Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Send-MailMessage'

PS C:\Users\wahid> Get-Help Send-MailMessage

```
NAME
    Send-MailMessage


SYNOPSIS
    Sends an email message.



SYNTAX
    Send-MailMessage [-To] <System.String[]> [-Subject] <System.String> [[-Body]
    <System.String>] [[-SmtpServer] <System.String>] [-Attachments
    <System.String[]>] [-Bcc <System.String[]>] [-BodyAsHtml] [-Cc
    <System.String[]>] [-Credential <System.Management.Automation.PSCredential>]
    [-DeliveryNotificationOption {None | OnSuccess | OnFailure | Delay | Never}]
    [-Encoding {ASCII | BigEndianUnicode | Default | OEM | Unicode | UTF7 | UTF8 |
    UTF32}] -From <System.String> [-Port <System.Int32>] [-Priority {Normal | High
    | Low}] [-UseSsl] [<CommonParameters>]



DESCRIPTION
    The `Send-MailMessage` cmdlet sends an email message from within PowerShell.
```

You must specify a Simple Mail Transfer Protocol (SMTP) server or the `Send-MailMessage` command fails. Use the SmtpServer parameter or set the `$PSEmailServer` variable to a valid SMTP server. The value assigned to `$PSEmailServer` is the default SMTP setting for PowerShell. For more information, see about_Preference_Variables (../Microsoft.PowerShell.Core/About/about_Preference_Variables.md).

> [!WARNING] > The `Send-MailMessage` cmdlet is obsolete. This cmdlet doesn't guarantee secure connections to > SMTP servers. While there is no immediate replacement available in PowerShell, we recommend you do > not use `Send-MailMessage`. For more information, see > Platform Compatibility note DE0005 (https://aka.ms/SendMailMessage).

PARAMETERS

-Attachments <System.String[]>

Specifies the path and file names of files to be attached to the email message. You can use this parameter or pipe the paths and file names to `Send-MailMessage`.

-Bcc <System.String[]>

Specifies the email addresses that receive a copy of the mail but aren't listed as recipients of the message. Enter names (optional) and the email address, such as `Name <someone@fabrikam.com>`.

-Body <System.String>

Specifies the content of the email message.

-BodyAsHtml <System.Management.Automation.SwitchParameter>

Specifies that the value of the Body parameter contains HTML.

-Cc <System.String[]>

Specifies the email addresses to which a carbon copy (CC) of the email message is sent. Enter names (optional) and the email address, such as `Name <someone@fabrikam.com>`.

-Credential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to perform this action. The default is the current user.

Type a user name, such as User01 or Domain01\User01 . Or, enter a PSCredential object, such as one from the `Get-Credential` cmdlet.

Credentials are stored in a PSCredential (/dotnet/api/system.management.automation.pscredential)object and the password is stored as a SecureString (/dotnet/api/system.security.securestring).

> [!NOTE] > For more information about SecureString data protection, see > How secure is SecureString? (/dotnet/api/system.security.securestring#how-secure-is-securestring).

-DeliveryNotificationOption <System.Net.Mail.DeliveryNotificationOptions>

Specifies the delivery notification options for the email message. You can specify multiple values. None is the default value. The alias for this parameter is DNO .

The delivery notifications are sent to the address in the From parameter.

The acceptable values for this parameter are as follows:

- `None`: No notification.

- `OnSuccess`: Notify if the delivery is successful.

- `OnFailure`: Notify if the delivery is unsuccessful.

- `Delay`: Notify if the delivery is delayed.

- `Never`: Never notify.

These values are defined as a flag-based enumeration. You can combine multiple values together to set multiple flags using this parameter. The values can be passed to the DeliveryNotification parameter as an array of values or as a comma-separated string of those values. The cmdlet will combine the values using a binary-OR operation. Passing values as an array is the simplest option and also allows you to use tab-completion on the values.

-Encoding <System.Text.Encoding>

Specifies the type of encoding for the target file. The default value is `Default`.

The acceptable values for this parameter are as follows:

- `ASCII` Uses ASCII (7-bit) character set.

- `BigEndianUnicode` Uses UTF-16 with the big-endian byte order.

- `Default` Uses the encoding that corresponds to the system's active code page (usually ANSI).

- `OEM` Uses the encoding that corresponds to the system's current OEM code page.

- `Unicode` Uses UTF-16 with the little-endian byte order.

- `UTF7` Uses UTF-7.

- `UTF8` Uses UTF-8.

- `UTF32` Uses UTF-32 with the little-endian byte order.

-From <System.String>

The From parameter is required. This parameter specifies the sender's email address. Enter a name (optional) and email address, such as `Name <someone@fabrikam.com>`.

-Port <System.Int32>

Specifies an alternate port on the SMTP server. The default value is 25, which is the default SMTP port.

-Priority <System.Net.Mail.MailPriority>

Specifies the priority of the email message. Normal is the default. The acceptable values for this parameter are Normal, High, and Low.

-SmtpServer <System.String>

Specifies the name of the SMTP server that sends the email message.

The default value is the value of the `$PSEmailServer` preference variable. If the preference variable isn't set and this parameter isn't used, the `Send-MailMessage` command fails.

-Subject <System.String>

The Subject parameter isn't required. This parameter specifies the subject of the email message.

-To <System.String[]>

The To parameter is required. This parameter specifies the recipient's email address. Enter names (optional) and the email address, such as `Name

<someone@fabrikam.com>`.

-UseSsl <System.Management.Automation.SwitchParameter>

The Secure Sockets Layer (SSL) protocol is used to establish a secure connection to the remote computer to send mail. By default, SSL isn't used.

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

-- Example 1: Send an email from one person to another person --

```
$sendMailMessageSplat = @{
    From = 'User01 <user01@fabrikam.com>'
    To = 'User02 <user02@fabrikam.com>'
    Subject = 'Test mail'
}
Send-MailMessage @sendMailMessageSplat
```

The `Send-MailMessage` cmdlet uses the From parameter to specify the message's sender. The To parameter specifies the message's recipient. The Subject parameter uses the text string Test mail as the message because the optional Body parameter isn't included.

---------------- Example 2: Send an attachment ----------------

```
$sendMailMessageSplat = @{
    From = 'User01 <user01@fabrikam.com>'
    To = 'User02 <user02@fabrikam.com>', 'User03 <user03@fabrikam.com>'
    Subject = 'Sending the Attachment'
    Body = "Forgot to send the attachment. Sending now."
    Attachments = '.\data.csv'
    Priority = 'High'
    DeliveryNotificationOption = 'OnSuccess', 'OnFailure'
    SmtpServer = 'smtp.fabrikam.com'
}
Send-MailMessage @sendMailMessageSplat
```

The `Send-MailMessage` cmdlet uses the From parameter to specify the message's sender. The To parameter specifies the message's recipients. The Subject parameter describes the content of the message. The Body parameter is the content of the message.

The Attachments parameter specifies the file in the current directory that's attached to the email message. The Priority parameter sets the message to `High` priority. The DeliveryNotificationOption parameter specifies two values, `OnSuccess` and `OnFailure`. The sender will receive email notifications to confirm the success or failure of the message delivery. The SmtpServer parameter sets the SMTP server to `smtp.fabrikam.com`.

----------- Example 3: Send email to a mailing list -----------

```
$sendMailMessageSplat = @{
    From = 'User01 <user01@fabrikam.com>'
    To = 'ITGroup <itdept@fabrikam.com>'
    Cc = 'User02 <user02@fabrikam.com>'
    Bcc = 'ITMgr <itmgr@fabrikam.com>'
    Subject = "Don't forget today's meeting!"
    Credential = 'domain01\admin01'
    UseSsl = $true
}
Send-MailMessage @sendMailMessageSplat
```

The `Send-MailMessage` cmdlet uses the From parameter to specify the message's sender. The To parameter specifies the message's recipients. The Cc parameter sends a copy of the message to the specified recipient. The Bcc parameter sends a blind copy of the message. A blind copy is an email address that's hidden from the other recipients. The Subject parameter is the message because the optional Body parameter isn't included.

The Credential parameter specifies a domain administrator's credentials are used to send the message. The UseSsl parameter specifies that Secure Socket Layer (SSL) creates a secure connection.

REMARKS

To see the examples, type: "get-help Send-MailMessage -examples".

For more information, type: "get-help Send-MailMessage -detailed".

For technical information, type: "get-help Send-MailMessage -full".

For online help, type: "get-help Send-MailMessage -online"