

MailStore Server Administration

API Commands

AttachStore

Attaches an archive store that has previously been detached.

```
AttachStore --name [--type] [--databaseName] [--databasePath] [--contentPath]
            [--indexPath] [--serverName] [--userName] [--password] [--requestedState]
```

Arguments

name

The name of the archive store to be attached. This does not necessarily have to match the name that the archive store originally had before detaching.

type (*optional*)

databaseName (*optional*)

databasePath (*optional*)

contentPath (*optional*)

indexPath (*optional*)

serverName (*optional*)

userName (*optional*)

password (*optional*)

For more information about these arguments, please refer to the documentation of the *CreateStore* method.

requestedState (*optional*)

The requested state to be set. The default value when attaching stores is *normal*. For a list of possible values, please refer to the documentation of the *CreateStore* method.

ClearUserPrivilegesOnFolders

Removes all privileges that a user has on archive folders.

```
ClearUserPrivilegesOnFolders --userName
```

Arguments

userName

The user name of the user whose privileges on archive folders should be removed.

CompactMasterDatabase

Compacts the master database.

CompactStore

Compacts an archive store.

```
CompactStore --id
```

Arguments

id

The identifier of the archive store to be compacted.

CreateStore

Creates a new archive store and attaches it afterwards.

```
CreateStore --name [--type] [--databaseName] [--databasePath] [--contentPath]
              [--indexPath] [--serverName] [--userName] [--password] [--requestedState]
```

Arguments

name

A meaningful name for the archive store. Examples: "Messages 2012" or "2012-01".

type *(optional)*

The archive store type. Possible values:

FileSystemStandard

Standard archive store. Recommended for most environments. This is the default value.

FileSystemAdvanced

Advanced file system-based archive store. The archive store will entirely be stored in the file system (local hard disk or network share).

SQLServer

The archive store will be stored in an external Microsoft SQL Server database. E-mail messages can either be stored in the database or in the file system.

PostgreSQL

The archive store will be stored in an external PostgreSQL database. E-mail messages can either be stored in the database or in the file system.

databasePath *(optional)*

Only valid for types *FileSystemStandard* and *FileSystemAdvanced*. The directory in which both folder information and e-mail metadata are stored.

serverName *(optional)*

Only valid for types *PostgreSQL* and *SQLServer*. The server name of the database server.

userName *(optional)*

Only valid for types *PostgreSQL* and *SQLServer*. The user name which will be used to connect to the database server.

password (*optional*)

Only valid for types *PostgreSQL* and *SQLServer*. The password which will be used to connect to the database server.

databaseName (*optional*)

Only valid for types *PostgreSQL* and *SQLServer*. The database on the database server in which both folder information and e-mail metadata are stored.

contentPath (*optional*)

The directory in which e-mail headers and contents are stored. If the archive store type is *PostgreSQL* or *SQLServer*, you don't need to specify this argument - in this case, e-mail headers and contents are stored in the database.

indexPath (*optional*)

The directory in which the full-text index is stored.

requestedState (*optional*)

The requested state. The default value when creating stores is *current*. List of possible values:

disabled

The archive store should be disabled. This causes the archive store to be closed if it is currently open.

writeProtected

The archive store should be write-protected.

normal

The archive store should be opened normally. Write access is possible, but new e-mail messages are not archived into this store.

current

New e-mail messages should be archived into this store.

CreateUser

Adds a new user to MailStore Server.

```
CreateUser --userName --privileges [--fullName] [--distinguishedName]
           [--authentication] [--password]
```

Arguments

userName

The name of the user to be created.

privileges

A comma-separated list of global privileges that the user should be granted. Possible values are:

none

The user is granted no global privileges. If specified, this value has to be the only value in the list.

admin

The user is granted administrator privileges. If specified, this value has to be the only value in the list.

login

The user can log on to MailStore Server.

changePassword

The user can change his own MailStore Server password. This only makes sense if the authentication is set to *integrated*.

archive

The user can run archiving profiles.

modifyArchiveProfiles

The user can create, modify and delete archiving profiles.

export

The user can run export profiles.

modifyExportProfiles

The user can create, modify and delete export profiles.

delete

The user can delete messages. Please note that a non-admin user can only delete messages in folders where he has been granted delete access. In addition, compliance settings may be in effect, keeping administrators and users from deleting messages even when they have been granted the privilege to do so.

fullName *(optional)*

The full name (display name) of the user, e.g. "John Doe".

distinguishedName *(optional)*

The LDAP distinguished name of the user. This is typically automatically specified when synchronizing with Active Directory or other LDAP servers.

authentication *(optional)*

The authentication mode. Possible values are:

integrated

Specifies MailStore-integrated authentication. This is the default value.

directoryServices

Specified Directory Services authentication. If this value is specified, the *password* is stored, but is ignored when the user logs on to MailStore Server.

password *(optional)*

The password that the user can use to log on to MailStore Server. This is only meaningful when *authentication* is set to *integrated*.

DeleteEmptyFolders

Deletes archive folders which don't contain any messages.

DeleteEmptyFolders [--folder]

Arguments

folder *(optional)*

If specified, only this folder and its subfolders are deleted if empty.

DeleteUser

Deletes a user from MailStore Server. Neither the user's archive nor the user's archived e-mail is deleted when deleting users.

```
DeleteUser --userName
```

Arguments

userName

The user name of the user to be deleted.

DetachStore

Detaches an archive store from MailStore Server.

```
DetachStore --id
```

Arguments

id

This identifier of the archive store to be detached.

GetActiveSessions

Retrieves a list of active logon sessions.

```
GetActiveSessions
```

GetChildFolders

Retrieves a list of child folders of a specific folder.

```
GetChildFolders [--folder] [--maxLevels]
```

Arguments

folder *(optional)*

The folder of which the child folders are to be retrieved. If you don't specify this parameter, the method returns the child folders of the root level (user archives).

maxLevels *(optional)*

If *maxLevels* is not specified, this method returns the child folders recursively, which means that you get the whole folder hierarchy starting at the folder specified. Set *maxLevels* to a value equal to or greater than 1 to limit the levels returned.

GetMethodList

Retrieves a list of API methods.

```
GetMethodList
```

GetServerInfo

Retrieves a list of server information.

```
GetServerInfo
```

GetStoreIndexList

Retrieves a list of full-text indexes that are stored within an archive store.

```
GetStoreIndexList --id
```

Arguments

id

The identifier of the archive store whose full-text indexes are to be returned.

GetStoreList

Retrieves a list of archive stores currently attached to MailStore Server.

```
GetStoreList
```

GetUserInfo

Retrieves a detailed user information object about a specific user.

```
GetUserInfo --userName
```

Arguments

userName

The user name of the user whose information object should be returned.

GetTimeZones

Retrieves a list of all available time zones. This is useful for GetWorkerResults

```
GetTimeZones
```

GetUserList

Retrieves a list of all users in MailStore Server.

```
GetUserList
```

GetWorkerResults

Retrieves a list of ended archiving jobs.

```
GetWorkerResults --fromIncluding --toExcluding --timZoneId
```

Arguments

fromIncluding

The date which indicates the beginning time, e.g. "2013-01-01T00:00:00".

toExcluding

The date which indicates the ending time, e.g. "2013-02-28T23:59:59".

timeZoneId

The time zone the date should be converted to, e.g. "\$Local", which represents the time zone of the operating system

MaintainFileSystemDatabases

Runs maintenance on all file system-based databases (Firebird Embedded .fdb files). Each database file will be rebuilt by this operation by creating a backup file and restoring from that backup file.

```
MaintainFileSystemDatabases
```

MoveFolder

Moves or renames an archive folder.

```
MoveFolder --fromFolder --toFolder
```

Arguments

fromFolder

The folder which should be moved or renamed, e.g. "johndoe/Outlook/Inbox".

toFolder

The target folder name, e.g. "johndoe/Outlook/Inbox-new".

Example 1: Rename a user archive

The following example renames the user archive "johndoe" to "john.doe".

```
MoveFolder --fromFolder="johndoe" --toFolder="john.doe"
```

Example 2: Rename a folder within the user archive

The following example renames the folder "Outlook" within the user archive "johndoe" to "Microsoft Outlook".

```
MoveFolder --fromFolder="johndoe/Outlook" --toFolder="johndoe/Microsoft Outlook"
```

Example 3: Move a folder within a user archive

The following example moves the folder "Project A" into the folder "Projects".

```
MoveFolder --fromFolder="johndoe/Outlook/Project A" --toFolder="johndoe/Outlook/Projects/Project A"
```

RebuildStoreIndex

Rebuilds a full-text index contained within an archive store.

```
RebuildStoreIndex --id --folder
```

Arguments

id

The identifier of the archive store that contains the full-text index to be rebuilt.

folder

The full-text index to be rebuilt, e.g. "johndoe".

RefreshAllStoreStatistics

Refreshes the statistics of all currently open archive stores.

```
RefreshAllStoreStatistics
```

RenameStore

This method is not documented.

```
RenameStore --id --name
```

Arguments

id

The identifier of the archive store to be renamed.

name

The new archive store name.

RenameUser

Renames a user. The user's archive will not be renamed by this method.

```
RenameUser --oldUserName --newUserName
```

Arguments

oldUserName

The user name of the user to be renamed.

newUserName

The new user name.

RetryOpenStores

Retries opening stores that could not be opened the last time.

```
RetryOpenStores
```

SetStoreProperties

Sets the properties of a store.

```
SetStoreProperties --id [--type] [--databaseName] [--databasePath] [--contentPath]
                    [--indexPath] [--serverName] [--userName] [--password]
```

Arguments

id

This argument is not documented.

type *(optional)*

databaseName *(optional)*

databasePath *(optional)*

contentPath *(optional)*

indexPath *(optional)*

serverName *(optional)*

userName *(optional)*

password *(optional)*

Please refer to the FileGroupCreate method documentation for information about these parameters.

SetStoreRequestedState

Sets the requested state of a store.

```
SetStoreRequestedState --id --requestedState
```

Arguments

id

The identifier of the archive store whose requested state should be set.

requestedState

The requested state to be set. For a list of possible values, please refer to the documentation of the *CreateStore* method.

SetUserAuthentication

Sets the authentication mode of a user.

```
SetUserAuthentication --userName --authentication
```

Arguments

userName

The user name of the user whose authentication mode should be set.

authentication

The authentication mode that should be set. For a list of possible values, please refer to the documentation of the *CreateUser* method.

SetUserDistinguishedName

Sets the distinguished name (DN) of a user.

```
SetUserDistinguishedName --userName [--distinguishedName]
```

Arguments

userName

The user name of the user whose distinguished name should be set (or removed).

distinguishedName *(optional)*

The distinguished name to be set. If this argument is not specified, the distinguished name of the specified user is removed.

SetUserEmailAddresses

Sets the e-mail addresses of a user.

```
SetUserEmailAddresses --userName [--emailAddresses]
```

Arguments

userName

The user name of the user whose e-mail addresses are to be set.

emailAddresses *(optional)*

A comma-separated list of e-mail addresses. The first e-mail address in the list must be the user's primary e-mail address.

SetUserFullName

Sets the full name (display name) of a user.

```
SetUserFullName --userName [--fullName]
```

Arguments

userName

The user name of the user whose full name (display name) should be set (or removed).

fullName *(optional)*

The full name to be set. If this argument is not specified, the full name of the specified user is removed.

SetUserPassword

Sets the password of a user.

```
SetUserPassword --userName --password
```

Arguments

userName

The user name of the user whose MailStore Server should be set.

password

The new password.

SetUserPop3UserNames

Sets the POP3 user names of a user (used for MailStore Proxy).

```
SetUserPop3UserNames --userName [- -pop3UserNames]
```

Arguments

userName

The user name of the user whose POP3 user names should be set.

pop3UserNames (*optional*)

A comma-separated list of POP3 user names that should be set.

SetUserPrivileges

Sets the privileges of a user.

```
SetUserPrivileges --userName --privileges
```

Arguments

userName

The user name of the user whose global privileges should be set.

privileges

A comma-separated list of global privileges. For a list of possible values, please refer to the documentation of the *CreateUser* method.

SetUserPrivilegesOnFolder

Sets a user's privileges on a specific folder.

```
SetUserPrivilegesOnFolder --userName --folder --privileges
```

Arguments

userName

The user name of the user who should be granted or denied privileges.

folder

The folder on which the user should be granted or denied privileges. In the current version, this can only be a top-level folder (user archive).

privileges

A comma-separated list of privileges that the specified user should be granted on the specified folder. Possible values are:

none

The user is denied access to the specified folder. If specified, this value has to be the only value in the list.

read

The user is granted read access to the specified folder.

write

The user is granted write access to the specified folder.

delete

The user is granted delete access to the specified folder.

SyncUsersWithDirectoryServices

Synchronizes with Active Directory or another directory service according to MailStore Server's configuration.

```
SyncUsersWithDirectoryServices [--dryRun]
```

Arguments

dryRun

if set (--dryRun=true) retrieves the user data from the directory service according to MailStore Server's configuration but does not sync the data.

UpgradeStore

Upgrades an archive store created in MailStore Server 5.x or earlier.

```
UpgradeStore --id
```

Arguments

id

The identifier of the archive store to be upgraded.

VerifyStore

Verifies the contents of an archive store.

Arguments

id

The identifier of the archive store to be verified.

🕒Revision #1

★Created 28 December 2021 17:18:45 by Mahesha Damayanthi

✎Updated 28 December 2021 17:51:58 by Mahesha Damayanthi