Document type that contains the details of the ActiveTransfer Server exclude file action.

Variables

None.

wm.mft.assets.rec.event.action.properties:ExecuteISService

Document type that contains the details of the execute Integration Server service action in ActiveTransfer Server.

Variables

serviceName	**String** Specifies the Integration Server service name. Includes the complete path of the service, starting from the package that contains the service you want to execute. For example, pub.event:addSubscriber
inputDataForService	**Object** Specifies the values for the input parameters of the service that you selected. For more information about the Integration Server services and their signatures, see *webMethods Integration Server Built-In Services Reference*.
serviceOutputMapping	**Document List** Specifies the values for the output parameters of the service that you selected. For more information about the Integration Server services and their signatures, see *webMethods Integration Server Built-In Services Reference*. Variables are:

- *paramName* **String** Name of the run-time variable used to store a output parameter of the Integration Server service.

- *outputPath* **String** The iData path to store the run-time variables used to store the output parameter of the Integration Server service.

includeFilePath	**String** (Optional) Set this parameter to true, if you want ActiveTransfer Server to provide the path of the target file to the respective service. The file path information is passed to the service as input parameter *filePath*.
includeFileContent	**String** (Optional) Set this parameter to true, to pass the file content to the service.
includeFileContentAs	**String** Specifies the transmission method for the file content. You can transmit the file content as:

- *stream* (default)

- *bytes*

wm.mft.assets.rec.event.action.properties:ExecuteScript

Document type that contains the details of the execute script action in ActiveTransfer Server.

Variables

command	**String** Specifies the command to be run by the action.
arguments	**String** (Optional) Specifies the arguments for the command. For example, {real_path}/archive/{name}: specifies that if the file is uploaded to /uploads/stuff.zip, it will be copied to /archive/stuff.zip.

separator   **String** (Optional) Specifies the separator character that you use to separate arguments.

workingDirectory   **String** Specifies the path to the directory where the command will execute. For example, when an application looks for a resource such as a configuration file, the application looks in the location specified here.

You should configure the variables of this document type depending on your operating system. One example each for the Windows and Unix/Linux platforms are listed below:

- **Windows Platform:** If you want to execute the batch file `C:\SAG\batchfiles\test.bat`, the properties that you need to specify for the Execute Script action are:

  command `C:\Windows\System32\cmd.exe`

  argument `/c;start;test.bat`

  separator `;`

  workingDirectory `C:\SAG\batchfiles\`

- **Unix/Linux Platforms:** You can directly specify the script file name. If you want to execute the batch file `/home/data/batchfiles/test.sh`, use the following settings in the Execute Script action.

  command `./test.sh`

  argument

  separator `;`

  workingDirectory `/home/data/batchfiles`

The above configuration settings can vary depending on the specific operating system that hosts your ActiveTransfer Server. In some of the operating systems, you might require an exit command at the end of the script file to properly terminate the command process.

wm.mft.assets.rec.event.action.properties:ExecuteTNService

Document type that contains the details of the execute Trading Networks service action in ActiveTransfer Server.

Variables

type   **String** (Optional) Specifies the document type:

- `XML` - (default) To execute the Trading Networks service wm.tn:receive to process XML document type.

- Flat File - To execute a particular Trading Networks service to process flat file document type.

- EDI - To execute the Trading Networks service wm.tn:receive to process EDI document type.

serviceName	**String** Specifies the service name only for Flat File document type. For EDI and XML, the *serviceName* is the value of wm.tn:receive by default.
inputDataForService	**Object** Specifies values for the input parameters of the service that you selected. You can add any content types as per the requirement. For more information about the Trading Networks services and their signatures, see *webMethods Trading Networks Built-In Services Reference*.

wm.mft.assets.rec.event.action.properties:FindFile

Document type that contains the details of the find file action in ActiveTransfer Server.

Variables

source	**Document** Specifies the path you want ActiveTransfer Server to search the file in. The variables are:

- *findUrl* **String** Specifies the complete path of the source folder to search the file in. The path could be on the local machine or network, or could be on a remote machine. Examples: FILE://c:/ProjectFolder/download/ or FILE:////<host>/SharedFolder/

  Note:
  Make sure that the OS user running the ActiveTransfer Server instance has full access to the shared location. When you enter file path locations, be sure to end the path with a slash character ("/") to identify the location as a folder and not a file.

- *isRemote* **String** Specifies if the source folder is a remote path.

  - false - (default) The source folder is in the same network.

  - true - The source folder is a remote path.

- *isVFS* **String** Specifies if the source folder is a VFS.

  - false - (default) The source folder is not a VFS.

  - true - The source folder is a VFS.

- *username* **String** (Optional) Specifies the username to access the remote system.

- *password* **String** (Optional) Specifies the password to the remote system.

> **Note:**
> If both *isRemote* and *isVFS* are set to true, then ActiveTransfer Server does not create the find file action.

folderDepth

String (Optional) Specifies the folder depth if you want to include subfolders in the search criteria for the find file action. The default value is 1 which restricts the search to the root folder.

maxItemToFind

String (Optional) Specifies the maximum number of items to find If you want to restrict the number of items in the find file action results. The default is 0 which includes all the items that match the search criteria for the find file action.

fileName

String (Optional) Specifies the name of the file to find. If you do not specify a file name, the find file action considers all files in the source location.

lastModification

Document (Optional) Specify the number of days, hours, and minutes, if you want to narrow the search by time period in which the file was last modified.

- *mode* The following modes are available:

  - Older than (default)

  - Earlier than

- *days* **String** (Optional) Number of days

- *hours* **String** (Optional) Number of hours

- *minutes* **String** (Optional) Number of minutes

failIfNoFiles

String Set this variable to true, if you want the find operation to fail if no files are found.

excludeChangingFiles

String (Optional) Set this parameter to true, if you want to remove files that are in use from the processing list.

- true (default)

- false

retryFactors

Document (Optional) Specifies when ActiveTransfer Server should retry the find file action.

- *retryWhenInterrupted* **String** Retry a failed find operation for the specified number of times.

  - false (default)

  - true

- *retryAttempts* **String** Specifies the number of retries.

■ *resumeFromPointofInterruption* **String** Specifies that an interrupted file find action should resume from the point of interruption.

- ■ false (default)

- ■ true

■ *retryInterval* **String** Specifies the time interval between retries.

sort   **Document** (Optional) Enables ActiveTransfer Server to find and sort files based on a particular order.

■ *sortFiles* **String** Enables or disables sorting of files.

- ■ false (default)

- ■ true

■ *sortField* **String** Sorts files based on last modified date, file size, and file name. The default is last modified date.

■ *sortOrder* **String** Sorts files based on ascending or descending order. The default is descending.

stability   **Document** (Optional) Specifies the file stability details. Set the following variables to true as per your requirement:

■ *delayTillNoChange* - If you want to delay processing of all files until no further file changes are made.

- ■ false (default)

- ■ true

■ *checkStabilityOnce* - If you want the find operation to scan and check the stability of the files one time only.

- ■ false (default)

- ■ true

■ *checkStabilityInterval* - If you want the find operation to check the stability of the files at regular intervals.

■ *checkStabilityTimeOut* - If you want the find operation to check the stability of the files at a certain time.

wm.mft.assets.rec.event.action.properties:Jump

Document type that contains the details of a jump action in ActiveTransfer Server.

Variables

jumpCondition   **Document** Specifies the condition based on which ActiveTransfer Server skips one or more actions and executes a designated action in the event. By default, a jump action is unconditional. Variables are:

- *variable1* **String** (Optional) Specifies the server variable in the jump condition. For a complete list of the server variables, see the "Server Variables" section in *Managing File Transfers with webMethods ActiveTransfer*.

- *condition* **String** Could be one of the following:

  - `Contains` (default) Includes items that contain a specified value.

  - `Does Not Contain` Excludes items that contain a specified value.

  - `Equals` Includes items that equal a specified value.

  - `Does Not Equal` Excludes items that equal a specified value.

  - `Matches Pattern` Uses pattern matching to include items that match a specified pattern.

  - `Does Not Match Pattern` Uses pattern matching to exclude items that match a specified pattern.

- *variable2* **String** (Optional) Specifies the value of the server variable in *variable1*.

jumpTo   **String** Specifies the action that ActiveTransfer Server will execute if the jump condition is met.

wm.mft.assets.rec.event.action.properties:MoveFile

Document type that contains the details of the move file action in ActiveTransfer Server.

Variables

destination   **Document** Specifies the location to move the file to. Variable are:

- *destinationUrl* **String** Specifies the complete path of the destination folder to move the file to. The path could be on the local machine or network, or could be on a remote machine. Examples: `FILE://c:/ProjectFolder/download/` or `FILE:////<host>/SharedFolder/`

  Note:
  Make sure that the OS user running the ActiveTransfer Server instance has full access to the shared location. When you enter file path locations, be sure to end the path with a slash character ("/") to identify the location as a folder and not a file.

- *createDirectory* **Boolean** (Optional) Specifies if, when the folder path given in **Destination URL** does not exist, ActiveTransfer must create the directory.

    - `false` - (default) ActiveTransfer does not create the missing directory.

    - `true` - ActiveTransfer creates the missing directory.

- *isRemote* **String** Specifies if the destination folder is a remote path.

    - `false` - (default) The destination folder is in the same network.

    - `true` - The destination folder is a remote path.

- *isVFS* **String** Specifies if the destination folder is a VFS.

    - *isRemote* **String** Specifies if the destination folder is a VFS.

        - `false` - (default) The destination folder is not a VFS.

        - `true` - The destination folder is a VFS.

- *username* **String** (Optional) Specifies the username to access the remote system.

- *password* **String** (Optional) Specifies the password to the remote system.

Note:
If both *isRemote* and *isVFS* are set to true, then ActiveTransfer Server does not create the move file action.

renameFileTo	**String** (Optional) Specifies the name to rename the file to.
waitTime	**String** (Optional) Specifies the number of seconds to wait before starting the move operation. This is to ensure that an outside process is not writing to the file.
giveUpAfter	**String** (Optional) Stops the move operation if it does not complete within the number of seconds specified in this parameter.
retryFactors	**Document** (Optional) Specifies when ActiveTransfer Server should retry the move file action.

- *retryWhenInterrupted* **String** Retry a failed move operation for the specified number of times.

    - `false` (default)

    - `true`

- *retryAttempts* **String** Specifies the number of retries.

- *resumeFromPointofInterruption* **String** Specifies that an interrupted file move action should resume from the point of interruption.

- - false (default)

  - true

  - *retryInterval* **String** Specifies the time interval between retries.

preserveFileDate   **String** (Optional) Set this option to true, if you want ActiveTransfer Server to retain the time stamp indicating when the file was last modified.

ftp   **Document** Specifies additional for FTP, FTPS, FTPES protocols. Variables are:

- *asciiTransfer* **String** (Optional) Set to true, to change the file transfer mode to ASCII.

  > **Note:**
  > The default transfer mode is *binary*.

- *simpleMode* **String** (Optional) Set to true, to change the file transfer mode to simple mode. Use this option if you are transferring files to AS/400 systems.

- *convertLineEnding* **String** (Optional) Set to true, for ActiveTransfer Server to change the line endings of the file.

  - None (default)

  - CRLF – Windows

  - CR – Mac OS Classic

  - LF – Unix

fileExistsAtDestination **Document**(Optional) Checks if the file already exists at the destination location. Variables are:

- *skipTransferIfFileExists* **String** Set to true, to skip the file transfer if the file already exists at the destination location.

- *failActionIfFileExists* **String** Set to true, to fail the action if the file already exists at the destination location.

> **Note:**
> If both *skipTransferIfFileExists* and *failActionIfFileExists* variables are set to true, then *skipTransferIfFileExists* will be considered.

wm.mft.assets.rec.event.action.properties:RenameFile

Document type that contains the details of the rename action in ActiveTransfer Server.

Variables

newFileName	**String** Specifies the new file name.
renameTimeOut	**String** (Optional) Specifies the time out duration for ActiveTransfer Server to rename the file.
retryFactors	**Document** (Optional) Specifies when ActiveTransfer Server should retry the rename file action.

- *retryWhenInterrupted* **String** Retry a failed rename operation for the specified number of times.

  - false (default)

  - true

- *retryAttempts* **String** Specifies the number of retries.

- *resumeFromPointofInterruption* **String** Specifies that an interrupted file rename action should resume from the point of interruption.

  - false (default)

  - true

- *retryInterval* **String** Specifies the time interval between retries.

skipSubItems	**String** (Optional) Set this parameter to true (default), if you want to rename a folder but not the files beneath the folder.

wm.mft.assets.rec.event.action.properties:SendBrokerNotification

Document type that contains the details of the send Universal Messaging or Broker notification action in ActiveTransfer Server.

Note:
This feature is deprecated.

Variables

docTypeName	**String** Specifies the Integration Server document type.
includeFilePath	**String** (Optional) Set this parameter to true if you want ActiveTransfer Server to provide the path of the target file to the respective service. The file path information is available as input parameter *filePath*.
includeFileContent	**String** (Optional) Set this parameter to true, if you want to populate the *fileContent* parameter.

inputData	**Object** Specifies the content for the document type that you selected. For more information about document types, Universal Messaging or Broker notifications, see the Universal Messaging, Broker, and Integration Server documentation.

wm.mft.assets.rec.event.action.properties:SendEmail

Document type that contains the details of the send email action in ActiveTransfer Server.

Variables

from	**String** (Optional) Specifies the email address of the sender. The value you specify here overrides the value specified in the mft.user.email.from server configuration parameter for this action.
to	**String** Specifies the email of the receiver.
cc	**String** (Optional) Specifies the email IDs to copy the email to.
bcc	**String** (Optional) Specifies the email IDs to blind copy the email to.
subject	**String** Specifies the text that will appear in the subject line of the email (for example, `Disconnect:?User %user_name%`). The value you specify here overrides the value specified in the mft.user.email.subject server configuration parameter for this action.
body	**String** Specifies the content that will appear in the body of the email. You can use server variables in the body of the email.

wm.mft.assets.rec.event.action.properties:UnzipFile

Document type that contains the details of the unzip file action in ActiveTransfer Server.

Variables

destination	**Document** Specifies the location to extract the zip file to. Variable are:

- *destinationUrl* Specifies the complete path of the destination folder to extract the zip file to. The path could be on the local machine or network, or could be on a remote machine. Examples: `FILE://c:/ProjectFolder/download/` or `FILE:////<host>/SharedFolder/`

> **Note:**
> Make sure that the OS user running the ActiveTransfer Server instance has full access to the shared location. When you enter file path locations, be sure to end the path with a slash character ("/") to identify the location as a folder and not a file.

- *isRemote* **String** Specifies if the destination folder is a remote path.

  - `true` - The destination folder is a remote path.

  - `false` - The destination folder is in the same network.

- *isVFS* **String** Specifies if the destination folder is a VFS.

  - *isRemote* **String** Specifies if the destination folder is a VFS.

    - `false` - (default) The destination folder is not a VFS.

    - `true` - The destination folder is a VFS.

- *username* **String** (Optional) Specifies the username to access the remote system.

- *password* **String** (Optional) Specifies the password to the remote system.

Note:
If both *isRemote* and *isVFS* are set to true, then ActiveTransfer Server does not create the unzip file action.

deleteOriginal **String** (Optional) Set this parameter to `true`, if you want to delete the original zip file after it is extracted to the *destination* folder.

wm.mft.assets.rec.event.action.properties:WriteFileToDB

Document type that contains the write file content to database action in ActiveTransfer Server.

Note:
This action delivers the contents of a file to an Integration Server service for the purpose of writing the content to the database. ActiveTransfer Server provides the content in byte or stream form to the service according to the format that the service's input signature requires.

Variables

serviceName **String** Specifies the Integration Server service name. Includes the complete path of the service, starting from the package that contains the service you want to execute. For example, pub.event:addSubscriber

inputDataForService **String** Specifies the values for the input parameters of the service that you selected. For more information about the Integration Server services and their signatures, see *webMethods Integration Server Built-In Services Reference*.

IncludeFileContent **String** (Optional) Set this parameter to `true`, to pass the file content to the service.

includeFileContent As **String** Specifies the transmission method for the file content. You can transmit the file content as:

- *stream* (default)

■   *bytes*

includeFilePath   **String** (Optional) Set this parameter to `true`, if you want ActiveTransfer Server to provide the path of the target file to the respective service. The file path information is passed to the service as input parameter *filePath*.

wm.mft.assets.rec.event.action.properties:WriteToFile

Document type that contains the details of the write content to file action in ActiveTransfer Server.

Variables

filePath   **String** Specifies the path containing the file to write to.

> **Note:**
> Be sure to end the path with a slash character ("/") to identify the location as a folder and not a file.

overwrite   **String** Set this parameter to `true` if the file already exists and you want to replace the entire contents of the existing file with the new content.

newFileContent   **String** Specifies the content to write to the new file if the file does not exist.

addBefore   **Document** Use this variable if you want to insert new content before the existing content in the file.

■   *beginningOfFile* - Set this variable to `true`, if you want to insert the content at the beginning of the file.

■   *addBeforeSearchText* - Set this variable to `true`, if you want to insert the content before a specific string of existing content in the file.

■   *addBeforeContent* - Set this variable to `true`, if you want to insert new content before existing content in the file.

addAfter   **Document** Use this variable if you want to insert new content after the existing content in the file.

■   *endOfFile* - Set this variable to `true`, if you want to insert the content at the end of the file.

■   *addAfterSearchText* - Set this variable to `true`, if you want to insert the content after a specific string of existing content in the file.

■   *addAfterContent* - Set this variable to `true`, if you want to insert new content after existing content in the file.

wm.mft.assets.rec.event.action.properties:ZipFile

Document type that contains the details of the zip file action in ActiveTransfer Server.

Variables

zipFilePath	**String** Specifies the path to which the file will be zipped to.
zipFileName	**String** (Optional) Specifies the zip file name. You can also provide a server variable name such as `{stem}.zip` for the zip file name.
createDirectory	**Boolean** (Optional) Specifies if, when the folder path given in **Zip File Path** does not exist, ActiveTransfer must create the directory.

- `false` - (default) ActiveTransfer does not create the missing directory.
- `true` - ActiveTransfer creates the missing directory.

isRemote	**String** Specifies if the source folder is a remote path.

- `false` - (default) The source folder is in the same network.
- `true` - The source folder is a remote path.

isVFS	**String** Specifies if the source folder is a VFS.

- `false` - (default) The source folder is not a VFS.
- `true` - The source folder is a VFS.

username	**String** (Optional) Specifies the username to access the remote system.
password	**String** (Optional) Specifies the password to the remote system.

Note:
If both *isRemote* and *isVFS* are set to true, then ActiveTransfer Server does not create the zip file action.

wm.mft.assets.rec.event:PostProcessEvent

Document type that contains the details of a ActiveTransfer Server post-processing event.

Variables

id	**String** (Optional) The internal identifier for the ActiveTransfer Server post-processing event.
name	**String** Name of the ActiveTransfer Server post-processing event.
description	**String** (Optional) The description of the ActiveTransfer Server post-processing event.
isActive	**String** (Optional) The status of the post-processing event in ActiveTransfer Server.

- `true` - The post-processing event is active.

- `false` - (default) The post-processing event is not active.

criteria   **Document** Specifies the file transfer criteria for the post-processing event. Variables are:

- *action* **String** (Optional) Specifies the action that will trigger the post-processing event:

  - `Uploads` (default) User uploads a file.

  - `Downloads` User downloads a file.

  - `Deletes` User deletes a file.

    Note:
    If you specify an event based on the deletion of a file, make sure that any subsequent actions you define for the event do not rely on the presence of the deleted file.

- *folderName* **String** (Optional) Specifies the file folder. You can specify one of the following:

  - Local file path

  - Remote file path

  - Virtual folder in a VFS

- *transferStatus* **String** (Optional) Specifies the status of the file transfer that should trigger the post-processing event:

  - `All` (default)

  - `Successful`

  - `NotSuccessful`

- *users* **String List** (Optional) The list of users who will carry out the file transfer operation.

- *executionMode* **String** (Optional) Specifies when to run the post-processing.

  - `immediate` (default) Run the post-processing event immediately

  - `userexit` Run the post-processing event after the user exits all sessions.

  - `useridle` Run the post-processing event after the user is idle for some seconds.

- *idlePeriodForAction* **String** (Optional) Specifies the idle period for the action.

actionList   **Document Reference List** Refers to the wm.mft.assets.rec.event.action:Action document type and contains the list of actions and the details of the actions in the post-processing event.

wm.mft.assets.rec.event:ScheduleCriteria

Document type that contains the schedule frequency and details for an ActiveTransfer Server scheduled event.

Variables

scheduleFrequency **String** (Optional) Specifies the frequency with which ActiveTransfer Server should run the schedule event configured. The schedule frequency options are as follows:

- `Daily` (default)

- `Hourly`

- `Run Once`

- `Monthly`

- `Manual`

- `Fixed Interval`

- `Yearly`

- `Weekly`

scheduleDetails **Object** Contains one of the document types in the wm.mft.assets.rec.event.scheduleData folder and is dependent on the *scheduleFrequency* selected for the scheduled event. The object can take one of the following values:

- daily

- hourly

- runOnce

- monthly

- manual

- fixedInterval

- yearly

- weekly

wm.mft.assets.rec.event:ScheduledEvent

Document type that contains the details of an ActiveTransfer Server scheduled event.

variables

id	**String** The internal identifier for the ActiveTransfer Server scheduled event.
name	**String** Name of the ActiveTransfer Server scheduled event.
description	**String** (Optional) The description of the ActiveTransfer Server scheduled event.
isActive	**String** (Optional) The status of the scheduled event in ActiveTransfer Server.

- `true` - The scheduled event is active.

- `false` - (default) The scheduled event is not active.

criteria	**Document Reference** (Optional) Refers to the wm.mft.assets.rec.event:ScheduleCriteria document type and specifies the scheduled event criteria.
actionList	**Document Reference List** Refers to the wm.mft.assets.rec.event.action:Action document type and contains the list of actions and the details of the actions in the scheduled event.

wm.mft.assets.rec.event.scheduleData

Document type that contains the details of the schedule for an ActiveTransfer Server scheduled event. This document type contains information on how often to execute an action.

Variables

daily	Specifies a date range and the times you want to execute the action each day. Variables are:

- *scheduleHours* - Specifies the hour portion of the time to execute an action.

- *scheduleMinutes* - Specifies the minute portion of the time to execute an action.

- *doNotOverlap* - Set this variable to `true` if you want ActiveTransfer Server to complete a running action before starting the next one.

  > **Note:**
  > Setting this variable to `true` might cause actions to start at other than specified times.

- *startDate* - Specifies the start date for the schedule.

- *startTime* - Specifies the start time for the schedule.

■ *endDate* - Specifies the end date of the schedule.

■ *endTime* - Specifies the end time for the schedule.

Note:
The default date and time formats are set in My webMethods.

fixedInterval   Specifies the time interval that ActiveTransfer Server should wait (for example, 10 seconds) before executing the next action for a scheduled event. Variables are:

■ *intervalType* - Select one of the following:

  ■ Seconds

  ■ Minutes

  ■ Hours

  ■ Days

  ■ Weeks

■ *intervalValue* - Specifies the number of seconds, minutes, hours, weeks, or days that ActiveTransfer Server should wait before executing the next action in a scheduled event.

■ *doNotOverlap* - Set this variable to true if you want ActiveTransfer Server to complete a running action before starting the next one.

Note:
Setting this variable to true might cause actions to start at other than specified times.

■ *startDate* - Specifies the start date for the schedule.

■ *startTime* - Specifies the start time for the schedule.

■ *endDate* - Specifies the end date of the schedule.

■ *endTime* - Specifies the end time for the schedule.

Note:
The default date and time formats are set in My webMethods.

hourly   Specifies a date range and the times you want to execute the action each hour. Variables are:

■ *scheduleMinutes* - Specifies the minutes in the hour at which the action is to be executed.

■ *doNotOverlap* - Set this variable to true if you want ActiveTransfer Server to complete a running action before starting the next one.

> **Note:**
> Setting this variable to `true` might cause actions to start at other than specified times.

- *startDate* - Specifies the start date for the schedule.

- *startTime* - Specifies the start time for the schedule.

- *endDate* - Specifies the end date of the schedule.

- *endTime* - Specifies the end time for the schedule.

> **Note:**
> The default date and time formats are set in My webMethods.

manual   Specifies that you want to use the wm.mft.schedule:executeEvent service to manually execute the actions defined for this event.

monthly   Specify a date range, the days within the month, and the times you want to execute the action each month. Variables are:

- *scheduleWeek* - Specifies the weeks in which to execute a monthly action.

- *scheduleDays* - Specifies the days of the week (for example, "second Tuesday of the month") to execute a monthly action.

- *scheduleHours* - Specifies the hour portion of the time to execute an action.

- *scheduleMinutes* - Specifies the minute portion of the time to execute an action.

- *doNotOverlap* - Set this variable to `true`, if you want ActiveTransfer Server to complete a running action before starting the next one.

> **Note:**
> Setting this variable to `true` might cause actions to start at other than specified times.

- *startDate* - Specifies the start date for the schedule.

- *startTime* - Specifies the start time for the schedule.

- *endDate* - Specifies the end date of the schedule.

- *endTime* - Specifies the end time for the schedule.

runOnce   Specifies the date and time to execute the action:

- *scheduleDate* - Specifies the date to execute the action.

- *scheduleTime* - Specifies the time to execute the action.

weekly   Specifies a date range, the days of the week, and the times you want to execute the action each week:

■ *scheduleDays* - Specifies the days of the week to execute the weekly action.

■ *scheduleHours* - Specifies the hour portion of the time to execute an action.

■ *scheduleMinutes* - Specifies the minute portion of the time to execute an action.

■ *doNotOverlap* - Set this variable to `true` (default), if you want ActiveTransfer Server to complete a running action before starting the next one.

> **Note:**
> Setting this variable to `true` might cause actions to start at other than specified times.

■ *startDate* - Specifies the start date for the schedule.

■ *startTime* - Specifies the start time for the schedule.

■ *endDate* - Specifies the end date of the schedule.

■ *endTime* - Specifies the end time for the schedule.

yearly   Specifies a date range, the months, the days within the month, and the times you want to execute the action each year. Variables are:

■ *scheduleMonths* - Specifies the month to execute a yearly action.

■ *scheduleWeek* - Specifies the weeks in a month to execute a yearly action.

■ *scheduleDays* - Specifies the days in a week to execute a yearly action.

■ *scheduleHours* - Specifies the hour portion of the time to execute an action.

■ *scheduleMinutes* - Specifies the minute portion of the time to execute an action.

■ *doNotOverlap* - Set this variable to `true`, if you want ActiveTransfer Server to complete a running action before starting the next one.

> **Note:**
> Setting this variable to `true` might cause actions to start at other than specified times.

■ *startDate* - Specifies the start date for the schedule.

■ *startTime* - Specifies the start time for the schedule.

■ *endDate* - Specifies the end date of the schedule.

■ *endTime* - Specifies the end time for the schedule.

wm.mft.assets.rec.event:MonitorFolderAction

Document type that contains the details of a ActiveTransfer Server monitor folder action.

Variables

id	**String** (Optional) The internal identifier for the ActiveTransfer Server monitor folder action.
name	**String** Name of the ActiveTransfer Server monitor folder action.
description	**String** (Optional) The description of the ActiveTransfer Server monitor folder action.
isActive	**String** (Optional) The status of the monitor folder action in ActiveTransfer Server.

- `true` - The monitor folder action is active.

- `false` - (default) The monitor folder action is not active.

criteria	**Document** Specifies the file monitoring criteria for the monitor folder action. Variables are:

- *fileOperation* **String** Specifies the action that will trigger the monitor folder action:

  - `CREATE`: ActiveTransfer Server will trigger the monitor folder action on creating a file in the monitored folder.

  - `DELETE`: ActiveTransfer Server will trigger the monitor folder action on deleting a file in the monitored folder.

- *fileNameFilter* **String** (Optional) The name of the file whose transfer will trigger this action. By default, ActiveTransfer Server considers all the files.

  > **Note:**
  > You can use wildcard characters to filter the file names. For example, enter `*.zip` to trigger the action only when zip files are created or deleted.

- *monitorFolder* **String**

  Specifies the folder to be monitored by the monitor folder action.

- *completionFolder* **String** (Optional)

  Specifies the folder to move the file after the successful execution of an action.

  .

- *errorFolder* **String** (Optional)

  Specifies the folder to move the file after the execution fails for an action.

- *watchFrequency* **String**

Specifies the time interval in seconds to monitor the folder.

- *retentionPeriod* **String**

  Specifies the number of days you want to retain the files in Completion folder or Error folder

actionList **Document Reference List** Refers to the wm.mft.assets.rec.event.action:Action document type and contains the list of actions and the details of the actions in the monitor folder action.

Partner Document Type

ActiveTransfer Server provides a document type which is referred to in the partner services.

wm.mft.assets.rec.partner:Partner

Document type that contains the details of an ActiveTransfer Server partner.

Variables

id **String** (Optional) The internal identifier for a partner created in ActiveTransfer Server. If you are using the trading partner information from Trading Networks, *id* is the Trading Networks identifier for the enterprise or partner profile.

name **String** The partner name in ActiveTransfer Server.

isTNPartner **String** If the partner is a Trading Networks partner.

- `false` - (default) The partner is a partner created in ActiveTransfer Server.

- `true` - The partner is a Trading Networks partner.

isEnterprise **String** If the partner is an enterprise.

- `false` - (default) The partner is not an enterprise.

- `true` - The partner is an enterprise.

Port Document Types

ActiveTransfer Server provides document types which are referred to in the port services. The following table lists the port document types:

Document Type	Description
wm.mft.assets.rec.port:FTPDetails	Document type that contains the details of a FTP port configured in ActiveTransfer Server.

Document Type	Description
wm.mft.assets.rec.port:HTTPDetails	Document type that contains the details of a HTTP port configured in ActiveTransfer Server.
wm.mft.assets.rec.port:PortInfo	Document type that contains the complete details of a port configured in ActiveTransfer Server.
wm.mft.assets.rec.port:SFTPDetails	Document type that contains the details of a SFTP port configured in ActiveTransfer Server.
wm.mft.assets.rec.port:SSLDetails	Document type that contains the global SSL settings for ActiveTransfer Server ports.

wm.mft.assets.rec.port:FTPDetails

Document type that contains the details of a FTP port configured in ActiveTransfer Server.

Variables

loginMessage **String** (Optional) Modifies the default login message displayed in the FTP client console when a user logs in to the ActiveTransfer Server.

commandDelayInterval **String** Adds a pause between each command to slow down clients that continually access the server. The command delay interval is specified in milliseconds.

requireEncryption **String** (Optional) Indicates if an encryption method is set for the FTP port. Default value is `true`.

implicitSSL **String** Use implicit SSL as the encryption mode. If this parameter is set to `true` (default), SSL is used on all the clients in each session.

explicitSSL **String** (Optional) Use SSLv3 in the explicit SSL encryption mode. Default value is `true`.

> **Note:**
> The explicit mode requires the client to use the data transfer encryption mode while connecting to the FTP server. In this mode, the client has the option to switch off the channel encryption.

explicitTLS **String** (Optional) Use TLSv1 in the explicit SSL encryption mode. Default is `true`.

ftpRouterAware **String** (Optional) Specifies if your router or firewall is FTP aware. Default is `true`.

> **Note:**

> FTP-aware routers and firewalls will inspect the FTP command and response, and might modify the response. Check your firewall configuration before selecting this option.

passiveIP	**String** (Optional) The external IP address of the FTP server. You can specify the value Auto, if you want ActiveTransfer Server to automatically identify the external IP address of the server, or type the IP address in, manually.

> **Note:**
> If you are using a passive IP configuration, be sure to provide proper access for the port(s) in your firewall settings. Otherwise, connections between the client machine and ActiveTransfer Server might be blocked.

passivePortFrom	**String** (Optional) Specifies the starting number of the range of port numbers that can be used for passive port connections.
passivePortTo	**String** (Optional) Specifies the last number of the range of port numbers that can be used for passive port connections
ftpsDetails	**Document Reference** A document reference to the wm.mft.assets.rec.port:SSLDetails document type which contains the SSL details for the port.

wm.mft.assets.rec.port:HTTPDetails

Document type that contains the details of a HTTP port configured in ActiveTransfer Server.

Variables

commandDelayInterval	**String** Adds a pause between each command to slow down clients that continually access the server. The command delay interval is specified in milliseconds.
httpsDetails	**Document Reference** A document reference to the wm.mft.assets.rec.port:SSLDetails which contains the SSL details for the HTTP port.
supportSAMLSSO	**Boolean** To enable single sign-on. You can set one of the following values:

- true: To enable single sign-on.

- false: To disable single sign-on. This is the default.

wm.mft.assets.rec.port:PortInfo

Document type that contains the complete details of a port configured in ActiveTransfer Server.

Variables

portId	**String** The internal identifier for the ActiveTransfer Server port.
portName	**String** The name of the ActiveTransfer Server port.
host	**String** The IP address of the machine on which the ActiveTransfer Server is installed.
port	**String** Port number assigned to the port.

> **Note:**
> Make sure that the port you specify is not being used by any application, including the default ports used for ActiveTransfer Server and ActiveTransfer Gateway (2080 and 8500, respectively).

protocol	**String** The transfer protocol to be used by the port to transfer files. The options are:

- FTP (default)

- SFTP

- HTTP

- HTTPS

instanceId	**String** (Optional) The internal identifier for the ActiveTransfer Server instance.
instanceName	**String** (Optional) Name of the ActiveTransfer Server instance.
active	**Boolean** Port is activated and started in all the ActiveTransfer instances. The default is true.
defaultInEmail	**String** (Optional) If the port information should be included by default in the user credential email sent to the ActiveTransfer Server user. ActiveTransfer Server sends an email alert to the user when you create a new user account or edit the credentials or server connection details for a user. Using this parameter, you can specify to include the port name, protocol, and host, and port information in these alert emails. The default is true.
ftpDetails	**Document Reference** (Optional) A document reference to the wm.mft.assets.rec.port:FTPDetails document type which contains the details of an ActiveTransfer Server port configured for FTP protocol.

> **Note:**
> Use this document type only if the ActiveTransfer Server port is configured for FTP file transfer protocol.

sftpDetails	**Document Reference** (Optional) A document reference to the wm.mft.assets.rec.port:SFTPDetails document type which contains the details of an ActiveTransfer Server port configured for SFTP protocol.

> **Note:**
> Use this document type only if the ActiveTransfer Server port is configured for SFTP file transfer protocol.

httpDetails **Document Reference** (Optional) A document reference to the wm.mft.assets.rec.port:HTTPDetails document type which contains the details of an ActiveTransfer Server port configured for HTTP protocol.

> **Note:**
> Use this document type only if the ActiveTransfer Server port is configured for HTTP file transfer protocol.

wm.mft.assets.rec.port:SFTPDetails

Document type that contains the details of a SFTP port configured in ActiveTransfer Server.

variables

commandDelayInterval **String** Adds a pause between each command to slow down clients that continually access the server. The command delay interval is specified in milliseconds.

enableDSA **String** (Optional) Specifies if the DSA encryption algorithm is to be enabled. Default value is `true`.

DSAKeyLocation **String** (Optional) The full path of the file that contains the key for the DSA algorithm.

DSAPassPhrase **String** (Optional) The password for the DSA key.

enableRSA **String** (Optional) Specifies if the RSA encryption algorithm is to be enabled. Default value is `true`.

RSAKeyLocation **String** (Optional) The full path of the file that contains the key for the RSA algorithm.

RSAPassPhrase **String** (Optional) The password for the RSA key.

mac **String** (Optional) To configure the HMACs for SFTP listeners.

ciphers **String** Specifies the SSH ciphers supported for encrypting or decrypting data.

requirePassword **String** (Optional) Set this parameter to `true`, if you want to make the password mandatory when the certificate handshake is passed.

requirePublicKey **String** (Optional) Set this parameter to `true`, if you want to make the public key mandatory when the certificate handshake is passed. If the public key is not provided for authentication, the user is authenticated using the password.

idleTimeout	**String** (Optional) The timeout value in seconds after which ActiveTransfer Server will disconnect an idle connection.
synchronous	**String** (Optional) Set this variable to `true`, if the SFTP port should use synchronous threads for communication.
textEncoding	**String** (Optional) The text encoding format to work with. Most clients use the `UTF-8` encoding.

wm.mft.assets.rec.port:SSLDetails

Document type that contains the global SSL settings for ActiveTransfer Server ports.

Variables

requireClientCert	**String** (Optional) If you want to block all connections from the client when the client does not have a valid client certificate key password. Default is `false`.

> **Note:**
> If this parameter is set to `true`, ActiveTransfer Server expects the clients requesting a server connection to present a valid certificate. The certificate should match one of the certificates stored in the truststore. To store valid certificates, you must create a truststore file in the same location as the keystore file, with the name keystoreName _trust. For example, if the keystore file name is server_ks.jks, the truststore name should be server_ks.jks_trust. You should add all of the valid client certificates to this truststore.

keystoreLocation	**String** (Optional) Specifies the path to the keystore file. For example, "C:\keystore" on Windows and "/usr/keystore" on UNIX.
keystorePassword	**String** (Optional) Specifies the keystore password.
keyPassword	**String** (Optional) Specifies the private key password.

User Document Types

ActiveTransfer Server provides document types which are referred to in the user and user template services. The following table lists the user document types:

Document Type	Description
wm.mft.assets.rec.user:User	Document type that contains the complete details of an ActiveTransfer Server user.

Document Type	Description
wm.mft.assets.rec.user:UserInfo	Document type that contains information for an ActiveTransfer Server user.
wm.mft.assets.rec.user:UserTemplate	Document type that contains the ActiveTransfer Server user template details.

wm.mft.assets.rec.user:User

Document type that contains the complete details of an ActiveTransfer Server user.

Variables

userId	**String** The internal identifier for an ActiveTransfer Server user.
disableLogin	**String** (Optional) Set this parameter to `true`, if you want to disable the user's ID and prevent that user from logging in to the server.
templateId	**String** (Optional) The internal identifier for the ActiveTransfer Server template to be used for this user. ActiveTransfer Server uses the default template if no information is provided.
partnerId	**String** The internal identifier of the ActiveTransfer Server partner associated with the user.
userDetails	**Document** Specifies the complete details of an ActiveTransfer Server user. Variables are:

- firstName

- lastName

- emailAddress

- password

wm.mft.assets.rec.user:UserInfo

Document type that contains information for an ActiveTransfer Server user.

Variables

userId	**String** The internal identifier for an ActiveTransfer Server user.
disableLogin	**String** (Optional) Set this parameter to `true`, if you want to disable the user's ID and prevent that user from logging in to the server.

templateId	**String** (Optional) The internal identifier for the ActiveTransfer Server template to be used for this user. ActiveTransfer Server uses the default template if no information is provided.
partnerId	**String** The internal identifier of the ActiveTransfer Server partner associated with the user.

wm.mft.assets.rec.user:UserTemplate

Document type that contains the ActiveTransfer Server user template details.

Variables

id	**String** (Optional) The internal identifier of an ActiveTransfer Server user template.
name	**String** Name of the user template.
description	**String** (Optional) Description of the ActiveTransfer Server user template.
defaultTemplate	**String** (Optional) Set this variable to `true`, if the template configured is the default template for a ActiveTransfer Server user.

VFS Document Types

ActiveTransfer Server provides document types which are referred to in the VFS services. The following table lists the VFS document types:

Document Type	Description
wm.mft.assets.rec.vfs:FTP	Document type that is used to set the passive FTP mode in ActiveTransfer Server.
wm.mft.assets.rec.vfs:LocalPath	Document type that contains the path in the local file system for a virtual folder in ActiveTransfer Server.
wm.mft.assets.rec.vfs:RemoteVfs	Document type that contains the path in the remote file system for a virtual folder in ActiveTransfer Server.
wm.mft.assets.rec.vfs:SFTP	Document type that contains the SFTP settings for a virtual folder in ActiveTransfer Server.
wm.mft.assets.rec.vfs:SSL	Document type that contains the SSL settings for the virtual folder in ActiveTransfer Server. This document type applies to the FTPES, FTPS, and HTTPS file transfer protocols.
wm.mft.assets.rec.vfs:VfsDetails	Document type that contains the details of a virtual folder in ActiveTransfer Server.

Document Type	Description
wm.mft.assets.rec.vfs:VfsInfo	Document type that contains the complete structure of a virtual folder created in ActiveTransfer Server.
wm.mft.assets.rec.vfs:VfsUser	Document type that contains the details of a user associated with the virtual folder in ActiveTransfer Server.

wm.mft.assets.rec.vfs:FTP

Document type that is used to set the passive FTP mode in ActiveTransfer Server.

Variables

passive   **String** Use this variable to specify if you want to connect to a remote server using passive FTP mode. The default is `true`.

wm.mft.assets.rec.vfs:LocalPath

Document type that contains the details of the virtual folder in ActiveTransfer Server, which points to a local file system.

Variables

filePath   **String** Specifies the file path in the local file system which is associated with the virtual folder.

wm.mft.assets.rec.vfs:RemoteVfs

Document type that contains the details of the virtual folder in ActiveTransfer Server, which points to a remote server.

Variables

url   **String** Specifies the URL path to the remote file system associated with the virtual fold Server. The URL is specified in the following format: *<protocol>://<localhost|machine_name|IP|ATG_machine_name|ATG_IP>:<port>/VFS_nam*

user   **String** The username to access the remote file system.

password   **String** The password to access the remote file system.

hADownloadRecovery   **String** Set this variable to `true`, if you want ActiveTransfer Server to recover from a d not completed.

hAUploadRecovery	**String** Set this variable to `true`, if you want ActiveTransfer Server to recover from a upload th completed.

> **Note:**
> The High Availability Upload/Download Recovery options provide recovery from short dro connections (30 to 60 seconds) during file transfers. ActiveTransfer Server maintains an inter for the transfer-in-progress which is used for automatic retries to re-connect to the external s connection goes down. And, when the connection is re-established, ActiveTransfer Server re file transfer from the point of failure. This automatic resume is triggered by ActiveTransfer Serv the intervention of the requestor ensuring uninterrupted file transfer.

ftp	**Document Reference** A document reference to the wm.mft.assets.rec.vfs:FTP document type wh the FTP settings for a virtual folder.

> **Note:**
> Use this variable only if the virtual folder points to a remote FTP or FTPS server.

ssl	**Document Reference** A document reference to the wm.mft.assets.rec.vfs:SSL document type wh the SSL settings for a virtual folder.

> **Note:**
> Use this variable only if the virtual folder points to a remote FTPS (FTP over SSL) or HTTPS (SSL) server.

sftp	**Document Reference** A document reference to the wm.mft.assets.rec.vfs:SFTP document type wh the SFTP settings for a virtual folder.

> **Note:**
> Use this variable only if the virtual folder points to a remote SFTP server.

wm.mft.assets.rec.vfs:SFTP

Document type that contains the SFTP settings for a virtual folder in ActiveTransfer Server.

Variables

privateKeyPath	**String** Specifies the path to the private key file set for the virtual folder in ActiveTransfer Server.
privateKeyPassword	**String** Specifies the private key password to access the private key file that you have set for the virtual folder in ActiveTransfer Server.
enableTwoFactorAuth	**String** (Optional) Specifies if the authentication should be done using both the private key and password. Default is `false`.

wm.mft.assets.rec.vfs:SSL

Document type that contains the SSL settings for the virtual folder in ActiveTransfer Server. This document type applies to the FTPES, FTPS, and HTTPS file transfer protocols.

Variables

keystorePath	**String** Specifies the path to the keystore set for the virtual folder in ActiveTransfer Server.
keystorePassword	**String** Specifies the password to access the keystore set for the virtual folder in ActiveTransfer Server.
keyPassword	**String** Specifies the key password for the SSL certificate associated with the virtual folder in ActiveTransfer Server.

wm.mft.assets.rec.vfs:VfsDetails

Document type that contains the details of a virtual folder in ActiveTransfer Server.

Variables

vfsId	**String** The internal identifier of the virtual folder in ActiveTransfer Server.
vfsName	**String** The name of the virtual folder in ActiveTransfer Server.
partnerId	**String** The internal identifier for a partner in ActiveTransfer Server.
vfsType	**String** Specifies the folder type.

- `Virtual` (default) The folder is a virtual folder.

- `Local` The folder is in the local system.

- `Remote` The folder is linked to a remote location.

vfsLocation	**Document** (Optional) Specifies the details of the physical location of the virtual folder. The physical location could be a:

- *localPath* Virtual folder path in a local file system specified by wm.mft.assets.rec.vfs:LocalPath.

- *remoteServerDetails* Virtual folder path in a remote file system specified by wm.mft.assets.rec.vfs:RemoteVfs.

> **Note:**
> You can also have virtual folders with no physical location which are used to represent a collection of physical folders and files located on one or more local or remote file systems.

users	**Document Reference** A document reference to the wm.mft.assets.rec.vfs:VfsUser document type which specifies the details of a user associated with a virtual folder.

wm.mft.assets.rec.vfs:VfsInfo

Document type that contains the complete structure of a virtual folder created in ActiveTransfer Server. The document type refers to itself recursively as the parameter, *childVfsList* which refers to subfolders that can be created under a root virtual folder.

Variables

vfsId	**String** The internal identifier of the virtual folder in ActiveTransfer Server.
vfsName	**String** The name of the virtual folder in ActiveTransfer Server.
virtualFolder	**String** Indicates if this folder has a physical location or not. You can also have virtual folders with no physical location which are used to represent a collection of physical folders and files located on one or more local or remote file systems.

- true (default)

- false

partnerId	**String** The internal identifier of the ActiveTransfer Server partner associated with the user.
partnerName	**String** The name of the ActiveTransfer Server partner associated with the user.
users	**String List** The list of all the user associated with the virtual folder.
childVfsList	**Document Reference List** A document reference to the wm.mft.assets.rec.vfs:VfsInfo document type which contains the details of all the child virtual folder under this virtual folder.

> **Note:**
> You cannot add a virtual folder beneath a virtual folder that is associated with a remote physical location.

wm.mft.assets.rec.vfs:VfsUser

Document type that contains the details of a user associated with the virtual folder in ActiveTransfer Server.

Variables

userId	**String** Internal identifier for the user associated with a ActiveTransfer Server virtual folder.

quotaLimit	**String** (Optional) Specifies the total amount of space, in megabytes to make available for file transfers in a specific virtual folder for the user identified by *userId*.
inheritPermissions	**String** Specifies if the user should inherit access privileges from the parent folder or not.

- false - (default) Do not inherit the user access privileges from the parent folder.

- true - Inherit the user access privileges set in the parent folder.

> **Note:**
> If this variable is set to true, ActiveTransfer Server will not consider any of the user access permissions explicitly set for the folder.

traverseFolder	**String** Specifies if the user has only folder traversal permission for a folder. Folder traversal permission lets the user navigate from the root folder to the subfolder to which the user has access privileges explicitly assigned.

- false (default)

- true

> **Note:**
> The value provided for this variable is ignored if *inheritPermssions* is true.

permissions	**Document** (Optional) Specifies the user access privileges for the user identified by *userId* in the ActiveTransfer Server virtual folder. Variables are:

- *view*

- *download*

- *upload*

- *delete*

- *createDirectory*

- *deleteDirectory*

- *rename*

- *resumeTransfer*

- *sharePublish*

- *slideShow*

Set the variables to true based on the access privileges assigned to the user in a virtual folder.

> **Note:**
> The access privileges set using this variable are ignored if *inheritPermssions* is `true` or *traverseFolder* is `true`.

Service Response Document Type

wm.mft.assets.rec:svcResponse

Document type that contains the response from the services under wm.mft.assets.

Variables

info	**Document List** Contains the information messages. The list contains one entry for each message. The variables are:

- *key* **String** The message key.

- *message* **String** The information message.

warning	**Document List** Contains the warning messages. The list contains one entry for each warning. The variables are:

- *key* **String** The message key.

- *message* **String** The warning message.

- *stack* **String** The stack trace of the exception.

error	**Document List** Contains the error messages. The list contains one entry for each error. The variables are:

- *key* **String** The error key.

- *message* **String** The error message.

- *stack* **String** The stack trace of the exception.