

MailEnable PowerShell Reference

10th April 2019

Overview

MailEnable's PowerShell interface allows system administrators and developers to manage MailEnable via PowerShell.

Administrators can perform typical actions via scripts and can develop and extend these to these commands via scripting. PowerShell significantly improves automation and integrated management of MailEnable.

PowerShell support is available for all editions of MailEnable since version 9.5.

Launching PowerShell

You can launch PowerShell either by:

1. Selecting **Windows PowerShell** from the Start screen, or:
2. Selecting **Windows PowerShell** from the taskbar.

Executing MailEnable PowerShell Commands

Once PowerShell opens, use the following command to add the MailEnable PowerShell snap-in to the current PowerShell session:

```
Add-PSSnapin MailEnable.Provision.Command
```

You can then issue specific commands depending on the area of MailEnable you wish to configure. You can add this to your PowerShell profile so this line is not needed each time.

The first command to issue is the Help command. It provides a comprehensive list of the settings that can be manipulated via PowerShell.

```
Example: PS> Get-MailEnablePlatform -Help "*"
```

The asterisk tells PowerShell to return all settings.

You can also filter them by simply providing part of the setting name.

```
Example: PS>Get-MailEnablePlatform -Help "Skin"
```

```
SettingId      : sysSkinCatalogueEnabled
SettingType    : MailEnablePlatform
SettingDataType : dword
SettingControl : 0=Disabled, 1=Enabled
Purpose        : Determines whether the Management Console shows an online skin catalogue
```

The Help command returns the SettingId (the name of the setting), and the SettingType (which is the Command that is used to Set it).

PowerShell Commands are prefixed with the action you wish to perform, i.e.: New, Get, Set, Remove, etc.

As such, to enable the Skin Catalogue within MailEnable, you would execute the following command:

```
Set-MailEnablePlatform -Setting sysSkinCatalogueEnabled -Value 1
```

MailEnable PowerShell Command Overview

MailEnable's PowerShell interface is compact but provides the ability to change thousands of settings. It provides the following cmdlets, each of which can be passed parameters to perform configuration changes.

The platform cmdlets manage server wide settings and options, such as service configuration.

Get-MailEnablePlatform

Set-MailEnablePlatform

The postoffice cmdlets manage postoffices and their settings.

Get-MailEnablePostoffice

Set-MailEnablePostoffice

Add-MailEnablePostoffice

Remove-MailEnablePostoffice

The mailbox cmdlets manage users and their settings.

Get-MailEnableMailbox

Set-MailEnableMailbox

Add-MailEnableMailbox

Remove-MailEnableMailbox

The domain cmdlets manage domain settings

Get-MailEnableDomain

Set-MailEnableDomain

Add-MailEnableDomain

Remove-MailEnableDomain

The address map cmdlets

MailEnableAddressMap - manage email addresses

The group cmdlets

MailEnableGroup - manage distribution groups

The group member cmdlets

MailEnableGroupMember - manage distribution group membership

The list cmdlets

MailEnableList - manage distribution lists

The list member cmdlets

MailEnableListMember - manage distribution group membership

PowerShell Command Line Examples

The following examples show some of the more common commands you may wish to execute via PowerShell.

Before executing the commands, you should ensure that the extension is loaded using the following PowerShell command: `Add-PSSnapin MailEnable.Provision.Command`

Platform Examples

Get the current IMAP alternate port number

```
Get-MailEnablePlatform -Setting imapAlternatePort
```

Set the current IMAP alternate port number

```
Set-MailEnablePlatform -Setting imapAlternatePort -Value 993
```

Postoffice Examples

```
# Create a new Postoffice with domain
```

```
New-MailEnablePostoffice -Domain "example.com" -Postoffice "example.com"
```

```
# Create a new Postoffice, domain and mailbox as ADMIN user
```

```
New-MailEnablePostoffice -Domain "example.com" -Postoffice "example.com" -AdminUserName "adminuser" -AdminPassword "pwd1234"
```

```
# Create a new Postoffice, domain and mailbox as ADMIN user, and mailbox as USER user
```

```
New-MailEnablePostoffice -Domain "example.com" -Postoffice "example.com" -AdminUserName "adminuser" -AdminPassword "pwd1234" -UserName "sales" -Password "pwd1234"
```

```
# Create a new Postoffice, domain and multiple mailboxes
```

```
New-MailEnablePostoffice -Domain "example.com" -Postoffice "example.com" -UserName "sales,support,accounts,marketing" -Password "pwd1234,pwd4321,pwd6789,pwd3456"
```

```
# Enable ActiveSync for a Postoffice
```

```
Set-MailEnablePostoffice -Postoffice "example.com" -Setting poActiveSync -Value 1
```

```
# Get the state of ActiveSync for a Postoffice
```

```
Get-MailEnablePostoffice -Postoffice "example.com" -Setting poActiveSync
```

Mailbox Examples

```
# Create a new mailbox for a Postoffice/domain
```

```
New-MailEnableMailbox -Mailbox "Joe" -Domain "example.com" -Password "password" -Right "USER"
```

```
# Change user password
```

```
Set-MailEnableMailbox -Postoffice "example.com" -Mailbox "Joe" -Setting "mailboxPassword" -Value "blah123"
```

```
# Change user display name
```

```
Set-MailEnableMailbox -Postoffice "example.com" -Mailbox "Joe" -Setting "mailboxDisplayName" -Value "mydisplayname"
```

```
# Change user timezone
```

```
Set-MailEnableMailbox -Postoffice "example.com" -Mailbox "Joe" -Setting "mailboxTimeZone" -Value "Mountain Standard Time"
```

```
# Set a user's mailbox quota to 50MB
```

```
Set-MailEnableMailbox -Postoffice MailEnable -Mailbox Andrew -Setting mailboxLimit -Value 50000
```

```
# Set a user's mailbox quota to unlimited
```

```
Set-MailEnableMailbox -Postoffice MailEnable -Mailbox Andrew -Setting mailboxLimit -Value "-1"
```

```
# Disable login access to a mailbox
```

```
Set-MailEnableMailbox -Postoffice MailEnable -Mailbox Andrew -Setting mailboxLoginStatus -Value 0
```

```
# Disable a mailbox (prevents message delivery to a mailbox)
```

```
Set-MailEnableMailbox -Postoffice MailEnable -Mailbox Andrew -Setting mailboxStatus -Value 0
```

Domain Examples

List the domains for the platform

```
Get-MailEnableDomain
```

Redirect domain to another host

```
Set-MailEnableDomain -Domain "example32.com" -Setting "domainRedirectionHosts" -Value  
"test.example.com,test2.example.com"
```

```
Set-MailEnableDomain -Domain "example32.com" -Setting "domainRedirectionStatus" -Value "1"
```

Disable domain redirection

```
Set-MailEnableDomain -Domain "example32.com" -Setting "domainRedirectionStatus" -Value "0"
```

List domains

```
Get-MailEnableDomain -Domain "example32.com"
```

Get properties of a domain object

```
Get-MailEnableDomain -Domain "example32.com"
```

Get a setting for a domain

```
Get-MailEnableDomain -Domain "example32.com" -Setting "domainRedirectionStatus"
```

List the domains for a given Postoffice

```
Get-MailEnableDomain -Postoffice "example.com"
```

Mail Distribution Group Examples

Create a new group

```
New-MailEnableGroup -Postoffice "example.com" -GroupName "myGroup222" -EmailAddress  
"mygroup@example.com"
```

Remove a group

```
Remove-MailEnableGroup -Postoffice "example.com" -GroupName "myGroup222"
```

Add a member to a group

```
New-MailEnableGroupMember -Postoffice "example.com" -GroupName "myGroup222" -EmailAddress  
"member@example.com"
```

Remove a member from a group

```
Remove-MailEnableGroupMember -Postoffice "example.com" -GroupName "myGroup222" -EmailAddress  
"member@example.com"
```

List the members of a group

```
Get-MailEnableGroupMember -Postoffice "example.com" -GroupName "myGroup222"
```

Mail Distribution List Examples

Create a Distribution List

```
New-MailEnableList -Postoffice "example.com" -ListName "myList222" -EmailAddress "mylist@example.com" -  
Description "my example list"
```

Remove list

```
Remove-MailEnableList -Postoffice "example.com" -ListName "myList222"
```

Add a member to a list

```
New-MailEnableListMember -Postoffice "example.com" -ListName "myList222" -EmailAddress "member@example.com"
```

```
# Remove a member from a list
```

```
Remove-MailEnableListMember -Postoffice "example.com" -ListName "myList222" -EmailAddress "member@example.com"
```

```
# List the members of a list
```

```
Get-MailEnableListMember -Postoffice "example.com" -ListName "myList222"
```

Mail Address Examples

```
# Add a mapping to an SMTP address to a local mailbox
```

```
New-MailEnableAddressMap -EmailAddress "Email@example.com" -Postoffice "example.com" -Mailbox "Joe"
```

```
# Remove a mapping to an SMTP address to a local mailbox
```

```
Remove-MailEnableAddressMap -EmailAddress "Email@example.com" -Postoffice "example.com" -Mailbox "Joe"
```

```
# Get the Email addresses associated with a mailbox
```

```
Get-MailEnableAddressMap -Postoffice "example.com" -Mailbox "Joe"
```

PowerShell Script Example

The following example shows how you can perform batch operations within PowerShell scripts.

The example below shows how you can iterate all the postoffices within MailEnable and enable public folder support. You can use the PowerShell ISE to develop your PowerShell scripts.

```
#####  
# Example script: Enable Public Folders for all Postoffices  
#####  
# Ensure the snapin module is loaded (if it is not already)  
if ((Get-PSSnapin -Name MailEnable.Provision.Command -ErrorAction SilentlyContinue) -eq $null )  
{  
    Add-PsSnapin MailEnable.Provision.Command  
}  
# Get the list of PostOffices  
$lstPostoffices = Get-MailEnablePostoffice  
  
ForEach($oPostoffice in $lstPostoffices)  
{  
    # Set the Public Folder Status Flag for each Postoffice  
    Set-MailEnablePostoffice -Postoffice $oPostoffice.Name -Setting "poPublicFolders" -Value 1  
}
```

A similar script can also be used to change settings on all mailboxes in a postoffice. The following script disables webmail for all mailboxes in a postoffice. Although the postoffice setting is checked first, this may be useful if you need to disable a feature for all mailboxes and enable ones on a case by case basis.

```
# Ensure the snapin module is loaded (if it is not already)  
if ((Get-PSSnapin -Name MailEnable.Provision.Command -ErrorAction SilentlyContinue) -eq $null)  
{  
    Add-PSSnapin MailEnable.Provision.Command  
}  
  
# Get the list of mailboxes in the postoffice  
$lstMailboxes = Get-MailEnableMailbox -Postoffice "example.com"  
  
ForEach($oMailbox in $lstMailboxes)
```

```
{  
    # Set the webmail setting  
    Set-MailEnableMailbox -Postoffice "example.com" -Mailbox $oMailbox.MailboxName -Setting  
"mailboxWebMail" -Value 0  
}
```

More Information

For more information on configuring MailEnable via PowerShell and integrating with other cmdlets, please refer to the Microsoft PowerShell reference.