

# Veritas Enterprise Vault™

## PowerShell Cmdlets

12

# Veritas Enterprise Vault: PowerShell Cmdlets

Last updated: 2015-12-17.

## Legal Notice

Copyright © 2015 Veritas Technologies LLC. All rights reserved.

Veritas, the Veritas Logo, Enterprise Vault, Compliance Accelerator, and Discovery Accelerator are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This product may contain third party software for which Veritas is required to provide attribution to the third party ("Third Party Programs"). Some of the Third Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. For more information on the Third Party Programs, please see the Third Party Notice document for this Veritas product that is available at <http://www.veritas.com/about/legal/license-agreements#3rd-party>.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Veritas Technologies LLC and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. VERITAS TECHNOLOGIES LLC SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Veritas Technologies LLC  
500 E Middlefield Road  
Mountain View, CA 94043

<http://www.veritas.com>

# Technical Support

Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within the company to answer your questions in a timely fashion.

Our support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about our support offerings, you can visit our website at the following URL:

[www.veritas.com/support](http://www.veritas.com/support)

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

## Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

[www.veritas.com/support](http://www.veritas.com/support)

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information
- Available memory, disk space, and NIC information

- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Technical Support
  - Recent software configuration changes and network changes

## Licensing and registration

If your product requires registration or a license key, access our technical support webpage at the following URL:

[www.veritas.com/support](http://www.veritas.com/support)

## Customer service

Customer service information is available at the following URL:

[www.veritas.com/support](http://www.veritas.com/support)

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Advice about technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs, DVDs, or manuals

## Support agreement resources

If you want to contact us regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Worldwide (except Japan)

[CustomerCare@veritas.com](mailto:CustomerCare@veritas.com)

Japan

[CustomerCare\\_Japan@veritas.com](mailto:CustomerCare_Japan@veritas.com)

# Contents

Technical Support .....	4	
<b>Chapter 1</b>	<b>About this guide .....</b>	<b>10</b>
	Introducing this guide .....	10
	Where to get more information about Enterprise Vault .....	10
	“How To” articles on the Veritas Support website .....	12
	Enterprise Vault training modules .....	13
	Comment on the documentation .....	13
<b>Chapter 2</b>	<b>Accessing the Enterprise Vault cmdlets .....</b>	<b>14</b>
	Enterprise Vault Management Shell .....	14
	Importing modules .....	14
<b>Chapter 3</b>	<b>Help for Enterprise Vault PowerShell</b>	
	<b>cmdlets .....</b>	<b>16</b>
	Getting help for Enterprise Vault PowerShell cmdlets .....	16
<b>Chapter 4</b>	<b>Enterprise Vault PowerShell cmdlet</b>	
	<b>reference .....</b>	<b>17</b>
	Get-EVArchive .....	20
	Set-EVArchive .....	23
	Get-EVClassificationPolicy .....	26
	New-EVClassificationPolicy .....	28
	Remove-EVClassificationPolicy .....	31
	Set-EVClassificationPolicy .....	32
	Import-EVClassificationRules .....	34
	Publish-EVClassificationRules .....	37
	Get-EVClassificationTags .....	40
	Get-EVClassificationTestMode .....	41
	Set-EVClassificationTestMode .....	43
	Get-EVComputers .....	44
	Get-EVDatabase .....	45
	Get-EVDatabaseDetail .....	46

Set-EVDatabaseDetail .....	48
Get-EVDatabaseFileInfo .....	50
Start-EVDatabaseUpgrade .....	52
Get-EVDependencyService .....	53
Get-EVDependencyServiceState .....	54
Remove-EVExchangeFolderPolicy .....	55
Get-EVFileLocation .....	56
Get-EVFSAVolume .....	58
New-EVFSAVolume .....	60
Remove-EVFSAVolume .....	63
Set-EVFSAVolume .....	64
Get-EVIIWebsite .....	67
Set-EVIMAPServerDisabled .....	68
Set-EVIMAPServerEnabled .....	69
Get-EVIMAPUsers .....	70
Get-EVIMAPUserSettings .....	72
Get-EVIndexLocation .....	74
New-EVMDSBuildTask .....	75
Get-EVMDSStatus .....	78
Add-EVPstComputer .....	80
Add-EVPstFile .....	81
Get-EVRBARole .....	83
Add-EVRBARoleMember .....	86
Get-EVRBARoleMember .....	89
Remove-EVRBARoleMember .....	92
Get-EVRBAAzStoreXml .....	95
Set-EVRBAAzStoreXml .....	97
Get-EVRetentionPlan .....	98
New-EVRetentionPlan .....	101
Remove-EVRetentionPlan .....	104
Set-EVRetentionPlan .....	105
Get-EVServer .....	109
Get-EVService .....	110
Get-EVServiceState .....	111
Get-EVSite .....	112
Get-EVSMTPHoldingFolder .....	113
Get-EVSMTPPolicy .....	114
New-EVSMTPPolicy .....	116
Remove-EVSMTPPolicy .....	118
Set-EVSMTPPolicy .....	119
Get-EVSMTPServerSettings .....	122
New-EVSMTPServerSettings .....	124
Set-EVSMTPServerSettings .....	130

Sync-EVSMTPServerSettings .....	135
Get-EVSMTPTarget .....	137
New-EVSMTPTarget .....	140
Remove-EVSMTPTarget .....	143
Set-EVSMTPTarget .....	144
Get-EVStorageDatabase .....	147
Get-EVTask .....	148
Get-EVTaskState .....	150
Get-EVVaultStore .....	151
Get-EVVaultStorePartition .....	152
Clear-IndexLocationBackupMode .....	155
Get-IndexLocationBackupMode .....	157
Set-IndexLocationBackupMode .....	159
Set-IndexMetadataSyncLevel .....	160
Get-IndexServerForIndexLocation .....	162
Start-PartitionRollover .....	163
Clear-VaultStoreBackupMode .....	163
Get-VaultStoreBackupMode .....	166
Set-VaultStoreBackupMode .....	168
Index .....	170

# About this guide

This chapter includes the following topics:

- [Introducing this guide](#)
- [Where to get more information about Enterprise Vault](#)
- [Comment on the documentation](#)

## Introducing this guide

This guide provides usage information for all the Enterprise Vault PowerShell cmdlets.

## Where to get more information about Enterprise Vault

[Table 1-1](#) lists the documentation that accompanies Enterprise Vault.

**Table 1-1** Enterprise Vault documentation set

Document	Comments
Veritas Enterprise Vault Documentation Library	<p>Includes all the following documents in Windows Help (.chm) format so that you can search across them all. It also includes links to the guides in Acrobat (.pdf) format.</p> <p>You can access the library in several ways, including the following:</p> <ul style="list-style-type: none"> <li>■ On the Windows <b>Start</b> menu, click <b>Start &gt; Programs &gt; Enterprise Vault &gt; Documentation</b>.</li> <li>■ In Windows Explorer, browse to the <code>Documentation\language</code> subfolder of the Enterprise Vault installation folder, and then open the <code>EV_Help.chm</code> file.</li> <li>■ On the <b>Help</b> menu in the Administration Console, click <b>Help on Enterprise Vault</b>.</li> </ul>
<i>Introduction and Planning</i>	Provides an overview of Enterprise Vault functionality.
<i>Deployment Scanner</i>	Describes how to check the required software and settings before you install Enterprise Vault.
<i>Installing and Configuring</i>	Provides detailed information on setting up Enterprise Vault.
<i>Upgrade Instructions</i>	Describes how to upgrade an existing Enterprise Vault installation to the latest version.
<i>Setting up Domino Server Archiving</i>	Describes how to archive items from Domino mail files and journal databases.
<i>Setting up Exchange Server Archiving</i>	Describes how to archive items from Microsoft Exchange user mailboxes, journal mailboxes, and public folders.
<i>Setting up File System Archiving</i>	Describes how to archive the files that are held on network file servers.
<i>Setting up IMAP</i>	Describes how to configure IMAP client access to Exchange archives and Internet mail archives.
<i>Setting up SMTP Archiving</i>	Describes how to archive SMTP messages from other messaging servers.
<i>Setting up SharePoint Server Archiving</i>	Describes how to archive content from Microsoft SharePoint servers.
<i>Administrator's Guide</i>	Describes how to perform day-to-day administration procedures.

**Table 1-1** Enterprise Vault documentation set (*continued*)

Document	Comments
<i>Backup and Recovery</i>	Describes how to implement an effective backup strategy to prevent data loss, and how to provide a means for recovery in the event of a system failure.
<i>Classification</i>	Describes how to assign classification values to the metadata properties of all new and existing archived items. Users of applications such as Enterprise Vault Search and Compliance Accelerator can then use the classification values to filter the items when they conduct searches or reviews.
<i>NSF Migration</i>	Describes how to migrate content from Domino and Notes NSF files into Enterprise Vault archives.
<i>PST Migration</i>	Describes how to migrate content from Outlook PST files into Enterprise Vault archives.
<i>Reporting</i>	Describes how to implement Enterprise Vault Reporting, which provides reports on the status of Enterprise Vault servers, archives, and archived items. If you configure FSA Reporting, additional reports are available for file servers and their volumes.
<i>Utilities</i>	Describes the Enterprise Vault tools and utilities.
<i>PowerShell Cmdlets</i>	Describes how to perform various administrative tasks by running the Enterprise Vault PowerShell cmdlets.
<i>Registry Values</i>	A reference document that lists the registry values with which you can modify many aspects of Enterprise Vault behavior.
Help for Administration Console	The online Help for the Enterprise Vault Administration Console.
Help for Enterprise Vault Operations Manager	The online Help for Enterprise Vault Operations Manager.

For the latest information on supported devices and versions of software, see the *Enterprise Vault Compatibility Charts* book, which is available from this address:

<http://www.veritas.com/docs/000097605>

## “How To” articles on the Veritas Support website

Most of the information in the Enterprise Vault administration guides is also available online as articles on the Veritas Support website. You can access these articles by

searching the Internet with any popular search engine, such as Google, or by following the procedure below.

**To access the “How To” articles on the Veritas Support website**

- 1** Type the following in the address bar of your web browser, and then press **Enter**:  
[http://www.veritas.com/support/en\\_US/products-a-z](http://www.veritas.com/support/en_US/products-a-z)
- 2** In the **Products A-Z** page, choose the required product, such as Enterprise Vault for Microsoft Exchange.
- 3** Search for a word or phrase by using the Knowledge Base Search feature, or browse the list of most popular subjects.

## Enterprise Vault training modules

The Enterprise Vault and eDiscovery Tech Center (<http://www.veritas.com/elibrary>) is an eLibrary of self-paced learning modules developed around key features, best practices, and common technical support questions.

More advanced instructor-led training, virtual training, and on-demand classes are also available. For information about them, see <http://www.veritas.com/education-services/training-courses>.

## Comment on the documentation

Let us know what you like and dislike about the documentation. Were you able to find the information you needed quickly? Was the information clearly presented? Report errors and omissions, or tell us what you would find useful in future versions of our guides and online help.

Please include the following information with your comment:

- The title and product version of the guide on which you want to comment.
- The topic (if relevant) on which you want to comment.
- Your name.

Email your comment to [evdocs@veritas.com](mailto:evdocs@veritas.com). Please only use this address to comment on product documentation.

We appreciate your feedback.

# Accessing the Enterprise Vault cmdlets

This chapter includes the following topics:

- [Enterprise Vault Management Shell](#)
- [Importing modules](#)

## Enterprise Vault Management Shell

Many of Enterprise Vault's PowerShell cmdlets are available as part of the Enterprise Vault Management Shell.

When you run the Enterprise Vault Management Shell, it loads the Enterprise Vault snapin, `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which makes available the cmdlets it contains, in addition to PowerShell's native cmdlets.

In this guide, each cmdlet's reference section indicates whether or not it is provided as part of the Enterprise Vault Management Shell.

## Importing modules

To use the Enterprise Vault PowerShell cmdlets that are not loaded automatically by the Enterprise Vault Management Shell, you must load a separate module. You can do this in an Enterprise Vault Management Shell, or directly in a PowerShell shell.

In this guide, if it is necessary to load a module to use a cmdlet, the cmdlet's reference section tells you which module you must load.

### **To load an Enterprise Vault PowerShell cmdlet module**

- 1** Run PowerShell, or the Enterprise Vault Management Shell.
- 2** At the command prompt, run the `Import-Module` cmdlet and provide the name of the module you want to load.

For example, to load the Monitoring module, use the following command:

```
Import-Module .\Symantec.EnterpriseVault.PowerShell.Monitoring.dll
```

When you have loaded the module, you can use all the Enterprise Vault cmdlets that it contains.

# Help for Enterprise Vault PowerShell cmdlets

This chapter includes the following topics:

- [Getting help for Enterprise Vault PowerShell cmdlets](#)

## Getting help for Enterprise Vault PowerShell cmdlets

Help is available for all Enterprise Vault's PowerShell cmdlets. For example, the following command shows the detailed help for `Clear-VaultStoreBackupMode`:

```
Get-Help Clear-VaultStoreBackupMode -detailed
```

You can also use the `-examples` parameter to output only the examples of cmdlet usage from the help, and the `-full` parameter to output the most detailed help available.

For more information about using PowerShell's `Get-Help` cmdlet, use the following command:

```
Get-Help Get-Help
```

# Enterprise Vault PowerShell cmdlet reference

This chapter includes the following topics:

- [Get-EVArchive](#)
- [Set-EVArchive](#)
- [Get-EVClassificationPolicy](#)
- [New-EVClassificationPolicy](#)
- [Remove-EVClassificationPolicy](#)
- [Set-EVClassificationPolicy](#)
- [Import-EVClassificationRules](#)
- [Publish-EVClassificationRules](#)
- [Get-EVClassificationTags](#)
- [Get-EVClassificationTestMode](#)
- [Set-EVClassificationTestMode](#)
- [Get-EVComputers](#)
- [Get-EVDatabase](#)
- [Get-EVDatabaseDetail](#)
- [Set-EVDatabaseDetail](#)

- [Get-EVDatabaseFileInfo](#)
- [Start-EVDatabaseUpgrade](#)
- [Get-EVDependencyService](#)
- [Get-EVDependencyServiceState](#)
- [Remove-EVExchangeFolderPolicy](#)
- [Get-EVFileLocation](#)
- [Get-EVFSAVolume](#)
- [New-EVFSAVolume](#)
- [Remove-EVFSAVolume](#)
- [Set-EVFSAVolume](#)
- [Get-EVIISWebsite](#)
- [Set-EVIMAPServerDisabled](#)
- [Set-EVIMAPServerEnabled](#)
- [Get-EVIMAPUsers](#)
- [Get-EVIMAPUserSettings](#)
- [Get-EVIndexLocation](#)
- [New-EVMDSBuildTask](#)
- [Get-EVMDSStatus](#)
- [Add-EVPstComputer](#)
- [Add-EVPstFile](#)
- [Get-EVRBARole](#)
- [Add-EVRBARoleMember](#)
- [Get-EVRBARoleMember](#)
- [Remove-EVRBARoleMember](#)
- [Get-EVRBAAzStoreXml](#)
- [Set-EVRBAAzStoreXml](#)
- [Get-EVRetentionPlan](#)

- [New-EVRetentionPlan](#)
- [Remove-EVRetentionPlan](#)
- [Set-EVRetentionPlan](#)
- [Get-EVServer](#)
- [Get-EVService](#)
- [Get-EVServiceState](#)
- [Get-EVSite](#)
- [Get-EVSMTPHoldingFolder](#)
- [Get-EVSMTPPolicy](#)
- [New-EVSMTPPolicy](#)
- [Remove-EVSMTPPolicy](#)
- [Set-EVSMTPPolicy](#)
- [Get-EVSMTPServerSettings](#)
- [New-EVSMTPServerSettings](#)
- [Set-EVSMTPServerSettings](#)
- [Sync-EVSMTPServerSettings](#)
- [Get-EVSMTPTarget](#)
- [New-EVSMTPTarget](#)
- [Remove-EVSMTPTarget](#)
- [Set-EVSMTPTarget](#)
- [Get-EVStorageDatabase](#)
- [Get-EVTask](#)
- [Get-EVTaskState](#)
- [Get-EVVaultStore](#)
- [Get-EVVaultStorePartition](#)
- [Clear-IndexLocationBackupMode](#)
- [Get-IndexLocationBackupMode](#)

- [Set-IndexLocationBackupMode](#)
- [Set-IndexMetadataSyncLevel](#)
- [Get-IndexServerForIndexLocation](#)
- [Start-PartitionRollover](#)
- [Clear-VaultStoreBackupMode](#)
- [Get-VaultStoreBackupMode](#)
- [Set-VaultStoreBackupMode](#)

## Get-EVArchive

`Get-EVArchive` returns a list of the Enterprise Vault archives in which a specified user has various permissions, either directly or through membership of an Active Directory group. `Get-EVArchive` also returns a list of the archives whose name exactly matches the specified name. You can filter the list by a number of archive properties. For example, you can choose to list only those archives in which the user has permission to delete the archived items, or only those archives that are on legal hold.

`Get-EVArchive` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

### Syntax

```
Get-EVArchive [[-Server] <String>] [[-User] <String>]
[-DeleteProtected [<Boolean>]] [-DeleteExpiredItems [<Boolean>]]
[-OnHold [<Boolean>]] [-Permissions [<String>]] [<CommonParameters>]
```

```
Get-EVArchive [[-Server] <String>] [[-ArchiveName] <String>]
[-DeleteProtected [<Boolean>]] [-DeleteExpiredItems [<Boolean>]]
[-OnHold [<Boolean>]] [<CommonParameters>]
```

### Parameters

**Table 4-1** Get-EVArchive parameters

Parameter	Description
<code>-Server</code> (required)	Identifies the Enterprise Vault server for which to retrieve the archive list. For example: <code>localhost</code> , <code>evserver.domain.com</code> .

**Table 4-1** Get-EVArchive parameters (*continued*)

Parameter	Description
-User (required)	Identifies the user for which to retrieve the archive list. Type the user's Windows logon name, user principal name, or Domino user ID. For example: user@domain.com, domain\user.
-ArchiveName (required)	Specifies the name of an archive for which to retrieve the details.
-DeleteProtected	Filters the archive list according to whether the user can manually delete the items in the archives. Set to <code>\$false</code> to list the archives in which the user can manually delete items, or <code>\$true</code> to list the archives in which the user cannot delete items.
-DeleteExpiredItems	Filters the archive list according to whether Enterprise Vault can automatically delete the archived items whose retention periods have expired. Set to <code>\$false</code> to list the archives in which Enterprise Vault does not delete expired items, or <code>\$true</code> to list the archives in which it does.
-OnHold	Filters the archive list according to the legal hold status of the archives. Set to <code>\$false</code> to list the archives in which either the user or Enterprise Vault (or both) can delete items, or <code>\$true</code> to list the archives that are on legal hold. (Legal hold is equivalent to <code>DeleteProtected</code> set to <code>\$true</code> and <code>DeleteExpiredItems</code> set to <code>\$false</code> .)
-Permissions	Filters the archive list according to the permissions that the user has on the archives. Specify the permissions as a combination of <code>r</code> (for read), <code>w</code> (for write), and <code>d</code> (for delete). For example: <code>rw</code> and <code>rwd</code> . A blank value ( <code></code> ) denotes any permission. The default permission if not specified is <code>d</code> (delete).

## Examples

- `Get-EVArchive MyEVServer Domain\User`

Lists the archives on the Enterprise Vault server `MyEVServer` in which the user `Domain\User` has delete permissions.

- `Get-EVArchive localhost user@domain.com -DeleteProtected:$false -DeleteExpiredItems:$false -Permissions rw`  
 Lists the archives on the local server in which the user `user@domain.com` has both read and write permissions. `Get-EVArchive` filters the list to show only those archives in which the user can manually delete items but in which Enterprise Vault cannot automatically delete expired items.
- `Get-EVArchive localhost Domain\User -OnHold:$false | format-table -auto`  
 Lists the archives on the local server in which the user `Domain\User` has delete permissions. `Get-EVArchive` excludes from the list all the archives that are on legal hold and writes the output to `format-table`.
- `Get-EVArchive -Server localhost -ArchiveName msmith | format-table -auto`  
 Lists the archives on the local server that have the name "msmith".

## Output

[Table 4-2](#) lists the properties that are available.

**Table 4-2** Get-EVArchive properties

Name	Type	Description
Archiveld	String	The Archiveld of an archive to which the specified user has access.
ArchiveName	String	The ArchiveName of an archive to which the specified user has access.
ArchiveType	ArchiveType	The type of archive. Possible values: "Exchange", "Internet Mail".
DeleteExpiredItems	Boolean	Indicates whether Enterprise Vault can automatically delete items from the archive. Possible values are <code>\$true</code> (can delete), <code>\$false</code> (cannot delete).
DeleteProtected	Boolean	Indicates whether the user can manually delete items from the archive.

**Table 4-2** Get-EVArchive properties (*continued*)

Name	Type	Description
OnHold	Boolean	Indicates whether the archive contains items that are on legal hold. Possible values are <code>\$true</code> (archive does contain items on legal hold), <code>\$false</code> (archive does not contain items on legal hold).
Status	String	Indicates the status of the archive. Can be any of the following: "Available", "Unavailable", "Closed", "Being created", "Being deleted".
RetentionPlanName	String	The name of the retention plan that has been applied to the archive.

### Related cmdlets

- See "[Set-EVArchive](#)" on page 23.

## Set-EVArchive

`Set-EVArchive` sets a number of properties of the nominated archive, including whether users can manually delete the items in the archive and whether Enterprise Vault can automatically delete the archived items when their retention period has expired. You can also specify a description and administrative note for the archive for display in the Administration Console.

`Set-EVArchive` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

### Syntax

```
Set-EVArchive [[-ArchiveID] <String>] [-DeleteProtected [<Boolean>]]  
[-DeleteExpiredItems [<Boolean>]] [-Description [<String>]]  
[-AdminNote [<String>]] [-RetentionPlanName [<String>]]  
[<CommonParameters>]
```

## Parameters

**Table 4-3** Set-EVArchive parameters

Parameter	Description
-ArchiveID (required)	Specifies the ID of the archive for which to set the properties. You can obtain the required ID with the <code>Get-EVArchive</code> cmdlet.
-DeleteProtected	Specifies whether to permit users manually to delete the items in the archive. Set to <code>\$false</code> to permit users to delete items, or <code>\$true</code> to prevent deletion.
-DeleteExpiredItem	Specifies whether to permit Enterprise Vault automatically to delete the archived items when their retention period has expired. Set to <code>\$false</code> to prevent deletion, or <code>\$true</code> to permit Enterprise Vault to delete the expired items.
-Description	Provides a short description for display in the archive properties in the Administration Console.
-AdminNote	Provides an administrative note for display in the archive properties in the Administration Console.
-RetentionPlanName	Specifies the name of a retention plan to apply to the archive.  Note that when the archiving task next performs mailbox synchronization, it may overwrite the specified retention plan with the retention settings of the appropriate provisioning group. For an Exchange mailbox archive, this is only the case if you have unchecked the <b>Overall lock</b> option in the Exchange mailbox policy settings.

## Examples

- `Set-EVArchive 14B...EVSERVER.LOCAL -DeleteProtected:$true -DeleteExpiredItems:$false`

Stops both users and Enterprise Vault from deleting the items in the nominated archive, and so places it on legal hold.

- `Set-EVArchive 14B...EVSERVER.LOCAL -DeleteProtected:$false -AdminNote "Item deletion allowed $(get-date) by $(gc env:username) "`  
 Allows users manually to delete the items in the nominated archive, and modifies its administrative note.
- `Get-EVArchive localhost Domain\User -OnHold:$false | Set-EVArchive -DeleteProtected:$true -DeleteExpiredItems:$false -Description "Archive placed on legal hold $(get-date) by $(gc env:username) from $(gc env:computername) "`  
 With `Get-EVArchive`, generates a list of the archives on the local Enterprise Vault server which are not on legal hold and in which the user `Domain\User` has delete permissions. `Get-EVArchive` passes the list to `Set-EVArchive`, which places the archives on hold and sets the archive description accordingly.
- `Set-EVArchive 14B...EVSERVER.LOCAL -RetentionPlanName RPFForManagers`  
 Applies the retention plan "RPFForManagers" to the nominated archive.
- `Set-EVArchive 14B...EVSERVER.LOCAL -RetentionPlanName ""`  
 Removes any retention plan that was previously applied to the nominated archive.

## Output

Table 4-4 lists the properties that are available.

**Table 4-4** Set-EVArchive properties

Name	Type	Description
ArchivedId	String	The ArchivedId of the archive to modify.
ArchiveName	String	The name of the archive to modify.
Updated	Boolean	Indicates whether the archive was updated by the command. Possible values are: <code>\$true</code> (archive was updated), <code>\$false</code> (archive was not updated).

## Related cmdlets

- See [“Get-EVArchive”](#) on page 20.

# Get-EVClassificationPolicy

`Get-EVClassificationPolicy` returns a list of all the classification policies that are configured in an Enterprise Vault site. You can also return the properties of a specific classification policy using the `-Name` parameter.

`Get-EVClassificationPolicy` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVClassificationPolicy [[-SiteId] <String>] [[-Name] <String>]  
[<CommonParameters>]
```

## Parameters

**Table 4-5** Get-EVClassificationPolicy parameters

Parameter	Description
<code>-SiteId</code>	The ID of the Enterprise Vault site for which to return the classification policy details. If you omit this parameter, and the command cannot determine the ID by looking in the registry, then <code>Get-EVClassificationPolicy</code> prompts you to enter the required ID.  You can use <code>Get-EVSite</code> to obtain the site ID.
<code>-Name</code>	The name of a specific classification policy whose properties you want to return.

## Examples

- `Get-EVClassificationPolicy`  
Returns a list of all the classification policies that are configured in the Enterprise Vault site. As no site ID is specified, the command first looks for it in the registry and then, if it cannot find the ID there, prompts you for it.
- `Get-EVClassificationPolicy -SiteId 13E...EV.example.com`  
Returns a list of all the classification policies that are configured in the specified Enterprise Vault site.
- `Get-EVClassificationPolicy -SiteId 13E...EV.example.com -Name "Classification policy"`  
Returns the properties of the classification policy that is named "Classification policy". For example:

```
Name           : Classification policy
EntryId        : 125...EV.example.com
IsADefaultPolicy : True
DuringIndexing : True
DetermineRC    : True
RCDuringDeletion : True
RCDuringExpiry : True
Description    : Classification policy
SiteId         : 13E...EV.example.com
```

- You can use the following sequence of commands to add the classification policy called "Classification policy" into the variable \$sp, and then rename it.
  - `$sp = Get-EVClassificationPolicy -SiteId 13E...EV.example.com -Name "Classification policy"`
  - `Set-EVClassificationPolicy -AdminObject $sp -name "Renamed Classification policy"`

## Output

This command returns an object of type `Symantec.EnterpriseVault.Admin.ClassificationPolicy`, which has the following properties.

**Table 4-6** Get-EVClassificationPolicy properties

Name	Type	Description
Name	String	Name of the classification policy.
EntryId	String	Provides the directory entry ID of the classification policy.
IsADefaultPolicy	Boolean	True if the classification policy is a default policy; false otherwise.
DuringIndexing	Boolean	Whether to classify items during indexing, and reclassify them during an index rebuild.
DetermineRC	Boolean	Whether classification is used to determine the retention category.
RCDuringDeletion	Boolean	If DetermineRC is \$True then whether items are classified during user deletion.
RCDuringExpiry	Boolean	If DetermineRC is \$True then whether items are classified during automatic expiry.

**Table 4-6** Get-EVClassificationPolicy properties (*continued*)

Name	Type	Description
Description	String	Description of the classification policy.
SiteId	String	The site ID to which the classification policy belongs.
Identity	Number	The identity number of the classification policy.

### Related cmdlets

- See [“New-EVClassificationPolicy”](#) on page 28.
- See [“Remove-EVClassificationPolicy”](#) on page 31.
- See [“Set-EVClassificationPolicy”](#) on page 32.

## New-EVClassificationPolicy

`New-EVClassificationPolicy` creates a classification policy for an Enterprise Vault site.

`New-EVClassificationPolicy` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

### Syntax

```
New-EVClassificationPolicy [[-SiteId] <String>] [-Name] <String>
[-Description <string>] [-DuringIndexing <Boolean>] [-DetermineRC
<Boolean>] [-RCDuringDeletion <Boolean>] [-RCDuringExpiry <Boolean>]
[<CommonParameters>]
```

### Parameters

**Table 4-7** New-EVClassificationPolicy parameters

Parameter	Description
<code>-SiteId</code>	The ID of the Enterprise Vault site for which to create the classification policy. If you omit this parameter, and the command cannot determine the ID by looking in the registry, then <code>New-EVClassificationPolicy</code> prompts you to enter the required ID.  You can use <code>Get-EVSite</code> to obtain the site ID.

**Table 4-7** New-EVClassificationPolicy parameters (*continued*)

Parameter	Description
-Name (required)	The name of the classification policy. The name must be unique, and it can contain up to 40 alphanumeric or space characters.
-Description	The description to set for the classification policy. The description can contain up to 127 alphanumeric, space, or special characters.
-DuringIndexing	Specifies whether Enterprise Vault should classify items at the point that it indexes them ( <code>\$true</code> ) or not ( <code>\$false</code> ). The default is <code>\$true</code> .  This setting also determines whether Enterprise Vault reclassifies items when you rebuild the indexes.
-DetermineRC	Specifies whether to allow the classification feature to update the retention categories of items ( <code>\$true</code> ) or not ( <code>\$false</code> ). The default is <code>\$true</code> .
-RCDuringDeletion	When <code>DetermineRC</code> is <code>\$true</code> , specifies whether to enable classification on user deletion ( <code>\$true</code> ) or not ( <code>\$false</code> ). The default is <code>\$false</code> .  You cannot set <code>RCDuringDeletion</code> to <code>\$true</code> when <code>DetermineRC</code> is set to <code>\$false</code> .
-RCDuringExpiry	When <code>DetermineRC</code> is <code>\$true</code> , specifies whether to enable classification on automatic expiry ( <code>\$true</code> ) or not ( <code>\$false</code> ). The default is <code>\$false</code> .  Note the following: <ul style="list-style-type: none"> <li>You cannot set <code>RCDuringExpiry</code> to <code>\$true</code> when <code>DetermineRC</code> is set to <code>\$false</code>.</li> <li>You must set <code>RCDuringExpiry</code> to <code>\$true</code> when <code>DuringIndexing</code> is <code>\$false</code> and <code>DetermineRC</code> is <code>\$true</code>.</li> </ul>

## Examples

- ```
New-EVClassificationPolicy -SiteId 13E...EV.example.com -Name "Classification policy" -Description "Classification policy created using PowerShell"
```

Creates a classification policy that is named "Classification policy" in the specified Enterprise Vault site. The new policy has the description "Classification policy created using PowerShell".
- ```
New-EVClassificationPolicy -Name "Classification policy" -DuringIndexing $true -DetermineRC $false
```

Creates a classification policy that is named "Classification policy". This policy does classify items during indexing but does not use classification to determine their retention categories.

## Output

This command returns an object of type `Symantec.EnterpriseVault.Admin.ClassificationPolicy`, which has the following properties.

**Table 4-8** New-EVClassificationPolicy properties

Name	Type	Description
Name	String	Name of the classification policy.
EntryId	String	Provides the directory entry ID of the classification policy.
IsADefaultPolicy	Boolean	True if the classification policy is a default policy; false otherwise.
DuringIndexing	Boolean	Whether to classify items during indexing, and reclassify them during an index rebuild.
DetermineRC	Boolean	Whether classification is used to determine the retention category.
RCDuringDeletion	Boolean	If DetermineRC is \$True then whether items are classified during user deletion.
RCDuringExpiry	Boolean	If DetermineRC is \$True then whether items are classified during automatic expiry.
Description	String	Description of the classification policy.
SiteId	String	The site ID to which the classification policy belongs.
Identity	Number	The identity number of the classification policy.

## Related cmdlets

- See [“Get-EVClassificationPolicy”](#) on page 26.
- See [“Remove-EVClassificationPolicy”](#) on page 31.
- See [“Set-EVClassificationPolicy”](#) on page 32.

# Remove-EVClassificationPolicy

`Remove-EVClassificationPolicy` removes the specified classification policy, if it is not in use. The command prompts you to confirm the removal of the classification policy.

`Remove-EVClassificationPolicy` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Remove-EVClassificationPolicy [[-SiteId] <String>] [-Name] <String>
[<CommonParameters>]
```

## Parameters

**Table 4-9** Remove-EVClassificationPolicy parameters

Parameter	Description
<code>-SiteId</code>	The ID of the Enterprise Vault site to which the classification policy belongs. If you omit this parameter, and the command cannot determine the ID by looking in the registry, then <code>Remove-EVClassificationPolicy</code> prompts you to enter the required ID.  You can use <code>Get-EVSite</code> to obtain the site ID.
<code>-Name (required)</code>	The name of the classification policy to remove.

## Examples

- `Remove-EVClassificationPolicy -SiteId 13E...EV.example.com -Name "Classification policy"`  
Removes the classification policy that is named "Classification policy" from the specified Enterprise Vault site.
- You can use the following sequence of commands to add the classification policy called "Classification policy" into the variable `$sp`, and then delete it.
  - `$sp = Get-EVClassificationPolicy -SiteId 13E...EV.example.com -Name "Classification policy"`
  - `Remove-EVClassificationPolicy -AdminObject $sp`

## Output

None.

## Related cmdlets

- See “[Get-EVClassificationPolicy](#)” on page 26.
- See “[New-EVClassificationPolicy](#)” on page 28.
- See “[Set-EVClassificationPolicy](#)” on page 32.

# Set-EVClassificationPolicy

`Set-EVClassificationPolicy` sets or updates the properties of an existing classification policy.

`Set-EVClassificationPolicy` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-EVClassificationPolicy [[-SiteId] <String>] [-Name] <String>
[-Description <string>] [-DuringIndexing <Boolean>] [-DetermineRC
<Boolean>] [-RCDuringDeletion <Boolean>] [-RCDuringExpiry <Boolean>]
[<CommonParameters>]
```

## Parameters

**Table 4-10** Set-EVClassificationPolicy parameters

Parameter	Description
-SiteId	The ID of the Enterprise Vault site for which to set or update the classification policy details. If you omit this parameter, and the command cannot determine the ID by looking in the registry, then <code>Set-EVClassificationPolicy</code> prompts you to enter the required ID.  You can use <code>Get-EVSite</code> to obtain the site ID.
-Name (required)	The name of a specific classification policy whose properties you want to set or update. If you want to rename the policy then the new name must be unique, and it can contain up to 40 alphanumeric or space characters.
-Description	The description to set for the classification policy. The description can contain up to 127 alphanumeric, space, or special characters.

**Table 4-10** Set-EVClassificationPolicy parameters (*continued*)

Parameter	Description
-DuringIndexing	Specifies whether Enterprise Vault should classify items at the point that it indexes them ( <code>\$true</code> ) or not ( <code>\$false</code> ). The default is <code>\$true</code> .  This setting also determines whether Enterprise Vault reclassifies items when you rebuild the indexes.
-DetermineRC	Specifies whether to allow the classification feature to update the retention categories of items ( <code>\$true</code> ) or not ( <code>\$false</code> ). The default is <code>\$true</code> .
-RCDuringDeletion	When <code>DetermineRC</code> is <code>\$true</code> , specifies whether to enable classification on user deletion ( <code>\$true</code> ) or not ( <code>\$false</code> ). The default is <code>\$false</code> .  You cannot set <code>RCDuringDeletion</code> to <code>\$true</code> when <code>DetermineRC</code> is set to <code>\$false</code> .
-RCDuringExpiry	When <code>DetermineRC</code> is <code>\$true</code> , specifies whether to enable classification on automatic expiry ( <code>\$true</code> ) or not ( <code>\$false</code> ). The default is <code>\$false</code> .  Note the following: <ul style="list-style-type: none"> <li>■ You cannot set <code>RCDuringExpiry</code> to <code>\$true</code> when <code>DetermineRC</code> is set to <code>\$false</code>.</li> <li>■ You must set <code>RCDuringExpiry</code> to <code>\$true</code> when <code>DuringIndexing</code> is <code>\$false</code> and <code>DetermineRC</code> is <code>\$true</code>.</li> </ul>

## Examples

- `Set-EVClassificationPolicy -SiteId 13E...EV.example.com -Name "Classification policy" -Description "Classification example policy"`  
Updates the description of an existing classification policy that is named "Classification policy" in the specified Enterprise Vault site.
- You can use the following sequence of commands to add the classification policy called "Classification policy" into the variable `$sp`, and then rename it.
  - `$sp = Get-EVClassificationPolicy -SiteId 13E...EV.example.com -Name "Classification policy"`
  - `Set-EVClassificationPolicy -AdminObject $sp -name "Renamed Classification policy"`

## Output

There is a confirmation message on completion.

## Related cmdlets

- See “[Get-EVClassificationPolicy](#)” on page 26.
- See “[New-EVClassificationPolicy](#)” on page 28.
- See “[Remove-EVClassificationPolicy](#)” on page 31.

# Import-EVClassificationRules

`Import-EVClassificationRules` imports all the Enterprise Vault classification properties and rules from a file into the target servers. Before the command does this, it clears any existing properties and rules from those servers.

This command performs a similar function to the `Publish-EVClassificationRules` command. The difference is that `Publish-EVClassificationRules` first exports the classification properties and rules to an XML file before it publishes them to the target servers; `Import-EVClassificationRules` uses an existing XML file to perform the import part of the operation only.

Note the following:

- To run this command, you must have the system administrator role on both the server where you run the command and on all the target servers.
- Install the Microsoft Data Classification Toolkit on the server where you run this command. You can download the toolkit from the following webpage: <http://www.microsoft.com/download/details.aspx?id=27123>
- If you do not run the command on an Enterprise Vault server, you must specify either the `-SiteId` or `-Servers` parameter. If you run the command on an Enterprise Vault server and omit these parameters, the command uses the site of the current server to publish to all the other Enterprise Vault servers in the site.
- In a cluster configuration (either Windows Server Failover Clustering or Veritas Cluster Server), if you import the classification properties and rules into one cluster node then all the other nodes are also updated. So, after a failover to another node, classification continues with the same rules as before.
- In an Enterprise Vault Building Blocks environment, this command imports only to servers that are currently hosting Enterprise Vault tasks and services.

- This command stops the Enterprise Vault Storage service on each target server and then, after it has imported the classification properties and rules, restarts the service.

---

**Note:** Other methods for publishing the classification properties and rules do not automatically stop and then restart the Storage service, and this can lead to classification errors. For example, this is the case if you use the PowerShell cmdlets that come with the Microsoft Data Classification Toolkit. Therefore, we strongly recommend that you use `Import-EVClassificationRules` (or `Publish-EVClassificationRules`) to publish the classification properties and rules.

---

`Import-EVClassificationRules` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Import-EVClassificationRules [-ImportRulesFile <string>] [-SiteId <string>] [-Servers <string>] [-TimeoutSecs <integer>] [-Confirm <Boolean>]
```

## Parameters

**Table 4-11** Import-EVClassificationRules parameters

Parameter	Description
<code>-ImportRulesFile</code> (required)	Specifies the file from which the command imports the classification properties and rules. The file must have a <code>.xml</code> file name extension.
<code>-SiteId</code>	Identifies the site to which you want to publish the classification properties and rules.  If you set this parameter, you cannot set the <code>-Servers</code> parameter as well.
<code>-Servers</code>	Nominates the servers that will receive the set of classification properties and rules. Type the NETBIOS name, IP address, or fully-qualified domain name of each server in a comma-separated list. To specify the local computer, type the computer name "localhost".  If you set this parameter, you cannot set the <code>-SiteId</code> parameter as well.

**Table 4-11** Import-EVClassificationRules parameters (*continued*)

Parameter	Description
-TimeoutSecs	Sets the timeout value in seconds when the command stops or starts the Enterprise Vault Storage service on each of the target servers. The default is 300 seconds.  <b>Note:</b> If the command fails to restart a service within the specified period, check the state of the classification rules and Storage services on the failed servers. A server can be left without classification rules if the command clears the existing rules without also importing the new ones.
-Confirm	When set to <code>\$True</code> (the default value), causes the command to prompt you for confirmation before it imports the classification properties and rules. Set to <code>\$False</code> to suppress the prompts.

## Examples

- `Import-EVClassificationRules -ImportRulesFile c:\Data\RulesFile.xml`  
Imports the classification properties and rules that are in the specified file into all the Enterprise Vault servers that are in the current site (that is, the same site as the server on which you run the command).
- `Import-EVClassificationRules -ImportRulesFile c:\Data\RulesFile.xml -SiteId 13E...EV.example.com`  
Imports the classification properties and rules that are in the specified file into all the Enterprise Vault servers that are in the specified site.
- `Import-EVClassificationRules -ImportRulesFile c:\Data\RulesFile.xml -Servers SERVER1,SERVER2.ABC.DEF.COM`  
Imports the classification properties and rules that are in the specified file into all the specified servers.

## Output

This command returns the objects of type `Symantec.EnterpriseVault.PowerShell.Commands.ServerInfo`, which has the following default properties.

**Table 4-12** Import-EVClassificationRules properties

Name	Type	Description
ServerName	String	The name of the Enterprise Vault server.

**Table 4-12** Import-EVClassificationRules properties (*continued*)

Name	Type	Description
ServerFQDN	String	The fully qualified domain name of the Enterprise Vault server.
Result	String	The import result (Succeeded/Failed/DuplicateServer).
ErrorMessage	String	The error reason, if the import to the server was not successful.

### Related cmdlets

- See “[Publish-EVClassificationRules](#)” on page 37.

## Publish-EVClassificationRules

`Publish-EVClassificationRules` exports all the Enterprise Vault classification properties and rules from a nominated server to an XML file in the specified location. Enterprise Vault then uses this file to import the classification properties and rules into the target servers. Before the command does this, it clears any existing properties and rules from those servers.

This command performs a similar function to the `Import-EVClassificationRules` command. However, `Import-EVClassificationRules` does not create the XML file that Enterprise Vault subsequently imports into the target servers; the command uses an existing XML file to perform the import part of the operation only.

Note the following:

- You can run the command on a different server from the server on which you have configured the classification properties and rules.
- To run this command, you must have the system administrator role on both the server where you run the command and on all the target servers.
- You must install the Microsoft Data Classification Toolkit on the computer where you run this command. You can download the toolkit from the following webpage: <http://www.microsoft.com/download/details.aspx?id=27123>
- If you do not run the command on an Enterprise Vault server, you must specify either the `-SiteId` or `-Servers` parameter. If you run the command on an Enterprise Vault server and omit these parameters, the command uses the site of the current server to publish to all the other Enterprise Vault servers in the site.

- In a cluster configuration (either Windows Server Failover Clustering or Veritas Cluster Server), if you publish the classification properties and rules to one cluster node then all the other nodes are also updated. So, after a failover to another node, classification continues with the same rules as before.
- In an Enterprise Vault Building Blocks environment, this command imports only to servers that are currently hosting Enterprise Vault tasks and services.
- This command stops the Enterprise Vault Storage service on each target server and then, after it has imported the classification properties and rules, restarts the service.

---

**Note:** Other methods for publishing the classification properties and rules do not automatically stop and then restart the Storage service, and this can lead to classification errors. For example, this is the case if you use the PowerShell cmdlets that come with the Microsoft Data Classification Toolkit. Therefore, we strongly recommend that you use `Publish-EVClassificationRules` (or `Import-EVClassificationRules`) to publish the classification properties and rules.

---

`Publish-EVClassificationRules` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Publish-EVClassificationRules [-StagingServer <string>]
[-ExportRulesFile <string>] [-SiteId <string>] [-Servers <string>]
[-TimeoutSecs <integer>] [-Confirm <Boolean>]
```

## Parameters

**Table 4-13** Publish-EVClassificationRules parameters

Parameter	Description
<code>-StagingServer</code> (required)	Specifies the name of the server on which you have configured the classification properties and rules and from which you now want to export them.
<code>-ExportRulesFile</code> (required)	Specifies the path to a file to which the command exports the classification properties and rules, before importing them into the target servers. The command creates the file locally, so you must specify a local path such as <code>c:\Data\RulesFile.xml</code> . Ensure that the file name has a <code>.xml</code> extension.

**Table 4-13** Publish-EVClassificationRules parameters (*continued*)

Parameter	Description
-SiteId	Identifies the Enterprise Vault site to which you want to publish the classification properties and rules.  If you set this parameter, you cannot set the -Servers parameter as well.
-Servers	Nominates the servers that will receive the set of classification properties and rules. Type the NETBIOS name, IP address, or fully-qualified domain name of one or more servers in a comma-separated list. To specify the local computer, type the computer name "localhost".  If you set this parameter, you cannot set the -SiteId parameter as well.
-TimeoutSecs	Sets the timeout value in seconds when stopping or starting the Enterprise Vault Storage service on each of the target servers. The default is 300 seconds.  <b>Note:</b> If the command fails to restart a service within the specified period, check the state of the classification rules and Storage services on the failed servers. A server can be left without classification rules if the command clears the existing rules without also importing the new ones.
-Confirm	When set to \$True (the default value), causes the command to prompt you for confirmation before it publishes the classification data. Set to \$False to suppress the prompts.

## Examples

- `Publish-EVClassificationRules -StagingServer SERVERXYZ -ExportRulesFile c:\Data\RulesFile.xml`  
Exports the classification properties and rules from server SERVERXYZ to the specified local file. The command then publishes the properties and rules to all the Enterprise Vault servers that are in the current site (that is, the same site as the server on which you run the command).
- `Publish-EVClassificationRules -StagingServer SERVERXYZ -ExportRulesPath c:\Data\RulesFile.xml -SiteId 13E...EV.example.com`  
Publishes the exported classification properties and rules to all the Enterprise Vault servers that are in the specified site.

- `Publish-EVClassificationRules -StagingServer SERVERXYZ -ExportRulesPath c:\Data\RulesFile.xml -Servers SERVER1,SERVER2.ABC.DEF.COM`  
Publishes the exported classification properties and rules to the specified servers.

## Output

This command returns the objects of type `Symantec.EnterpriseVault.PowerShell.Commands.ServerInfo`, which has the following default properties.

**Table 4-14** Publish-EVClassificationRules properties

Name	Type	Description
ServerName	String	The name of the Enterprise Vault server.
ServerFQDN	String	The fully qualified domain name of the Enterprise Vault server.
Result	String	The publish result (Succeeded/Failed/DuplicateServer).
ErrorMessage	String	The error reason, if the import to the server was not successful.

## Related cmdlets

- See [“Import-EVClassificationRules”](#) on page 34.

# Get-EVClassificationTags

For the specified plain-text (.txt) file in the classification cache folder, `Get-EVClassificationTags` returns details of the rules that it matches and the associated classification properties and property values. You may find this useful when you create a classification rule, as the command lets you verify that the rule is operating correctly without repeatedly having to rearchive the same item.

By default, Enterprise Vault empties the cache folder at the first opportunity. However, you can configure it to retain the cache contents by choosing a setting in the Administration Console.

`Get-EVClassificationTags` is provided by

`Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVClassificationTags [-File] <string>
```

## Parameters

**Table 4-15** Get-EVClassificationTags parameters

Parameter	Description
-File (required)	The path to the plain-text file for which to return the classification details. Only .txt files are eligible. File names that contain a dollar sign (\$) must be escaped using single quotation marks (').

## Examples

- `Get-EVClassificationTags -File E:\EVCache\Classification\ClassificationFile.txt`  
Returns the classification details for the file `ClassificationFile.txt`.
- `Get-EVClassificationTags -File 'E:\EVCache\Classification\EV$90B2291D1E3417B67AB88BDDC2195091~02B5EDB8.txt'`  
Returns the classification details for a file that has a dollar sign in its name. For this reason, the entire path is enclosed in single quotation marks.

## Output

This command returns an array of objects of type `Symantec.EnterpriseVault.PowerShell.Commands.EVClassificationProperty`, which have the following properties.

**Table 4-16** Get-EVClassificationTags properties

Name	Type	Description
RuleName	String	The classification rule that the file has matched.
PropertyName	String	The classification property in which the rule assigns one or more values.
PropertyValue	String	The values that the rule assigns to the classification property.

# Get-EVClassificationTestMode

`Get-EVClassificationTestMode` reports on whether the Enterprise Vault classification feature is operating in test mode in the nominated archive. In test

mode, the classification feature generates a report that lists the planned changes instead of applying classification tags and other changes to the items in the archive.

`Get-EVClassificationTestMode` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVClassificationTestMode [-ArchiveID] <string>
```

## Parameters

**Table 4-17** Get-EVClassificationTestMode parameters

Parameter	Description
<code>-ArchiveID</code> (required)	Specifies the ID of the archive for which to get the status of classification test mode.

## Examples

- `Get-EVClassificationTestMode -ArchiveID 19D...EVServer1`  
Gets the current status of classification test mode for the specified archive.

## Output

[Table 4-18](#) lists the properties that are available.

**Table 4-18** Get-EVClassificationTestMode properties

Name	Type	Description
<code>ArchiveID</code>	String	The ID of the archive for which to get the test mode status.
<code>ArchiveName</code>	String	The name of the archive for which to get the test mode status.
<code>TestMode</code>	Boolean	The current status of classification test mode for the archive: enabled ( <code>\$true</code> ) or disabled ( <code>\$false</code> ).

## Related cmdlets

- See [“Set-EVClassificationTestMode”](#) on page 43.

# Set-EVClassificationTestMode

`Set-EVClassificationTestMode` specifies whether the Enterprise Vault classification feature should operate in test mode in the nominated archive. In test mode, the classification feature generates a report that lists the planned changes instead of applying classification tags and other changes to the items in the archive. You can then run `Get-EVClassificationTestMode` on the same archive to check that the outcome is satisfactory.

`Set-EVClassificationTestMode` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-EVClassificationTestMode [-ArchiveID] <string> [-Enabled  
<Boolean>]
```

## Parameters

**Table 4-19** Set-EVClassificationTestMode parameters

Parameter	Description
<code>-ArchiveID</code> (required)	Specifies the ID of the archive for which to set the test mode status.
<code>-Enabled</code> (required)	Specifies whether to enable classification test mode for the archive ( <code>\$true</code> ) or disable it ( <code>\$false</code> ).

## Examples

- `Set-EVClassificationTestMode -ArchiveID 1E...EVServer1 -Enabled $true`  
Specifies that the classification feature should operate in test mode in the nominated archive.

## Output

Returns an exception in the event of failure but otherwise provides no output.

## Related cmdlets

- See [“Get-EVClassificationTestMode”](#) on page 41.

# Get-EVComputers

`Get-EVComputers` returns the collection of computers listed in the `ComputerEntry` table of the Directory database.

`Get-EVComputers` is provided by `Symantec.EnterpriseVault.PowerShell.IMAP.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVComputers [<CommonParameters>]
```

## Parameters

PowerShell common parameters only.

## Examples

- `Get-EVComputers`  
Returns the collection of computers listed in the `ComputerEntry` table of the Directory database.
- `Get-EVComputers | Where-Object {$_.EnabledForIMAP -eq 1} | ForEach-Object {Set-EVIMAPServerDisabled $_.ComputerNameAlternate}`  
This command finds all Enterprise Vault servers that are enabled for IMAP and disables each one in turn, stopping the IMAP server in the process.

## Output

[Table 4-20](#) lists the properties that are available.

**Table 4-20** Get\_EVComputers properties

Name	Type	Description
ComputerIdentity	Int32	An Enterprise Vault unique identifier for this server.
ComputerName	String	The computer name.
ComputerNameAlternate	String	The computer alternate name.
EnabledForIMAP	Boolean	Indicates whether the server has been enabled for IMAP. Possible values <code>\$true</code> (server is enabled for IMAP), <code>\$false</code> (server is not enabled for IMAP), <code>null</code> (server has never been enabled for IMAP).

**Table 4-20** Get\_EVComputers properties (*continued*)

Name	Type	Description
IMAPEndPointEntryId	String	The IMAP Endpoint Entry Id.
IMAPEndpointName	String	The IMAP Endpoint name.

## Related cmdlets

- See “[Set-EVIMAPServerDisabled](#)” on page 68.
- See “[Set-EVIMAPServerEnabled](#)” on page 69.
- See “[Get-EVIMAPUsers](#)” on page 70.
- See “[Get-EVIMAPUserSettings](#)” on page 72.

# Get-EVDatabase

`Get-EVDatabase` returns Enterprise Vault Directory, monitoring, and auditing database details for an Enterprise Vault server.

`Get-EVDatabase` is provided by

`Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

## Syntax

`Get-EVDatabase`

## Parameters

PowerShell common parameters only.

## Examples

- `Get-EVDatabase`  
Returns Directory, monitoring, and auditing database details.

## Output

`Get-EVDatabase` returns an object of type

`Symantec.EnterpriseVault.PowerShell.Core.Output.Database`, which has the following properties.

**Table 4-21** Get-EVDatabase properties

Name	Type	Description
DBName	String	The name of the Enterprise Vault database. For example, "EnterpriseVaultDirectory".
IsClustered	Boolean	Indicates whether SQL Server is clustered.
SQLInstanceName	String	The name of the SQL Server instance that hosts the Directory database.
SQLServerName	String	The name of the SQL Server computer that hosts the Directory database.
SQLServerVersion	String	The version of SQL Server on the SQL Server computer that hosts the Directory database.
Type	EVDatabaseType	The possible values are "Audit", "Directory", and "Monitoring".

### Related cmdlets

- See "[Get-EVDatabaseDetail](#)" on page 46.
- See "[Get-EVDatabaseFileInfo](#)" on page 50.
- See "[Get-EVStorageDatabase](#)" on page 147.

## Get-EVDatabaseDetail

`Get-EVDatabaseDetail` gets Enterprise Vault directory, monitoring, auditing, reporting, vault store, and fingerprint database details for an Enterprise Vault server.

`Get-EVDatabaseDetail` is provided by `Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

### Syntax

```
Get-EVDatabaseDetail [<CommonParameters>]
```

### Parameters

PowerShell common parameters only.

## Examples

- `Get-EVDatabaseDetail`  
Gets Enterprise Vault directory, monitoring, auditing, reporting, vault store, and fingerprint database details for an Enterprise Vault server.
- `Get-EVDatabaseDetail | Where-Object {$_.Type -eq "Fingerprint" -and ($_.IsInAG -eq 0 -or $_.IsAccessible -eq 0)}`  
Gets all Enterprise Vault fingerprint databases that are either not in an availability group (`IsInAG=0`), or are not accessible (`IsAccessible=0`). This information can be used to get a list of Enterprise Vault databases that need to be added to an availability group.
- `Get-EVDatabaseDetail | Where-Object {$_.IsCollationOK -eq 0}`  
Finds Enterprise Vault databases with collation issues.

## Output

`Get-EVDatabaseDetail` returns an object of type

`Symantec.EnterpriseVault.PowerShell.Core.Output.DatabaseDetail`, which has the following properties.

**Table 4-22** Get-EVDatabaseDetail properties

Name	Type	Description
DBName	String	The name of the Enterprise Vault database. For example, "EnterpriseVaultDirectory".
SQLInstanceName	String	This is the SQL server instance where the database is located.
SQLServerVersion	String	This is the SQL server product version (SERVERPROPERTY('ProductVersion')).
Type	EVDatabaseType	This is the Enterprise Vault database type enumeration. Possible values are Directory (0), vault store (1), fingerprint (2), audit (3), monitoring (4), and reporting (5).
SQLServerName	String	This is the name of the physical SQL server where the database is located. (SERVERPROPERTY('MachineName')).
IsClustered	Boolean	Indicates whether the SQL server is clustered.
IsInAG	Boolean	Indicates whether the database is in an availability group.

**Table 4-22** Get-EVDatabaseDetail properties (*continued*)

Name	Type	Description
IsAccessible	Boolean	Indicates whether the database is currently accessible.
IsCollationOK	Boolean	False indicates an issue with database collation.

### Related cmdlets

- See “[Set-EVDatabaseDetail](#)” on page 48.
- See “[Get-EVDatabase](#)” on page 45.
- See “[Get-EVDatabaseFileInfo](#)” on page 50.
- See “[Get-EVStorageDatabase](#)” on page 147.

## Set-EVDatabaseDetail

`Set-EVDatabaseDetail` updates the SQL server connection information for an Enterprise Vault database to a new SQL server name. You can update to use either a server name, instance name, listener or a virtual network name.

`Set-EVDatabaseDetail` takes the following mandatory input parameters:

- `-ServerName`: This can be an instance, listener, server or virtual network name and must be a valid accessible server.
- `-InputObject`: This is a `DatabaseDetail` object of type `Symantec.EnterpriseVault.PowerShell.Core.Output.DatabaseDetail`.

`Set-EVDatabaseDetail` is provided by

`Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

### Syntax

```
Set-EVDatabaseDetail -ServerName <string> -InputObject <string>  
[<CommonParameters>]
```

## Parameters

**Table 4-23** Set-EVDatabaseDetail parameters

Parameter	Description
-ServerName (required)	The name of the SQL server you are migrating to. This can be a listener, instance, server or virtual network name.
-InputObject (required)	A valid DatabaseDetail object of type Symantec.EnterpriseVault.PowerShell.Core.Output.DatabaseDetail.  See the help for Get-EVDatabaseDetail for a description of the properties of this object.  See <a href="#">“Get-EVDatabaseDetail”</a> on page 46.

## Examples

- ```
Get-EVDatabaseDetail | where {$_.SQLInstanceName -eq 'SVR55\SQL1' } | Set-EVDatabaseDetail -ServerName AGlistener
```

Updates SQL server connection details for all databases on a specific SQL server instance “SVR55\SQL1” to use “AGlistener” listener. This example uses the Get-EVDatabaseDetail cmdlet as the pipeline input to the Set-EVDatabaseDetail cmdlet.
- ```
Get-EVDatabaseDetail | where {$_.type -eq 'directory'} | Set-EVDatabaseDetail -ServerName AGlistener
```

Updates the “SQLServer Name” for the “HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\KVS\Enterprise Vault\Directory\DirectoryService” key on all Enterprise Vault servers to “AGlistener”. This example uses the Get-EVDatabaseDetail cmdlet as the pipeline input to the Set-EVDatabaseDetail cmdlet.
- ```
New-Object -TypeName Symantec.EnterpriseVault.PowerShell.Core.Output.DatabaseDetail("SVR33", "EnterpriseVaultDirectory", 0) | Set-EVDatabaseDetail -ServerName AGlistener
```

Updates the “SQLServer Name” for the “HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\KVS\Enterprise Vault\Directory\DirectoryService” key on all Enterprise Vault servers from “SVR33” to “AGlistener”. In this example we create a new object of type Symantec.EnterpriseVault.PowerShell.Core.Output.DatabaseDetail, setting the SQLInstanceName, DBName and Type mandatory properties as the pipeline input. Use this method for updating connection details if your EnterpriseVaultDirectory database server is no longer accessible.

- `Get-EVDatabaseDetail | where {$_.sqlservername -eq 'sql-svr1' -and $_.SQLInstanceName -ne 'AGlistener' -and $_.IsAccessible -eq 'true' } | Set-EVDatabaseDetail -ServerName AGlistener`

Updates SQL server connection details for all accessible databases on a specific SQL server machine “sql-svr1” that do not currently use a listener to use “AGlistener” listener. This example uses the `Get-EVDatabaseDetail` cmdlet as the pipeline input to the `Set-EVDatabaseDetail` cmdlet.

## Output

None.

## Related cmdlets

- See “[Get-EVDatabaseDetail](#)” on page 46.
- See “[Get-EVDatabase](#)” on page 45.
- See “[Get-EVDatabaseFileInfo](#)” on page 50.
- See “[Get-EVStorageDatabase](#)” on page 147.

# Get-EVDatabaseFileInfo

`Get-EVDatabaseFileInfo` returns database file and disk space information for the Enterprise Vault databases on the current Enterprise Vault server.

`Get-EVDatabaseFileInfo` is provided by `Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

## Syntax

```
Get-EVDatabaseFileInfo [-DatabaseType] <string> [<CommonParameters>]
```

```
Get-EVDatabaseFileInfo [-DatabaseType] <string> [-EntryId] <string>  
[-StorageServiceEntryId] <string> [<CommonParameters>]
```

## Parameters

**Table 4-24** Get-EVDatabaseFileInfo properties

| Parameter                           | Description                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-DatabaseType</code>          | EVDatabaseFileInfo. The type of database for which you want details. You can specify any of the following: <ul style="list-style-type: none"><li>■ Directory. Returns details of the Directory database.</li><li>■ VaultStore. Returns details of Vault Store databases.</li><li>■ Fingerprint. Returns details of fingerprint databases.</li></ul> |
| <code>-EntryID</code>               | To find the values for <code>EntryId</code> you can use the <code>Get-EVStorageDatabase</code> cmdlet.<br><br>See <a href="#">“Get-EVStorageDatabase”</a> on page 147.                                                                                                                                                                              |
| <code>-StorageServiceEntryId</code> | To find the value for <code>StorageServiceEntryId</code> you can use the <code>Get-EVStorageDatabase</code> cmdlet.<br><br>See <a href="#">“Get-EVStorageDatabase”</a> on page 147.                                                                                                                                                                 |

## Examples

- `Get-EVDatabaseFileInfo -DatabaseType Directory`  
Returns database file and disk space information for the Enterprise Vault Directory database.
- `Get-EVDatabaseFileInfo -DatabaseType VaultStore -EntryId 21E...08002B30909D -StorageServiceEntryId 21E...08002B30309D`  
Returns the database file and disk space information for a specific vault store database. The command specifies the Storage service to use to obtain the information.  
  
To find the values for `EntryId` and `StorageServiceEntryId` you can use the `Get-EVStorageDatabase` cmdlet.

## Output

[Table 4-25](#) lists the properties that are available.

**Table 4-25** Get-EVDatabaseFileInfo properties

| Name                      | Type           | Description                                                                                                  |
|---------------------------|----------------|--------------------------------------------------------------------------------------------------------------|
| DatabaseName              | String         | The name of the Enterprise Vault database. For example, "EnterpriseVaultDirectory".                          |
| FileType                  | EVDatabaseType | One of the following: "Log", "Data".                                                                         |
| HoursSinceLastBackup      | Int32          | The number of hours since the file was last backed up.                                                       |
| LastBackupDate            | DateTime       | The date and time when the file was last backed up. This value is null if the file has never been backed up. |
| LogicalFileName           | String         | The logical file name.                                                                                       |
| PhysicalFileName          | String         | The full path to the physical file.                                                                          |
| TotalQuotaGBytesAvailable | Double         | The total space available, in gigabytes. This value can be null.                                             |
| TotalQuotaGBytesFree      | Double         | The amount of unused space, in gigabytes. This value can be null.                                            |
| TotalQuotaGBytesUsed      | Double         | The amount of used space, in gigabytes. This value can be null.                                              |

### Related cmdlets

- See [“Get-EVDatabase”](#) on page 45.
- See [“Get-EVStorageDatabase”](#) on page 147.

## Start-EVDatabaseUpgrade

`Start-EVDatabaseUpgrade` upgrades all the Enterprise Vault databases following an Enterprise Vault upgrade, and returns the status for each database.

You must log in using the vault service account to run this cmdlet.

Stop all Enterprise Vault services before you start the upgrade, and ensure that an upgrade is not already running.

`Start-EVDatabaseUpgrade` is provided by `DBUpgraderPowerShell.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Start-EVDatabaseUpgrade [<CommonParameters>]
```

## Parameters

PowerShell common parameters only.

## Examples

- `Start-EVDatabaseUpgrade`  
Starts an upgrade of Enterprise Vault databases, and returns the status of each.

# Get-EVDependencyService

Gets the dependency services information for all the Enterprise Vault services that are on the current Enterprise Vault server.

`Get-EVDependencyService` is provided by `Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

## Syntax

```
Get-EVDependencyService [<CommonParameters>]
```

## Parameters

PowerShell common parameters only.

## Examples

- `Get-EVDependencyService`  
Gets the dependency information for all the Enterprise Vault services that are on the current Enterprise Vault server.

## Output

[Table 4-26](#) lists the properties that are available.

**Table 4-26** Get-EVDependencyService properties

| Name        | Type   | Description                                                                                       |
|-------------|--------|---------------------------------------------------------------------------------------------------|
| DisplayName | String | The name of the dependency service, such as "Message Queuing" or "World Wide Publishing Service". |

**Table 4-26** Get-EVDependencyService properties (*continued*)

| Name    | Type        | Description                                        |
|---------|-------------|----------------------------------------------------|
| EntryID | String      | This value is always empty.                        |
| Name    | String      | The dependency name, such as "MSMQ" or "W3SVC".    |
| Type    | ServiceType | One of the following: "MSMQService", "IISService". |

### Related cmdlets

- See [“Get-EVDependencyServiceState”](#) on page 54.

## Get-EVDependencyServiceState

Gets the current state for the specified dependency service. The state can be either "Running" or "Stopped".

`Get-EVDependencyServiceState` is provided by

`Symantec.EnterpriseVault.PowerShell.Monitoring.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

### Syntax

```
Get-EVDependencyServiceState [-ServiceName] <String>
```

### Parameters

[Table 4-27](#) lists the properties that are available.

**Table 4-27** Get-EVDependencyServiceState parameters

| Parameter                 | Description                                                                                                                                     |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-ServiceName</code> | The name of the service whose details you require.<br><br>You can use the <code>Get-EVDependencyService</code> cmdlet to get the service names. |

### Examples

- `Get-EVDependencyServiceState msmq`  
Gets the state of the Message Queuing service.

- `Get-EVDependencyServiceState W3SVC`  
Gets the state of the World Wide Web Publishing Service (IIS).

## Output

Table 4-28 lists the properties that are available.

**Table 4-28** Get-EVDependencyServiceState properties

| Name  | Type                    | Description                                                                                                             |
|-------|-------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Value | ServiceControllerStatus | One of the following: "Stopped", "StartPending", "StopPending", "Running", "ContinuePending", "PausePending", "Paused". |

## Related cmdlets

- See ["Get-EVDependencyService"](#) on page 53.

# Remove-EVExchangeFolderPolicy

`Remove-EVExchangeFolderPolicy` removes Enterprise Vault filter settings from folders within a mailbox. The folders affected can be restricted to only include Exchange Managed Folders.

`Remove-EVExchangeFolderPolicy` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Remove-EVExchangeFolderPolicy -PrimarySMTPAddress <String>
[-DomainController <String>] [-DontUseSSL] [-Credential
<PSCredential>] [-ManagedFolders] [-MailServer <String>] [-LogFile
<String>] [-ExchangeWebServicesURL <String>] [<CommonParameters>]
```

## Parameters

**Table 4-29** Remove-EVExchangeFolderPolicy parameters

| Parameter                                   | Description                                                   |
|---------------------------------------------|---------------------------------------------------------------|
| <code>-PrimarySMTPAddress</code> (required) | Specifies the primary SMTP address of the mailbox to be used. |

**Table 4-29** Remove-EVExchangeFolderPolicy parameters (continued)

| Parameter               | Description                                                                                                                                                                                             |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -DomainController       | Specifies the domain controller that the LDAP lookup query should be run against.                                                                                                                       |
| -DontUseSSL             | Specifies that SSL should not be used when connecting to the Exchange Server.                                                                                                                           |
| -Credential             | Specifies the PSCredentials that should be used to connect to the Exchange Server. This should be your Enterprise Vault administrator account.                                                          |
| -ManagedFolders         | Restricts the delete to only folders that have an Exchange Managed Folder policy set against them.                                                                                                      |
| -MailServer             | Overrides the mailbox server value associated with the mailbox and attempts to use the server specified.                                                                                                |
| -LogFile                | Log the output to the file specified instead of to the console.                                                                                                                                         |
| -ExchangeWebServicesURL | Specifies the Exchange Web Services URL in Exchange 2010 environments. Use this parameter if <code>Remove-EVExchangeFolderPolicy</code> fails to determine the Exchange Web Services URL automatically. |

## Examples

- `Remove-EVExchangeFolderPolicy -PrimarySMTPAddress user1@example.com -ManagedFolders`  
This command removes any Enterprise Vault filter settings from Exchange Managed Folders located in the mailbox with the SMTP address of `user1@example.com`.

# Get-EVFileLocation

Gets details of the Enterprise Vault file locations that are configured on the current Enterprise Vault server.

`Get-EVFileLocation` is provided by

`Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

## Syntax

```
Get-EVFileLocation [[-Type] <string>] [<CommonParameters>]
```

## Parameters

**Table 4-30** Get-EVFileLocation parameters

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -Type     | The Enterprise Vault file location for which you want details. You can specify any of the following: <ul style="list-style-type: none"><li>■ Application. Gets the path to the Enterprise Vault program folder.</li><li>■ Cache. Gets the path to the Enterprise Vault cache location.</li><li>■ Temporary. Gets the path to the folder that Enterprise Vault uses for temporary files.</li></ul> |

## Examples

- `Get-EVFileLocation`  
Gets details of the Enterprise Vault file locations that are configured on the current server.

## Output

[Table 4-31](#) lists the properties that are available.

**Table 4-31** Get-EVFileLocation properties

| Name                      | Type         | Description                                                |
|---------------------------|--------------|------------------------------------------------------------|
| Path                      | String       | Path to the folder location.                               |
| TotalQuotaGBytesAvailable | Double       | The total available space at the location, in gigabytes.   |
| TotalQuotaGBytesFree      | Double       | The free space at the location, in gigabytes.              |
| TotalQuotaGBytesUsed      | Double       | The space used at the location, in gigabytes.              |
| Type                      | FileLocation | One of the following: "Application"; "Cache"; "Temporary". |

# Get-EVFSAVolume

`Get-EVFSAVolume` retrieves details of all the volumes set up for File System Archiving in an Enterprise Vault site. This list can be filtered based on a file server and a volume name.

`Get-EVFSAVolume` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVFSAVolume [-SiteId <string>] [<CommonParameters>]
```

```
Get-EVFSAVolume [-SiteId <string>] [-FileName] <string> [[-Name] <string>] [<CommonParameters>]
```

```
Get-EVFSAVolume [-SiteId <string>] -FileServerId <string> [[-Name] <string>] [<CommonParameters>]
```

## Parameters

**Table 4-32** Get-EVFSAVolume parameters

| Parameter                             | Description                                                                                                                                                                                                                                                                             |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-SiteId</code> (required)       | The ID of the Enterprise Vault site under which the file server is configured. If you omit this parameter, <code>Get-EVFSAVolume</code> uses the ID of the site of the Enterprise Vault Server where the command is running. You can use <code>Get-EVSite</code> to obtain the site ID. |
| <code>-FileName</code> (required)     | The fully-qualified DNS name or the alias of the file server target configured under the Enterprise Vault site, which hosts the volume share.                                                                                                                                           |
| <code>-FileServerId</code> (required) | The ID of the file server that hosts the volume share.                                                                                                                                                                                                                                  |
| <code>-Name</code>                    | The name of a share folder added as an FSA volume.                                                                                                                                                                                                                                      |

## Examples

- `Get-EVFSAVolume -SiteId 198...example.com`  
Retrieves all the volumes hosted on all the file servers in the specified Enterprise Vault site.
- `Get-EVFSAVolume -SiteId 198...example.com -FileServerId 1FB...example.com`  
Retrieves all the volumes hosted on the file server with ID '1FB...example.com'.

- `Get-EVFSAVolume -SiteId 198...example.com -FileServerId 1FB...example.com -Name Sales`  
Retrieves the properties of the volume 'Sales' hosted on the file server with ID '1FB...example.com'.
- `Get-EVFSAVolume -SiteId 198...example.com -FileServerName 'WindowsFiler.example.com'`  
Retrieves all the volumes hosted on the file server with the name 'WindowsFiler.example.com'.
- `Get-EVFSAVolume -SiteId 198...example.com -FileServerName \\WindowsFiler`  
Retrieves all the volumes hosted on the file server with the name '\\WindowsFiler'.
- `Get-EVFSAVolume -SiteId 198...example.com -FileServerName 'WindowsFiler.example.com' -Name Marketing`  
Retrieves the properties of the volume 'Marketing' hosted on the file server with the name 'WindowsFiler.example.com'.

## Output

[Table 4-33](#) lists the properties that are available.

**Table 4-33** Get-EVFSAVolume properties

| Name           | Type   | Description                                                                                       |
|----------------|--------|---------------------------------------------------------------------------------------------------|
| VolumeName     | String | The name of the FSA volume.                                                                       |
| FileServerId   | String | The ID of the file server that hosts the volume share.                                            |
| FileServerName | String | The name of the file server that hosts the volume as configured in the Enterprise Vault Site.     |
| PolicyId       | String | The ID of the file system archiving policy that is used when archiving from this volume.          |
| PolicyName     | String | The name of the file system archiving volume policy that is used when archiving from this volume. |
| VaultStoreId   | String | The ID of the vault store that is used to store files archived from this volume.                  |
| VaultStoreName | String | The name of the vault store that is used to store files archived from this volume.                |
| TaskId         | String | The ID of the File System Archiving task that is used to process this volume.                     |

**Table 4-33** Get-EVFSAVolume properties (*continued*)

| Name              | Type            | Description                                                                     |
|-------------------|-----------------|---------------------------------------------------------------------------------|
| TaskName          | String          | The name of the File System Archiving task that is used to process this volume. |
| DisableArchiving  | SwitchParameter | Specifies whether file archiving from this volume is disabled.                  |
| PassthroughRecall | SwitchParameter | Specifies whether pass-through recall for this volume is enabled.               |

### Related cmdlets

- See [“New-EVFSAVolume”](#) on page 60.
- See [“Remove-EVFSAVolume”](#) on page 63.
- See [“Set-EVFSAVolume”](#) on page 64.

## New-EVFSAVolume

`New-EVFSAVolume` helps to set up File System Archiving for a volume on the selected file server. If the share you want to add is defined as a safety folder, then archived items will not be deleted when placeholders are deleted. For Celerra shares, if the volume policy has both pass-through and DOD enabled, then you are prompted to confirm the action because it can lead to data loss.

`New-EVFSAVolume` is provided by

`Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

### Syntax

```
New-EVFSAVolume [-SiteId <String>] [-Name] <String> [-FileServerName]
<String> [-VaultStoreName] <String> [-TaskName] <String> [-PolicyName]
<String> [-DisableArchiving [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]] [-Confirm [<SwitchParameter>]]
[<CommonParameters>]
```

```
New-EVFSAVolume [-SiteId <String>] [-Name] <String> -FileServerId
<String> -VaultStoreId <String> -TaskId <String> -PolicyId <String>
[-DisableArchiving [<SwitchParameter>]] [-WhatIf [<SwitchParameter>]]
[-Confirm [<SwitchParameter>]] [<CommonParameters>]
```

## Parameters

**Table 4-34** New-EVFSAVolume parameters

| Parameter                  | Description                                                                                                                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -SiteId (required)         | The ID of the Enterprise Vault site under which the file server is configured. If you omit this parameter, <code>New-EVFSAVolume</code> uses the ID of the site of the Enterprise Vault Server where the command is running. You can use <code>Get-EVSite</code> to obtain the site ID. |
| -Name (required)           | The name of the share that you want to configure as a volume target                                                                                                                                                                                                                     |
| -FileServerName (required) | The name of the file server whose volume you want to retrieve.                                                                                                                                                                                                                          |
| -FileServerId (required)   | The ID of the file server whose volume you want to retrieve.                                                                                                                                                                                                                            |
| -VaultStoreName (required) | The name of the vault store that you want to use for archiving files from the new volume.                                                                                                                                                                                               |
| VaultStoreId (required)    | The ID of the vault store that you want to use for archiving files from the new volume.                                                                                                                                                                                                 |
| -TaskName (required)       | The name of the File System Archiving task that you want to use for this volume. The task should exist on the same computer as that of the vault store.                                                                                                                                 |
| -TaskId (required)         | The ID of the File System Archiving task that you want to use for this volume. The task should exist on the same computer as that of the vault store.                                                                                                                                   |
| -PolicyName (required)     | The name of the FSA volume policy that you want to use for the new volume.                                                                                                                                                                                                              |
| -PolicyId (required)       | The ID of the FSA volume policy that you want to use for the new volume.                                                                                                                                                                                                                |
| -DisableArchiving          | If specified, disables file archiving from this volume.                                                                                                                                                                                                                                 |

## Examples

- ```
New-EVFSAVolume -SiteId 198...example.com -Name Sales -FileServerId 1FB...example.com -PolicyId 10C...example.com -VaultStoreId 1AC...example.com -TaskId 120...example.com
```

Adds the existing share 'Sales' as a volume for archiving on the file server with ID '1FB...example.com', with the specified properties.

- `New-EVFSAVolume -SiteId 198...example.com -Name Marketing -FileServerId 1FB...example.com -PolicyId 10C...example.com -VaultStoreId 1AC...example.com -TaskId 120...example.com -DisableArchiving`

Adds the existing share 'Marketing' as a volume on the file server with ID '1FB...example.com', with the specified properties and prevents archiving from the volume.
- `New-EVFSAVolume -SiteId 198...example.com -Name Finance -FileServerName WindowsFiler.example.com -PolicyName 'FSA Volume Policy' -VaultStoreName VSForShares -TaskName 'File System Archiving Task'`

Adds the existing share 'Finance' as a volume for archiving on the file server with the fully-qualified DNS name 'WindowsFiler.example.com', with the specified properties.
- `New-EVFSAVolume -SiteId 198...example.com -Name HumanResources -FileServerName \\WindowsFiler -PolicyName 'FSA Volume Policy' -VaultStoreName VSForShares -TaskName 'File System Archiving Task'`

Adds the existing share 'HumanResources' as a volume for archiving on the file server with the alias '\\WindowsFiler', with the specified properties.

## Output

[Table 4-35](#) lists the properties that are available.

**Table 4-35** New-EVFSAVolume properties

Name	Type	Description
VolumeName	String	The name of the share added as a volume.
FileServerId	String	The ID of the file server that hosts the volume.
FileServerName	String	The name of the file server that hosts the volume as configured in the Enterprise Vault site.
PolicyId	String	The ID of the file system archiving policy that is used when archiving from this volume
PolicyName	String	The name of the file system archiving policy that is used when archiving from this volume.
VaultStoreId	String	The ID of the vault store that is used to store files archived from this volume.
VaultStoreName	String	The name of the vault store that is used to store files archived from this volume.

**Table 4-35** New-EVFSAVolume properties (*continued*)

Name	Type	Description
TaskId	String	The ID of the File System Archiving task that is used to process this volume.
TaskName	String	The name of the File System Archiving task that is used to process this volume.
DisableArchiving	SwitchParameter	Specifies whether file archiving from this volume is disabled.
PassthroughRecall	SwitchParameter	Specifies whether pass-through recall for this volume is enabled.

### Related cmdlets

- See [“Get-EVFSAVolume”](#) on page 58.
- See [“Remove-EVFSAVolume”](#) on page 63.
- See [“Set-EVFSAVolume”](#) on page 64.

## Remove-EVFSAVolume

`Remove-EVFSAVolume` removes the specified volume from Enterprise Vault configuration. You are prompted to confirm the removal of the volume. You cannot remove a volume if the File System Archiving task that is used to process this volume is not in stopped state.

`Remove-EVFSAVolume` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

### Syntax

```
Remove-EVFSAVolume [-SiteId <String>] [-Name] <String>  
[-FileServerName] <String> [-WhatIf [<SwitchParameter>]] [-Confirm  
[<SwitchParameter>]] [<CommonParameters>]
```

```
Remove-EVFSAVolume [-SiteId <String>] [-Name] <String> -FileServerId  
<String> [-WhatIf [<SwitchParameter>]] [-Confirm [<SwitchParameter>]]  
[<CommonParameters>]
```

## Parameters

**Table 4-36** Remove-EVFSAVolume parameters

Parameter	Description
-SiteId (required)	The ID of the Enterprise Vault site under which the file server is configured. If you omit this parameter, <code>Remove-EVFSAVolume</code> uses the ID of the site of the Enterprise Vault Server where the command is running. You can use <code>Get-EVSite</code> to obtain the site ID.
-FileServerName (required)	The fully qualified DNS name or the alias of the file server target configured under the Enterprise Vault site that hosts the volume.
-FileServerId (required)	The ID of the file server target that hosts the volume share.
-Name (required)	The name of the volume that you want to remove.

## Examples

- `Remove-EVFSAVolume -SiteId 198...example.com -Name Sales -FileServerId 1FB...example.com`  
Removes the volume 'Sales' on the file server with ID '1FB...example.com'.
- `Remove-EVFSAVolume -SiteId 198...example.com -Name Finance -FileServerName WindowsFiler.example.com`  
Removes the volume 'Finance' on the file server 'WindowsFiler.example.com'.

## Output

`Remove-EVFSAVolume` returns no output.

## Related cmdlets

- See "[New-EVFSAVolume](#)" on page 60.
- See "[Get-EVFSAVolume](#)" on page 58.
- See "[Set-EVFSAVolume](#)" on page 64.

# Set-EVFSAVolume

`Set-EVFSAVolume` updates the specified settings of an existing file server volume. For Celerra shares, if the volume policy has both pass-through and DOD enabled, then you are prompted to confirm the action because it can lead to data loss.

Set-EVFSAVolume is provided by

Symantec.EnterpriseVault.PowerShell.AdminAPI.dll, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-EVFSAVolume [-SiteId <String>] [-Name] <String> [-FileServerName] <String> [[-TaskName] <String>] [[-PolicyName] <String>] [-DisableArchiving <SwitchParameter>] [-PassthroughRecall <SwitchParameter>] [-WhatIf <SwitchParameter>] [-Confirm <SwitchParameter>] [<CommonParameters>]
```

```
Set-EVFSAVolume [-SiteId <String>] [-Name] <String> -FileServerId <String> [-TaskId <String>] [-PolicyId <String>] [-DisableArchiving <SwitchParameter>] [-PassthroughRecall <SwitchParameter>] [-WhatIf <SwitchParameter>] [-Confirm <SwitchParameter>] [<CommonParameters>]
```

## Parameters

**Table 4-37** Set-EVFSAVolume parameters

Parameter	Description
-SiteId (required)	The ID of the Enterprise Vault site under which the file server is configured. If you omit this parameter, Set-EVFSAVolume uses the ID of the site of the Enterprise Vault Server where the command is running. You can use Get-EVSite to obtain the site ID.
-Name (required)	The name of the volume that you want to update.
-FileServerName (required)	The fully-qualified DNS name or the alias of the file server target configured under the Enterprise Vault site that hosts the volume.
-FileServerId (required)	The ID of the file server target that hosts the volume share.
-TaskName	The name of the File System Archiving task that you want to use for this volume. The task should exist on the same computer as that of the vault store.
-TaskId	The ID of the File System Archiving task that you want to use for this volume. The task should exist on the same computer as that of the vault store.
-PolicyName	The name of the FSA volume policy that you want to use for the new volume.

**Table 4-37** Set-EVFSAVolume parameters (*continued*)

Parameter	Description
-PolicyId	The ID of the FSA volume policy that you want to use for the new volume.
-DisableArchiving	If specified, disables file archiving from this volume.
-PassthroughRecall	If specified, enables pass-through recall with this volume. This option is configurable only for volume shares that are hosted on a Windows file server. It is available only if you have configured pass-through recall on the file server.

## Examples

- `Set-EVFSAVolume -SiteId 198...example.com -Name Sales -FileServerId 1FB...example.com -PolicyId 10C...example.com -TaskId 120...example.com`  
Updates the policy and task of the volume 'Sales' on the file server with ID '1FB...example.com', using ID values of policy and task.
- `Set-EVFSAVolume -SiteId 198...example.com -Name Finance -FileServerName WindowsFiler.example.com -PolicyName 'FSA Volume Policy' -TaskName 'File System Archiving Task'`  
Updates the policy and task of the volume 'Finance' on the file server with the fully-qualified DNS name 'WindowsFiler.example.com', using name values.
- `1. Set-EVFSAVolume -SiteId 198...example.com -Name HumanResources -FileServerName \\WindowsFiler -DisableArchiving -PassthroughRecall`  
Disables archiving and enables pass-through recall for the volume 'HumanResources' on the file server '\\WindowsFiler'.

## Output

[Table 4-38](#) lists the properties that are available.

**Table 4-38** Set-EVFSAVolume properties

Name	Type	Description
VolumeName	String	The name of the share added as a volume.
FileServerId	String	The ID of the file server that hosts the volume.
FileServerName	String	The name of the file server that hosts the volume as configured in the Enterprise Vault site.
PolicyId	String	The ID of the file system archiving policy that is used when archiving from this volume

**Table 4-38** Set-EVFSAVolume properties (*continued*)

Name	Type	Description
PolicyName	String	The name of the file system archiving policy that is used when archiving from this volume.
VaultStoreId	String	The ID of the vault store that is used to store files archived from this volume.
VaultStoreName	String	The name of the vault store that is used to store files archived from this volume.
TaskId	String	The ID of the File System Archiving task that is used to process this volume.
TaskName	String	The name of the File System Archiving task that is used to process this volume.
DisableArchiving	SwitchParameter	Specifies whether file archiving from this volume is disabled.
PassthroughRecall	SwitchParameter	Specifies whether pass-through recall for this volume is enabled.

### Related cmdlets

- See [“New-EVFSAVolume”](#) on page 60.
- See [“Get-EVFSAVolume”](#) on page 58.
- See [“Remove-EVFSAVolume”](#) on page 63.

## Get-EVIISWebsite

Gets details of the Enterprise Vault virtual directories that are configured on the current Enterprise Vault server.

`Get-EVIISWebsite` is provided by

`Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

### Syntax

```
Get-EVIISWebsite [<CommonParameters>]
```

### Parameters

PowerShell common parameters only.

## Examples

- `Get-EVIISWebsite`  
Gets details of the Enterprise Vault virtual directories that have been configured on the current Enterprise Vault server.

## Output

[Table 4-39](#) lists the properties that are available.

**Table 4-39** Get-EVIISWebsite properties

Name	Type	Description
EnterpriseVaultModule	EVWebsiteModule	One of the following: "EVMOD_ENTERPRISEVAULT"; "EVMOD_FSAREPORTING"; "EVMOD_EVINDEXING"; "EVMOD_ENTERPRISEVAULTAPI"; "EVMOD_EVSEARCHCLIENT"; "EVMOD_EVSEARCHSERVER".
WebAddress	String	An Enterprise Vault URL.

# Set-EVIMAPServerDisabled

`Set-EVIMAPServerDisabled` disables the specified server for IMAP and stops the IMAP server.

`Set-EVIMAPServerDisabled` is provided by `Symantec.EnterpriseVault.PowerShell.IMAP.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-EVIMAPServerDisabled -ComputerNameAlternate [<CommonParameters>]
```

## Parameters

**Table 4-40** Set-EVIMAPServerDisabled parameters

Parameter	Description
<code>-ComputerNameAlternate</code> (required)	The alias for the Enterprise Vault server on which you want to stop and disable the IMAP server.

## Examples

- `Get-EVComputers | Where-Object {$_.EnabledForIMAP -eq 1} |  
ForEach-Object {Set-EVIMAPServerDisabled $_.ComputerNameAlternate}`  
This command finds all Enterprise Vault servers that are enabled for IMAP and disables each one in turn, stopping the IMAP server in the process.
- `Set-EVIMAPServerDisabled ACME_LAB_1`  
This command finds the Enterprise Vault server with the Computer name 'ACME\_LAB\_1' and disables it for IMAP, stopping the IMAP server in the process.

## Output

There is a confirmation message on completion.

## Related cmdlets

- See [“Get-EVComputers”](#) on page 44.
- See [“Set-EVIMAPServerEnabled”](#) on page 69.
- See [“Get-EVIMAPUsers”](#) on page 70.
- See [“Get-EVIMAPUserSettings”](#) on page 72.

# Set-EVIMAPServerEnabled

`Set-EVIMAPServerEnabled` enables the specified server for IMAP and starts the IMAP server.

`Set-EVIMAPServerEnabled` is provided by

`Symantec.EnterpriseVault.PowerShell.IMAP.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-EVIMAPServerEnabled -ComputerNameAlternate [<CommonParameters>]
```

## Parameters

**Table 4-41** Set-EVIMAPServerEnabled parameters

Parameter	Description
<code>-ComputerNameAlternate</code> (required)	The alias for the Enterprise Vault server on which you want to enable and start the IMAP server.

## Examples

- `Get-EVComputers | Where-Object {$_.EnabledForIMAP -eq 0} | ForEach-Object {Set-EVIMAPServerEnabled $_.ComputerNameAlternate}`  
This command finds all Enterprise Vault servers that are not enabled for IMAP and enables each one in turn, starting the IMAP server in the process.
- `Set-EVIMAPServerEnabled ACME_LAB_1`  
This command finds the Enterprise Vault server with the Computer name 'ACME\_LAB\_1' and enables it for IMAP, starting the IMAP server in the process.

## Output

There is a confirmation message on completion.

## Related cmdlets

- See [“Get-EVComputers”](#) on page 44.
- See [“Set-EVIMAPServerDisabled”](#) on page 68.
- See [“Get-EVIMAPUsers”](#) on page 70.
- See [“Get-EVIMAPUserSettings”](#) on page 72.

# Get-EVIMAPUsers

`Get-EVIMAPUsers` gets the list of users enabled for IMAP access.

`Get-EVIMAPUsers` is provided by

`Symantec.EnterpriseVault.PowerShell.IMAP.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVIMAPUsers -ArchiveName -NTUserName [<CommonParameters>]
```

## Parameters

**Table 4-42** Get-EVIMAPUsers parameters

Parameter	Description
<code>-ArchiveName</code>	Use <code>-ArchiveName</code> to display only the IMAP enabled users whose archive name matches the value you provide.

**Table 4-42** Get-EVIMAPUsers parameters (*continued*)

Parameter	Description
-NTUserName	Use <code>-NTUserName</code> to display only the IMAP enabled users whose Active Directory user name matches the value you provide.

## Examples

- `Get-EVIMAPUsers -ArchiveName John`  
This command returns the users matching the Archive name 'John'.
- `Get-EVIMAPUsers -NTUserName "Steve_B"`  
This command returns the users matching the username entry 'Steve\_B'.

## Output

[Table 4-43](#) lists the properties that are available.

**Table 4-43** Get-EVIMAPUsers properties

Name	Type	Description
EnabledForIMAP	Boolean	Indicates whether the user is enabled for IMAP access. Possible values are: <code>\$true</code> (user is enabled), <code>\$false</code> (user is not enabled).
MbxArchiveName	String	The name of the user's mailbox archive.
MbxNTDomain	String	The Windows domain.
MbxNTUser	String	The user's login name.
ReadyForIMAP	Boolean	Indicates whether the metadata store has been built for the archive. Possible values: <code>\$true</code> (the metadata store has been built), <code>\$false</code> (the metadata store has not yet been built). Internet mail archives always have a metadata store, so ReadyForIMAP is always true for Internet mail archives.
SID	String	The SID (security identifier) of the Active Directory user.
Type	String	The type of archive. Possible values: "Exchange", "Internet Mail".

## Related cmdlets

- See “[Get-EVComputers](#)” on page 44.
- See “[Set-EVIMAPServerDisabled](#)” on page 68.
- See “[Set-EVIMAPServerEnabled](#)” on page 69.
- See “[Get-EVIMAPUserSettings](#)” on page 72.

# Get-EVIMAPUserSettings

`Get-EVIMAPUserSettings` gets the IMAP/SMTP client settings for the given user SID.

`Get-EVIMAPUserSettings` is provided by `Symantec.EnterpriseVault.PowerShell.IMAP.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVIMAPUserSettings -SID [<CommonParameters>]
```

## Parameters

**Table 4-44** Get-EVIMAPUserSettings parameters

Parameter	Description
-SID (required)	The SID (security identifier) of the Active Directory user whose IMAP settings you want to display.

## Examples

- `Get-EVIMAPUserSettings S-1-5-21-750406912-2207124990-322363385-3682`  
This command returns the IMAP/SMTP server settings for the archives that the user represented by the SID parameter has access to.

## Output

[Table 4-45](#) lists the properties that are available.

**Table 4-45** Get-EVIMAPUserSettings properties

Name	Type	Description
Archivename	String	The name of the user's archive.

**Table 4-45** Get-EVIMAPUserSettings properties (*continued*)

Name	Type	Description
ArchiveOwner	Boolean	Indicates whether the user is the owner of the archive. Possible values: <code>\$true</code> (user owns the archive), <code>\$false</code> (user does not own the archive).
ArchiveType	String	The type of archive. Possible values: "Exchange", "Internet Mail".
IMAP_Connection_Security	String	The security to use for the IMAP connection.
IMAP_Password	String	Specifies which account's password to use when connecting to the archive.
IMAP_Port	Int32	The port number to use for the IMAP connection.
IMAP_Server	String	The IMAP server to which to connect.
IMAP_UserName	String	The account name to use when connecting to the IMAP server.
ReadyForIMAP	Boolean	Indicates whether the metadata store has been built for the archive. Possible values: <code>\$true</code> (the metadata store has been built), <code>\$false</code> (the metadata store has not yet been built). Internet mail archives always have a metadata store, so ReadyForIMAP is always true for Internet mail archives.
SMTP_Connection_Security	String	The type of security to use for the SMTP connection.
SMTP_Password	String	Possible values: "SMTP account username" (the SMTP server requires authentication), "Not applicable" (the SMTP server does not require authentication).
SMTP_Port	Int32	The port to use for the SMTP connection.
SMTP_Server	String	The SMTP server to which to connect.
SMTP_UserName	String	If the SMTP server requires authentication this string contains "SMTP account username".
WindowsUser	String	The Windows account to use to connect to the IMAP server.

## Related cmdlets

- See “[Get-EVComputers](#)” on page 44.
- See “[Set-EVIMAPServerDisabled](#)” on page 68.
- See “[Set-EVIMAPServerEnabled](#)” on page 69.
- See “[Get-EVIMAPUsers](#)” on page 70.

# Get-EVIndexLocation

Gets details of the Enterprise Vault index locations that are configured on the current Enterprise Vault server.

`Get-EVIndexLocation` is provided by

`Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

## Syntax

```
Get-EVIndexLocation [<CommonParameters>]
```

## Parameters

PowerShell common parameters only.

## Examples

- `Get-EVIndexLocation`  
Gets details of the Enterprise Vault index locations that are configured on the current Enterprise Vault server.

## Output

[Table 4-46](#) lists the properties that are available.

**Table 4-46** Get-EVIndexLocation properties

Name	Type	Description
BackupMode	IndexLocationBackupMode	Index location backup mode. One of the following: "On"; "Off".
ComputerName	String	The name of the computer that hosts the index location.

**Table 4-46** Get-EVIndexLocation properties (*continued*)

Name	Type	Description
IndexingGroup	String	The name of the indexing group that the location is in, if any.
IndexRootPath	String	The index root path. For example, E:\EVStorage\Index\index8
IndexRootPathEntryId	String	The Index Root Path Entry Id.
IndexServiceEntryID	String	The Index Service Entry ID.
Status	IndexLocationStatus	The index location status. One of the following: "Open"; "Closed".
TotalQuotaGBytesAvailable	Double	The total amount of space available, in gigabytes. This value can be null.
TotalQuotaGBytesFree	Double	The amount of unused space, in gigabytes. This value can be null
TotalQuotaGBytesUsed	Double	The amount of used space, in gigabytes. This value can be null.

## New-EVMDSBuildTask

`New-EVMDSBuildTask` creates an indexing task that builds (or rebuilds) a Metadata Store for an archive. The command must be run by a user with permission to manage indexes.

`New-EVMDSBuildTask` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

### Syntax

```
New-EVMDSBuildTask [-ArchiveId] <String> -Mode <MDSBuildTaskMode>
[-VerifyIndex [<MDSBuildTaskVerifyIndex>]] [-Force
[<SwitchParameter>]] [-WhatIf [<SwitchParameter>]]
[<CommonParameters>]
```

## Parameters

**Table 4-47** New-EVMDSBuildTask parameters

Parameter	Description
-ArchiveId (required)	Identifies an Enterprise Vault archive.
-Mode (required)	Build, Rebuild. Use <b>Build</b> to create a new Metadata Store for the archive, or fill in gaps in the existing one. Use <b>Rebuild</b> to recreate the entire Metadata Store for the archive, overwriting any existing data.
-VerifyIndex	Default, Verify, NoVerify. In some cases the Metadata Store build task triggers an index verify task to ensure the Metadata Store is complete. This parameter can be used to enable or disable this behavior, overriding any default.
-Force	Creates the Metadata Store build task without prompting for confirmation.
-WhatIf	Describes what would happen if you executed the command without actually executing the command.

## Examples

- `New-EVMDSBuildTask -Mode Build -ArchiveId 14B...EVSERVER.LOCAL`  
This command creates an indexing task that builds the Metadata Store for the specified archive. If the archive already has a Metadata Store, any missing items will be added where possible, but existing data will not be recreated.
- `New-EVMDSBuildTask -Mode Rebuild -ArchiveId 14B...EVSERVER.LOCAL`  
This command creates an indexing task that rebuilds the Metadata Store for the specified archive. Any existing data in the archive's Metadata Store will be overwritten.
- `New-EVMDSBuildTask -Mode Build -VerifyIndex NoVerify -ArchiveId 14B...EVSERVER.LOCAL`  
This command creates an indexing task that builds the Metadata Store for the specified archive, but will NOT allow the task to trigger a verification of the archive's index.
- `Import-Csv c:\archives.csv | New-EVMDSBuildTask -Mode Build`

This command reads a list of archives from a CSV file, and creates a Metadata Store build task for each archive. One of the columns in the CSV file must be called 'ArchiveId'.

- `Get-Content c:\archives.txt | New-EVMDSBuildTask -Mode Build`  
This command reads a list of archive IDs from a text file (one per line), and creates a Metadata Store build task for each archive.
- `New-EVMDSBuildTask -Force -Mode Build -ArchiveId  
14B...EVSERVER.LOCAL`  
This command uses the `-Force` switch to suppress confirmation prompting.
- `New-EVMDSBuildTask -WhatIf -Mode Build -ArchiveId  
14B...EVSERVER.LOCAL`  
This command uses the `-WhatIf` switch to display the Metadata Store build tasks that would be created by the command, without actually creating the tasks.

## Output

`NewMDSBuildTaskResult`

[Table 4-48](#) lists the properties that are available.

**Table 4-48** New-EVMDSBuildTask properties

Name	Type	Description
ArchiveId	String	The ArchiveId of the archive for which a task has been created.
ArchiveName	String	The name of the archive for which a task has been created.
Mode	MDSBuildTaskMode	The mode that is being used when building the archive. Possible values: "Build", "Rebuild".
VerifyIndex	MDSBuildTaskVerifyIndex	Indicates whether an index verify task will be created to verify that the Metadata Store is complete. Possible values: "Default", "Verify", "NoVerify".

## Related cmdlets

- See [“Get-EVMDSStatus”](#) on page 78.

# Get-EVMDSStatus

`Get-EVMDSStatus` gets the current status of the metadata store for an archive. The command can also be used to get the number of items that are missing from the metadata store for an archive.

`Get-EVArchive` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVMDSStatus [[-ArchiveId] <String>] [[-VaultStoreId] <String>]
[[-SiteId] <String>] [-IncludeMissingItems [<SwitchParameter>]]
[<CommonParameters>]
```

## Parameters

**Table 4-49** Get-EVMDSStatus parameters

Parameter	Description
<code>-ArchiveId</code> (required)	Identifies an archive. This is the default operation.
<code>-VaultStoreId</code> (required)	Identifies a vault store. If this parameter is used instead of <code>ArchiveId</code> , the command will operate on all applicable archives in the specified vault store.
<code>-SiteId</code> (required)	Identifies a site. If this parameter is used the command operates on all appropriate archives in the specified site.
<code>-IncludeMissingItems</code>	Whether the command should return the count of items that are present in the archive but are missing from the metadata store. This option can be slow if you have specified thousands of archives.

## Examples

- `Get-EVMDSStatus -ArchiveId 14B...EVSERVER.LOCAL`  
Gets the current status of the metadata store for the specified archive.
- `Get-EVMDSStatus -IncludeMissingItems -VaultStoreId 14B...EVSERVER.LOCAL`

Gets the current status of the metadata store for all archives supported by metadata store in the specified vault store. Also includes a count of the number of items that are present in the archive but are missing from the metadata store.

- `Import-CSV example.csv | Get-EVMDSStatus`  
Gets the current status of the metadata store for each archive that is specified in a CSV file.
- `Import-CSV example.csv | Select VaultStoreEntryId | Get-EVMDSStatus`  
Imports a record set from a CSV file and, for each value in the `VaultStoreEntryId` column, returns the current status of the metadata store for each archive that is supported by metadata store.
- `Get-EVMDSStatus -VaultStoreId 14B...EVSERVER.LOCAL -IncludeMissingItems | Where-Object {$_.MDSStatus -ne 'EV_MDS_DISABLED' -and $_.MDSMissingItems -gt 0} | New-EVMDSBuildTask -Mode Build -Force`  
Creates a metadata store build task for each Fast Browsing or IMAP enabled archive in the specified vault store, if there are items missing from that archive's metadata store.

## Output

MDSStatus

MDSMissingItems

[Table 4-50](#) lists the properties that are available.

**Table 4-50** Get-EVMDSStatus properties

Name	Type	Description
ArchiveId	String	The ArchiveId.
MDSStatus	MDSStatus	The current status of the metadata store. Possible values: "EV_MDS_DISABLED", "EV_MDS_BUILD_PENDING", "EV_MDS_BUILDING", "EV_MDS_READY", "EV_MDS_BUILD_FAILED".

## Related cmdlets

- See "[New-EVMDSBuildTask](#)" on page 75.

# Add-EVPstComputer

`Add-EVPstComputer` adds a computer as a locate and migrate PST migration target. Depending on configuration, Enterprise Vault can search the computer for PST files to be migrated to Enterprise Vault. For example, you can add computers that belong to individual users, and you can add file servers that host PST files belonging to many users.

`Add-EVPstComputer` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Add-EVPstComputer -Name <String> [-Mailbox <String>] [-EnableSearch]
[-SiteName <String>] [-DirectoryServer <String>] [<CommonParameters>]
```

## Parameters

**Table 4-51** Add-EVPstComputer parameters

Parameter	Description
<code>-Name</code> (required)	The name of the computer you want to add as a PST migration target.
<code>-Mailbox</code>	The name of the mailbox in which Enterprise Vault places shortcuts for items migrated from PST files found on the computer. If you do not specify a mailbox, <code>Add-EVPstComputer</code> uses the mailbox that each PST file is associated with. Typically, you would not specify a mailbox for any computer that hosts PST files that belong to different users, such as a file server.
<code>-EnableSearch</code>	Use <code>-EnableSearch</code> if you want Enterprise Vault to search the target computer for PST files.
<code>-SiteName</code>	If you have more than one Enterprise Vault site in your environment, use <code>-SiteName</code> to specify the site in which you want to add the computer.
<code>-DirectoryServer</code>	If you execute <code>Add-EVPstComputer</code> from a server other than an Enterprise Vault server, specify the name of the Enterprise Vault server that <code>Add-EVPstComputer</code> should connect to.

## Examples

- `Add-EVPstComputer -Name FileServer1 -EnableSearch`

Adds FileServer1 as a PST migration target, and specifies that Enterprise Vault should search the server for PST files.

- `Add-EVPstComputer -Name JohnDoeLaptop -Mailbox "John Doe" -EnableSearch`

Adds John Doe's laptop as a PST migration target, and also specifies John's mailbox, in which Enterprise Vault will place shortcuts for migrated items.

## Related cmdlets

- See ["Add-EVPstFile"](#) on page 81.

# Add-EVPstFile

`Add-EVPstFile` specifies the UNC path and name of a PST file whose contents you want to migrate to Enterprise Vault.

`Add-EVPstFile` is provided by

`Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Add-EVPstFile -UNCPath <String> [-Mailbox <String>] [-Archive
<String>] [-ArchiveType <String>] [-RetentionCategory <String>]
[-PasswordProtected] [-Language <String>] [-Priority <String>]
[-SiteName <String>] [-DirectoryServer <String>] [<CommonParameters>]
```

## Parameters

**Table 4-52** Add-EVPstFile parameters

Parameter	Description
<code>-UNCPath</code> (required)	The complete UNC path to the PST file you want to migrate, including the name of the PST file itself.
<code>-Mailbox</code>	The name of the mailbox in which Enterprise Vault places shortcuts for the migrated items. If you do not specify a mailbox, <code>Add-EVPstFile</code> uses the mailbox that the PST file is associated with.
<code>-Archive</code>	The name of the archive to which you want to migrate the contents of the PST file. If you do not specify the archive, <code>Add-EVPstFile</code> migrates items to the archive that is associated with the mailbox that owns the PST file.

**Table 4-52** Add-EVPstFile parameters (*continued*)

Parameter	Description
-ArchiveType	The type of archive. You can specify either "Exchange Mailbox" or "Internet Mail". If you specify a mailbox, the PST is migrated to the associated Exchange Mailbox archive. If you do not specify a mailbox, Enterprise Vault looks for the archive in both Exchange Mailbox archives and Internet Mail archives. If only one archive is found, Enterprise Vault sets the archive type as that of the found archive and migrates the file to that archive. If multiple entries are found, then Enterprise Vault migrates the file to the Exchange Mailbox archive. Note that you cannot associate a mailbox with an Internet Mail archive.
-RetentionCategory	The retention category that is applied to the migrated items. If you do not specify a retention category, Add-EVPstFile uses the retention category set in the applicable PST migration policy.
-PasswordProtected	Indicates that the PST file is password protected. If you use -PasswordProtected, Add-EVPstFile prompts for the password.
-Language	The language used in the PST file. If you do not specify a language, Add-EVPstFile uses the language set in the applicable PST migration policy.
-Priority	Sets the PST file's migration priority. You can specify any one of the following priorities, or their associated numerical values which are shown in brackets: Critical (1), Important (2), High (3), Medium (4), Low (5), or Lowest (6). The default priority is Medium (4).
-SiteName	If you have more than one Enterprise Vault site in your environment, use -SiteName to specify the site in which you want to migrate the PST file.
-DirectoryServer	If you execute Add-EVPstFile from a server other than an Enterprise Vault server, specify the name of the Enterprise Vault server that Add-EVPstFile should connect to.

## Examples

- `Add-EVPstFile -UNCPath \\FileServer1\UserShares\VIPs\JohnDoe\PSTs\2012.pst`  
Adds the specified PST file for migration to Enterprise Vault.
- `Add-EVPstFile -UNCPath \\abc.xyz.com\c$\user1.pst -Archive IMAP1 -ArchiveType "Internet Mail"`

Migrates the PST file to the specified Internet Mail archive.

## Related cmdlets

- See “[Add-EVPstComputer](#)” on page 80.

# Get-EVRBARole

`Get-EVRBARole` gets all the Enterprise Vault RBA roles from the Enterprise Vault Directory. You can also get specific RBA roles by providing a full or partial role name using the `-Name` parameter.

You can specify the Enterprise Vault Directory using the `-EVDirectoryServer` parameter. If you do not specify the `-EVDirectoryServer` parameter, and `Get-EVRBARole` cannot determine the directory, you are prompted to enter a value for `-EVDirectoryServer`.

If you execute `Get-EVRBARole` from a server other than an Enterprise Vault server, it prompts for the `-EVDirectoryServer` parameter, unless `-EVDirectoryServer` is already specified as a parameter.

If you execute the cmdlet from a server other than an Enterprise Vault server, and where an object of the type `Symantec.EnterpriseVault.Admin.EVRbaRole` is created and any of its public method is expected to be called directly in the PowerShell scripts, the `EVDirectoryServer` property of the object must be set before calling the method on the object.

When an object of type `Symantec.EnterpriseVault.Admin.EVRbaRole` is created and used in any PowerShell scripts, such as when it is passed to the cmdlet using the `-AdminObject` parameter, it is the object's `AdminObjectCollection` that is populated with the criteria specified. This is because the cmdlet allows wildcards on the `Name` property. However, the collection can be accessed by calling the method `GetAdminObjectCollection()` on the object.

Permission to execute this cmdlet is restricted to the Vault Service Account.

`Get-EVRBARole` is provided by

`Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVRBARole [[-Name] <string>] -EVDirectoryServer <string>  
[<CommonParameters>]
```

```
Get-EVRBARole [-AdminObject
<Symantec.EnterpriseVault.Admin.EVRbaRole>] -EVDirectoryServer
<string> [<CommonParameters>]
```

## Parameters

**Table 4-53** Get-EVRBARole parameters

Parameter	Description
-Name	Identifies all or part of the name of an Enterprise Vault RBA role.
-EVDirectoryServer (required)	Specifies the host name, fully qualified domain name, or DNS alias of an Enterprise Vault Directory server. Note that, although this parameter is required, the cmdlet determines it automatically if it can.
-AdminObject	Identifies an object of the type <code>Symantec.EnterpriseVault.Admin.EVRbaRole</code> .

## Examples

- `Get-EVRBARole`  
Gets all the Enterprise Vault RBA roles from the current Enterprise Vault Directory.
- `Get-EVRBARole -Name "File Server Administrator"`  
Gets the Enterprise Vault RBA role called "File Server Administrator" from the current Enterprise Vault Directory.
- `Get-EVRBARole ex*`  
Performs a wildcard search and gets the Enterprise Vault RBA role objects whose names begin with "ex". For example:
  - Exchange Administrator
  - Extension Content Provider Administrator
  - Extension Content Provider Application
 Note that the parameter `-Name` is a positioned parameter (at position 0), so this example works without it being specified.
- In the preceding examples, you can also use the `-EVDirectoryServer` parameter to connect to a specified Enterprise Vault Directory.
  - `Get-EVRBARole -EVDirectoryServer evs2.example.com`  
Gets all the Enterprise Vault RBA roles from the Enterprise Vault Directory identified by the `-EVDirectoryServer` parameter.

- `Get-EVRBARole ex* -EVDirectoryServer evs2.example.com`  
Gets the Enterprise Vault RBA role objects whose names begin with “ex” in the Enterprise Vault Directory identified by `EVDirectoryServer` parameter.
- **AdminObject usage examples:**
  - `$r = New-Object Symantec.EnterpriseVault.Admin.EVRbaRole`  
`Get-EVRBARole -AdminObject $r`  
Gets all the Enterprise Vault RBA roles from the current directory into `$r`.
  - `$r = New-Object Symantec.EnterpriseVault.Admin.EVRbaRole`  
`Get-EVRBARole -AdminObject $r -EVDirectoryServer`  
`"evs2.example.com"`  
Gets all the Enterprise Vault RBA roles from the directory identified by the `-EVDirectoryServer` parameter into `$r`.
  - `$r = New-Object Symantec.EnterpriseVault.Admin.EVRbaRole`  
`$r.Name = "*co*"`  
`Get-EVRBARole -AdminObject $r`  
Gets the Enterprise Vault RBA role objects whose names contain “co” into `$r`.
- **String piping example:**  
`"ex*", "file server administrator" | Get-EVRBARole`  
Gets the matching Enterprise Vault RBA role objects.

## Output

`Get-EVRBARole` returns objects of the type `Symantec.EnterpriseVault.Admin.EVRbaRole`, which has the following properties.

**Table 4-54** Symantec.EnterpriseVault.Admin.EVRbaRole properties

Name	Type	Description
Name	String	The name of the Enterprise Vault RBA role.
RoleGuid	String	The GUID of the Enterprise Vault RBA role.
TaskLinkDescription	String	The description of the associated TaskLink.
TaskLinkGuid	String	The GUID of the associated TaskLink.

## Related cmdlets

- See [“Add-EVRBARoleMember”](#) on page 86.
- See [“Get-EVRBARoleMember”](#) on page 89.

- See “[Remove-EVRBARoleMember](#)” on page 92.

## Add-EVRBARoleMember

`Add-EVRBARoleMember` adds one or more Active Directory users, groups, or built-in security principals as new members of an Enterprise Vault RBA role.

The `-Identity` parameter specifies the Enterprise Vault RBA role to access. You can identify an RBA role by its name or its GUID. You can also specify the role by passing a role object through the pipeline. For example, you can use `Get-EVRBARole` to retrieve a role object and then pass the object through the pipeline to `Add-EVRBARoleMember`. If you do not specify the `-Identity` or the `-InputObject` parameter, you are prompted to enter the value for `-Identity`.

The `-Members` parameter specifies the new members to add to a role. You can identify a new member by its security identifier (SID) or its SAM account name. If you specify more than one new member, use a comma-separated list. You cannot pass member objects through the pipeline to this cmdlet. If you do not specify the `-Members` parameter, you are prompted to enter the value for `-Members`.

You can specify the Enterprise Vault directory server using the `-EVDirectoryServer` parameter. If you do not use the `-EVDirectoryServer` parameter, and `Add-EVRBARoleMember` cannot determine the server, you are prompted to enter the Enterprise Vault directory server.

Permission to execute this cmdlet is restricted to the Vault Service Account.

`Add-EVRBARoleMember` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded automatically by the Enterprise Vault Management Shell.

### Syntax

```
Add-EVRBARoleMember [-Identity] <string> [-Members] <string[]>  
-EVDirectoryServer <string> [-PassThru <Switch>] [-Confirm <Switch>]  
[-WhatIf <Switch>] [<CommonParameters>]
```

```
Add-EVRBARoleMember [-InputObject]  
<Symantec.EnterpriseVault.Admin.EVRbaRole> [-Members] <string[]>  
-EVDirectoryServer <string> [-PassThru <Switch>] [-Confirm <Switch>]  
[-WhatIf <Switch>] [<CommonParameters>]
```

## Parameters

**Table 4-55** Add-EVRBARoleMember parameters

Parameter	Description
-Identity (required)	Specifies an Enterprise Vault RBA role by its name or GUID.
-InputObject (required)	Specifies an Enterprise Vault RBA role object. This parameter can get this object through the pipeline or you can set this parameter to an object instance.
-Members (required)	<p>Specifies a set of Active Directory users, groups, or built-in security principal objects in a comma-separated list to add to a role.</p> <p>To identify each object, use one of the following property values. Note that the identifier in parentheses is the LDAP display name. The acceptable values for this parameter are:</p> <ul style="list-style-type: none"> <li>■ Security Identifier (objectSid)</li> <li>■ SAM Account Name (sAMAccountName)</li> </ul> <p>The following example shows how to specify this parameter. This example specifies a user and a group by its SAM Account Name property and another user by its SID property.</p> <pre>-Members "fsaAdmin@example.com", "NT AUTHORITY\This Organization", "S-1-5-21-1529523603-1500826627-74573220-1119"</pre> <p>You cannot provide objects to this parameter. You cannot pass objects through the pipeline to this parameter.</p>
-EVDirectoryServer (required)	Specifies the host name, fully qualified domain name, or DNS alias of an Enterprise Vault Directory server. Note that, although this parameter is required, the cmdlet determines it automatically if it can.
-PassThru	Returns an object representing the role with which you are working. By default, this cmdlet does not generate any output.
-Confirm	Prompts you for confirmation before running the cmdlet.
-WhatIf	Shows what would happen if the cmdlet runs. The cmdlet is not actually run.

## Examples

- `Add-EVRBARoleMember -Identity "File Server Administrator" -Members "fsaAdmin@example.com", "example\fsaGroup", "s-1-5-21-1529523603-1500826627-74573220-1119"`  
Adds the specified Active Directory members to an Enterprise Vault RBA role called "File Server Administrator".
- `Add-EVRBARoleMember "File Server Administrator"`  
Prompts for the members to be added and adds them to an Enterprise Vault RBA role called "File Server Administrator". Note that the parameter `-Identity` is a positioned parameter (at position 0), so this example works without it being specified.
- Additionally, `-EVDirectoryServer` parameter can be specified to connect to the desired Enterprise Vault directory:  
`Add-EVRBARoleMember "File Server Administrator" -EVDirectoryServer evs2.example.com`  
Prompts for the members to be added and adds them to an Enterprise Vault RBA role called "File Server Administrator" in the Enterprise Vault directory identified by the `-EVDirectoryServer` parameter.
- `-InputObject` usage example:  
`$r = Get-EVRBARole "file server administrator"`  
`Add-EVRBARoleMember -InputObject $r`  
Prompts for the members to be added and adds them to an Enterprise Vault RBA role called "File Server Administrator" in the current Enterprise Vault directory. Note that the parameter `-InputObject` is a positioned parameter (again at position 0), so this example will work without it being specified.
- String piping example:  
`"file server administrator" | Add-EVRBARoleMember`  
Prompts for the members to be added and adds them to an Enterprise Vault RBA role called "File Server Administrator" in the current Enterprise Vault Directory.
- EVRbaRole piping example:  
`Get-EVRBARole "file server administrator" -EVDirectoryServer evs2.example.com | Add-EVRBARoleMember -Members "fsaAdmin1", "fsaAdmin2" -EVDirectoryServer evs2.example.com`  
Pipes the EVRbaRole object obtained from the `Get-EVRBARole` cmdlet to the `Add-EVRBARoleMember` cmdlet and adds the specified members to that role. Also note that the parameter `-EVDirectoryServer` specifies the Enterprise Vault directory for each cmdlet to use.

## Output

`Add-EVRBARoleMember` returns no output, or a modified object of the type `Symantec.EnterpriseVault.Admin.EVRbaRole` when the `-PassThru` parameter is specified. `Symantec.EnterpriseVault.Admin.EVRbaRole` has the following properties.

**Table 4-56** Symantec.EnterpriseVault.Admin.EVRbaRole properties

Name	Type	Description
Name	String	The name of the Enterprise Vault RBA role.
RoleGuid	String	The GUID of the Enterprise Vault RBA role.
TaskLinkDescription	String	The description of the associated TaskLink.
TaskLinkGuid	String	The GUID of the associated TaskLink.

## Related cmdlets

- See [“Get-EVRBARole”](#) on page 83.
- See [“Get-EVRBARoleMember”](#) on page 89.
- See [“Remove-EVRBARoleMember”](#) on page 92.

# Get-EVRBARoleMember

`Get-EVRBARoleMember` gets the members of an Enterprise Vault RBA role from the Enterprise Vault Directory. Members can be Active Directory users, groups, or built-in security principals.

The `-Identity` parameter specifies the Enterprise Vault RBA role to access. You can identify an RBA role by its name, or GUID. You can also specify the role by passing a role object through the pipeline. For example, you can use `Get-EVRBARole` to retrieve a role object and then pass the object through the pipeline to `Get-EVRBARoleMember`. If you do not specify the `-Identity` or the `-InputObject` parameter, you are prompted to enter the value for `-Identity`.

You can specify the Enterprise Vault Directory using the `-EVDirectoryServer` parameter. If you do not specify the `-EVDirectoryServer` parameter, and `Get-EVRBARoleMember` cannot determine the directory, you are prompted to enter a value for `-EVDirectoryServer`.

If you execute `Get-EVRBARoleMember` from a server other than an Enterprise Vault server, it prompts for the `-EVDirectoryServer` parameter, unless `-EVDirectoryServer` is already specified as a parameter.

Permission to execute this cmdlet is restricted to the Vault Service Account.

`Get-EVRBARoleMember` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded automatically by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVRBARoleMember [-Identity] <string> -EVDirectoryServer <string>
[<CommonParameters>]
```

```
Get-EVRBARoleMember [-InputObject]
<Symantec.EnterpriseVault.Admin.EVRbaRole> -EVDirectoryServer <string>
[<CommonParameters>]
```

## Parameters

**Table 4-57** Get-EVRBARoleMember parameters

Parameter	Description
<code>-Identity</code> (required)	Specifies an Enterprise Vault RBA role by its name or GUID.
<code>-InputObject</code> (required)	Specifies an Enterprise Vault RBA role object. This parameter can get this object through the pipeline or you can set this parameter to an object instance.
<code>-EVDirectoryServer</code> (required)	Specifies the host name, fully qualified domain name, or DNS alias of an Enterprise Vault Directory server. Note that, although this parameter is required, the cmdlet determines it automatically if it can.

## Examples

- `Get-EVRBARoleMember -Identity "File Server Administrator"`  
Gets the members of the Enterprise Vault RBA role called “File Server Administrator” from the current Enterprise Vault Directory.
- `Get-EVRBARoleMember "File Server Administrator"`  
Gets the members of the Enterprise Vault RBA role called “File Server Administrator” from the current Enterprise Vault Directory. Note that the `-Identity` parameter is a positioned parameter (at position 0), so this example works without it being specified.
- Additionally, `-EVDirectoryServer` parameter can be specified to connect to the desired Enterprise Vault directory.

```
Get-EVRBARoleMember "File Server Administrator" -EVDirectoryServer
evs2.example.com
```

Gets the members of the Enterprise Vault RBA role called “File Server Administrator” from the Enterprise Vault Directory identified by the `-EVDirectoryServer` parameter.

- String piping example:

```
"file server administrator" | Get-EVRBARoleMember
```

Gets the members of the Enterprise Vault RBA role called “File Server Administrator” from the current Enterprise Vault Directory.

- `-InputObject` usage example:

```
$r = Get-EVRBARole "file server administrator"
```

```
C:\PS> Get-EVRBARoleMember -InputObject $r
```

Gets the members of the Enterprise Vault RBA role as specified by the input object `$r` from the current Enterprise Vault Directory. Note that the parameter `-InputObject` is a positioned parameter (again at position 0), so this example will work without it being specified.

- `EVRbaRole` piping example:

```
Get-EVRBARole "file server administrator" | Get-EVRBARoleMember
```

Pipes the `EVRbaRole` object obtained from the `Get-EVRBARole` cmdlet to the `Get-EVRBARoleMember` cmdlet to get the role members for the specified identity.

## Output

`Get-EVRBARoleMember` returns objects of the type `Symantec.EnterpriseVault.Admin.EVRbaRoleMember`, which has the following properties.

**Table 4-58** Symantec.EnterpriseVault.Admin.EVRbaRoleMember properties

Name	Type	Description
Name	String	The NTAccount name of the member.
SID	String	The security identifier (SID) of the member.
MemberOf	String	The name of the role to which the member belongs.

## Related cmdlets

- See “[Get-EVRBARole](#)” on page 83.
- See “[Add-EVRBARoleMember](#)” on page 86.
- See “[Remove-EVRBARoleMember](#)” on page 92.

# Remove-EVRBARoleMember

`Remove-EVRBARoleMember` removes one or more Active Directory users, groups, or built-in security principals from an Enterprise Vault RBA role.

The `-Identity` parameter specifies the Enterprise Vault RBA role to access. You can identify an RBA role by its name or its GUID. You can also specify the role by passing a role object through the pipeline. For example, you can use `Get-EVRBARole` to retrieve a role object and then pass the object through the pipeline to `Remove-EVRBARoleMember`. If you do not specify the `-Identity` or the `-InputObject` parameter, you are prompted to enter the value for `-Identity`.

The `-Members` parameter specifies the members to remove from the role. You can identify a member by its security identifier (SID) or its SAM account name. If you are specifying more than one member, use a comma-separated list. You cannot pass member objects through the pipeline to this cmdlet. If you do not specify the `-Members` parameter, you are prompted to enter the value for `-Members`.

You can specify the Enterprise Vault Directory using the `-EVDirectoryServer` parameter.

If you do not specify the `-EVDirectoryServer` parameter, and `Remove-EVRBARoleMember` cannot determine the server, you are prompted to enter the Enterprise Vault directory server.

If you execute `Remove-EVRBARoleMember` from a server other than an Enterprise Vault server, it prompts for the `-EVDirectoryServer` parameter, unless `-EVDirectoryServer` is already specified as a parameter.

Permission to execute this cmdlet is restricted to the Vault Service Account.

`Remove-EVRBARoleMember` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Remove-EVRBARoleMember [-Identity] <string> [-Members] <string[]>  
-EVDirectoryServer <string> [-PassThru] [-Confirm] [-WhatIf]  
[<CommonParameters>]
```

```
Remove-EVRBARoleMember [-InputObject]  
<Symantec.EnterpriseVault.Admin.EVRbaRole> [-Members] <string[]>  
-EVDirectoryServer <string> [-PassThru] [-Confirm] [-WhatIf]  
[<CommonParameters>]
```

## Parameters

**Table 4-59** Remove-EVRBARoleMember parameters

Parameter	Description
-Identity (required)	Specifies an Enterprise Vault RBA role by its name or GUID.
-InputObject (required)	Specifies an Enterprise Vault RBA role object. This parameter can get this object through the pipeline or you can set this parameter to an object instance.
-Members (required)	<p>Specifies a set of Active Directory users, groups, or built-in security principal objects in a comma-separated list to remove from a role.</p> <p>To identify each object, use one of the following property values. Note that the identifier in parentheses is the LDAP display name. The acceptable values for this parameter are:</p> <ul style="list-style-type: none"> <li>■ Security Identifier (objectSid)</li> <li>■ SAM Account Name (sAMAccountName)</li> </ul> <p>The following example shows how to specify this parameter. This example specifies a user and a group by their SAM Account Name property and another user by its SID property.</p> <pre>-Members "fsaadmin1@example.com", "NT AUTHORITY\This Organization", "S-1-5-21-1529523603-1500826627-74573220-1119"</pre> <p>You cannot provide objects to this parameter. You cannot pass objects through the pipeline to this parameter.</p>
-EVDirectoryServer (required)	Specifies the host name, fully qualified domain name, or DNS alias of an Enterprise Vault directory server. Note that, although this parameter is required, the cmdlet determines it automatically if it can.
-PassThru	Returns an object representing the role with which you are working. By default, this cmdlet does not generate any output.
-Confirm	Prompts you for confirmation before running the cmdlet.
-WhatIf	Shows what would happen if the cmdlet runs. The cmdlet is not actually run.

## Examples

- `Remove-EVRBARoleMember -Identity "File Server Administrator" -Members "fsaAdmin1@example.com", "example\fsaGroup1", "S-1-5-21-1529523603-1500826627-74573220-1119"`  
Removes the specified Active Directory members from an Enterprise Vault RBA role called "File Server Administrator".
- `Remove-EVRBARoleMember "File Server Administrator"`  
Prompts for the members to be removed and removes them from an Enterprise Vault RBA role called "File Server Administrator". Note that the parameter `-Identity` is a positioned parameter (at position 0), so this example works without it being specified.
- Additionally, `-EVDirectoryServer` parameter can be specified to connect to the desired Enterprise Vault directory.  
`Remove-EVRBARoleMember "File Server Administrator" -EVDirectoryServer evs2.example.com`  
Prompts for the members to be removed and removes them from an Enterprise Vault RBA role called "File Server Administrator" in the Enterprise Vault Directory identified by the `-EVDirectoryServer` parameter.
- `-InputObject` usage example:  
`$r = Get-EVRBARole "file server administrator"`  
`Remove-EVRBARoleMember -InputObject $r`  
Prompts for the members to be removed and removes them from an Enterprise Vault RBA role called "File Server Administrator" in the current Enterprise Vault Directory. Note that the parameter `-InputObject` is a positioned parameter (again at position 0), so this example will work without it being specified.
- String piping example:  
`"file server administrator" | Remove-EVRBARoleMember`  
Prompts for the members to be removed and removes them from an Enterprise Vault RBA role called "File Server Administrator" in the current Enterprise Vault Directory.
- `EVRbaRole` piping example:  
`Get-EVRBARole "file server administrator" -EVDirectoryServer evs2.example.com | Remove-EVRBARoleMember -Members "fsaAdmin1", "fsaAdmin2" -EVDirectoryServer evs2.example.com`  
Pipes the `EVRbaRole` object obtained from the `Get-EVRBARole` cmdlet to the `Remove-EVRBARoleMember` cmdlet and removes the specified members from that role. Also note that the parameter `-EVDirectoryServer` specifies the Enterprise Vault Directory for each cmdlet to use.

## Output

`Remove-EVRBARoleMember` returns no output, or a modified object of the type `Symantec.EnterpriseVault.Admin.EVRbaRole` when the `-PassThru` parameter is specified. `Symantec.EnterpriseVault.Admin.EVRbaRole` has the following properties.

**Table 4-60** Symantec.EnterpriseVault.Admin.EVRbaRole properties

Name	Type	Description
Name	String	The name of the Enterprise Vault RBA role.
RoleGuid	String	The GUID of the Enterprise Vault RBA role.
TaskLinkDescription	String	The description of the associated TaskLink.
TaskLinkGuid	String	The GUID of the associated TaskLink.

## Related cmdlets

- See [“Get-EVRBARole”](#) on page 83.
- See [“Add-EVRBARoleMember”](#) on page 86.
- See [“Get-EVRBARoleMember”](#) on page 89.

# Get-EVRBAAzStoreXml

`Get-EVRBAAzStoreXml` downloads the RBA store from the Enterprise Vault Directory database to a file called `EvAzStore.xml` in the specified folder.

Specify the Enterprise Vault Directory database using the `-EVDirectoryServer` parameter. If the folder specified already contains a file called `EvAzStore.xml`, `Get-EVRBAAzStoreXml` overwrites it.

You can edit `EvAzStore.xml` using a suitable editor, then upload it to the Enterprise Vault Directory database using `Set-EVRBAAzStoreXml`.

Permission to execute this cmdlet is restricted to the Vault Service Account.

If you execute `Get-EVRBAAzStoreXml` from a server other than an Enterprise Vault server, it prompts for the `-EVDirectoryServer` parameter, unless `-EVDirectoryServer` is already specified as a parameter.

`Get-EVRBAAzStoreXml` is provided by

`Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVRBAAzStoreXml [-FolderPath] <string> -EVDirectoryServer <string>  
[<CommonParameters>]
```

## Parameters

**Table 4-61** Get-EVRBAAzStoreXml parameters

Parameter	Description
-FolderPath (required)	Specifies the folder to which EvAzStore.xml will be downloaded.
-EVDirectoryServer (required)	Specifies the host name, fully qualified domain name, or DNS alias of an Enterprise Vault Directory server. Note that although this parameter is required, the cmdlet determines it automatically if it can.

## Examples

- `Get-EVRBAAzStoreXml "C:\EvAzStoreXmlLocation"`  
Downloads EvAzStore.xml from the locally installed Enterprise Vault server, to C:\EvAzStoreXmlLocation.
- `Get-EVRBAAzStoreXml "C:\EvAzStoreXmlLocation" -EVDirectoryServer "evserver.example.com"`  
Downloads EvAzStore.xml from evserver.example.com to C:\EvAzStoreXmlLocation.
- `Get-EVRBAAzStoreXml -FolderPath "C:\EvAzStoreXmlLocation" -EVDirectoryServer "evserver.example.com"`  
Downloads EvAzStore.xml from evserver.example.com to C:\EvAzStoreXmlLocation. Note the use of named parameters.

## Output

None.

## Related cmdlets

- See "[Set-EVRBAAzStoreXml](#)" on page 97.

# Set-EVRBAAzStoreXml

Set-EVRBAAzStoreXml uploads the RBA store from the specified folder to the Enterprise Vault Directory database. Set-EVRBAAzStoreXml requires that the RBA store is in a file called `EvAzStore.xml`, and that it is in the folder you specify.

Specify the Enterprise Vault Directory using the `-EVDirectoryServer` parameter.

Permission to execute this cmdlet is restricted to the Vault Service Account.

If you execute the cmdlet from a server other than an Enterprise Vault server, the cmdlet prompts for the `-EVDirectoryServer` parameter, unless `-EVDirectoryServer` is already specified as a parameter.

You can only upload changes to latest version of `EvAzStore.xml`. If Enterprise Vault has a newer version of `EvAzStore.xml`, you must get the latest RBA store using `Get-EVRBAAzStoreXml`, make the changes and upload `EvAzStore.xml` again.

`Get-EVRBAAzStoreXml` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-EVRBAAzStoreXml [-FolderPath] <string> -EVDirectoryServer <string>
[<CommonParameters>]
```

## Parameters

**Table 4-62** Set-EVRBAAzStoreXml parameters

Parameter	Description
<code>-FolderPath</code> (required)	Specifies the folder from which <code>EvAzStore.xml</code> will be uploaded.
<code>-EVDirectoryServer</code> (required)	Specifies the host name, fully qualified domain name, or DNS alias of an Enterprise Vault Directory server. Note that although this parameter is required, the cmdlet determines it automatically if it can.

## Examples

- `Set-EVRBAAzStoreXml "C:\EvAzStoreXmlLocation"`  
Uploads `EvAzStore.xml` from `C:\EvAzStoreXmlLocation` to the locally installed Enterprise Vault server.
- `Set-EVRBAAzStoreXml "C:\EvAzStoreXmlLocation" -EVDirectoryServer "evserver.example.com"`

Uploads `EvAzStore.xml` from `C:\EvAzStoreXmlLocation` to `evserver.example.com`.

- `Set-EVRBAzStoreXml -FolderPath "C:\EvAzStoreXmlLocation" -EVDirectoryServer "evserver.example.com"`

Uploads `EvAzStore.xml` from `C:\EvAzStoreXmlLocation` to `evserver.example.com`. Note the use of named parameters.

## Output

None.

## Related cmdlets

- See “[Get-EVRBAzStoreXml](#)” on page 95.

# Get-EVRetentionPlan

`Get-EVRetentionPlan` returns a list of all the retention plans that are configured in an Enterprise Vault site. You can filter the list of returned retention plans by description, retention category, override retention flag, and classification policy. You can also return the properties of a specific retention plan.

`Get-EVRetentionPlan` is provided by

`Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVRetentionPlan [-SiteId <String>] [-Name <String>] [-Description <String>] [-RetentionCategory <String>] [-OverrideRetention [<SwitchParameter>]] [-ClassificationPolicy <String>] [<CommonParameters>]
```

## Parameters

**Table 4-63** Get-EVRetentionPlan parameters

Parameter	Description
<code>-SiteId</code> (required)	The ID of the Enterprise Vault site for which to return the retention plan details. If you omit this parameter, and the command cannot determine the ID by looking in the registry, then <code>Get-EVRetentionPlan</code> prompts you to enter the required ID.  You can use <code>Get-EVSite</code> to obtain the site ID.

**Table 4-63** Get-EVRetentionPlan parameters (*continued*)

Parameter	Description
-Name	The name of a specific retention plan whose properties you want to return.
-Description	The retention plan description with which to filter the list of returned plans.
-RetentionCategory	The name of a retention category with which to filter the list of returned retention plans.
-OverrideRetention	If specified, return only those retention plans for which you have set the override retention flag. This flag instructs Enterprise Vault to expire each item according to the retention category that you have set with the retention plan. This retention category overrides the item's individual retention category.  <b>Note:</b> The override retention flag does not take effect if there is an associated classification policy that allows the classification rules to update the retention category of items. In this case, Enterprise Vault expires the items according to the retention categories that the classification rules apply to them.
-ClassificationPolicy	The name of a classification policy with which to filter the list of returned retention plans.

## Examples

- `Get-EVRetentionPlan`  
Returns a list of all the retention plans that are configured in the Enterprise Vault site. As no site ID is specified, the command first looks for it in the registry and then, if it cannot find the ID there, prompts you for it.
- `Get-EVRetentionPlan -SiteId 13E...localdomain.com`  
Returns a list of all the retention plans that are configured in the specified Enterprise Vault site.
- `Get-EVRetentionPlan -SiteId 13E...localdomain.com -Name ManagersPlan`  
Returns the properties of the retention plan that is named "ManagersPlan".
- `Get-EVRetentionPlan -SiteId 13E...localdomain.com -Description RetentionPlanForManagers`  
Restricts the list of returned retention plans to those that have a description of "RetentionPlanForManagers".

- `Get-EVRetentionPlan -SiteId 13E...localdomain.com -RetentionCategory 3years`  
Restricts the list of returned retention plans to those that have a retention category of "3years".
- `Get-EVRetentionPlan -OverrideRetention`  
Restricts the list of returned retention plans to those for which you have set the override retention flag.
- `Get-EVRetentionPlan -ClassificationPolicy "Managers CP"`  
Restricts the list of returned retention plans to those that have a classification policy name of "Managers CP".
- `Get-EVRetentionPlan -ClassificationPolicy ""`  
Restricts the list of returned retention plans to those for which no classification policy has been specified.

## Output

[Table 4-64](#) lists the properties that are available.

**Table 4-64** Get-EVRetentionPlan properties

Name	Type	Description
Name	String	The name of the retention plan.
Description	String	The description of the retention plan.
RetentionCategory	String	The name of the retention category that is associated with the retention plan.
OverrideRetention	SwitchParameter	Whether you have specified the override retention flag for this retention plan.  Note that this flag does not take effect if there is an associated classification policy that allows the classification rules to update the retention categories of items.
ClassificationPolicy	String	The name of the classification policy that is associated with the retention plan. If specified, Enterprise Vault sends for classification all the archived items that have this retention plan and tags them according to the rules specified in the classification policy.

If no retention plans are available, or if the command cannot find any plans that match the specified criteria, then it outputs a suitable error message.

## Related cmdlets

- See “[New-EVRetentionPlan](#)” on page 101.
- See “[Remove-EVRetentionPlan](#)” on page 104.
- See “[Set-EVRetentionPlan](#)” on page 105.
- See “[Get-EVSite](#)” on page 112.

# New-EVRetentionPlan

`New-EVRetentionPlan` creates a retention plan for an Enterprise Vault site.

`New-EVRetentionPlan` is provided by

`Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
New-EVRetentionPlan [-SiteId <String>] [-Name <String>] [-Description
<String>] [-RetentionCategory <String>] [-OverrideRetention
[<SwitchParameter>]] [-ClassificationPolicy <String>]
[<CommonParameters>]
```

## Parameters

**Table 4-65** New-EVRetentionPlan parameters

Parameter	Description
<code>-SiteId</code> (required)	The ID of the Enterprise Vault site for which to create the retention plan. If you omit this parameter, and the command cannot determine the ID by looking in the registry, then <code>New-EVRetentionPlan</code> prompts you to enter the required ID.  You can use <code>Get-EVSite</code> to obtain the site ID.
<code>-Name</code> (required)	The name of the retention plan. The name must be unique, and it can contain up to 40 alphanumeric or space characters.
<code>-Description</code> (required)	The description of the retention plan. The description can contain up to 127 alphanumeric, space, or special characters.
<code>-RetentionCategory</code> (required)	The name of a retention category with which to associate the retention plan.

**Table 4-65** New-EVRetentionPlan parameters (*continued*)

Parameter	Description
<code>-OverrideRetention</code>	<p>Add this to override the current retention categories of the affected items with the retention category associated with this plan.</p> <p><b>Note:</b> This flag does not take effect if there is an associated classification policy that allows the classification rules to update the retention category of items. In this case, Enterprise Vault expires the items according to the retention categories that the classification rules apply to them.</p>
<code>-ClassificationPolicy</code>	<p>The name of a classification policy with which to associate the retention plan. If specified, Enterprise Vault sends for classification all the archived items that have this retention plan and tags them according to the rules specified in the classification policy.</p> <p><b>Note:</b> Classification operates in test mode if you have yet to install a license for the Enterprise Vault retention feature, or the existing license has expired. Instead of applying classification tags and other changes to the archived items, the classification feature generates a report that lists the planned changes.</p>

## Examples

- ```
New-EVRetentionPlan -SiteId 13E...localdomain.com -Name "Projects Retention Plan" -Description "Plan created for managing projects" -RetentionCategory "5 years"
```

Creates a retention plan that is named "Projects Retention Plan" in the specified Enterprise Vault site. The new plan has the description "Plan created for managing projects" and the retention category "5 years".
- ```
New-EVRetentionPlan -Name "Projects Retention Plan" -Description "Plan created for managing projects" -RetentionCategory "5 years" -OverrideRetention -ClassificationPolicy "Managers CP"
```

Creates a retention plan with the specified description and retention category. As no site ID is specified, the command first looks for it in the registry and then, if it cannot find the ID there, prompts you for it.

Enterprise Vault expires the affected items according to the "5 years" retention category rather than their individual retention categories. However, this is not the case if the "Managers CP" classification policy allows the classification rules to update the retention categories of the items. In this instance, Enterprise Vault

expires the items according to the retention categories that the classification rules apply to them.

## Output

Table 4-66 lists the properties that are available.

**Table 4-66** New-EVRetentionPlan properties

Name	Type	Description
Name	String	The name of the retention plan.
Description	String	The description of the retention plan.
RetentionCategory	String	The name of the retention category that is associated with the retention plan.
OverrideRetention	SwitchParameter	Whether you have specified the override retention flag for this retention plan. If specified, this flag causes Enterprise Vault to override the current retention categories of the affected items with the retention category associated with this plan.  <b>Note:</b> This flag does not take effect if there is an associated classification policy that allows the classification rules to update the retention category of items. In this case, Enterprise Vault expires the items according to the retention categories that the classification rules apply to them.
ClassificationPolicy	String	The name of a classification policy with which the retention plan is associated. If specified, Enterprise Vault sends for classification all the archived items that have this retention plan and tags them according to the rules specified in the classification policy.  <b>Note:</b> Classification operates in test mode if you have yet to install a license for the Enterprise Vault retention feature, or the existing license has expired. Instead of applying classification tags and other changes to the archived items, the classification feature generates a report that lists the planned changes.

## Related cmdlets

- See “[Get-EVRetentionPlan](#)” on page 98.
- See “[Remove-EVRetentionPlan](#)” on page 104.
- See “[Set-EVRetentionPlan](#)” on page 105.
- See “[Get-EVSite](#)” on page 112.

# Remove-EVRetentionPlan

`Remove-EVRetentionPlan` removes the specified retention plan, if it is not in use. The command prompts you to confirm the removal of the retention plan.

You cannot remove a retention plan while you have associated it with any of the following:

- An Exchange, Domino, or IMAP provisioning group
- An FSA volume or folder policy
- A public folder target
- A SharePoint target or site collection
- An archive to which you have assigned the retention plan through the `Set-EVArchive` PowerShell cmdlet

You must disassociate the retention plan from these entities and then run the appropriate provisioning or archiving task to remove the association with the target archives.

`Remove-EVRetentionPlan` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Remove-EVRetentionPlan [-SiteId <String>] [-Name <String>]  
[<CommonParameters>]
```

## Parameters

**Table 4-67** Remove-EVRetentionPlan parameters

Parameter	Description
-SiteId (required)	The ID of the Enterprise Vault site to which the retention plan belongs. If you omit this parameter, and the command cannot determine the ID by looking in the registry, then <code>Remove-EVRetentionPlan</code> prompts you to enter the required ID.  You can use <code>Get-EVSite</code> to obtain the site ID.
-Name (required)	The name of the retention plan to remove.

## Examples

- `Remove-EVRetentionPlan -SiteId 13E...localdomain.com -Name "PlanForCEOs"`  
Removes the retention plan that is named "PlanForCEOs".

## Output

There is a confirmation message on completion.

## Related cmdlets

- See [“Get-EVRetentionPlan”](#) on page 98.
- See [“New-EVRetentionPlan”](#) on page 101.
- See [“Set-EVRetentionPlan”](#) on page 105.
- See [“Set-EVArchive”](#) on page 23.
- See [“Get-EVSite”](#) on page 112.

# Set-EVRetentionPlan

`Set-EVRetentionPlan` sets or updates the properties of an existing retention plan.

`Set-EVRetentionPlan` is provided by

`Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-EVRetentionPlan [-SiteId <string>] [-Name <string>] [-Description <string>] [-RetentionCategory <string>] [-OverrideRetention
```

```
[<SwitchParameter>]] [-ClassificationPolicy <String>]  
[<CommonParameters>]
```

## Parameters

**Table 4-68** Set-EVRetentionPlan parameters

Parameter	Description
-SiteId (required)	The ID of the Enterprise Vault site for which to set or update the retention plan details. If you omit this parameter, and the command cannot determine the ID by looking in the registry, then <code>Set-EVRetentionPlan</code> prompts you to enter the required ID.  You can use <code>Get-EVSite</code> to obtain the site ID.
-Name (required)	The name of a specific retention plan whose properties you want to set or update.
-Description	The description to set for the retention plan. The description can contain up to 127 alphanumeric, space, or special characters.
-RetentionCategory	The name of a retention category with which to associate the retention plan.
-OverrideRetention	Add this to override the current retention categories of the affected items with the retention category associated with this plan.  <b>Note:</b> This flag does not take effect if there is an associated classification policy that allows the classification rules to update the retention category of items. In this case, Enterprise Vault expires the items according to the retention categories that the classification rules apply to them.

**Table 4-68** Set-EVRetentionPlan parameters (*continued*)

Parameter	Description
-ClassificationPolicy	<p>The name of a classification policy with which to associate the retention plan. If specified, Enterprise Vault sends for classification all the archived items that have this retention plan and tags them according to the rules specified in the classification policy.</p> <p><b>Note:</b> Classification operates in test mode if you have yet to install a license for the Enterprise Vault retention feature, or the existing license has expired. Instead of applying classification tags and other changes to the archived items, the classification feature generates a report that lists the planned changes.</p>

## Examples

- `Set-EVRetentionPlan -SiteId 13E...localdomain.com -Name "Projects Retention Plan" -Description "Plan created for managing projects" -RetentionCategory "5 years"`

Updates the properties of an existing retention plan, "Projects Retention Plan", in the specified Enterprise Vault site. The plan has the description "Plan created for managing projects" and the retention category "5 years".
- `Set-EVRetentionPlan -Name "Projects Retention Plan" -ClassificationPolicy CP`

Updates the classification policy property of the specified retention plan.
- `Set-EVRetentionPlan -Name "Projects Retention Plan" -ClassificationPolicy CP -OverrideRetention`

Sets the override retention flag for the specified retention plan. When items are manually deleted or automatically expired, the retention category that you have set with the plan overrides their current retention categories. Note that this flag does not take effect if the "CP" classification policy allows the classification rules to update the retention categories of items. In this case, Enterprise Vault expires the items according to the retention categories that the classification rules apply to them.
- `Set-EVRetentionPlan -SiteId 13E9...localdomain.com -Name "Projects Retention Plan" -Description "Plan updated for managing projects" -RetentionCategory "6 years" -OverrideRetention -ClassificationPolicy CP2`

Updates the properties of an existing retention plan that is named "Projects Retention Plan" in the specified Enterprise Vault site. The plan has the description "Plan updated for managing projects" and the retention category "6 years".

- `Set-EVRetentionPlan -SiteId 13E...localdomain.com -Name "Projects Retention Plan" -ClassificationPolicy ''`

Removes the previously-applied classification policy so that classification no longer occurs.

## Output

[Table 4-69](#) lists the properties that are available.

**Table 4-69** Set-EVRetentionPlan properties

Name	Type	Description
Name	String	The name of the retention plan.
Description	String	The description of the retention plan.
RetentionCategory	String	The name of the retention category that is associated with the retention plan.
OverrideRetention	SwitchParameter	Whether you have specified the override retention flag for this retention plan. If specified, this flag causes Enterprise Vault to override the current retention categories of the affected items with the retention category associated with this plan.  <b>Note:</b> This flag does not take effect if there is an associated classification policy that allows the classification rules to update the retention category of items. In this case, Enterprise Vault expires the items according to the retention categories that the classification rules apply to them.

**Table 4-69** Set-EVRetentionPlan properties (*continued*)

Name	Type	Description
ClassificationPolicy	String	<p>The name of a classification policy with which the retention plan is associated. If specified, Enterprise Vault sends for classification all the archived items that have this retention plan and tags them according to the rules specified in the classification policy.</p> <p><b>Note:</b> Classification operates in test mode if you have yet to install a license for the Enterprise Vault retention feature, or the existing license has expired. Instead of applying classification tags and other changes to the archived items, the classification feature generates a report that lists the planned changes.</p>

### Related cmdlets

- See “[Get-EVRetentionPlan](#)” on page 98.
- See “[New-EVRetentionPlan](#)” on page 101.
- See “[Remove-EVRetentionPlan](#)” on page 104.
- See “[Get-EVSite](#)” on page 112.

## Get-EVServer

Gets information about the current Enterprise Vault server.

`Get-EVServer` is provided by `Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

### Syntax

```
Get-EVServer [<CommonParameters>]
```

### Parameters

PowerShell common parameters only.

### Examples

- `Get-EVServer`  
Gets details of the current Enterprise Vault server.

## Output

[Table 4-70](#) lists the properties that are available.

**Table 4-70** Get-EVServer properties

Name	Type	Description
Aliases	List<String>	Alias names for the Enterprise Vault server.
ID	String	The server Id.
Name	String	The server name.
Version	String	The version of Enterprise Vault that is on the server.

## Get-EVService

Gets details of the Enterprise Vault services that are configured on the current Enterprise Vault server.

`Get-EVService` is provided by `Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

### Syntax

```
Get-EVService [<CommonParameters>]
```

### Parameters

PowerShell common parameters only.

### Examples

- `Get-EVService`  
Lists the Enterprise Vault services that are configured on the current Enterprise Vault server.

### Output

[Table 4-71](#) lists the properties that are available.

**Table 4-71** Get-EVService properties

Name	Type	Description
DisplayName	String	The service display name as shown in the Administration Console. For example, "Enterprise Vault Directory Service".
EntryID	String	The service EntryID.
Name	String	The service name. For example, "EnterpriseVaultAdminService".
Type	ServiceType	One of the following: "DirectoryService"; "AdminService"; "ShoppingService"; "IndexingService"; "StorageService"; "TaskControllerService"

## Get-EVServiceState

Gets the current state of the specified Enterprise Vault service. The state can be either "Running" or "Stopped".

`Get-EVServiceState` is provided by

`Symantec.EnterpriseVault.PowerShell.Monitoring.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

### Syntax

```
Get-EVServiceState [-ServiceName] <String> [<CommonParameters>]
```

### Parameters

[Table 4-72](#) lists the properties that are available.

**Table 4-72** Get-EVServiceState properties

Parameter	Description
<code>-ServiceName</code>	The name of the service whose details you require.

### Examples

- `Get-EVServiceState -ServiceName EnterpriseVaultShoppingService`  
Returns the current state of the `EnterpriseVaultShoppingService`. You can use the `Get-EVService` to find the names of the Enterprise Vault services.

## Output

Table 4-73 lists the properties that are available.

**Table 4-73** Get-EVServiceState properties

Name	Type	Description
Value	Service Controller Status	One of the following: "Stopped"; "StartPending"; "StopPending"; "Running"; "ContinuePending"; "PausePending"; "Paused".

## Related cmdlets

- See ["Get-EVService"](#) on page 110.

# Get-EVSite

Gets the Enterprise Vault site information for the current Enterprise Vault server.

`Get-EVSite` is provided by `Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

## Syntax

```
Get-EVSite [<CommonParameters>]
```

## Parameters

PowerShell common parameters only.

## Examples

- `Get-EVSite`  
Gets details of the Vault Site of which the current server is a member.

## Output

Table 4-74 lists the properties that are available.

**Table 4-74** Get-EVSite properties

Name	Type	Description
DirectorySQLServer	String	The name of the SQL Server computer that hosts the Directory database.

**Table 4-74** Get-EVSite properties (*continued*)

Name	Type	Description
ID	String	The Site ID.
Name	String	The name of the Enterprise Vault site.

## Get-EVSMTPHoldingFolder

`Get-EVSMTPHoldingFolder` displays details of the SMTP holding folder that is configured on the Enterprise Vault server where you run the command. If the SMTP Archiving task is not configured, `Get-EVSMTPHoldingFolder` produces no output. If the SMTP Archiving task cannot locate the holding folder, the total values displayed by `Get-EVSMTPHoldingFolder` are zero.

`Get-EVSMTPHoldingFolder` is provided by `Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this module.

### Syntax

```
Get-EVSMTPHoldingFolder [<CommonParameters>]
```

### Parameters

PowerShell common parameters only.

### Examples

- `Get-EVSMTPHoldingFolder`  
Displays details of the SMTP holding folder that is configured on the Enterprise Vault server where you run the command.

### Output

`Get-EVSMTPHoldingFolder` returns an object of the type `Symantec.EnterpriseVault.PowerShell.Core.Output.SMTPHoldingFolder`, which contains the properties listed in the following table.

**Table 4-75** Get-EVSMTPHoldingFolder properties

Name	Type	Description
TaskName	String	The name of the SMTP Archiving task that is using the holding folder.

**Table 4-75** Get-EVSMTPHoldingFolder properties (*continued*)

Name	Type	Description
TaskEntryId	String	The directory entry ID of the SMTP Archiving task that is using the holding folder.
Path	String	The full path to the SMTP holding folder.
TotalQuotaGBytesFree	Double	The total space unused, in gigabytes
TotalQuotaGBytesUsed	Double	The total space used, in gigabytes.
TotalQuotaGBytesAvailable	Double	The total space available, in gigabytes.

## Get-EVSMTPPolicy

`Get-EVSMTPPolicy` retrieves details of all the SMTP policies for the Enterprise Vault site you specify. If you do not specify a site ID using the `-SiteID` parameter, and `Get-EVSMTPPolicy` cannot determine the site, you are prompted to enter the site ID. You can also retrieve the properties of a specific policy using the `-Name` parameter.

`Get-EVSMTPPolicy` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

### Syntax

```
Get-EVSMTPPolicy [-SiteId] <string> [[-Name] <string>]
[<CommonParameters>]
```

### Parameters

**Table 4-76** Get-EVSMTPPolicy parameters

Parameter	Description
<code>-SiteId</code> (required)	The ID of the site whose policies you want to retrieve.
<code>-Name</code>	The name of the SMTP policy you want to retrieve.

### Examples

- `Get-EVSMTPPolicy -SiteId 13E...example.com`  
Retrieves all the SMTP policies configured in the specified Enterprise Vault site.
- `Get-EVSMTPPolicy -SiteId 13E...example.com -Name "SMTP policy"`

Or:

```
Get-EVSMTPPolicy "SMTP policy"
```

Retrieves the properties of the SMTP policy, "SMTP policy". For example:

```
SiteId           : 13E...example.com
Name             : SMTP policy
Description      : SMTP policy
IsADefaultPolicy : True
EntryId         : 125...example.com
```

- You can use the following sequence of commands to view the properties of an SMTP policy and its X-Header details:

- `$sp = Get-EVSMTPPolicy 13E...example.com "SMTP policy"`

Retrieves the SMTP policy into the `$sp` variable.

- `$sp`

Displays the information contained in the `$sp` variable. For example:

```
SiteId           : 13E...example.com
Name             : SMTP policy
Description      :
IsADefaultPolicy : False
EntryId         : 170...example.com
```

- `$sp.XHeaders`

Displays X-Headers. For example:

```
HeadersList
-----
{x-sender}
```

- `$sp.XHeaders.HeadersList`

Displays a detailed listing of X-Headers. For example:

Name	ValueType	Searchable	Retrievable
x-sender	0	0	0

- You can use the following sequence of commands to list the SMTP targets associated with the specified SMTP policy:

- `$sp = Get-EVSMTPPolicy 13E...example.com "SMTP policy"`

Retrieves the SMTP policy into the `$sp` variable.

- `$sp.GetTargets()`

Lists all the SMTP targets associated with the policy. For example:

```
user1@example.com
user2@example.com
user3@example.com
```

## Output

`Get-EVSMTPPolicy` returns objects of the type

`Symantec.EnterpriseVault.Admin.SMTPPolicy`, which contains the properties listed in the following table.

**Table 4-77** Symantec.EnterpriseVault.Admin.SMTPPolicy properties

Name	Type	Description
Description	String	The description of the SMTP policy.
EntryId	String	The directory entry ID of the SMTP policy.
IsADefaultPolicy	Boolean	<code>\$true</code> if the SMTP policy is a default policy, or <code>\$false</code> if it is not.
Name	String	The name of the SMTP policy.
SiteId	String	The site ID to which the SMTP policy belongs.
XHeaders	Object	<code>Symantec.EnterpriseVault.Admin.SMTPXHeaders</code> : Provides a list of X-Headers associated with the SMTP policy. For more information on X-Header lists, type <code>get-help about_SMTPXHeaders</code> at the PowerShell prompt.

## Related cmdlets

- See [“New-EVSMTPPolicy”](#) on page 116.
- See [“Remove-EVSMTPPolicy”](#) on page 118.
- See [“Set-EVSMTPPolicy”](#) on page 119.

# New-EVSMTPPolicy

`New-EVSMTPPolicy` creates a new SMTP policy with the specified name, description, and X-Header list. If you do not specify a site ID using the `-SiteID` parameter, and `New-EVSMTPPolicy` cannot determine the site, you are prompted to enter the site ID. `New-EVSMTPPolicy` fails if you specify a policy name that already exists.

New-EVSMTPPolicy is provided by

Symantec.EnterpriseVault.PowerShell.AdminAPI.dll, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
New-EVSMTPPolicy [-SiteId] <string> [-Name] <string> [-Description
<string>] [-XHeaders <Symantec.EnterpriseVault.Admin.SMTPXHeaders>]
[<CommonParameters>]
```

## Parameters

**Table 4-78** New-EVSMTPPolicy parameters

Parameter	Description
-SiteId (required)	The ID of the site to which you want to add the new SMTP policy.
-Name (required)	The name of the new SMTP policy.
-Description	The description for the new SMTP policy.
-XHeaders	The X-Headers to add to the new SMTP policy.  For more information about the management of X-Header lists, type <code>get-help about_SMTPXHeaders</code> at the PowerShell prompt.

## Examples

- `New-EVSMTPPolicy -SiteId 13E...example.com -Name "SMTP policy" -Description "SMTP policy created using PowerShell"`  
This command creates a new SMTP policy called “SMTP policy”, with the description “SMTP policy created using PowerShell”.
- You can use the following sequence of commands to create an X-Headers object, add two X-Headers to the object, then create a new SMTP policy:
  - `$xheaders = New-Object Symantec.EnterpriseVault.Admin.SMTPXHeaders`  
Creates an X-Headers object.
  - `$xheaders.Add('x-code1')`  
`$xheaders.Add('x-code2')`  
These commands add X-Headers “x-code1” and “x-code2” to the X-Headers object.
  - `New-EVSMTPPolicy -Name "SMTP policy" -XHeaders $xheaders`  
Creates a new SMTP policy called “SMTP policy”, adds the new X-Headers to it, and produces the following output:

```

SiteId           : 13E...example.com
EntryId         : 11E...example.com
IsADefaultPolicy : False
XHeaders        : Symantec.EnterpriseVault.Admin.SMTPXHeaders
Name            : SMTP policy
Description     :

```

## Output

`New-EVSMTPPolicy` returns an object of the type `Symantec.EnterpriseVault.Admin.SMTPPolicy`, which contains the properties listed in the following table.

**Table 4-79** Symantec.EnterpriseVault.Admin.SMTPPolicy properties

Name	Type	Description
Description	String	The description of the SMTP policy.
EntryId	String	The directory entry ID of the SMTP policy.
IsADefaultPolicy	Boolean	\$true if the SMTP policy is a default policy, or \$false if it is not.
Name	String	The name of the SMTP policy.
SiteId	String	The site ID to which the SMTP policy belongs.
XHeaders	Object	Symantec.EnterpriseVault.Admin.SMTPXHeaders: Provides a list of X-Headers associated with the SMTP policy. For more information on X-Header lists, type <code>get-help about_SMTPXHeaders</code> at the PowerShell prompt.

## Related cmdlets

- See [“Get-EVSMTPPolicy”](#) on page 114.
- See [“Remove-EVSMTPPolicy”](#) on page 118.
- See [“Set-EVSMTPPolicy”](#) on page 119.

# Remove-EVSMTPPolicy

`Remove-EVSMTPPolicy` deletes the SMTP policy you specify. You are prompted to confirm the deletion. If you do not specify a site ID using the `-SiteID` parameter, and `Remove-EVSMTPPolicy` cannot determine the site, you are prompted to enter the site ID.

`Remove-EVSMTPPolicy` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Remove-EVSMTPPolicy [-SiteId] <string> [-Name] <string>  
[<CommonParameters>]
```

## Parameters

**Table 4-80** Remove-EVSMTPPolicy parameters

Parameter	Description
<code>-SiteId</code> (required)	The ID of the site to which the SMTP policy belongs.
<code>-Name</code> (required)	The name of SMTP policy you want to remove.

## Examples

- `Remove-EVSMTPPolicy -SiteId 13E...example.com -Name "SMTP policy"`  
Deletes the SMTP policy called “SMTP policy”.
- You can use the following sequence of commands to add the SMTP policy called “SMTP policy” into the variable `$sp`, then delete it.
  - `$sp = Get-EVSMTPPolicy 13E...example.com 'SMTP policy'`
  - `Remove-EVSMTPPolicy -AdminObject $sp`

## Output

`Remove-EVSMTPPolicy` returns no output.

## Related cmdlets

- See [“Get-EVSMTPPolicy”](#) on page 114.
- See [“New-EVSMTPPolicy”](#) on page 116.
- See [“Set-EVSMTPPolicy”](#) on page 119.

# Set-EVSMTPPolicy

`Set-EVSMTPPolicy` sets or updates the properties of an existing SMTP policy. If you do not specify a site ID using the `-SiteID` parameter, and `Set-EVSMTPPolicy` cannot determine the site, you are prompted to enter the site ID. If you specify X-Headers, `Set-EVSMTPPolicy` overwrites the existing X-Headers in the policy.

Set-EVSMTPPolicy is provided by Symantec.EnterpriseVault.PowerShell.AdminAPI.dll, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-EVSMTPPolicy [-SiteId] <string> [-Name] <string> [-Description <string>] [-XHeaders <Symantec.EnterpriseVault.Admin.SMTPXHeaders>] [<CommonParameters>]
```

## Parameters

**Table 4-81** Set-EVSMTPPolicy parameters

Parameter	Description
-SiteId (required)	The ID of the site to which the SMTP policy belongs.
-Name (required)	The name of the SMTP policy you want to update.
-Description	The new description for the SMTP policy.
-XHeaders	The new X-Header list you want to add to the SMTP policy. Set-EVSMTPPolicy automatically removes any existing X-Headers, and replaces them with the new list.  For more information about the management of X-Header lists, type <code>get-help about_SMTPXHeaders</code> at the PowerShell prompt.

## Examples

- `Set-EVSMTPPolicy -SiteId '13E...example.com' -Name 'SMTP policy' -Description 'SMTP journal archiving policy'`  
Updates the description of the SMTP policy called “SMTP policy”. The existing X-Headers are unchanged.
- You can use the following sequence of commands to add a new X-Header to the SMTP policy called “SMTP policy”:
  - `$xh = New-Object Symantec.EnterpriseVault.Admin.SMTPXHeader 'x-Encoding'`  
The `New-Object` cmdlet creates a new X-Header object with the following properties: Name: 'x-Encoding', ValueType: 0 (string), Searchable: 0 (not searchable), Retrievable: 0 (not retrievable).
  - `$sp = Get-EVSMTPPolicy '13E...example.com' 'SMTP policy'`  
Retrieves the SMTP policy into the `$sp` variable.
  - `$sp.XHeaders.Add($xh)`

Adds the newly created X-Header to the list of X-Headers using the `Add()` function of the `X-Headers` object. `$sp` now holds the modified SMTP policy.

- `Set-EVSMTPPolicy -AdminObject $sp`  
Or:  
`Set-EVSMTPPolicy -SiteId '13E...example.com' -Name 'SMTP policy' -X-Headers $sp.X-Headers`  
Overwrites the Enterprise Vault copy of the SMTP policy with the new version. The first use of `Set-EVSMTPPolicy` updates the whole policy object. The second use of `Set-EVSMTPPolicy` updates the policy object according to the parameters that are provided.
- You can use the following sequence of commands to modify the X-Header that is already associated with the SMTP policy called “SMTP policy”.
  - `$sp = Get-EVSMTPPolicy 13E...example.com "SMTP policy"`  
Retrieves the SMTP policy into the `$sp` variable.
  - `$xh = $sp.X-Headers.GetXHeader('x-Encoding')`  
Retrieves the X-Header called “x-Encoding” into the `$xh` variable using `GetXHeader()` function of the `X-Headers` object.
  - `$xh.Name = 'x-Encoder'`  
`$xh.Searchable = 0`  
Updates the in-memory copy of the X-Header with new values for the `Name` property and then the `Searchable` property.
  - `$sp.X-Headers.SetXHeader($xh.Name, $xh.ValueType, $xh.Searchable, $xh.Retrievable)`  
Or:  
`$sp.X-Headers.SetXHeader($xh)`  
Updates the in-memory X-Header list of the SMTP policy with the in-memory copy of the X-Header.
  - `Set-EVSMTPPolicy -AdminObject $sp`  
Updates the SMTP policy with the in-memory copy of the policy.

## Output

`Set-EVSMTPPolicy` returns no output.

## Related cmdlets

- See [“Get-EVSMTPPolicy”](#) on page 114.
- See [“New-EVSMTPPolicy”](#) on page 116.
- See [“Remove-EVSMTPPolicy”](#) on page 118.

# Get-EVSMTPServerSettings

`Get-EVSMTPServerSettings` displays the settings for all Enterprise Vault SMTP servers in the specified site. If you do not specify a site ID, `Get-EVSMTPServerSettings` uses the ID of the site to which the local Enterprise Vault server belongs. If you run the command a remote computer, `Get-EVSMTPServerSettings` prompts for a site ID.

`Get-EVSMTPServerSettings` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVSMTPServerSettings [[-SiteId] <string>] [<CommonParameters>]
```

## Parameters

**Table 4-82** Get-EVSMTPServerSettings parameters

Parameter	Description
<code>-SiteId</code>	The ID of the site to which the SMTP settings belong.

## Examples

- `Get-EVSMTPServerSettings -SiteId 13E...example.com`  
This command returns the SMTP server settings that apply to all the SMTP servers in the specified site.
- You can use the following sequence of commands to view the SMTP server settings for the site:

- `$sp = Get-EVSMTPServerSettings 13E...example.com`

Retrieves the SMTP server settings into the `$sp` variable.

- `$sp`

Displays the information held in the `$sp` variable. For example:

```
SiteId                : 13E...example.com
Port                  : 25
MaxMessageSize       : 0
Authentication        : EV_SMTP_AUTHENTICATION_ENABLE
AuthenticationEmailAddress : user@example.com
AuthenticationAllowPlainText : EV_SMTP_AUTHENTICATION
                        _ALLOWPLAINTEXT_TLS
Tls                   : EV_SMTP_TLS_ENCRYPTED
```

```
TlsCertificateInfo           : Symantec.EnterpriseVault.Admin
                             .TlsCertificate
SMTPConnectionControl       : Symantec.EnterpriseVault.Admin
                             .SMTPConnectionControlList
```

- `$sp.SMTPConnectionControl.SMTPConnectionList`

Displays the connection control list. The list contains the names or IP addresses of the hosts that are allowed to connect to the Enterprise Vault SMTP servers. The host names should be fully qualified domain names.

```
Connections
-----
{test.example.com}
```

- `$sp.SMTPConnectionControl.SMTPConnectionList.Connections`

Displays a detailed list of connection controls. Type shows the value of the `_EV_SMTP_CONNECTION` enumeration.

```
Type                               Value
----                               -
_EV_SMTP_CONNECTION_HOSTNAME      test.example.com
```

## Output

`Get-EVSMTPServerSettings` returns an object of the type `Symantec.EnterpriseVault.Admin.SMTPServerSettings`, which contains the properties listed in the following table.

**Table 4-83** Symantec.EnterpriseVault.Admin.SMTPServerSettings properties

Name	Type	Description
Authentication	Object	<code>Symantec.EnterpriseVault.Admin._EV_SMTP_AUTHENTICATION</code> enumeration value: Specifies whether authentication is required for incoming connections. For details of the enumeration values, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt.
AuthenticationEmailAddress	String	The identity of the authentication user, if authentication is enabled.

**Table 4-83** Symantec.EnterpriseVault.Admin.SMTPServerSettings properties  
(continued)

Name	Type	Description
AuthenticationAllowPlainText	Object	Symantec.EnterpriseVault.Admin. _EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT enumeration value: Specifies whether plain text authentication is allowed for incoming connections. If authentication is disabled this value defaults to EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT_NEVER. For details of the enumeration values, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt.
MaxMessageSize	Integer	Maximum SMTP message size (in MB).
Port	Integer	The port number on which the SMTP servers listen.
SiteId	String	The ID of the site to which the SMTP server settings belong.
SMTPConnectionControl	Object	Symantec.EnterpriseVault.Admin.SMTPConnectionControlList: Provides a list of allowed connections to the SMTP servers. For more information on managing the list of allowed connections, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt.
Tls	Object	Symantec.EnterpriseVault.Admin._EV_SMTP_TLS enumeration value: Enumeration value that specifies the support for encrypted and unencrypted connections to the SMTP servers. For details of the enumeration values, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt.
TlsCertificateInfo	Object	Symantec.EnterpriseVault.Admin.TlsCertificate: Provides information about the TLS certificate used for SMTP connections.

## Related cmdlets

- See [“New-EVSMTPServerSettings”](#) on page 124.
- See [“Set-EVSMTPServerSettings”](#) on page 130.
- See [“Sync-EVSMTPServerSettings”](#) on page 135.

# New-EVSMTPServerSettings

`New-EVSMTPServerSettings` creates the initial settings for all Enterprise Vault SMTP servers in the specified site. If you do not specify a site ID using the `-SiteID` parameter, and `New-EVSMTPServerSettings` cannot determine the site, you are prompted to enter the site ID.

This command fails if SMTP settings for site already exist.

New-EVSMTPServerSettings is provided by Symantec.EnterpriseVault.PowerShell.AdminAPI.dll, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
New-EVSMTPServerSettings [[-SiteId] <string>] -Port <integer>
[-MaxMessageSize <integer>] -Authentication
<Symantec.EnterpriseVault.Admin._EV_SMTP_AUTHENTICATION enumeration
value> [-AuthenticationCredentials
<System.Management.Automation.PSCredential>]
-AuthenticationAllowPlainText
<Symantec.EnterpriseVault.Admin._EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT
enumeration value> -Tls <Symantec.EnterpriseVault.Admin._EV_SMTP_TLS
enumeration value> [-TlsCertificateFile <string>]
[-TlsSecurePassphrase <System.Security.SecureString>]
[-SMTPConnectionControl
<Symantec.EnterpriseVault.Admin.SMTPConnectionControlList>]
[<CommonParameters>]
```

## Parameters

**Table 4-84** New-EVSMTPServerSettings parameters

Parameter	Description
-SiteId	The ID of the site to which the SMTP settings apply.
-Port (required)	The port number to be used by the SMTP servers in the site.
-MaxMessageSize	The maximum message size that the SMTP servers accept (in MB). The SMTP servers will reject messages that are larger than the specified size.

**Table 4-84** New-EVSMTPServerSettings parameters (*continued*)

Parameter	Description
-Authentication (required)	<p>The enumeration value specifies whether authentication is enabled for SMTP connections to the SMTP servers. If the value <code>EV_SMTP_AUTHENTICATION_ENABLE</code> is specified, hosts that connect to the SMTP servers must supply the credentials specified by the parameter <code>-AuthenticationCredentials</code>. If the value <code>EV_SMTP_AUTHENTICATION_DISABLE</code> is specified, then <code>-AuthenticationCredentials</code> is not required. For details of the enumerations used, type <code>get-help about_SMTPEnumerations</code>.</p>
-AuthenticationCredentials	<p>The credentials that hosts must use when connecting to SMTP servers in the site. This parameter is only required if <code>-Authentication</code> is set to <code>EV_SMTP_AUTHENTICATION_ENABLE</code>.</p>
-AuthenticationAllowPlainText (required)	<p>Enumeration value that specifies whether plain text authentication is enabled for SMTP connections to the SMTP servers. If <code>-Authentication</code> is set to <code>EV_SMTP_AUTHENTICATION_DISABLE</code>, this value is always set to <code>EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT_NEVER</code>.</p> <p>If <code>EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT_TLS</code> is specified, then TLS must be enabled on the SMTP servers or an appropriate exception message is generated.</p> <p>For details of the enumerations used, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt.</p>

**Table 4-84** New-EVSMTPServerSettings parameters (*continued*)

Parameter	Description
-Tls (required)	The enumeration value specifies the support for encrypted and unencrypted connections to the SMTP servers. The protocol supported for encrypted connections is Transport Layer Security (TLS). If encrypted connections are supported, then you must specify values for the parameters <code>-TlsCertificateFile</code> and <code>-TlsSecurePassphrase</code> . For details of the enumerations used, type <code>get-help about_SMTPEnumerations</code> .
-TlsCertificateFile	The path of the TLS certificate file to use for encrypted connections to the SMTP servers. This parameter is not required if only unencrypted connections are supported.
-TlsSecurePassphrase	The TLS certificate file secure passphrase required for encrypted connections to the SMTP servers. This parameter is not required if only unencrypted connections are supported.
-SMTPConnectionControl	A list of hosts that can connect to the SMTP servers. For details of <code>SMTPConnectionControlList</code> object, type <code>get-help about_SMTPConnectionControlList</code> .

## Examples

- ```

New-EVSMTPServerSettings -Port 25 -Authentication
EV_SMTP_AUTHENTICATION_ENABLE -AuthenticationCredentials
(New-Object System.Management.Automation.PSCredential -ArgumentList
"user@example.com", ("admin@123" | ConvertTo-SecureString
-AsPlainText -Force)) -AuthenticationAllowPlainText
EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT_TLS -Tls
EV_SMTP_TLS_ENCRYPTED_AND_UNENCRYPTED -TlsCertificateFile
"C:\rsa.p12" -TlsSecurePassphrase ("admin@123" |
ConvertTo-SecureString -AsPlainText -Force)

```

Creates the initial SMTP server settings for all SMTP servers in the specified site. The SMTP servers listen on port 25, authentication is enabled, plain text authentication is enabled on TLS, and both encrypted and unencrypted connections are supported.

```
■ C:\PS> $xConnectionList = new-object
Symantec.EnterpriseVault.Admin.SMTPConnectionControlList
C:\PS> $xConnectionList.Add("EV_SMTP_CONNECTION_HOSTNAME", "Test")
C:\PS> $xConnectionList.Add("EV_SMTP_CONNECTION_HOSTNAMESUFFIX",
"smtp.example.com")
C:\PS> $xConnectionList.Add("EV_SMTP_CONNECTION_HOSTNAMEPATTERN",
"as-[a-z0-9]*-[a-z0-9]*.example.local")
C:\PS> $xConnectionList.Add("EV_SMTP_CONNECTION_IPV4",
"192.168.1.1")
C:\PS> $xConnectionList.Add("EV_SMTP_CONNECTION_IPV4INCIDR",
"192.168.1.1/24")
C:\PS> $xConnectionList.Add("EV_SMTP_CONNECTION_IPV6",
"0000:0000:0000:0000:0000:0000:0000:0001")
C:\PS> $xConnectionList.Add("EV_SMTP_CONNECTION_IPV6INCIDR",
"0000:0000:0000:0000:0000:0000:0000:0001/128")
C:\PS> New-EVSMTPServerSettings -Port 25 -Authentication
EV_SMTP_AUTHENTICATION_ENABLE -AuthenticationCredentials
(New-Object System.Management.Automation.PSCredential -ArgumentList
"user@example.com", ("admin@123" | ConvertTo-SecureString
-AsPlainText -Force)) -Tls EV_SMTP_TLS_ENCRYPTED_AND_UNENCRYPTED
-TlsCertificateFile "C:\rsa.pl2" -TlsSecurePassphrase ("admin@123"
| ConvertTo-SecureString -AsPlainText -Force)
-SMTPConnectionControl $xConnectionList

SiteId                : 13E9...EV1101.example.com
Port                  : 25
MaxMessageSize       : 0
Authentication       : EV_SMTP_AUTHENTICATION_ENABLE
AuthenticationEmailAddress : user@example.com
AuthenticationAllowPlainText : EV_SMTP_AUTHENTICATION
                        _ALLOWPLAINTEXT_TLS
Tls                  : EV_SMTP_TLS_ENCRYPTED_AND_UNENCRYPTED
TlsCertificateInfo   : Symantec.EnterpriseVault.Admin
                        .TlsCertificate
SMTPConnectionControl : Symantec.EnterpriseVault.Admin
                        .SMTPConnectionControlList
```

This sequence of commands creates the initial SMTP server settings for all SMTP servers in the specified site. The SMTP servers listen on port 25, authentication is enabled, plain text authentication is enabled on TLS, and both encrypted and unencrypted connections are supported. The allowed connections

are added to the SMTPConnectionControlList object using different address formats.

## Output

New-EVSMTPServerSettings returns an object of the type `Symantec.EnterpriseVault.Admin.SMTPServerSettings`, which contains the properties listed in the following table.

**Table 4-85** Symantec.EnterpriseVault.Admin.SMTPServerSettings properties

| Name                         | Type    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Authentication               | Object  | <code>Symantec.EnterpriseVault.Admin._EV_SMTP_AUTHENTICATION</code> enumeration value: Specifies whether authentication is required for incoming connections. For details of the enumeration values, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt.                                                                                                                                                                               |
| AuthenticationEmailAddress   | String  | The identity of the authentication user, if authentication is enabled.                                                                                                                                                                                                                                                                                                                                                                                         |
| AuthenticationAllowPlainText | Object  | <code>Symantec.EnterpriseVault.Admin._EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT</code> enumeration value: Enumeration value that specifies whether plain text authentication is allowed for incoming connections.<br><br>If authentication is disabled this value defaults to <code>EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT_NEVER</code> .<br><br>For details of the enumeration values, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt. |
| MaxMessageSize               | Integer | Maximum SMTP message size (in MB).                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Port                         | Integer | The port number on which the SMTP servers listen.                                                                                                                                                                                                                                                                                                                                                                                                              |
| SiteId                       | String  | The ID of the site to which the SMTP server settings belong.                                                                                                                                                                                                                                                                                                                                                                                                   |
| SMTPConnectionControl        | Object  | <code>Symantec.EnterpriseVault.Admin.SMTPConnectionControlList</code> : Provides a list of allowed connections to the SMTP servers. For more information on managing the list of allowed connections, type <code>get-help about_SMTPConnectionControlList</code> at the PowerShell prompt.                                                                                                                                                                     |
| Tls                          | Object  | <code>Symantec.EnterpriseVault.Admin._EV_SMTP_TLS</code> enumeration value: Enumeration value that specifies the support for encrypted and unencrypted connections to the SMTP servers. For details of the enumeration values, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt.                                                                                                                                                     |

**Table 4-85** Symantec.EnterpriseVault.Admin.SMTPServerSettings properties  
(continued)

| Name               | Type   | Description                                                                                                                                                                                                                                                  |
|--------------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TlsCertificateInfo | Object | Symantec.EnterpriseVault.Admin.TlsCertificate: Provides information about the TLS certificate used for SMTP connections.<br><br>For more information on the TlsCertificate object, type <code>get-help about_TlsCertificate</code> at the PowerShell prompt. |

### Related cmdlets

- See “[Get-EVSMTPServerSettings](#)” on page 122.
- See “[Set-EVSMTPServerSettings](#)” on page 130.
- See “[Sync-EVSMTPServerSettings](#)” on page 135.

## Set-EVSMTPServerSettings

`Set-EVSMTPServerSettings` updates the settings for all Enterprise Vault SMTP servers in the specified site. If you do not specify a site ID using the `-SiteID` parameter, and `Set-EVSMTPServerSettings` cannot determine the site, you are prompted to enter the site ID.

This command fails if SMTP server settings for site do not exist.

`Set-EVSMTPServerSettings` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

### Syntax

```
Set-EVSMTPServerSettings [[-SiteId] <string>] [-Port <integer>]
[-MaxMessageSize <integer>] [-Authentication
<Symantec.EnterpriseVault.Admin._EV_SMTP_AUTHENTICATION enumeration
value>] [-AuthenticationCredentials
<System.Management.Automation.PSCredential>]
[-AuthenticationAllowPlainText
<Symantec.EnterpriseVault.Admin._EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT
enumeration value>] [-Tls <Symantec.EnterpriseVault.Admin._EV_SMTP_TLS
enumeration value>] [-TlsCertificateFile <string>]
[-TlsSecurePassphrase <System.Security.SecureString>]
[-SMTPConnectionControl
<Symantec.EnterpriseVault.Admin.SMTPConnectionControlList>]
[<CommonParameters>]
```

## Parameters

**Table 4-86** Set-EVSMTPServerSettings parameters

| Parameter                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -SiteId                    | The ID of the site to which the SMTP settings apply.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| -Port                      | The port number to be used by the SMTP servers in the site.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| -MaxMessageSize            | The maximum message size that the SMTP servers will accept (in MB). The SMTP servers will reject messages that are larger than the specified size.                                                                                                                                                                                                                                                                                                                                                                                                       |
| -Authentication            | The enumeration value specifies whether authentication is enabled for SMTP connections to the SMTP servers. If the value <code>EV_SMTP_AUTHENTICATION_ENABLE</code> is specified, hosts that connect to the SMTP servers must supply the credentials specified by the parameter <code>-AuthenticationCredentials</code> . If the value <code>EV_SMTP_AUTHENTICATION_DISABLE</code> is specified, then <code>-AuthenticationCredentials</code> is not required. For details of the enumerations used, type <code>get-help about_SMTPEnumerations</code> . |
| -AuthenticationCredentials | The Authentication credentials of SMTP servers in the site. This parameter is only required if <code>-Authentication</code> is set to <code>EV_SMTP_AUTHENTICATION_ENABLE</code> .                                                                                                                                                                                                                                                                                                                                                                       |

**Table 4-86** Set-EVSMTPServerSettings parameters (*continued*)

| Parameter                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-AuthenticationAllowPlainText</code> | <p>Enumeration value that specifies whether plain text authentication is enabled for SMTP connections to the SMTP servers.</p> <p>If <code>-Authentication</code> is set to <code>EV_SMTP_AUTHENTICATION_DISABLE</code>, this value is always set to <code>EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT_NEVER</code>.</p> <p>If <code>EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT_TLS</code> is specified, then <code>-Tls</code> must be enabled on the SMTP servers or an appropriate exception message is generated.</p> <p>For details of the enumerations used, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt.</p> |
| <code>-Tls</code>                          | <p>The enumeration value specifies the support for encrypted and unencrypted connections to the SMTP servers. The protocol supported for encrypted connections is Transport Layer Security (TLS). If encrypted connections are supported, then you must specify values for the parameters <code>-TlsCertificateFile</code> and <code>-TlsSecurePassphrase</code>. For details of the enumerations used, type <code>get-help about_SMTPEnumerations</code>.</p>                                                                                                                                                                          |
| <code>-TlsCertificateFile</code>           | <p>The path of the TLS certificate file to use for encrypted connections to the SMTP servers. This parameter is not required if only unencrypted connections are supported.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>-TlsSecurePassphrase</code>          | <p>The TLS certificate file secure passphrase required for encrypted connections to the SMTP servers. This parameter is not required if only unencrypted connections are supported.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>-SMTPConnectionControl</code>        | <p>A list of hosts that can connect to the SMTP servers. For details of <code>SMTPConnectionControlList</code> object, type <code>get-help about_SMTPConnectionControlList</code>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

## Examples

- ```
Set-EVSMTPServerSettings -SiteId 13E...example.com -Authentication
EV SMTP AUTHENTICATION_ENABLE -AuthenticationCredentials
(New-Object System.Management.Automation.PSCredential -ArgumentList
"user@example.com", ("admin@123" | ConvertTo-SecureString
-AsPlainText -Force)) -AuthenticationAllowPlainText
EV SMTP AUTHENTICATION_ALLOWPLAINTEXT_TLS -Tls EV SMTP TLS ENCRYP
TED_AND_UNENCRYPTED -TlsCertificateFile "C:\rsa.p12"
-TlsSecurePassphrase ("admin@123" | ConvertTo-SecureString
-AsPlainText -Force)
```

This command updates SMTP server settings for all SMTP servers in the specified site. The SMTP servers continue to listen on the port previously configured, authentication is enabled, and both encrypted and unencrypted connections are supported.

- ```
C:\PS> $xConnectionList = new-object
Symantec.EnterpriseVault.Admin.SMTPConnectionControlList
C:\PS> $xConnectionList.Add("EV SMTP_CONNECTION_HOSTNAME", "Test")
C:\PS> $xConnectionList.Add("EV SMTP_CONNECTION_HOSTNAMESUFFIX",
"smtp.example.com")
C:\PS> $xConnectionList.Add("EV SMTP_CONNECTION_HOSTNAMEPATTERN",
"as-[a-z0-9]*-[a-z0-9]*.example.Local")
C:\PS> $xConnectionList.Add("EV SMTP_CONNECTION_IPV4",
"192.168.1.1")
C:\PS> $xConnectionList.Add("EV SMTP_CONNECTION_IPV4INCIDR",
"192.168.1.1/24")
C:\PS> $xConnectionList.Add("EV SMTP_CONNECTION_IPV6",
"0000:0000:0000:0000:0000:0000:0000:0001")
C:\PS> $xConnectionList.Add("EV SMTP_CONNECTION_IPV6INCIDR",
"0000:0000:0000:0000:0000:0000:0000:0001/128")
C:\PS> Set-EVSMTPServerSettings -Port 25 -Authentication
EV SMTP AUTHENTICATION_ENABLE -AuthenticationCredentials
(New-Object System.Management.Automation.PSCredential -ArgumentList
"user@example.com", ("admin@123" | ConvertTo-SecureString
-AsPlainText -Force)) -AuthenticationAllowPlainText
EV SMTP AUTHENTICATION_ALLOWPLAINTEXT_NEVER -Tls
EV SMTP TLS ENCRYPTED_AND_UNENCRYPTED -TlsCertificateFile
"C:\rsa.p12" -TlsSecurePassphrase ("admin@123" |
ConvertTo-SecureString -AsPlainText -Force) -SMTPConnectionControl
$xConnectionList
```

```
C:\PS> Get-EVSMTPServerSettings

SiteId                : 13E...example.com
Port                  : 25
MaxMessageSize       : 0
Authentication        : EV_SMTP_AUTHENTICATION_ENABLE
AuthenticationEmailAddress : user@example.com
AuthenticationAllowPlainText : EV_SMTP_AUTHENTICATION
                        _ALLOWPLAINTEXT_NEVER
Tls                   : EV_SMTP_TLS_ENCRYPTED_AND_UNENCRYPTED
TlsCertificateInfo    : Symantec.EnterpriseVault.Admin
                        .TlsCertificate
SMTPConnectionControl : Symantec.EnterpriseVault.Admin
                        .SMTPConnectionControlList
```

This sequence of commands updates SMTP server settings for all SMTP servers in the specified site. The SMTP servers listen on port 25, authentication is enabled, plain text authentication is not allowed, and both encrypted and unencrypted connections are supported. The allowed connections are added to the SMTPConnectionControlList object using different address formats.

## Output

`Set-EVSMTPServerSettings` returns an object of the type `Symantec.EnterpriseVault.Admin.SMTPServerSettings`, which contains the properties listed in the following table.

**Table 4-87** Symantec.EnterpriseVault.Admin.SMTPServerSettings properties

| Name                         | Type   | Description                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Authentication               | Object | Symantec.EnterpriseVault.Admin._EV_SMTP_AUTHENTICATION enumeration value: Specifies whether authentication is required for incoming connections. For details of the enumeration values, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt.                                                                                                                                          |
| AuthenticationEmailAddress   | String | The identity of the authentication user, if authentication is enabled.                                                                                                                                                                                                                                                                                                                                       |
| AuthenticationAllowPlainText | Object | Symantec.EnterpriseVault.Admin._EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT enumeration value: Specifies whether plain text authentication is allowed for incoming connections. If authentication is disabled this value defaults to <code>EV_SMTP_AUTHENTICATION_ALLOWPLAINTEXT_NEVER</code> . For details of the enumeration values, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt. |

**Table 4-87** Symantec.EnterpriseVault.Admin.SMTPServerSettings properties  
(continued)

| Name                  | Type    | Description                                                                                                                                                                                                                                                                                                |
|-----------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MaxMessageSize        | Integer | Maximum SMTP message size (in MB).                                                                                                                                                                                                                                                                         |
| Port                  | Integer | The port number on which the SMTP servers listen.                                                                                                                                                                                                                                                          |
| SiteId                | String  | The ID of the site to which the SMTP server settings belong.                                                                                                                                                                                                                                               |
| SMTPConnectionControl | Object  | <code>Symantec.EnterpriseVault.Admin.SMTPConnectionControlList</code> : Provides a list of allowed connections to the SMTP servers. For more information on managing the list of allowed connections, type <code>get-help about_SMTPConnectionControlList</code> at the PowerShell prompt.                 |
| Tls                   | Object  | <code>Symantec.EnterpriseVault.Admin._EV_SMTP_TLS</code> enumeration value: Enumeration value that specifies the support for encrypted and unencrypted connections to the SMTP servers. For details of the enumeration values, type <code>get-help about_SMTPEnumerations</code> at the PowerShell prompt. |
| TlsCertificateInfo    | Object  | <code>Symantec.EnterpriseVault.Admin.TlsCertificate</code> : Provides information about the TLS certificate used for SMTP connections.<br><br>For more information on the <code>TlsCertificate</code> object, type <code>get-help about_TlsCertificate</code> at the PowerShell prompt.                    |

## Related cmdlets

- See [“Get-EVSMTPServerSettings”](#) on page 122.
- See [“New-EVSMTPServerSettings”](#) on page 124.
- See [“Sync-EVSMTPServerSettings”](#) on page 135.

# Sync-EVSMTPServerSettings

`Sync-EVSMTPServerSettings` synchronizes the SMTP server settings in the Enterprise Vault directory with those on the SMTP server you specify. If you do not specify a server, `Sync-EVSMTPServerSettings` synchronizes the settings on the server on which you run the command.

If you use `-Recreate $True`, `Sync-EVSMTPServerSettings` recreates all the SMTP server settings on the specified server using the settings that are stored in the Enterprise Vault directory.

`Sync-EVSMTPServerSettings` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Sync-EVSMTPServerSettings [[-Server] <string>] [[-Recreate] <boolean>]  
[<CommonParameters>]
```

## Parameters

**Table 4-88** Sync-EVSMTPServerSettings parameters

| Parameter | Description                                                                                                                                                                           |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -Server   | The name of the SMTP server on which you want to synchronize or recreate the SMTP server settings. You must specify the server's host name or alias as a fully qualified domain name. |
| -Recreate | Forces recreation the SMTP server settings, rather than synchronization.                                                                                                              |

## Examples

- `Sync-EVSMTPServerSettings`  
Synchronizes the SMTP server settings on the server where you run the command.
- `Sync-EVSMTPServerSettings -Server smtp.example.com`  
Synchronizes the SMTP server settings on smtp.example.com.
- `Sync-EVSMTPServerSettings -Recreate $True`  
Deletes the existing SMTP server settings on the server where you run the command, and recreates them from the SMTP server settings that are stored in the Enterprise Vault directory.
- `Sync-EVSMTPServerSettings -Server smtp.example.com -Recreate $True`  
Deletes the existing SMTP server settings on smtp.example.com, and recreates them from the SMTP server settings that are stored in the Enterprise Vault directory.

## Output

`Sync-EVSMTPServerSettings` returns no output.

## Related cmdlets

- See [“Get-EVSMTPServerSettings”](#) on page 122.
- See [“New-EVSMTPServerSettings”](#) on page 124.
- See [“Set-EVSMTPServerSettings”](#) on page 130.

# Get-EVSMTPTarget

`Get-EVSMTPTarget` returns all the SMTP target addresses that are configured in the Enterprise Vault site. If you do not specify a site ID using the `-SiteID` parameter, and `Get-EVSMTPTarget` cannot determine the site, you are prompted to enter the site ID. If you specify the name of an SMTP target, then `Get-EVSMTPTarget` gets the properties of that target address. See the help for `New-EVSMTPTarget` for a description of the properties.

When you run `New-EVSMTPTarget`, `Set-EVSMTPTarget`, or `Remove-EVSMTPTarget`, Enterprise Vault updates the SMTP target information in the Enterprise Vault directory, and on each Enterprise Vault SMTP server in the site. If Enterprise Vault is unable to update the SMTP target on any of the SMTP servers, you can use the `GetFailedEVServerList` function to list the servers that Enterprise Vault could not update.

`Get-EVSMTPTarget` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-EVSMTPTarget [-SiteId] <string> [[-Name] <string>]  
[<CommonParameters>]
```

## Parameters

**Table 4-89** Get-EVSMTPTarget parameters

| Parameter                       | Description                                                |
|---------------------------------|------------------------------------------------------------|
| <code>-SiteId</code> (required) | The ID of the site whose SMTP target you want to retrieve. |
| <code>-Name</code>              | The address of the SMTP target to retrieve.                |

## Examples

- `Get-EVSMTPTarget -SiteId 13E...example.com`  
This command returns all the SMTP targets that are configured in the Enterprise Vault site.
- You can use the following sequence of commands to return information about the SMTP target `JohnDoe@example.com`:
  - `$st = New-Object Symantec.EnterpriseVault.Admin.SMTPTarget`  
Uses `New-Object` to create a new variable called `$st` to hold an `SMTPTarget` object.

- `$st.Name = "JohnDoe@example.com"`  
Specifies `JohnDoe@example.com` as the target to be retrieved, using the `Name` parameter.
- `$st.Read()`  
Retrieves information about the specified SMTP target into `$st` using the `Read` method.
- `$st`  
Displays the target information contained in `$st` variable. For example:

```
SiteId           : 13E...example.com
Name             : JohnDoe@example.com
PolicyId        : 138...example.com
PolicyName      : Default SMTP Policy
RetentionCategoryId : 100...example.com
RetentionCategory : Default Retention Category
ArchiveId       : 11F...example.com
ArchiveType     : SMTP
ArchiveName     : SMTPLocal
VaultStoreName  : Express Vault Store
ArchivingEnabled : true
RetryCount      : 0
```

You can also display this information using either of the following commands:

- `Get-EVSMTPTarget -SiteId 13E...example.com -Name JohnDoe@example.com`
- `Get-EVSMTPTarget 13E...example.com JohnDoe@example.com`
- You can use the following sequence of commands to return information about an SMTP target, and find the list of servers on which the target was not up dated by any of `New-EVSMTPTarget`, `Set-EVSMTPTarget`, or `Remove-EVSMTPTarget`:

- `$st = Get-EVSMTPTarget 13E...example.com JohnDoe@example.com`  
Retrieves the SMTP target into the `$st` variable.

- `$st`  
Displays the target information held in `$st`. For example:

```
SiteId           : 13E...example.com
Name             : JohnDoe@example.com
PolicyId        : 138...example.com
PolicyName      : Default SMTP Policy
RetentionCategoryId : 100...example.com
```

```
RetentionCategory : Default Retention Category
ArchiveId         : 11F...example.com
ArchiveType       : SMTP
ArchiveName       : SMTPLocal
VaultStoreName    : Express Vault Store
ArchivingEnabled  : true
RetryCount        : 0
```

- `$st.GetFailedEVServerList()`

Displays the list of Enterprise Vault servers on which the target was not updated. For example:

```
EV1101, EVSMTP11
```

## Output

`Get-EVSMTPTarget` returns objects of the type

`Symantec.EnterpriseVault.Admin.SMTPTarget`, which contains the properties listed in the following table.

**Table 4-90** Symantec.EnterpriseVault.Admin.SMTPTarget properties

| Name                | Type    | Description                                                                                                                            |
|---------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------|
| ArchiveId           | String  | The ID of the archive associated with the SMTP target.                                                                                 |
| ArchiveName         | String  | The name of the archive associated with the SMTP target.                                                                               |
| ArchiveType         | Object  | <code>_DV_DS_E_VAULT_TYPE</code> : The type of the archive associated with the SMTP target.                                            |
| ArchivingEnabled    | Boolean | <code>\$true</code> if messages sent to or from the SMTP target are currently being archived, or <code>\$false</code> if they are not. |
| Name                | String  | The SMTP target address.                                                                                                               |
| PolicyId            | String  | The ID of the SMTP policy associated with the SMTP target.                                                                             |
| PolicyName          | String  | The SMTP policy associated with the SMTP target.                                                                                       |
| RetentionCategory   | String  | The name of the retention category associated with the SMTP target.                                                                    |
| RetentionCategoryId | String  | The ID of the retention category associated with the SMTP target.                                                                      |
| SiteId              | String  | The ID of the site to which the target belongs.                                                                                        |

**Table 4-90** Symantec.EnterpriseVault.Admin.SMTPTarget properties  
(continued)

| Name           | Type   | Description                                                  |
|----------------|--------|--------------------------------------------------------------|
| VaultStoreName | String | The name of the vault store associated with the SMTP target. |

## Related cmdlets

- See [“New-EVSMTPTarget”](#) on page 140.
- See [“Remove-EVSMTPTarget”](#) on page 143.
- See [“Set-EVSMTPTarget”](#) on page 144.

# New-EVSMTPTarget

`New-EVSMTPTarget` adds a new SMTP target with the specified SMTP address and other properties. If you do not specify a site ID using the `-SiteID` parameter, and `New-EVSMTPTarget` cannot determine the site, you are prompted to enter the site ID.

`New-EVSMTPTarget` adds the SMTP target information to the directory, and to each Enterprise Vault SMTP server in the site.

To add a new SMTP target you must specify the following:

- The target SMTP email address (use `-Name`)
- An existing SMTP policy (use `-PolicyName` or `-PolicyId`)
- An existing retention category (use `-RetentionCategory` or `-RetentionCategoryId`)
- An existing archive (use `-ArchiveName` or `-ArchiveId`)
- The type of archive (use `-ArchiveType` or `-ArchiveId`)

You can specify each of the policy, retention category, and archive using its name, its ID, or both. If you use both a name and an ID, the two must relate to the same policy, retention category, or archive.

Use `-ArchivingEnabled $True` to enable archiving for the target, or `-ArchivingEnabled $False` to disable archiving.

`New-EVSMTPTarget` fails if you specify a target that already exists, or if you specify SMTP settings that do not exist in the site.

If `New-EVSMTPTarget` fails to update the SMTP settings on any of the Enterprise Vault SMTP servers, the `GetFailedEVServerList` function can list the servers that were not updated. See the help for `Get-EVSMTPTarget` for an example.

`New-EVSMTPTarget` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
New-EVSMTPTarget [-SiteId] <string> [-Name] <string> [-PolicyId <string>] [-PolicyName <string>] [-RetentionCategoryId <string>] [-RetentionCategory <string>] [-ArchiveId <string>] [-ArchiveName <string>] -ArchiveType <string> [-ArchivingEnabled <Boolean>] [-RetryCount <string>] [<CommonParameters>]
```

## Parameters

**Table 4-91** New-EVSMTPTarget parameters

| Parameter                            | Description                                                                                                                                           |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-SiteId</code> (required)      | The ID of the site in which you want to add the SMTP target.                                                                                          |
| <code>-Name</code> (required)        | The SMTP address of the target.                                                                                                                       |
| <code>-PolicyId</code>               | The ID of an existing SMTP policy to be associated with the SMTP target.                                                                              |
| <code>-PolicyName</code>             | The name of an existing SMTP policy to be associated with the SMTP target.                                                                            |
| <code>-RetentionCategoryId</code>    | The ID of an existing retention category to be associated with the SMTP target.                                                                       |
| <code>-RetentionCategory</code>      | The name of an existing retention category to be associated with the SMTP target.                                                                     |
| <code>-ArchiveId</code>              | The ID of the archive in which Enterprise Vault will archive the target's messages.                                                                   |
| <code>-ArchiveName</code>            | The name of the archive in which Enterprise Vault will archive the target's messages.                                                                 |
| <code>-ArchiveType</code> (required) | The type of archive in which Enterprise Vault will archive the target's messages.                                                                     |
| <code>-ArchivingEnabled</code>       | Set to <code>\$True</code> to archive the target's messages (incoming and outgoing). Set to <code>\$False</code> to disable archiving for the target. |

**Table 4-91** New-EVSMTPTarget parameters (continued)

| Parameter   | Description                                                                                                                                               |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| -RetryCount | The maximum number of times <code>New-EVSMTPTarget</code> will attempt to update the target information on the Enterprise Vault SMTP servers in the site. |

## Examples

- ```
New-EVSMTPTarget -SiteId 13E...example.com -Name
JohnDoe@example.com -PolicyName 'SMTP policy' -RetentionCategory
'Standard retention' -ArchiveName SMTPLocal -ArchiveType SMTP
-ArchivingEnabled $true
```

Adds a new SMTP target, “JohnDoe@example.com”, with the specified properties.

## Output

`New-EVSMTPTarget` returns an object of the type `Symantec.EnterpriseVault.Admin.SMTPTarget`, which contains the properties listed in the following table.

**Table 4-92** Symantec.EnterpriseVault.Admin.SMTPTarget properties

Name	Type	Description
ArchiveId	String	The ID of the archive associated with the SMTP target.
ArchiveName	String	The name of the archive associated with the SMTP target.
ArchiveType	Object	<code>_DV_DS_E_VAULT_TYPE</code> : The type of the archive associated with the SMTP target.
ArchivingEnabled	Boolean	<code>\$true</code> if messages sent to or from the SMTP target are currently being archived, or <code>\$false</code> if they are not.
Name	String	The SMTP target address.
PolicyId	String	The ID of the SMTP policy associated with the SMTP target.
PolicyName	String	The SMTP policy associated with the SMTP target.
RetentionCategory	String	The name of the retention category associated with the SMTP target.
RetentionCategoryId	String	The ID of the retention category associated with the SMTP target.

**Table 4-92** Symantec.EnterpriseVault.Admin.SMTPTarget properties  
(continued)

Name	Type	Description
SiteId	String	The ID of the site to which the target belongs.
VaultStoreName	String	The name of the vault store associated with the SMTP target.

### Related cmdlets

- See [“Get-EVSMTPTarget”](#) on page 137.
- See [“Remove-EVSMTPTarget”](#) on page 143.
- See [“Set-EVSMTPTarget”](#) on page 144.

## Remove-EVSMTPTarget

`Remove-EVSMTPTarget` removes the specified SMTP target. You are prompted to confirm the removal of the target. If you do not specify a site ID using the `-SiteID` parameter, and `Remove-EVSMTPTarget` cannot determine the site, you are prompted to enter the site ID.

Enterprise Vault removes the SMTP target information from the directory, and from each Enterprise Vault SMTP server in the site. If `Remove-EVSMTPTarget` fails to update the SMTP settings on any of the Enterprise Vault SMTP servers, the `GetFailedEVServerList` function can list the servers that were not updated. See the help for `Get-EVSMTPTarget` for an example.

`Remove-EVSMTPTarget` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

### Syntax

```
Remove-EVSMTPTarget [-SiteId] <string> [-Name] <string> [-RetryCount <string>] [<CommonParameters>]
```

### Parameters

**Table 4-93** Remove-EVSMTPTarget parameters

Parameter	Description
<code>-SiteId</code> (required)	The ID of the site from which to remove the SMTP target.

**Table 4-93** Remove-EVSMTPTarget parameters (*continued*)

Parameter	Description
-Name (required)	The SMTP address of the target you want to remove.
-RetryCount	The maximum number of times <code>Remove-EVSMTPTarget</code> will attempt to update the target information on the Enterprise Vault SMTP servers in the site.

## Examples

- `Remove-EVSMTPTarget -SiteId 13E...example.com -Name JohnDoe@example.com`  
This command removes the SMTP target, “JohnDoe@example.com”.
- This sequence of commands first adds the SMTP target “JohnDoe@example.com” into the `$st` variable, then removes it.
  - `$st = Get-EVSMTPTarget 13E...example.com JohnDoe@example.com`  
Adds the SMTP target “JohnDoe@example.com” into the `$st` variable.
  - `Remove-EVSMTPTarget -AdminObject $st -RetryCount 3`  
Makes a maximum of three attempts to remove the target.

## Output

`Remove-EVSMTPTarget` returns no output.

## Related cmdlets

- See “[Get-EVSMTPTarget](#)” on page 137.
- See “[New-EVSMTPTarget](#)” on page 140.
- See “[Set-EVSMTPTarget](#)” on page 144.

# Set-EVSMTPTarget

`Set-EVSMTPTarget` sets or updates the specified properties of an existing SMTP target. If you do not specify a site ID using the `-SiteID` parameter, and `Set-EVSMTPTarget` cannot determine the site, you are prompted to enter the site ID.

`Set-EVSMTPTarget` sets the SMTP target information in the directory, and on each Enterprise Vault SMTP server in the site.

Use `-ArchivingEnabled $True` to enable archiving for the target, or `-ArchivingEnabled $False` to disable archiving.

If `Set-EVSMTPTarget` fails to update the SMTP settings on any of the Enterprise Vault SMTP servers, the `GetFailedEVServerList` function can list the servers that were not updated. See the help for `Get-EVSMTPTarget` for an example.

`Set-EVSMTPTarget` is provided by `Symantec.EnterpriseVault.PowerShell.AdminAPI.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-EVSMTPTarget [-SiteId] <string> [-Name] <string> [-PolicyId
<string>] [-PolicyName <string>] [-RetentionCategoryId <string>]
[-RetentionCategory <string>] [-ArchiveId <string>] [-ArchiveName
<string>] [-ArchiveType <_DV_DS_E_VAULT_TYPE>] [-ArchivingEnabled
<Boolean>] [<CommonParameters>]
```

## Parameters

**Table 4-94** New-EVSMTPTarget parameters

Parameter	Description
<code>-SiteId</code> (required)	The ID of the site in which you want update the SMTP target.
<code>-Name</code> (required)	The SMTP address of the target.
<code>-PolicyId</code>	The ID of an existing SMTP policy to be associated with the SMTP target.
<code>-PolicyName</code>	The name of an existing SMTP policy to be associated with the SMTP target.
<code>-RetentionCategoryId</code>	The ID of an existing retention category to be associated with the SMTP target.
<code>-RetentionCategory</code>	The name of an existing retention category to be associated with the SMTP target.
<code>-ArchiveId</code>	The ID of the archive in which Enterprise Vault will archive the target's messages.
<code>-ArchiveName</code>	The name of the archive in which Enterprise Vault will archive the target's messages.
<code>-ArchiveType</code>	The type of archive in which Enterprise Vault will archive the target's messages.
<code>-ArchivingEnabled</code>	Set to <code>\$True</code> to archive the target's messages (incoming and outgoing). Set to <code>\$False</code> to disable archiving for the target.

## Examples

- `Set-EVSMTPTarget -SiteId 13E...example.com -Name JohnDoe@example.com -PolicyName 'SMTP policy' -RetentionCategory 'Standard retention' -ArchiveName SMTPLocal -ArchiveType SMTP -ArchivingEnabled $true`  
 Updates the SMTP target, “JohnDoe@example.com” with the properties specified.
- `Set-EVSMTPTarget 13E...example.com -Name JohnDoe@example.com -PolicyId 162...example.com -RetentionCategoryId 11B...example.com -ArchiveName SMTPLocal -ArchiveType SMTP -ArchivingEnabled $True`  
 Updates the policy, retention category, and archive of the SMTP target, “JohnDoe@example.com”, and enables archiving. The new policy and retention category are specified using ID values.
- You can use the following sequence of commands to associate a new SMTP policy called "Sales SMTP policy" with the target:
  - `$st = Get-EVSMTPTarget 13E...example.com 'JohnDoe@example.com'`  
 Retrieves the SMTP target, “JohnDoe@example.com”, into the `$st` variable.
  - `$st.PolicyName="Sales SMTP policy"`  
 Updates the `$st` variable with the new SMTP policy.
  - `Set-EVSMTPTarget -AdminObject $st`  
 Uses the copy of the SMTP target held in `$st` to update the SMTP target information in the directory and on each Enterprise Vault SMTP server in the site.

## Output

`Set-EVSMTPTarget` returns an object of the type `Symantec.EnterpriseVault.Admin.SMTPTarget`, which contains the properties listed in the following table.

**Table 4-95** Symantec.EnterpriseVault.Admin.SMTPTarget properties

Name	Type	Description
ArchiveId	String	The ID of the archive associated with the SMTP target.
ArchiveName	String	The name of the archive associated with the SMTP target.
ArchiveType	Object	<code>_DV_DS_E_VAULT_TYPE</code> : The type of the archive associated with the SMTP target.

**Table 4-95** Symantec.EnterpriseVault.Admin.SMTPTarget properties  
(continued)

Name	Type	Description
ArchivingEnabled	Boolean	<code>\$true</code> if messages sent to or from the SMTP target are currently being archived, or <code>\$false</code> if they are not.
Name	String	The SMTP target address.
PolicyId	String	The ID of the SMTP policy associated with the SMTP target.
PolicyName	String	The SMTP policy associated with the SMTP target.
RetentionCategory	String	The name of the retention category associated with the SMTP target.
RetentionCategoryId	String	The ID of the retention category associated with the SMTP target.
SiteId	String	The ID of the site to which the target belongs.
VaultStoreName	String	The name of the vault store associated with the SMTP target.

### Related cmdlets

- See [“Get-EVSMTPTarget”](#) on page 137.
- See [“New-EVSMTPTarget”](#) on page 140.
- See [“Remove-EVSMTPTarget”](#) on page 143.

## Get-EVStorageDatabase

Gets vault store and fingerprint database details for the current Enterprise Vault server.

`Get-EVStorageDatabase` is provided by

`Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

### Syntax

```
Get-EVStorageDatabase [<CommonParameters>]
```

### Parameters

PowerShell common parameters only.

## Examples

- `Get-EVStorageDatabase`  
This command gets vault store and fingerprint databases details for an Enterprise Vault server.

## Output

[Table 4-96](#) lists the properties that are available.

**Table 4-96** Get-EVStorageDatabase properties

Name	Type	Description
DBName	String	A database name.
EntryId	String	The vault store Entry Id.
IsClustered	Boolean	Indicates whether SQL Server is clustered.
SQLInstanceName	String	The name of the SQL Instance that hosts the database.
SQLServerName	String	The name of the SQL Server that hosts the database.
SQLServerVersion	String	The version of SQL Server on the server that hosts the database.
StorageServiceEntryId	String	The Storage Service Entry Id.
Type	EVDATABASETYPE	One of the following: "VaultStore"; "Fingerprint".
VaultStoreGroupName	String	The name of the Vault Store Group that hosts the vault store.
VaultStoreName	String	The name of the Vault Store.

## Related cmdlets

- See ["Get-EVDatabase"](#) on page 45.
- See ["Get-EVDatabaseFileInfo"](#) on page 50.

# Get-EVTask

Gets details of the Enterprise Vault tasks that are configured on the current Enterprise Vault server.

`Get-EVTask` is provided by `Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

## Syntax

```
Get-EVTask [<CommonParameters>]
```

## Parameters

PowerShell common parameters only.

## Examples

- `Get-EVTask`  
This command gets details of the Enterprise Vault tasks that are configured on the current server.

## Output

[Table 4-97](#) lists the properties that are available.

**Table 4-97** Get-EVTask properties

Name	Type	Description
Description	String	The description of the task, as shown the task's properties in the Administration Console.
EntryID	String	The task's Entry ID.
Name	String	The task's name, as shown in the Administration Console.

**Table 4-97** Get-EVTask properties (*continued*)

Name	Type	Description
Type	String	One of the following: "EV_TASK_TYPE_ARCHIVING"; "EV_TASK_TYPE_CLIENTACCESS_POLICY_SYNC"; "EV_TASK_TYPE_DOMINO_MAILBOX_ARCHIVING"; "EV_TASK_TYPE_DOMINO_MAILBOX_POLICY_SYNC"; "EV_TASK_TYPE_EXCHANGE_POLICY_SYNC"; "EV_TASK_TYPE_FSA"; "EV_TASK_TYPE_INDEXING"; "EV_TASK_TYPE_JOURNALING"; "EV_TASK_TYPE_MOVE_ARCHIVE"; "EV_TASK_TYPE_NOTES_JOURNALING"; "EV_TASK_TYPE_PSTCOLLECTOR"; "EV_TASK_TYPE_PSTLOCATOR"; "EV_TASK_TYPE_PSTMIGRATOR"; "EV_TASK_TYPE_PUBLICFOLDER"; "EV_TASK_TYPE_SHAREPOINT".

## Related cmdlets

- See ["Get-EVTaskState"](#) on page 150.

# Get-EVTaskState

Gets the current state of the specified Enterprise Vault task.

`Get-EVDatabase` is provided by

`Symantec.EnterpriseVault.PowerShell.Monitoring.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

## Syntax

```
Get-EVTaskState [-TaskEntryID] <String> [<CommonParameters>]
```

## Parameters

**Table 4-98** Get-EVTaskState parameters

Parameter	Description
<code>-TaskEntryID</code>	The TaskEntryID of the task whose details you require. You can use the <code>Get-EVTask</code> cmdlet to get the TaskEntryID.

## Examples

- `Get-EVTask | foreach($_.EntryID;Get-EVTaskState $_.EntryID}`  
Uses `Get-EVTask` to get the `TaskEntryID` of each task on the current Enterprise Vault

## Output

[Table 4-99](#) lists the properties that are available.

**Table 4-99** Get-EVTaskState properties

Name	Type	Description
(None)	String	A text string that describes the current status of the task. Possible values: "Stopping", "Stopped", "Running", "Refreshing", "Paused", "Loading", "Failed", "Disabled", "CntrSvcNotRunning", "Error", "Processing", "Invalid".

## Related cmdlets

- See [“Get-EVTask”](#) on page 148.

# Get-EVVaultStore

Gets information about the Vault Stores that are hosted by the current Enterprise Vault server.

`Get-EVVaultStore` is provided by

`Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL module.

## Syntax

```
Get-EVVaultStore [<CommonParameters>]
```

## Parameters

PowerShell common parameters only.

## Examples

- `Get-EVVaultStore`  
Returns details of all the Vault Stores on the current Enterprise Vault server.

## Output

Table 4-100 lists the properties that are available.

**Table 4-100** Get-EVVaultStore properties

Name	Type	Description
BackupMode	Boolean	One of the following: "True" (backup mode is on); "False" (backup mode is off).
Computer	String	The name of the server that hosts the vault store.
DatabaseName	String	The name of the vault store database.
EntryId	String	The vault store Entry Id.
Indexed	Boolean	This property is "True" when a vault store is enabled for indexing, otherwise it is "False".
Name	String	The vault store name.
RemovalStatus	RemovalStatus	The safety copy removal setting. One of the following: "Never", "Immediately", "AfterBackup", "AfterBackupImmediateForJournaling".
SharingLevel	VaultStoreSharingLevel	One of the following: "ShareWithinVaultStore"; "ShareWithinGroup"; "NoSharing".
SQLServer	String	The SQL Server that hosts the vault store.
Status	VaultStoreStatus	One of the following: "Available", "MarkedForDeletion", "BackupMode", "BeingCreated".
VaultStoreGroupName	String	The name of the vault store group that the vault store is in.

## Get-EVVaultStorePartition

Gets information about Vault Store partitions.

`Get-EVVaultStorePartition` is provided by `Symantec.EnterpriseVault.PowerShell.Core.dll`, which is not imported automatically by the Enterprise Vault Management Shell. You must import this DLL file.

## Syntax

```
Get-EVVaultStorePartition [<CommonParameters>]
```

## Parameters

PowerShell common parameters only.

## Examples

- `Get-EVVaultStorePartition`  
Returns database details about all the Vault Store partitions on the current server.

## Output

[Table 4-101](#) lists the properties that are available.

**Table 4-101** Get-EVVaultStorePartition properties

Name	Type	Description
CollectorType	String	An indication of the collector type.
Computer	String	The name of the server that hosts the partition.
CreationDate	DateTime	The date and time that the partition was created.
DeviceType	String	The type of device on which the partition resides. For example, "NTFS Volume".
EntryId	String	The vault store partition Entry Id.
HoursSinceLastBackup	Int32	The number of hours since the partition was last backed up.
InterfaceType	VaultStorePartition InterfaceType	One of the following: "Unknown", "NonWorm", "VolumeLevelRetPolicy", "Snaplock", "Snaplike", "Atime", "Streamer", "Max".

**Table 4-101** Get-EVVaultStorePartition properties (*continued*)

Name	Type	Description
LastBackupDate	DateTime	The date and time that the partition was last backed up. This value is null if the partition has never been backed up.
LastOpenedDate	DateTime	The date and time that the partition was last placed in an open state. This value is null if the partition has never been open.
Location	String	The path to the partition.
MigratorType	String	An indication of the migrator type.
Name	String	The name of the partition.
RollOver	RolloverType	One of the following: "NotEnabled", "EnabledBasedOnVolumeSize", "EnabledBasedOnVolumePercentage", "EnabledBasedOnDateTime", "EnabledBasedOnVolumeSizeAndDateTime", "EnabledBasedOnVolumePercentageAndDateTime", "EnabledBasedOnTimerAfter", "EnabledBasedOnVolumeSizeAndTimerAfter", "EnabledBasedOnVolumePercentageAndTimerAfter".
Status	PartitionState	One of the following: "Undefined", "Closed", "Open", "Standby", "BeingDeleted", "Ready".
TotalQuotaGBytesAvailable	Double	The total amount of space available, in gigabytes. This value can be null.
TotalQuotaGBytesFree	Double	The total amount of free space, in gigabytes. This value can be null.
TotalQuotaGBytesUsed	Double	The amount of used space, in gigabytes. This value can be null.
VaultStoreEntryId	String	The vault store Entry Id.
VaultStoreGroupName	String	The vault store group name.
VaultStoreName	String	The vault store name.

# Clear-IndexLocationBackupMode

`Clear-IndexLocationBackupMode` decrements the backup mode count on one indexing location, on all the indexing locations associated with a server, or on all the indexing locations in a site.

Enterprise Vault maintains a count of backup mode requests for each indexing location. For example, if you use concurrent backup scripts in your environment, the backup mode count can be higher than 1. Backup mode is not cleared until the backup mode count reduces to 0.

`Clear-IndexLocationBackupMode` is provided by

`Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Clear-IndexLocationBackupMode [-EVServerName] <String>
[[-IndexRootPath] <String>] [-EVSiteName <String>]
[-ForceClearBackupMode <Boolean>] [-SuppressConfirm]
[<CommonParameters>]
```

```
Clear-IndexLocationBackupMode [-EntryId <String>]
[-ForceClearBackupMode <Boolean>] [-SuppressConfirm]
[<CommonParameters>]
```

## Parameters

**Table 4-102** Clear-IndexLocationBackupMode parameters

Parameter	Description
<code>-EVServerName</code> (required)	The Enterprise Vault server that owns the indexing locations. You must provide this parameter if you specify a site or indexing location by name rather than ID. This allows <code>Clear-IndexLocationBackupMode</code> to find the ID of the site or indexing location.
<code>-IndexRootPath</code>	The root path of an indexing location.
<code>-EVSiteName</code>	The name of the site.
<code>-EntryId</code>	The ID of the indexing location, service, server, or site.
<code>-ForceClearBackupMode</code>	Use <code>-ForceClearBackupMode:\$true</code> to forcibly clear backup mode. <code>Clear-IndexLocationBackupMode</code> then ignores the existing backup mode count and sets it to 0.

**Table 4-102** Clear-IndexLocationBackupMode parameters (*continued*)

Parameter	Description
<code>-SuppressConfirm</code>	Use <code>-SuppressConfirm</code> to suppress the prompt for confirmation when you use <code>-ForceClearBackupMode:\$true</code> . This parameter is only available when you use <code>-ForceClearBackupMode:\$true</code> .

## Examples

- `Clear-IndexLocationBackupMode EVServer-Win2k3`  
Decrements the backup mode count on all the indexing locations associated with server `EVServer-Win2k3`.
- `Clear-IndexLocationBackupMode EVServer-Win2k3 -ForceClearBackupMode:$true`  
Forcibly clears backup mode from all the indexing locations associated with server `EVServer-Win2k3`. `Clear-IndexLocationBackupMode` ignores the existing backup mode count and sets it to 0. In this example, Windows PowerShell prompts for confirmation before it executes the command.
- `Clear-IndexLocationBackupMode EVServer-Win2k3 -ForceClearBackupMode:$true -SuppressConfirm`  
Forcibly clears backup mode from all the indexing locations associated with server `EVServer-Win2k3`. `Clear-IndexLocationBackupMode` ignores the existing backup mode count and sets it to 0. In this example, Windows PowerShell does not prompt for confirmation before it executes the command.
- `Clear-IndexLocationBackupMode EVServer-Win2k3 -IndexRootPath f:\indexing\index0`  
Decrements the backup mode count on indexing location `f:\indexing\index0`.
- `Clear-IndexLocationBackupMode EVServer-Win2k3 -EVSiteName Site1`  
Decrements the backup mode count on all indexing locations in `Site1`.
- `Clear-IndexLocationBackupMode -EntryId 1F3...Domain.local`  
Decrements the backup mode count on the object that has the specified ID. The ID can be that of an indexing location, a server, or a site.  
If you specify the ID of an indexing location, backup mode is decremented on that indexing location.  
If you specify the ID of a server or a site, backup mode is decremented on the indexing locations associated with the server, or in the site.

## Output

There is a confirmation message on completion.

## Related cmdlets

- See “[Get-IndexLocationBackupMode](#)” on page 157.
- See “[Set-IndexLocationBackupMode](#)” on page 159.
- See “[Clear-VaultStoreBackupMode](#)” on page 163.
- See “[Get-VaultStoreBackupMode](#)” on page 166.
- See “[Set-VaultStoreBackupMode](#)” on page 168.

# Get-IndexLocationBackupMode

`Get-IndexLocationBackupMode` reports the current backup mode settings of one indexing location, of all the indexing locations associated with a server, or of all the indexing locations in a site.

`Get-IndexLocationBackupMode` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-IndexLocationBackupMode [-EVServerName] <String> [[-IndexRootPath] <String>] [-EVSiteName <String>] [<CommonParameters>]
```

```
Get-IndexLocationBackupMode [-EntryId <String>] [<CommonParameters>]
```

## Parameters

**Table 4-103** Get-IndexLocationBackupMode parameters

Parameter	Description
<code>-EVServerName</code> (required)	The Enterprise Vault server that owns the indexing locations. You must provide this parameter if you specify a site or indexing location by name rather than ID. This allows <code>Get-IndexLocationBackupMode</code> to find the ID of the site or indexing location.
<code>-IndexRootPath</code>	The root path of an indexing location.
<code>-EVSiteName</code>	The name of the site.
<code>-EntryId</code>	The ID of the indexing location, service, server, or site.

## Examples

- `Get-IndexLocationBackupMode EVServer-Win2k3`  
Reports the current backup mode settings of all the indexing locations associated with server EVServer-Win2k3.
- `Get-IndexLocationBackupMode EVServer-Win2k3 -IndexRootPath f:\indexing\index0`  
Reports the current backup mode setting of indexing location f:\indexing\index0.
- `Get-IndexLocationBackupMode EVServer-Win2k3 -EVSiteName Site1`  
Reports the current backup mode settings of all indexing locations in Site1.
- `Get-IndexLocationBackupMode -EntryId 1F3...Domain.local`  
Reports the current backup mode settings of the object that has the specified ID. The ID can be that of an indexing location, a service, a server, or a site. If you specify the ID of an indexing location, the current backup mode setting of that indexing location is reported. If you specify the ID of a service, server or a site, the current backup mode settings of the indexing locations associated with the server or site are reported.

## Output

Table 4-104 lists the properties that are available.

**Table 4-104** Get-IndexLocationBackupMode properties

Name	Type	Description
BackupMode	Boolean	Indicates whether the index location is in backup mode. Possible values: <code>\$true</code> (index location is in backup mode), <code>\$false</code> (index location is not in backup mode).
IndexingServiceId	String	The IndexingServiceId.
IndexRootPath	String	The IndexRootPath.
IndexRootPathId	String	The IndexRootPathId.

## Related cmdlets

- See [“Clear-IndexLocationBackupMode”](#) on page 155.
- See [“Set-IndexLocationBackupMode”](#) on page 159.
- See [“Clear-VaultStoreBackupMode”](#) on page 163.
- See [“Get-VaultStoreBackupMode”](#) on page 166.

- See “[Set-VaultStoreBackupMode](#)” on page 168.

## Set-IndexLocationBackupMode

`Set-IndexLocationBackupMode` increments the backup mode count on one indexing location, on all the indexing locations associated with a server, or on all the indexing locations in a site.

Enterprise Vault maintains a count of backup mode requests for each indexing location. For example, if you use concurrent backup scripts in your environment, the backup mode count can be higher than 1. Backup mode is not cleared until the backup mode count reduces to 0.

`Set-IndexLocationBackupMode` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

### Syntax

```
Set-IndexLocationBackupMode [-EVServerName] <String> [[-IndexRootPath]
<String>] [-EVSiteName <String>] [<CommonParameters>]
```

```
Set-IndexLocationBackupMode [-EntryId <String>] [<CommonParameters>]
```

### Parameters

**Table 4-105** Set-IndexLocationBackupMode parameters

Parameter	Description
<code>-EVServerName</code> (required)	The Enterprise Vault server that owns the indexing locations. You must provide this parameter if you specify a site or indexing location by name rather than ID. This allows <code>Set-IndexLocationBackupMode</code> to find the ID of the site or indexing location.
<code>-IndexRootPath</code>	The root path of an indexing location.
<code>-EVSiteName</code>	The name of the site.
<code>-EntryId</code>	The ID of the indexing location, service, server, or site.

### Examples

- `Set-IndexLocationBackupMode EVServer-Win2k3`  
Increments the backup mode count on all the indexing locations associated with server `EVServer-Win2k3`.

- `Set-IndexLocationBackupMode EVServer-Win2k3 -IndexRootPath f:\indexing\index0`  
Increments the backup mode count on indexing location `f:\indexing\index0`.
- `Set-IndexLocationBackupMode EVServer-Win2k3 -EVSiteName Site1`  
Increments the backup mode count on all indexing locations in `Site1`.
- `Set-IndexLocationBackupMode -EntryId 1F3...Domain.local`  
Increments the backup mode count on the object that has the specified ID. The ID can be that of an indexing location, a server, or a site.  
If you specify the ID of an indexing location, the backup mode count is incremented on that indexing location.  
If you specify the ID of a server or a site, the backup mode count is incremented on the indexing locations associated with the server, or in the site.

## Output

There is a confirmation message on completion.

## Related cmdlets

- See [“Clear-IndexLocationBackupMode”](#) on page 155.
- See [“Get-IndexLocationBackupMode”](#) on page 157.
- See [“Clear-VaultStoreBackupMode”](#) on page 163.
- See [“Get-VaultStoreBackupMode”](#) on page 166.
- See [“Set-VaultStoreBackupMode”](#) on page 168.

# Set-IndexMetadataSyncLevel

`Set-IndexMetadataSyncLevel` sets the index metadata synchronization flag to ensure that a synchronization between index volume entries in Directory database and those stored in Indexing engine's metadata takes place at next index service startup.

`Set-IndexMetadataSyncLevel` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-IndexMetadataSyncLevel -EntryId <String> -SyncLevel <String>  
[<CommonParameters>]
```

## Parameters

**Table 4-106** Set-IndexMetadataSyncLevel parameters

Parameter	Description
-EntryId (required)	The ID of the indexing service or EV Server hosting that indexing service.
-SyncLevel (required)	<p>The type of synchronization level to be done at startup of Indexing Service. Synchronization will be done for all index volumes linked to input indexing service entry Id.</p> <p>If this is set to 0 (EV_INDEX_METADATA_SYNC_NONE), no synchronization of index volume entries will take place.</p> <p>If this is set to 1 (EV_INDEX_METADATA_SYNC_BASIC), missing index volume entries that are in the Directory database and not in the indexing engine's metadata will be added to the index engine metadata. Those that are present in the indexing engine but not in the Directory database will be removed from the indexing engine metadata.</p> <p>If this is set to 2 (EV_INDEX_METADATA_SYNC_CHECK_LOCATION_PATH), additionally the index location of every index volume in the Directory database will be compared with that stored in the indexing engine's metadata and updated if required.</p>

## Examples

- `Set-IndexMetadataSyncLevel -EntryId 15B...Domain.local -SyncLevel 1`

Sets the index metadata synchronization level to 1 for the indexing service that has the specified entry ID. This ensures that a synchronization between index volume entries in Directory database and those stored in Indexing engine's metadata takes place at next index service startup.

As a result of synchronization missing index volume entries that are in Directory database and not in indexing engine's metadata will be added to index engine metadata and those present in indexing engine but not in Directory database will be removed from indexing engine metadata.

## Output

There is a confirmation message on completion.

## Related cmdlets

- See "[Get-IndexServerForIndexLocation](#)" on page 162.

# Get-IndexServerForIndexLocation

`Get-IndexServerForIndexLocation` gets the index server name and server entry ID for a given index location.

`Get-IndexServerForIndexLocation` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-IndexServerForIndexLocation -Location <String>  
[<CommonParameters>]
```

## Parameters

**Table 4-107** Get-IndexServerForIndexLocation parameters

Parameter	Description
<code>-Location</code> (required)	The root path entry id or root path of the index location.

## Examples

- `Get-IndexServerForIndexLocation -Location 10A...Domain.local`  
Gets the index server name and server entry id for a given index location as a root path entry ID.
- `Get-IndexServerForIndexLocation -Location "F:\Test Index Location\index7"`  
Gets the index server name and server entry id for a given index location as a root path.

## Output

[Table 4-108](#) lists the properties that are available.

**Table 4-108** Get-IndexServerForIndexLocation properties

Name	Type	Description
<code>ServerEntryId</code>	String	The <code>ServerEntryId</code> .
<code>ServerName</code>	String	The <code>ServerName</code> .

## Related cmdlets

- See [“Set-IndexMetadataSyncLevel”](#) on page 160.

# Start-PartitionRollover

`Start-PartitionRollover` forcibly closes the open partition in the specified vault store, and opens the first ready partition if one exists. You must also specify the server that runs the storage service for the vault store in which you want to force partition rollover.

`Start-PartitionRollover` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Start-PartitionRollover [-EVServerName] <String> [-VaultStore]
<String> [<CommonParameters>]
```

## Parameters

**Table 4-109** Start-PartitionRollover parameters

Parameter	Description
<code>-EVServerName</code> (required)	The Enterprise Vault server that runs the storage service for the vault store in which you want to force partition rollover.
<code>-VaultStore</code> (required)	The name or ID of the vault store in which you want to force partition rollover.

## Examples

- `Start-PartitionRollover EVServer-Win2k3 VS1`  
Connects to the Enterprise Vault server `EVServer-Win2k3` and verifies that vault store `VS1` uses the storage service on this server. If it does, `Start-PartitionRollover` forces partition rollover in vault store `VS1`.

## Output

There is a confirmation message on completion.

# Clear-VaultStoreBackupMode

`Clear-VaultStoreBackupMode` decrements the backup mode count on one vault store, on all the vault stores in a vault store group, or on all the vault stores in a site.

Enterprise Vault maintains a count of backup mode requests for each vault store. For example, if you use concurrent backup scripts in your environment, the backup

mode count can be higher than 1. Backup mode is not cleared until the backup mode count reduces to 0.

`Clear-VaultStoreBackupMode` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Clear-VaultStoreBackupMode [-Name] <String> [-EVServerName] <String>
[-EvObjectType] <EVOBJECTTYPE> [-ForceClearBackupMode <Boolean>]
[-SuppressConfirm] [<CommonParameters>]
```

```
Clear-VaultStoreBackupMode -EntryId <String> [-ForceClearBackupMode
<Boolean>] [-SuppressConfirm] [<CommonParameters>]
```

## Parameters

**Table 4-110** Clear-VaultStoreBackupMode parameters

Parameter	Description
<code>-Name</code> (required)	The name of the vault store, vault store group, or site.
<code>-EVServerName</code> (required)	The Enterprise Vault server that owns the vault store, vault store group, or site. You must provide this parameter if you specify a vault store, vault store group, or site by name rather than ID. This allows <code>Clear-VaultStoreBackupMode</code> to find the ID of the vault store, vault store group, or site.
<code>-EvObjectType</code> (required)	The type of object passed by the <code>Name</code> parameter. Acceptable values are <code>VaultStore</code> , <code>VaultStoreGroup</code> and <code>Site</code> . You must provide this parameter if you specify a vault store, vault store group, or site by name rather than ID.
<code>-EntryId</code> (required)	The ID of the vault store, vault store group, or site.
<code>-ForceClearBackupMode</code>	Use <code>-ForceClearBackupMode:\$true</code> to forcibly clear backup mode. <code>Clear-VaultStoreBackupMode</code> then ignores the existing backup mode count and sets it to 0.
<code>-SuppressConfirm</code>	Use <code>-SuppressConfirm</code> to suppress the prompt for confirmation when you use <code>-ForceClearBackupMode:\$true</code> . This parameter is only available when you use <code>-ForceClearBackupMode:\$true</code> .

## Examples

- `Clear-VaultStoreBackupMode VS1 EVServer-Win2k3 VaultStore`

Decrements the backup mode count on vault store VS1.

- `Clear-VaultStoreBackupMode VS1 EVServer-Win2k3 VaultStore -ForceClearBackupMode:$true`

Forcibly clears backup mode from vault store VS1. `Clear-VaultStoreBackupMode` ignores the existing backup mode count and sets it to 0. In this example, Windows PowerShell prompts for confirmation before it executes the command.

- `Clear-VaultStoreBackupMode VS1 EVServer-Win2k3 VaultStore -ForceClearBackupMode:$true -SuppressConfirm`

Forcibly clears backup mode from vault store VS1. `Clear-VaultStoreBackupMode` ignores the existing backup mode count and sets it to 0. In this example, Windows PowerShell does not prompt for confirmation before it executes the command.

- `Clear-VaultStoreBackupMode VSG1 EVServer-Win2k3 VaultStoreGroup`  
Decrements the backup mode count on all the vault stores in vault store group VSG1.
- `Clear-VaultStoreBackupMode Site1 EVServer-Win2k3 Site`  
Decrements the backup mode count on all the vault stores in Site1.
- `Clear-VaultStoreBackupMode -EntryId 1F3...Domain.local`  
Decrements the backup mode count on the object that has the specified ID. The ID can be that of a vault store, a vault store group, or a site.  
If you specify the ID of a vault store, the backup mode count is decremented on that vault store.  
If you specify the ID of a vault store group or a site, the backup mode count is decremented on the vault stores that belong to the group or site.

## Output

There is a confirmation message on completion.

## Related cmdlets

- See [“Clear-IndexLocationBackupMode”](#) on page 155.
- See [“Get-IndexLocationBackupMode”](#) on page 157.
- See [“Set-IndexLocationBackupMode”](#) on page 159.
- See [“Get-VaultStoreBackupMode”](#) on page 166.
- See [“Set-VaultStoreBackupMode”](#) on page 168.

# Get-VaultStoreBackupMode

`Get-VaultStoreBackupMode` reports the current backup mode settings of one vault store, of all the vault stores in a vault store group, or of all the vault stores in a site.

`Get-VaultStoreBackupMode` is provided by

`Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Get-VaultStoreBackupMode [-Name] <String> [-EVServerName] <String>
[-EvObjectType] <EVOBJECTTYPE> [<CommonParameters>]
```

```
Get-VaultStoreBackupMode -EntryId <String> [<CommonParameters>]
```

## Parameters

**Table 4-111** Get-VaultStoreBackupMode parameters

Parameter	Description
<code>-Name</code> (required)	The name of the vault store, vault store group, or site.
<code>-EVServerName</code> (required)	The Enterprise Vault server that owns the vault store, vault store group, or site.  You must provide this parameter if you specify a vault store, vault store group, or site by name rather than ID. This allows <code>Get-VaultStoreBackupMode</code> to find the ID of the vault store, vault store group, or site.
<code>-EvObjectType</code> (required)	The type of object passed by the Name parameter. Acceptable values are <code>VaultStore</code> , <code>VaultStoreGroup</code> and <code>Site</code> .  You must provide this parameter if you specify a vault store, vault store group, or site by name rather than ID.
<code>-EntryId</code> (required)	The ID of the vault store, vault store group, or site.

## Examples

- `Get-VaultStoreBackupMode VS1 EVServer-Win2k3 VaultStore`  
Reports the current backup mode setting of vault store VS1.

- `Get-VaultStoreBackupMode VSG1 EVServer-Win2k3 VaultStoreGroup`  
Reports the current backup mode settings of all the vault stores in vault store group VSG1.
- `Get-VaultStoreBackupMode Site1 EVServer-Win2k3 Site`  
Reports the current backup mode settings of all the vault stores in Site1.
- `Get-VaultStoreBackupMode -EntryId 1F3...Domain.local`  
Reports the current backup mode settings of the object that has the specified ID. The ID can be that of a vault store, a vault store group, or a site.  
If you specify the ID of a vault store, the current backup mode setting of that vault store is reported.  
If you specify the ID of a vault store group or a site, the current backup mode settings of the vault stores that belong to the group or site are reported.
- `Get-VaultStoreBackupMode -EVServerName EVServer-Win2k3 -EVOBJECTType VaultStore -Name VS1`  
Reports the current backup mode setting on vault store VS1.  
In this example, the parameters must be named because they are not in the default order.

## Output

[Table 4-112](#) lists the properties that are available.

**Table 4-112** Get-VaultStoreBackupMode properties

Name	Type	Description
BackupMode	Boolean	Indicates whether the vault store is in backup mode. Possible values: <code>\$true</code> (the vault store is in backup mode), <code>\$false</code> (the vault store is not in backup mode).
The VaultStoreEntryId.	String	The VaultStoreEntryId.
VaultStoreName	String	The name of the vault store.

## Related cmdlets

- See [“Clear-IndexLocationBackupMode”](#) on page 155.
- See [“Get-IndexLocationBackupMode”](#) on page 157.
- See [“Set-IndexLocationBackupMode”](#) on page 159.
- See [“Clear-VaultStoreBackupMode”](#) on page 163.
- See [“Set-VaultStoreBackupMode”](#) on page 168.

# Set-VaultStoreBackupMode

`Set-VaultStoreBackupMode` increments the backup mode count on one vault store, on all the vault stores in a vault store group, or on all the vault stores in a site.

Enterprise Vault maintains a count of backup mode requests for each vault store. For example, if you use concurrent backup scripts in your environment, the backup mode count can be higher than 1. Backup mode is not cleared until the backup mode count reduces to 0.

`Set-VaultStoreBackupMode` is provided by `Symantec.EnterpriseVault.PowerShell.Snapin.dll`, which is loaded by the Enterprise Vault Management Shell.

## Syntax

```
Set-VaultStoreBackupMode [-Name] <String> [-EVServerName] <String> [-EvObjectType] <EVObjectType> [<CommonParameters>]
```

```
Set-VaultStoreBackupMode -EntryId <String> [<CommonParameters>]
```

## Parameters

**Table 4-113** Set-VaultStoreBackupMode parameters

Parameter	Description
<code>-Name</code> (required)	The name of the vault store, vault store group, or site.
<code>-EVServerName</code> (required)	The Enterprise Vault server that owns the vault store, vault store group, or site.  You must provide this parameter if you specify a vault store, vault store group, or site by name rather than ID. This allows <code>Set-VaultStoreBackupMode</code> to find the ID of the vault store, vault store group, or site.
<code>-EvObjectType</code> (required)	The type of object passed by the <code>Name</code> parameter. Acceptable values are <code>VaultStore</code> , <code>VaultStoreGroup</code> and <code>Site</code> .  You must provide this parameter if you specify a vault store, vault store group, or site by name rather than ID.
<code>-EntryId</code> (required)	The ID of the vault store, vault store group, or site.

## Examples

- `Set-VaultStoreBackupMode VS1 EVServer-Win2k3 VaultStore`  
Increments the backup mode count on vault store VS1.

- `Set-VaultStoreBackupMode VSG1 EVServer-Win2k3 VaultStoreGroup`  
Increments the backup mode count on all the vault stores in vault store group VSG1.
- `Set-VaultStoreBackupMode Site1 EVServer-Win2k3 Site`  
Increments the backup mode count on all the vault stores in Site1.
- `Set-VaultStoreBackupMode -EntryId 1F3...Domain.local`  
Increments the backup mode count on the object that has the specified ID. The ID can be that of a vault store, a vault store group, or a site.  
If you specify the ID of a vault store, the backup mode count is incremented on that vault store.  
If you specify the ID of a vault store group or a site, the backup mode count is incremented on the vault stores that belong to the group or site.
- `Set-VaultStoreBackupMode -EVServerName EVServer-Win2k3  
-EVObjectType VaultStore -Name VS1`  
Increments the backup mode count on vault store VS1.  
In this example, the parameters must be named because they are not in the default order.

## Output

There is a confirmation message on completion.

## Related cmdlets

- See [“Clear-IndexLocationBackupMode”](#) on page 155.
- See [“Get-IndexLocationBackupMode”](#) on page 157.
- See [“Set-IndexLocationBackupMode”](#) on page 159.
- See [“Clear-VaultStoreBackupMode”](#) on page 163.
- See [“Get-VaultStoreBackupMode”](#) on page 166.

# Index

## A

- Add-EVPstComputer 80
  - examples 80
  - parameters 80
  - syntax 80
- Add-EVPstFile 81
  - examples 82
  - parameters 81
  - syntax 81
- Add-EVRBARoleMember 86
  - examples 88
  - output 89
  - parameters 87
  - related cmdlets 89
  - syntax 86

## C

- Clear-IndexLocationBackupMode 155
  - examples 156
  - output 156
  - parameters 155
  - syntax 155
- Clear-VaultStoreBackupMode 163
  - examples 164
  - output 165
  - parameters 164
  - syntax 164
- cmdlet 23, 78
  - Get-EVFSAVolume 58
  - Get-EVSMTPHoldingFolder 113
  - Get-EVSMTPPolicy 114
  - Get-EVSMTPServerSettings 122
  - Get-EVSMTPTarget 137
  - New-EVFSAVolume 61
  - New-EVSMTPPolicy 117
  - New-EVSMTPServerSettings 125
  - New-EVSMTPTarget 141
  - output 96, 98
  - Remove-EVFSAVolume 64
  - Remove-EVSMTPPolicy 119
  - Remove-EVSMTPTarget 143

## cmdlet (continued)

- Set-EVFSAVolume 65
- Set-EVSMTPPolicy 120
- Set-EVSMTPServerSettings 131
- Set-EVSMTPTarget 145
- Set-VaultStoreBackupMode 168
- Sync-EVSMTPServerSettings 136

## G

- Get-EVArchive 20
  - examples 21
  - output 22
  - parameters 20
  - syntax 20
- Get-EVClassificationPolicy 26
  - examples 26
  - output 27
  - parameters 26
  - related cmdlets 28
  - syntax 26
- Get-EVClassificationTags 40
  - examples 41
  - output 41
  - parameters 41
  - syntax 41
- Get-EVClassificationTestMode 41
  - examples 42
  - output 42
  - parameters 42
  - related cmdlets 42
  - syntax 42
- Get-EVComputers 44
  - examples 44
  - output 44
  - parameters 44
- Get-EVDatabase 45
  - examples 45
  - output 45
  - parameters 45
  - syntax 45

- Get-EVDatabaseDetail 46
  - examples 47
  - output 47
  - parameters 46
  - syntax 46
- Get-EVDatabaseFileInfo 50
  - examples 51
  - output 51
  - parameters 51
  - syntax 50
- Get-EVDependencyService 53
  - examples 53
  - output 53
  - parameters 53
  - syntax 53
- Get-EVDependencyServiceState 54
  - examples 54
  - output 55
  - parameters 54
  - syntax 54
- Get-EVFileLocation 56
  - examples 57
  - output 57
  - parameters 57
  - syntax 56
- Get-EVFSVolume 58
  - examples 58
  - output 59
  - syntax 58
- Get-EVIISWebsite 67
  - examples 68
  - output 68
  - parameters 67
  - syntax 67
- Get-EVIMAPUsers 70
  - examples 71
  - output 71
  - parameters 70
  - syntax 70
- Get-EVIMAPUserSettings 72
  - examples 72
  - output 72
  - parameters 72
  - syntax 72
- Get-EVIndexLocation 74
  - examples 74
  - output 74
  - parameters 74
  - syntax 74
- Get-EVMDSStatus
  - examples 78
  - output 79
  - parameters 78
  - syntax 78
- Get-EVRBAAzStoreXml 95
  - examples 96
  - parameters 96
  - related cmdlets 96
  - syntax 96
- Get-EVRBARole 83
  - examples 84
  - output 85
  - parameters 84
  - related cmdlets 85
  - syntax 83
- Get-EVRBARoleMember 89
  - examples 90
  - output 91
  - parameters 90
  - related cmdlets 91
  - syntax 90
- Get-EVRetentionPlan 98
  - examples 99
  - output 100
  - parameters 98
  - related cmdlets 101
  - syntax 98
- Get-EVServer 109
  - examples 109
  - output 110
  - parameters 109
  - syntax 109
- Get-EVService 110
  - examples 110
  - output 110
  - parameters 110
  - syntax 110
- Get-EVServiceState 111
  - examples 111
  - output 112
  - parameters 111
  - syntax 111
- Get-EVSite 112
  - examples 112
  - output 112
  - parameters 112
  - syntax 112

Get-EVSMTPHoldingFolder 113  
  examples 113  
  output 113  
  syntax 113

Get-EVSMTPPolicy 114  
  examples 114  
  output 116  
  syntax 114

Get-EVSMTPServerSettings 122  
  examples 122  
  output 123  
  syntax 122

Get-EVSMTPTarget 137  
  examples 137  
  output 139  
  syntax 137

Get-EVStorageDatabase 147  
  examples 148  
  output 148  
  parameters 147  
  syntax 147

Get-EVTask 148  
  examples 149  
  output 149  
  parameters 149  
  syntax 149

Get-EVTaskState 150  
  examples 151  
  output 151  
  parameters 150  
  syntax 150

Get-EVVaultStore 151  
  examples 151  
  output 152  
  parameters 151  
  syntax 151

Get-EVVaultStorePartition 152  
  examples 153  
  output 153  
  parameters 153  
  syntax 153

Get-IndexLocationBackupMode 157  
  examples 158  
  output 158  
  parameters 157  
  syntax 157

Get-IndexServerForIndexLocation 162  
  examples 162  
  output 162

Get-IndexServerForIndexLocation (continued)  
  parameters 162  
  syntax 162

Get-VaultStoreBackupMode 166  
  examples 166  
  output 167  
  parameters 166  
  syntax 166

## I

Import-EVClassificationRules 34  
  examples 36  
  output 36  
  parameters 35  
  related cmdlets 37  
  syntax 35

## N

New-EVClassificationPolicy 28  
  examples 29  
  output 30  
  parameters 28  
  related cmdlets 30  
  syntax 28

New-EVFSVolume  
  examples 61  
  output 62  
  syntax 60

New-EVMDSBuildTask 75  
  examples 76  
  output 77  
  parameters 76  
  syntax 75

New-EVRetentionPlan 101  
  examples 102  
  output 103  
  parameters 101  
  related cmdlets 104  
  syntax 101

New-EVSMTPPolicy 116  
  examples 117  
  output 118  
  syntax 117

New-EVSMTPServerSettings 124  
  examples 127  
  output 129  
  syntax 125

**New-EVSMTPTarget** 140

- examples 142
- output 142
- syntax 141

**P****Publish-EVClassificationRules** 37

- examples 39
- output 40
- parameters 38
- related cmdlets 40
- syntax 38

**R****Remove-EVClassificationPolicy** 31

- examples 31
- output 31
- parameters 31
- related cmdlets 32
- syntax 31

**Remove-EVExchangeFolderPolicy** 55

- examples 56
- parameters 55
- syntax 55

**Remove-EVFSAVolume**

- examples 64
- output 64
- syntax 63

**Remove-EVRBARoleMember** 92

- examples 94
- output 95
- parameters 93
- related cmdlets 95
- syntax 92

**Remove-EVRetentionPlan** 104

- examples 105
- output 105
- parameters 105
- related cmdlets 105
- syntax 104

**Remove-EVSMTPPolicy** 118

- examples 119
- output 119
- syntax 119

**Remove-EVSMTPTarget** 143

- examples 144
- output 144
- syntax 143

**S****Set-EVArchive**

- examples 24
- output 25
- parameters 24
- syntax 23

**Set-EVClassificationPolicy** 32

- examples 33
- output 34
- parameters 32
- related cmdlets 34
- syntax 32

**Set-EVClassificationTestMode** 43

- examples 43
- output 43
- parameters 43
- related cmdlets 43
- syntax 43

**Set-EVDatabaseDetail** 48

- examples 49
- output 50
- parameters 49
- syntax 48

**Set-EVFSAVolume**

- examples 66
- output 66
- syntax 65

**Set-EVIMAPServerDisabled** 68

- examples 69
- output 69
- parameters 68
- syntax 68

**Set-EVIMAPServerEnabled** 69

- examples 70
- output 70
- parameters 69
- syntax 69

**Set-EVRBAzStoreXml** 97

- examples 97
- parameters 97
- related cmdlets 98
- syntax 97

**Set-EVRetentionPlan** 105

- examples 107
- output 108
- parameters 106
- related cmdlets 109
- syntax 105

- Set-EVSMTPPolicy 119
  - examples 120
  - output 121
  - syntax 120
- Set-EVSMTPServerSettings 130
  - examples 133
  - output 134
  - syntax 130
- Set-EVSMTPTarget 144
  - examples 146
  - output 146
  - syntax 145
- Set-IndexLocationBackupMode 159
  - examples 159
  - output 160
  - parameters 159
  - syntax 159
- Set-IndexMetadataSyncLevel 160
  - examples 161
  - output 161
  - parameters 161
  - syntax 160
- Set-VaultStoreBackupMode 168
  - examples 168
  - output 169
  - syntax 168
- Start-EVComputers
  - syntax 44
- Start-EVDatabaseUpgrade 52
  - examples 53
  - parameters 53
  - syntax 53
- Start-PartitionRollover 163
  - examples 163
  - output 163
  - parameters 163
  - syntax 163
- Symantec.EnterpriseVault.Admin.EVRbaRole
  - properties 85, 89, 95
- Symantec.EnterpriseVault.Admin.EVRbaRoleMember
  - properties 91
- Sync-EVSMTPServerSettings 135
  - examples 136
  - output 136
  - syntax 136