

Veritas Enterprise Vault™

Utilities

12

Veritas Enterprise Vault: Utilities

Last updated: 2015-12-03.

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

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

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

Customer service

Customer service information is available at the following URL:

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

Japan

CustomerCare_Japan@veritas.com

Contents

Technical Support	4	
Chapter 1	About this guide	14
	About Enterprise Vault utilities	14
	Running the Enterprise Vault command-line utilities with administrator privileges	16
	Where to get more information about Enterprise Vault	17
	“How To” articles on the Veritas Support website	19
	Enterprise Vault training modules	19
	Comment on the documentation	19
Chapter 2	ArchivePoints	20
	About ArchivePoints	20
	ArchivePoints syntax	20
	ArchivePoints examples	24
Chapter 3	Audit Viewer	25
	About Audit Viewer	25
	Using Audit Viewer to run a report on audit data	25
	Copying the search results from Audit Viewer	27
	Changing Audit Viewer settings	27
Chapter 4	Backtrace	28
	About Backtrace	28
	Backtrace default settings	29
	Backtrace registry values	29
	Backtrace file name format	31
	Backtrace examples	31
Chapter 5	CenteraPing	34
	About CenteraPing	34
	CenteraPing syntax	34

Chapter 6	Domino Archive Exporter	36
	About Domino Archive Exporter	36
	Domino Archive Exporter syntax	36
	Domino Archive Exporter example	38
Chapter 7	Domino Profile Document Tool	39
	About Domino Profile Document Tool	39
	Domino Profile Document Tool syntax	39
	Domino Profile Document Tool examples	40
Chapter 8	Domino Retention Plan Tool	41
	About Domino retention plans	41
	Domino Retention Plan Tool permissions	42
	Defining a Domino retention plan	42
	EVDominoRetentionPlans.exe syntax	45
Chapter 9	DTrace	47
	About DTrace	47
	Running DTrace from the command line	48
	Running DTrace from the Administration Console	51
	About the DTrace log	52
	DTrace troubleshooting	52
Chapter 10	EVDominoExchangeMigration Tool	54
	About the EVDominoExchangeMigration tool	54
	Client requirements for the EVDominoExchangeMigration tool	55
	Adding the EVDominoExchangeMigration tool to the Windows Server firewall exceptions list	55
	EVDominoExchangeMigration tool and Binary Tree	56
	Using Quest Notes Migrator for Exchange and the EVDominoExchangeMigration tool	56
	Requirements for other migration software with the EVDominoExchangeMigration tool	57
	Running the EVDominoExchangeMigration tool	57
	Syntax for EVDominoExchangeMigration tool	58
	Log files for EVDominoExchangeMigration tool	59
	Limitations of EVDominoExchangeMigration tool	59

Chapter 11	EVDuplicateCleaner	61
	About EVDuplicateCleaner	61
	Configuring EVDuplicateCleaner	61
	Running EVDuplicateCleaner	63
	Fixing broken shortcuts after you have run EVDuplicateCleaner	64
Chapter 12	EVEARemovalUtility	66
	About EVEARemovalUtility	66
	EVEARemovalUtility prerequisites	67
	Running EVEARemovalUtility	68
	EVEARemovalUtility syntax	68
	Format of the EVEARemovalUtility output and log files	69
	EVEARemovalUtility usage examples	71
	EVEARemovalUtility example: processing a single file	71
	EVEARemovalUtility example: processing a folder and its subfolders	71
Chapter 13	EVFSASetRightsAndPermissions	73
	About EVFSASetRightsAndPermissions	73
	Running EVFSASetRightsAndPermissions	74
Chapter 14	EVrights	75
	About EVrights	75
	EVrights syntax	75
Chapter 15	EVservice	78
	About EVservice	78
	EVservice prerequisites	79
	EVservice syntax	79
	EVservice list file format	80
Chapter 16	EVSPShortcutManager	81
	About EVSPShortcutManager	81
	Permissions required to run EVSPShortcutManager	82
	EVSPShortcutManager syntax	82
	EVSPShortcutManager examples	84

Chapter 17	EVSVR	85
	About EVSVR	85
	About the checkpointing facility in EVSVR	87
	Note on performing EVSVR operations on CIFS and NTFS partitions	88
	Starting EVSVR	89
	EVSVR commands	89
	EVSVR application states	91
	Creating an EVSVR operation file	92
	Editing an EVSVR operation file in which you have enabled checkpointing	98
	Running an EVSVR operation	99
	About the EVSVR operation settings	100
	Report operations in EVSVR	100
	Verify operations in EVSVR	107
	Repair operations in EVSVR	114
	Using the output from one EVSVR operation as input for another operation	128
	About EVSVR item list files	128
	EVSVR operations that support item list processing	129
	Viewing the EVSVR output log file	130
	About the checkpointing information in the EVSVR log file	131
	About the item list information in the EVSVR log file	132
	Additional log file information when you run certain EVSVR Repair operations	133
	Running EVSVR in interactive mode	134
	DumpSaveSet command	135
	DumpSISPart command	138
	ExtractSaveSets command	141
	GetNativeItem command	143
	ListSaveSetLocations command	144
	Note on reviewing the messages in the EVSVR log files	145
	Improving EVSVR performance when processing CAB collections	146
Chapter 18	FSARunNow	148
	About FSARunNow	148
	Running FSARunNow	149
	FSARunNow syntax	149
	FSARunNow examples	152

Chapter 19	FSAUndelete	153
	About FSAUndelete	153
	Running FSAUndelete	154
	FSAUndelete syntax	155
	FSAUndelete examples	156
Chapter 20	FSAUtility	157
	About FSAUtility	157
	Running FSAUtility	158
	About using FSAUtility with EMC Celerra/VNX placeholders	159
	Configuring which API call FSAUtility uses to identify Celerra/VNX placeholders	160
	Example FSAUtility.exe.config file settings	161
	FSAUtility options	163
	Recreating archive points	164
	Recreating placeholders	165
	Moving placeholders and corresponding files	167
	Migrating placeholders	169
	Deleting orphaned placeholders	174
	Restoring archived files	175
	Recalling files corresponding to placeholders	177
Chapter 21	NTFS to Centera Migration	180
	About NTFS to Centera Migration	180
	Managing migrator jobs using NTFS to Centera Migration	180
	Creating migrator jobs using NTFS to Centera Migration	182
	Deleting active jobs using NTFS to Centera Migration	184
	Deleting source files after migration using NTFS to Centera Migration	185
	NTFS to Centera Migration log files	185
Chapter 22	Permission Browser	187
	About Permission Browser	187
	Running Permission Browser	187
Chapter 23	Policy Manager (EVPM)	188
	About Policy Manager	188
	Policy Manager syntax	189
	Saving a Policy Manager initialization file as a Unicode file	190
	Policy Manager initialization file syntax	190

Understanding the usage summary from Vault Store Usage Reporter	255
Checking that the IIS authentication method is correctly set for Vault Store Usage Reporter	256
Index	257

About this guide

This chapter includes the following topics:

- [About Enterprise Vault utilities](#)
- [Running the Enterprise Vault command-line utilities with administrator privileges](#)
- [Where to get more information about Enterprise Vault](#)
- [Comment on the documentation](#)

About Enterprise Vault utilities

Enterprise Vault provides a number of utilities with which you can test and log the performance of Enterprise Vault, run scripts to perform common tasks, and more.

[Table 1-1](#) lists the utilities that are available when you install Enterprise Vault.

Table 1-1 Available Enterprise Vault utilities

Use this utility	To do this
ArchivePoints	Create and manage "archive points"—the points marking the top of each folder structure that File System Archiving is to store in a single archive.
Audit Viewer	View and filter the data that is logged in an Enterprise Vault auditing database.
Backtrace	Obtain tracing information from Enterprise Vault processes. The trace starts automatically, just before a problem occurs.
CenteraPing	Test the connection to an EMC Centera cluster.

Table 1-1 Available Enterprise Vault utilities (*continued*)

Use this utility	To do this
Domino Archive Exporter	Export items from an Enterprise Vault Domino archive to a Notes database.
Domino Profile Document Tool	View the contents of the profile document that Enterprise Vault adds to a Domino mailbox.
Domino Retention Plan Tool	Upload to Enterprise Vault any new retention plans that you create.
DTrace	Run Enterprise Vault in debug mode by logging what processes are doing at the code level.
EVDominoExchangeMigration	Modify shortcuts in Exchange Server mailboxes that have been migrated from Domino to Exchange Server.
EVDuplicateCleaner	Find and delete duplicate savesets.
EVEARemovalUtility	Remove the extended attributes from files so that FSA can create placeholder shortcuts for them.
EVFSASetRightsAndPermissions	Configure the required permissions and privileges for a changed Vault Service account on a file server on which the FSA Agent is installed.
EVrights	Grant user rights to users and groups from a command line or batch file.
EVservice	Start and stop Windows services and Enterprise Vault tasks on local or remote computers.
EVSVR	Report on, verify, and repair Enterprise Vault storage. You can also perform a number of specialized activities such as retrieving the savesets of an archived item and extracting savesets from an EMC Centera data blob.
FSARunNow	Start archiving from a specified file server, synchronize permissions, and prune earlier versions of archived files.
FSAUndelete	Cancel the permanent deletion of the archived files for specified placeholders, or for all of the placeholders in a specified folder.

Table 1-1 Available Enterprise Vault utilities (*continued*)

Use this utility	To do this
FSAUtility	Recreate archive points and placeholders, move and delete placeholders, and restore archived files.
NTFS to Centera Migration	Copy Enterprise Vault savesets from an NTFS source partition to an EMC Centera destination partition.
Permission Browser	View the security identifiers (SIDs) and access permissions for the archives and archive folders in an Enterprise Vault directory database.
Policy Manager	Use scripts to modify and control mailboxes and archives so that they conform to your Enterprise Vault archiving policies. Additionally, you can use Policy Manager to migrate the contents of PST files to Enterprise Vault.
ResetEVClient	Fix a number of problems with the Enterprise Vault add-in to Microsoft Outlook.
Vault Store Usage Reporter	Obtain reports on current vault store usage.

Running the Enterprise Vault command-line utilities with administrator privileges

Many of the utilities that this guide describes are command-line utilities. On computers where User Account Control (UAC) is enabled, you must always run these utilities with administrator privileges. The Enterprise Vault utilities may not run properly without these elevated privileges.

To run an Enterprise Vault command-line utility with Administrator privileges

- 1 Right-click the **Command Prompt** shortcut, and then click **Run as Administrator**.
- 2 Change to the folder that contains the utility that you want to run, for example `C:\Program Files (x86)\Enterprise Vault`.
- 3 Type the command to start the utility.

Where to get more information about Enterprise Vault

Table 1-2 lists the documentation that accompanies Enterprise Vault.

Table 1-2 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"> ■ On the Windows Start menu, click Start > Programs > Enterprise Vault > Documentation. ■ 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. ■ On the Help menu in the Administration Console, click Help on Enterprise Vault.
<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.

Table 1-2 Enterprise Vault documentation set (*continued*)

Document	Comments
<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.
<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
- 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. Please only use this address to comment on product documentation.

We appreciate your feedback.

ArchivePoints

This chapter includes the following topics:

- [About ArchivePoints](#)
- [ArchivePoints syntax](#)
- [ArchivePoints examples](#)

About ArchivePoints

The ArchivePoints utility provides a convenient means to create and manage archive points, as an alternative to using the Administration Console. An archive point marks the top of a folder structure that File System Archiving stores in a single archive. You can use ArchivePoints to create, list, and delete archive points, and to update their attribute values.

ArchivePoints includes an `autoenable` option to create an auto-enabling folder. If you create an auto-enabling folder, the archiving task creates an archive point for each immediate subfolder, including new subfolders when they are added. An auto-enabling folder can be useful for example when you have a folder that holds a subfolder for each of a number of users.

Note: Take care when you create archive points not to overwrite any existing archive points. An overwritten archive point can result in archived data that is split across multiple archives.

ArchivePoints syntax

Use one of the following formats:

- To create archive points:

```
ArchivePoints create archive_point_path_share_name  
subfolders|nosubfolders [XML_template_file_path_name]
```

- To update the attributes of archive points:

```
ArchivePoints update archive_point_path_share_name  
subfolders|nosubfolders XML_template_file_path_name
```

- To list all the archive points beneath a specified network share:

```
ArchivePoints find archive_point_path_share_name
```

- To display the contents of the archive points:

```
ArchivePoints read archive_point_path_share_name
```

- To delete the archive points:

```
ArchivePoints delete archive_point_path_share_name
```

- To create an auto-enabling folder:

```
ArchivePoints autoenable autoenabling_folder_path on
```

- To turn off the auto-enabling property for a folder:

```
ArchivePoints autoenable autoenabling_folder_path off  
[subfolderdelete]
```

where the parameters are as follows:

<i>archive_point_path_share_name</i>	Specifies the UNC path to the network share to which the command applies. Enclose the path in quotes if it includes any non-alphanumeric characters.
subfolders nosubfolders	Specifies whether to create an archive point for each immediate subfolder.
<i>XML_template_file_path_name</i>	Specifies the full path to an XML template file of archive point attribute values, which override the default values or existing values.
<i>autoenabling_folder_path</i>	The full path to the folder on which you want to switch on or switch off the auto-enabling property. If you omit the on and off switches, the auto-enabling property for a folder will be switched off.
subfolderdelete	Deletes the archive point from each immediate subfolder when you switch off the auto-enabling property for a folder.

If you specify an XML template file, it must have the following format:

```
<archivePoint xmlns="urn:kvsplc-com:FileSystemFolderArchivePoint">
  <attribute>value</attribute>
  <attribute>value</attribute>
  <attribute>value</attribute>
  ...
</archivePoint>
```

where each attribute line specifies an archive point attribute and its value.

Note the following:

- You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.
 See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.
- If an attribute is not included in the XML template file when you create an archive point, the File System Archiving task uses the default value for that attribute.
- If an attribute is not included in the XML template file when you update an archive point, the File System Archiving task does not change the existing value.
- You cannot use an XML template file if you create an auto-enabling folder. The archive points associated with an auto-enabling folder must be created with default attribute values.

[Table 2-1](#) lists the attributes you can include in the XML template file.

Table 2-1 ArchivePoints template file attributes

Attribute	Description	Default value for new archive point
name	Specifies the name of the archive that is associated with this archive point, with any <code>prefix</code> if specified.	The name of the folder on which the archive point resides.
description	Provides a description for the archive if required. The description appears in the list of file system archives under Archives > File System in the Administration Console.	None.
owner	Specifies the account to use when billing archive usage.	None.
indexDisabled	Specifies whether to disable (<code>true</code>) or enable (<code>false</code>) indexing for the files in the archive.	false.

Table 2-1 ArchivePoints template file attributes (*continued*)

Attribute	Description	Default value for new archive point
indexingLevel	<p>Specifies the indexing level, which can be either <code>brief</code> or <code>full</code>.</p> <p><code>brief</code> indexes the metadata of archived items such as the file name and the item date, but not any content. A brief index is smaller than a full index, but users cannot search for any content in the archived items.</p> <p><code>full</code> indexes the metadata and the content of archived items. Users can search for the content of items.</p>	The setting on the Indexing tab of the Site properties in the Administration Console.
indexSnippetLength	<p>Specifies the preview length in characters. The preview length is the amount of text that Enterprise Vault shows in a search results list, when the indexing level is <code>full</code>. The preview length can only be <code>128</code> or <code>1000</code>. The size of an index increases when you increase the preview length.</p>	The setting on the Indexing tab of the Site properties in the Administration Console.
indexAttachment Snippet	<p>Specifies whether Enterprise Vault creates previews of attachment content when the indexing level is <code>full</code>. The value can be <code>true</code> (yes) or <code>false</code> (no). These previews cannot be viewed in this release of Enterprise Vault. The size of an index increases when you enable this option.</p>	The setting on the Indexing tab of the Site properties in the Administration Console.
deleteExpiredItems	<p>Specifies whether Enterprise Vault automatically deletes items from the archive when their retention periods expire. The value can be <code>true</code> (delete expired items) or <code>false</code> (do not delete expired items).</p>	<code>false</code> .
prefix	<p>Specifies a prefix that Enterprise Vault prepends to <code>name</code> to make the archive name. A prefix may be useful if you do not specify a <code>name</code> and you use the <code>subfolders</code> option to create an archive point for each immediate subfolder of the target folder.</p>	None.

For example, the following file sets the attribute values for an archive point:

```
<archivePoint xmlns="urn:kvsp1c-com:FileSystemFolderArchivePoint">
  <name>Newton archive</name>
  <description>Isaac Newton's User Archive</description>
  <owner>astronomy\newtoni</owner>
  <indexDisabled>false</indexDisabled>
  <indexingLevel>full</indexingLevel>
  <indexSnippetLength>1000</indexSnippetLength>
</archivePoint>
```

```
<indexAttachmentSnippet>>false</indexAttachmentSnippet>  
<deleteExpiredItems>>false</deleteExpiredItems>  
<prefix>User</prefix>  
</archivePoint>
```

ArchivePoints examples

The following are example ArchivePoints commands.

- To create an archive point on folder `\\myserver\users\jones`:
`ArchivePoints create \\myserver\users\jones nosubfolders`
- To create an archive point on each immediate subfolder of `\\myserver\users\`, and use an XML template file named `archiveptfile.xml` to override the default values of the archive point attributes:
`ArchivePoints create \\myserver\users subfolders "c:\Program Files (x86)\Enterprise Vault\archiveptfile.xml"`
- To list all archive points on share `\\myserver\users`:
`ArchivePoints find \\myserver\users`
- To auto-enable archive points for all immediate subfolders of the folder `\\myserver\development`:
`ArchivePoints autoenable \\myserver\development on`
- To switch off the auto-enable property for the folder `\\myserver\development` and delete the archive points from all its immediate subfolders:
`ArchivePoints autoenable \\myserver\development off subfolderdelete`

Audit Viewer

This chapter includes the following topics:

- [About Audit Viewer](#)
- [Using Audit Viewer to run a report on audit data](#)
- [Copying the search results from Audit Viewer](#)
- [Changing Audit Viewer settings](#)

About Audit Viewer

Audit Viewer lets you view and filter the data that is logged in an Enterprise Vault auditing database. The function of this database is to keep a record of Enterprise Vault activities such as archiving items and viewing and restoring archived items. You can specify the data that you want to view, sort the data, and copy it to the Windows Clipboard.

Using Audit Viewer to run a report on audit data

Follow the instructions in this section to open Audit Viewer and generate a report on the data in the auditing database.

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

To use Audit Viewer to run a report on audit data

- 1** In Windows Explorer, browse to the Enterprise Vault program folder (for example `C:\Program Files (x86)\Enterprise Vault`).
- 2** Double-click `AuditViewer.exe`.
- 3** In the Audit Viewer window, type or select the search criteria for the records that you want to view.

The following table provides information on each search term.

User Name	Specify the required user in the form <i>domain\username</i> .
Archive	You can use the Enterprise Vault Administration Console to determine the ID of the archive. Right-click the required archive, and then click Properties . The Advanced tab in the properties sheet shows the archive ID.
Category	Select a category of audit entries to search from the list. Audit Viewer lists only those categories that exist in the captured data.
Subcategory	After you have selected a category, select a subcategory from the list. <ul style="list-style-type: none"> ■ Item returns the summary information for a category. ■ If you select Detailed as a category, the additional information is held in Information records. ■ All returns both the summary and detailed records for selected categories.
Date (From), Date (To)	Define a date range and time range to search the audit records.
Information contains	Type a keyword for which to search in the audit records.
Status	Select a status from the list for the records that you want to view.
Server	Select the Enterprise Vault server that is the target of this search.
Audit ID	Type a range of numbers to indicate the audit records that you want to view.
Order By	Select the attribute by which to order the results and whether you want Audit Viewer to list the results in ascending order or descending order.

Maximum Results Select whether to view all the results that the search finds or a portion of those results.

- 4 Click **Search** to generate the report.

Copying the search results from Audit Viewer

Audit Viewer displays the records that match your search criteria in the Search Results window.

Click a column heading to sort the records according to the entries in that column.

You can copy the contents of this window to another application, such as a spreadsheet application.

To copy the search results from Audit Viewer

- 1 In the Search Results window, highlight the records that you want to copy.
- 2 Right-click the records, and then click **Copy**.
 You can also press Ctrl+A and Ctrl+C to copy all the search results to the Clipboard.
- 3 Paste the records into the destination document.

Changing Audit Viewer settings

You can change the auditing database that you want to search. Audit Viewer also provides the option to hide or show selected fields in the Search Results window.

To change Audit Viewer settings

- 1 In the main Audit Viewer window, click **Settings**.
- 2 In the Settings window, change the auditing database that you want to search. You can also check or uncheck the return fields that you want to show or hide.

Backtrace

This chapter includes the following topics:

- [About Backtrace](#)
- [Backtrace default settings](#)
- [Backtrace registry values](#)
- [Backtrace file name format](#)
- [Backtrace examples](#)

About Backtrace

Caution: Running the Backtrace utility can affect Enterprise Vault performance, so you should only enable it when necessary.

Backtrace enables you to obtain log files of tracing information from Enterprise Vault processes. Unlike the DTrace utility, Backtrace does the following:

- Provides tracing from the period before a problem occurs.
- Generates log files that contain tracing information from a single process.

Backtrace retains tracing information in memory until a previously defined trigger event occurs. Backtrace then writes a limited amount of DTrace information to a log file. The log file contains DTrace information from before and after the trigger event occurred.

When you enable Backtrace, the default is for it to create logs for all Enterprise Vault errors and warnings. You can modify this behavior as required. You can specify that particular events trigger Backtrace. Alternatively, you can make Backtrace create log files for all events except for those that you specifically exclude.

Note: Backtrace does not create a log file if there is less than 100 MB of free disk space. You cannot change this setting.

You control Backtrace by editing the Backtrace registry values.

If you run DTrace, Enterprise Vault automatically disables Backtrace while DTrace is running.

Backtrace default settings

The Backtrace default settings are as follows:

- Backtrace is disabled.
- Backtrace writes log files to folders in the `Backtrace` subfolder of the Enterprise Vault `Reports` folder.
Backtrace creates a new subfolder for each day. For example, on 16 November 2012 the default folder path is as follows:
`C:\Program Files (x86)\Enterprise Vault\Reports\Backtrace\20121116\`
- Backtrace creates log files for a maximum of five identical events each day for each Enterprise Vault process. The counter is reset when a process is restarted.
- The maximum total size of all Backtrace log files that can be generated in a single day is 200 MB.
- Backtrace keeps log files for seven days and then deletes them automatically.
- When Backtrace is enabled, all warnings and errors trigger Backtrace to create log files.

Backtrace registry values

Enterprise Vault automatically creates the Backtrace registry values on each Enterprise Vault server. By default, Backtrace is disabled. If you want to enable Backtrace, you must edit the registry on the server on which you require tracing.

The Backtrace registry values are in the following location on each Enterprise Vault server:

```
HKEY_LOCAL_MACHINE
  \SOFTWARE
    \Wow6432Node
      \KVS
        \Enterprise Vault
          \Backtrace
```

Table 4-1 describes the Backtrace registry values.

Table 4-1 Backtrace registry values

Registry value	Description
Enabled	<p>Controls whether Backtrace is enabled (1) or disabled (0, the default setting). Note that enabling Backtrace can affect performance, so you should only do so if you experience issues with Enterprise Vault.</p> <p>With Backtrace enabled, each Enterprise Vault process on the server maintains Backtrace information in memory. When a trigger event occurs, Backtrace writes trace information to a log file.</p>
Exclude	<p>Provides a semicolon-separated list of the events that must not trigger Backtrace. For example, 3310;3230;2776. Set RuleType to Exclude to activate this list.</p>
Include	<p>Provides a semicolon-separated list of events that must trigger Backtrace. For example, 3310;3230;2776. Set RuleType to Include to activate this list.</p>
LogFileKeepDays	<p>Specifies the number of days to keep Backtrace log files. Enterprise Vault automatically deletes old Backtrace log files. Backtrace checks for log files to delete when the Admin service starts and then every hour on the hour.</p>
LogFolderPath	<p>Specifies the location for Backtrace log files. If no value is specified for LogFolderPath, Backtrace stores its log files in the Backtrace subfolder of the Enterprise Vault Reports folder. You can edit LogFolderPath to set a different path.</p>
MaxEventsOfEachTypePerDay	<p>Specifies the maximum number of log files to create each day for each event. The default is 5.</p>
MaxEventsOfEachTypePerDayAcrossAllProcesses	<p>Specifies the maximum number of log files to create each day for each event for all processes. The default is 40.</p>
MaxSizeOfAllLogsPerDayMB	<p>Specifies the maximum total size of all Backtrace log files that can be generated in a single day. The default is 200 (megabytes).</p>

Table 4-1 Backtrace registry values (*continued*)

Registry value	Description
RuleType	Controls the manner in which Backtrace is triggered. When RuleType is set to Exclude (the default setting), all error events and warning events trigger Backtrace, except for those that are listed in the Exclude registry value. When RuleType is set to Include, all the events that are specified in the Include registry value trigger Backtrace. Other events do not trigger Backtrace.

See the "Backtrace" chapter in the *Registry Values* guide for details of the Backtrace registry values.

Backtrace file name format

The Backtrace log file names comprise the following items, separated by underscores:

- The name begins with "EV".
- Local date and time in the format *YYYYMMDD_HHMMSSmmm*
- Server name. The name of the server on which the process is running.
- Process name. The name of the process that is traced.
- Process ID. The ID of the process that is traced.
- Event IDs. The name contains a maximum of five IDs of the most recent events that are in the file.

The following example shows a log file name when error event 8938 from the Admin service triggers Backtrace on server "MYSERVER". The trigger event 8938 is followed by error event 8942:

```
EV201110908_095919784_MYSERVER_AdminService(2872)_8938E_8942E.log
```

By default Backtrace stores its log files in the `Backtrace` subfolder of the Enterprise Vault `Reports` folder. You can edit the `LogFolderPath` registry value to specify a different location.

Backtrace examples

[Table 4-2](#) shows an example of Backtrace registry values when **RuleType** is set to the default of 'Exclude'.

Table 4-2 Sample values when RuleType is set to Exclude

Registry value	Setting	Comments
Enabled	1	Backtrace is enabled. Backtrace is triggered according to the setting of RuleType.
RuleType	Exclude	Default of 'Exclude' for RuleType. All warnings and errors trigger Backtrace except for those that are listed in the Exclude value.
Exclude	None	By default there is no value for Exclude. All errors and warnings trigger Backtrace.
Include	3310;3230;2776	Backtrace ignores the Include setting because RuleType is set to 'Exclude'.
LogFileKeepDays	7	The default is to keep log files for seven days.
LogFolderPath	None	Default of no value for LogFolderPath. By default, Backtrace writes log files to the Backtrace subfolder of the Enterprise Vault Reports folder.
MaxEventsOfEachTypePerDay	5	Default of 5 for MaxEventsOfEachTypePerDay. Backtrace creates log files for a maximum of five identical events each day for each process. The counter is reset when a process is restarted.

Table 4-3 shows an example of Backtrace registry values when **RuleType** is set to 'Include'.

Table 4-3 Sample values when RuleType is set to Include

Registry value	Setting	Comments
Enabled	1	Backtrace is enabled. Backtrace is triggered according to the setting of RuleType.
RuleType	Include	Each of the events that are listed in Include triggers Backtrace.

Table 4-3 Sample values when RuleType is set to Include (*continued*)

Registry value	Setting	Comments
Exclude	None	Backtrace ignores the Exclude setting because RuleType is set to 'Include'. All errors and warnings trigger Backspace.
Include	3310;3230;2776	A semicolon-separated list of the events that trigger Backtrace when RuleType is set to 'Include'. No other events trigger Backtrace.
LogFileKeepDays	7	The default is to keep log files for seven days.
LogFolderPath	None	Default of no value for LogFolderPath. By default, Backtrace writes log files to the <code>Backtrace</code> subfolder of the Enterprise Vault <code>Reports</code> folder.
MaxEventsOfEachTypePerDay	5	Default of 5 for MaxEventsOfEachTypePerDay. Backtrace creates log files for a maximum of five identical events each day for each process. The counter is reset when a process is restarted.

CenteraPing

This chapter includes the following topics:

- [About CenteraPing](#)
- [CenteraPing syntax](#)

About CenteraPing

Use CenteraPing to test the connection to an EMC Centera™ cluster.

CenteraPing syntax

```
CenteraPing -address IP_address [-version|-help]
```

where *IP_address* is the address of one of the access nodes in the cluster that you want to examine.

CenteraPing tries to make a connection to the specified IP address. If this connection is successful, CenteraPing returns the following message:

```
IP_address is accessible
```

Otherwise, CenteraPing returns the following message:

```
IP_address Open Error: -10020 No connection with pool
```

The `-help` option returns the same information, together with the version number of the utility, and a usage statement. The `-version` option returns the version number, as well as the usage statement.

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

Domino Archive Exporter

This chapter includes the following topics:

- [About Domino Archive Exporter](#)
- [Domino Archive Exporter syntax](#)
- [Domino Archive Exporter example](#)

About Domino Archive Exporter

Domino Archive Exporter is a command-line utility with which you can export items from an Enterprise Vault Domino archive to a Notes database.

You can choose to export items as follows:

- To a specified local or remote Notes database
- With a specified retention category
- That were archived within a specified date range

You can stop the export process at any time by pressing Ctrl+C.

Domino Archive Exporter syntax

```
EVDominoExporter.exe /A archive /O destination database/I ID file /P  
ID file password [/T database template] [/R retention category] [/SD  
start date] [/ED end date]
```

[Table 6-1](#) lists the available parameters.

Table 6-1 EVDominoExporter.exe parameters

Parameter	Description
/A	Identifies the Enterprise Vault Domino archive from which to export items.
/ED	Specifies the end date and time for a range of items to archive, in the form <i>dd /mm /yyyy hh :mm :ss</i> . If you omit the time, the default time that is used is 00:00:00.
/I	Specifies the full path to a Notes authentication ID file.
/O	<p>Specifies the Notes database to open or create. If you want to open or create a local database, specify it as in this example:</p> <pre>/O "c:\Program Files (x86)\Enterprise Vault\dest.nsf"</pre> <p>If you omit the path to the database file, Domino Archive Exporter stores the file in the \Data folder of the Notes client. The utility creates the specified directory if it does not exist.</p> <p>To open or create a remote database on a Domino server, specify it as in this example:</p> <pre>/O Server1/Sales/ACME!!Restore\dest.nsf</pre> <p>This parameter instructs Domino Archive Exporter to export the items to the database <i>dest.nsf</i> in the folder \Data\Restore on the server <i>Server1/Sales/ACME</i>.</p>
/P	Specifies the password that is associated with the Notes authentication ID file.
/R	Filters the archive contents by retention category.
/SD	Specifies the start date and time for a range of items to archive, in the form <i>dd /mm /yyyy hh :mm :ss</i> . If you omit the time, the default time that is used is 00:00:00.
/T	<p>Specifies the mail template to use when you create the Notes database. For example, you can type the following to use a local template file:</p> <pre>/T mailbox.ntf</pre> <p>To use a template file on a Domino server, specify it as in this example:</p> <pre>/T Server1/Sales/ACME!!mailjrn.ntf</pre> <p>If you omit the /T parameter, Domino Archive Exporter uses the router mail template (<i>mailbox.ntf</i>).</p>

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

Domino Archive Exporter example

See FrameMaker source for another example. The following command exports the archive `L14` to the database `sample.nsf`, using the Notes authentication ID file `local_admin.id` and the associated password `W3lcome`. The only items that are exported are those marked with the retention category `Business` and archived between 10 A.M. on December 16 2005 and 4.56 P.M. on December 17 2005.

```
EVDominoExporter.exe /A L14 /O sample.nsf  
/I "d:\local_admin.id" /P W3lcome /R Business  
/SD "16/12/2005 10:00:00" /ED "17/12/2005 16:56:00"
```

Domino Profile Document Tool

This chapter includes the following topics:

- [About Domino Profile Document Tool](#)
- [Domino Profile Document Tool syntax](#)
- [Domino Profile Document Tool examples](#)

About Domino Profile Document Tool

This tool lets you view the contents of the profile document that Enterprise Vault adds to a Domino mailbox. If you suspect that the profile document is corrupt, you can also use this tool to delete it.

Domino Profile Document Tool syntax

```
EvLotusDominoProfileDocTool.exe server database id password  
<zap|view|clearlist>
```

where the parameters are as follows:

<i>server</i>	Specifies the server on which the appropriate Notes database resides.
<i>database</i>	Specifies the Notes database for the desired mailbox.
<i>id</i>	Specifies the location of the Notes authentication ID file, relative to the \Data folder.
<i>password</i>	Specifies the password that is associated with the Notes authentication ID file.

<code>zap</code>	Deletes the specified profile document.
<code>view</code>	Lists the contents of the specified profile document.
<code>clearblacklist</code>	Clears the list of items that Enterprise Vault has blacklisted. These items have been archived but the archiving task is unable to modify the original notes because their notes summary buffers are full. When you clear this list the archiving task archives the items again. Any items that cannot be modified at that time are blacklisted again.

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

Domino Profile Document Tool examples

The following are examples of how to run Domino Profile Document Tool.

- The following command lists the contents of the profile document in the database `mdavis.nsf`. The Notes authentication ID file is `admin.id`, and the associated password is `W3lcome`.

```
EvLotusDominoProfileDocTool.exe DomServer1/EU/Veritas  
mail\mdavis.nsf admin.id W3lcome view
```

- The following command deletes the profile document from the database `mdavis.nsf`.

```
EvLotusDominoProfileDocTool.exe DomServer1/EU/Veritas  
mail\mdavis.nsf admin.id W3lcome zap
```

Domino Retention Plan Tool

This chapter includes the following topics:

- [About Domino retention plans](#)
- [Domino Retention Plan Tool permissions](#)
- [Defining a Domino retention plan](#)
- [EVDominoRetentionPlans.exe syntax](#)

About Domino retention plans

The Retention Folder feature enables you to create a single folder or a hierarchy of folders automatically in users' mail files. Enterprise Vault archives these folders according to policies that you assign. If a user deletes any folders in the retention folder hierarchy, Enterprise Vault automatically recreates them.

You specify the retention folders and their retention categories in retention plans. You can create as many retention plans as you require.

You use Enterprise Vault provisioning groups to apply retention plans to mail files. Thus, different users can have different retention folders with the appropriate retention categories. You can also define a default retention plan that Enterprise Vault applies to all users for whom a specific plan is not defined.

You create an XML file in which you define the retention plans and assign retention plans to provisioning groups. You then use the `EVDominoRetentionPlans.exe` command line tool to upload the XML file to Enterprise Vault.

The process to create and apply a retention plan is as follows:

- 1 If you have existing retention plans you can use the `EVDominoRetentionPlans.exe` command line tool to extract the definition of the existing plans from Enterprise Vault. You extract the plans as a single XML file.
- 2 Edit the existing XML file or create new XML file as required to create the new retention plan.
- 3 Use `EVDominoRetentionPlans.exe` to load the XML file into Enterprise Vault. Enterprise Vault automatically validates the XML and does not accept an invalid file.
- 4 Enterprise Vault applies the plan on the next run of the provisioning task or the mailbox archiving task.

Domino Retention Plan Tool permissions

The retention folders are created in users' mail files by the Domino provisioning task or mailbox archiving task.

The ID that the provisioning task or mailbox archiving task uses must have the permission '**Access to current Database**' in the Execution Control List on every users' computer.

The account you use to run `EVDominoRetentionPlans.exe` must have the Enterprise Vault roles-based administration permission 'Domino Administrator'.

For information about roles-based administration, see the *Administrator's Guide*.

Defining a Domino retention plan

If you have previously created a retention plan XML file you can modify that file. If necessary, you can use the `EVDominoRetentionPlans.exe` tool to extract the existing retention plans from Enterprise Vault to a file that you can edit.

To extract the existing retention plans from Enterprise Vault and save them in the file `MyPlans.xml`:

```
EVDominoRetentionPlans.exe -save MyPlans.xml
```

In the Enterprise Vault program folder there is an example retention plans XML file that you can copy and modify as required. The file is `Example RetentionPlans.xml` in `.`

The example file defines retention plans 'All Users' and 'Projects'.

The 'All Users' retention plan does the following:

- Creates the retention folder 'Retention Folders' with the following subfolders:
 - 'Business Records' with a retention category of 'Business'.
 - 'Customer Mails' with a retention category of 'Customers'.

The 'Projects' retention plan does the following:

- Creates a retention folder that is called 'Retention Folders' and that has the following subfolders:
 - 'Business Records' with a retention category of 'Business'.
 - 'Customer Mails' with a retention category of 'Customers'.
- 'Projects' under 'Retention Folders' with a setting of `ARCHIVENOW="true"`. The plan also creates the following folders under 'Projects':
 - 'Project X'
 - 'Project Y'
- Deletes the retention folder 'Test'.

The XML file assigns retention plans to provisioning groups as follows:

- The 'Projects' retention plan is assigned to the provisioning group 'Project members'.

```
<?xml version="1.0"?>
<RETENTIONPLANCONFIG>

    <!-- Start of defining retention plans -->
    <RETENTIONPLANS>

        <!-- Start of 'All Users' retention plan -->
        <RETENTIONPLAN NAME="All Users">
            <FOLDER NAME="Retention Folders">
                <FOLDER NAME="Business Records" RETCAT="Business"/>
                <FOLDER NAME="Customer Mails" RETCAT="Customers"/>
            </FOLDER>
        </RETENTIONPLAN>
        <!-- End of 'All users' retention plan -->

        <!-- Start of 'Projects' retention plan -->
        <RETENTIONPLAN NAME="Projects">
            <FOLDER NAME="Retention Folders">
                <FOLDER NAME="Business Records" RETCAT="Business"/>
```

```
<FOLDER NAME="Customer Mails" RETCAT="Customers"/>
<FOLDER NAME="Projects" ARCHIVENOW="true">
  <FOLDER NAME="Project X" RETCAT="Project X"/>
  <FOLDER NAME="Project Y" RETCAT="Project Y"/>
</FOLDER>

<!-- Delete temporary folder 'Test' -->
<FOLDER NAME="Test" DELETE="true"/>
</FOLDER>
</RETENTIONPLAN>
<!-- End of 'Projects' retention plan -->

</RETENTIONPLANS>
<!-- End of defining retention plans -->

<!-- Assign retention plans to provisioning groups -->
<PROVISIONINGGROUPS>
  <DOMAIN NAME="ACME">
    <GROUP NAME="Project members" RETENTIONPLAN="Projects"/>
    <DEFAULT RETENTIONPLAN="All Users"/>
  </DOMAIN>
</PROVISIONINGGROUPS>
<!-- End of assigning retention plans to provisioning groups -->

</RETENTIONPLANCONFIG>
```

Note the following:

- To specify a standard folder or view, use the real name, not the displayed name. For example:
 - Use "`($Inbox)`" to specify the Inbox folder.
 - Use "`($ToDo)`" to specify the Tasks view.
 - Use "`($Calendar)`" to specify the Calendar.
- Use the `FOLDER` element to define both folders and views.
- `FOLDER` elements can contain other `FOLDER` elements. This feature enables you to define a hierarchy of folders or views.
- A parent folder's retention category applies to all its subfolders, unless specifically overridden for a particular folder.

- If the `ARCHIVENOW` attribute is set to true, documents present in the folder are archived on the next run of the archive task.
- If the `ARCHIVENOW` attribute is specified on a parent folder, it automatically applies to all subfolders, unless overridden at the subfolder level.
- If the `DELETE` attribute is set to True, Enterprise Vault deletes the folder and all its subfolders provided that Enterprise Vault created the folder originally. The `DELETE` attribute removes all subfolders, even those that a user created. `DELETE` does not remove notes; the folder contents are still available in the **All Documents** view.
- Optionally, you can specify a default plan for each domain. The default plan is applied if there is no plan for a provisioning group.
- A retention plan cannot contain multiple entries for the same folder or view.
- A provisioning group can have only one retention plan.
- In the `DOMAIN` section, provisioning group names must be unique.
- You can define some folder hierarchies that do not have retention categories assigned.

EVDominoRetentionPlans.exe syntax

You can use `EVDominoRetentionPlans.exe` as follows:

- To load a retention plan definition file into Enterprise Vault, enter the following:
`EVDominoRetentionPlans.exe -set pathToUploadXmlFile`
`pathToUploadXmlFile` is the path to the file that contains the retention plan definitions you want to load into Enterprise Vault.
This action overwrites all existing retention plans that are in the current Enterprise Vaultsite.
The change to new retention plans appears in the Enterprise Vault event log as event ID 41238 and event category 'Domino Retention Plan Tool'.
- To save the current retention plan definitions in a file, enter the following:
`EVDominoRetentionPlans.exe -save pathToDownloadXmlFile`
`pathToDownloadXmlFile` is the path to the file in which you want `EVDominoRetentionPlans.exe` to save a copy of the current retention plans.
- To delete all retention plans from Enterprise Vault, enter the following:
`EVDominoRetentionPlans.exe -clear`
This action deletes all retention plans in the current Enterprise Vault site but does not affect retention folders. If you want to delete a retention folder you must create a retention plan that specifies the `DELETE` attribute.

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

DTrace

This chapter includes the following topics:

- [About DTrace](#)
- [Running DTrace from the command line](#)
- [Running DTrace from the Administration Console](#)
- [About the DTrace log](#)
- [DTrace troubleshooting](#)

About DTrace

When an Enterprise Vault service, task, or process fails, it is important to diagnose what is wrong. The DTrace utility logs the activities of a process at the code level, and so provides a way to run Enterprise Vault in debug mode. DTrace lets you monitor multiple services simultaneously, write the trace to a file, filter for specific words, and trigger tracing that is based on filters. If you log a call with Veritas Support, you may be asked to run a trace to aid problem diagnosis.

You can run DTrace from the command line or from the Administration Console.

Note: Unlike DTrace, the Backtrace utility provides tracing from the period before a problem occurs. Therefore, you may prefer to implement tracing with Backtrace rather than DTrace.

See [“About Backtrace”](#) on page 28.

Running DTrace from the command line

By running DTrace from the command line, you can create more customized traces than those that you can create from the Administration Console.

To run DTrace from the command line

- 1 Log on to the Enterprise Vault server as the Vault Service account.
- 2 Click the **DTrace** shortcut.

The DTrace prompt (`DT>`) indicates that DTrace has loaded. Some commands change the prompt. For example, if you type `filter`, the prompt changes to `DT FILTER>`. To return to the `DT>` prompt, type `Quit` or `Exit`.

- 3 Type the required commands.

See [DTrace commands](#) below.

- 4 When you want to stop DTrace, press `Ctrl+C` to stop monitoring and then type `Quit` or `Exit`.

DTrace commands

[Table 9-1](#) describes the commands that you can type at the DTrace prompt.

Table 9-1 DTrace commands

Command	Description
<code>cls</code>	Clears the console.
<code>comment</code>	Lets you type a comment to add to the trace output.
<code>display</code>	Displays the selected trace entries. You can specify the start entries and end entries in a range, and choose whether to apply a filter to those entries.

Table 9-1 DTrace commands (*continued*)

Command	Description
filter	Lets you filter the contents of the trace by specifying the text strings that the entries either must contain (includes) or cannot contain (excludes). You can type the following commands at the <code>DT Filter></code> prompt:
<code>+ string [;string] or Include string [;string]</code>	Adds the nominated strings to the filter include list. These strings are case-sensitive.
<code>- string [;string] or exclude string [;string]</code>	Adds the nominated strings to the filter exclude list. These strings are case-sensitive.
<code>clear [Includes Excludes Both]</code>	Deletes all the include strings from the filter, all the exclude strings, or both.
<code>delete string</code>	Deletes the nominated string from the filter.
<code>exit or quit</code>	Exits filter management.
<code>reset</code>	Resets the filter to the default settings.
<code>view</code>	Displays the current filter settings.
log	Specifies the name (and optionally the full path) of the log file. The default location for the log file is the Enterprise Vault <code>Reports</code> folder (for example, <code>C:\Program Files (x86)\Enterprise Vault\Reports</code>).
monitor	Displays the trace live in the console but does not write it to disk. Press <code>Ctrl+C</code> to stop the console output.
open	Lists the available log files and lets you open them in a text editor.
pause	Pauses tracing for the specified period or until the current <code>watch</code> command has completed.
registry	Displays the entries under the following key in the Windows registry: <pre>HKEY_LOCAL_MACHINE \SOFTWARE \Wow6432Node \KVS \Enterprise Vault</pre>
reset	Resets the trace options.

Table 9-1 DTrace commands (*continued*)

Command	Description														
rollover	<p>Lets you view and edit the rollover settings for the DTrace log files. By default, DTrace automatically creates a new log file when the current one reaches 100 megabytes (MB) in size. You can enable and disable file rollover, specify the required size of the log files, and reset the rollover settings to the default settings.</p> <p>Type the following commands at the <code>DT Rollover></code> prompt:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><code>disable or off</code></td> <td style="padding: 2px;">Disables file rollover.</td> </tr> <tr> <td style="padding: 2px;"><code>enable or on</code></td> <td style="padding: 2px;">Enables file rollover.</td> </tr> <tr> <td style="padding: 2px;"><code>exit or quit</code></td> <td style="padding: 2px;">Exits file rollover management.</td> </tr> <tr> <td style="padding: 2px;"><code>help or ?</code></td> <td style="padding: 2px;">Displays online Help on the rollover commands.</td> </tr> <tr> <td style="padding: 2px;"><code>reset</code></td> <td style="padding: 2px;">Resets the rollover settings to the default settings.</td> </tr> <tr> <td style="padding: 2px;"><code>size number</code></td> <td style="padding: 2px;">Specifies the maximum size of each log file in megabytes.</td> </tr> <tr> <td style="padding: 2px;"><code>view</code></td> <td style="padding: 2px;">Displays the current rollover settings.</td> </tr> </table>	<code>disable or off</code>	Disables file rollover.	<code>enable or on</code>	Enables file rollover.	<code>exit or quit</code>	Exits file rollover management.	<code>help or ?</code>	Displays online Help on the rollover commands.	<code>reset</code>	Resets the rollover settings to the default settings.	<code>size number</code>	Specifies the maximum size of each log file in megabytes.	<code>view</code>	Displays the current rollover settings.
<code>disable or off</code>	Disables file rollover.														
<code>enable or on</code>	Enables file rollover.														
<code>exit or quit</code>	Exits file rollover management.														
<code>help or ?</code>	Displays online Help on the rollover commands.														
<code>reset</code>	Resets the rollover settings to the default settings.														
<code>size number</code>	Specifies the maximum size of each log file in megabytes.														
<code>view</code>	Displays the current rollover settings.														
save	<p>Specifies the name (and optionally the full path) of the file to which DTrace saves the selected trace entries. You can specify the start entries and end entries in a range, and choose whether to apply a filter to those entries.</p>														
set	<p>Sets the monitoring level for a service or component. The available levels are Off (o), Brief (b), Medium (m), and Verbose (v). Specify the monitoring level as follows:</p> <pre>set servicename_or_ID level</pre> <p>For example:</p> <pre>set ArchiveTask v</pre> <pre>set 59 m</pre> <p>All lines of code have a minimum monitoring level, and these are viewable within the DTrace log files. For example, if you set the logging level to Medium, only code lines that are marked for Brief and Medium logging show in the log file.</p>														
trigger	<p>Starts logging after a particular string appears in the trace. You set up triggers using the same syntax as for filters.</p>														

Table 9-1 DTrace commands (*continued*)

Command	Description
<code>version</code>	Displays version information on the executable files in the Enterprise Vault program folder (for example <code>C:\Program Files (x86)\Enterprise Vault</code>).
<code>view</code>	Lists all the available processes and services against which you can run DTrace. This list may change slightly depending on what is loaded or installed. It is always a good idea to use <code>view</code> first to see a current list of processes and their IDs. This is particularly important if you want to set a monitoring level with an ID rather than using the name of the process.
<code>watch</code>	After a trigger filter that you have defined with the <code>trigger</code> command has taken effect, records the specified number of trace entries in the log .

Running DTrace from the Administration Console

In the Administration Console, you can choose from a number of supplied DTrace scripts that collect tracing information.

To run DTrace from the Administration Console

- 1 In the Administration Console, expand the Enterprise Vault site until the **Enterprise Vault Servers** container is visible.
- 2 Expand the **Enterprise Vault Servers** container.
- 3 Expand the Enterprise Vault server on which you want to run a trace.
- 4 On the **Tools** menu, check **Advanced Features**.

Note that this setting is not remembered; it applies to the current session of the Administration Console only.

- 5 Press F5 to refresh the view. A **Traces** container appears underneath the server.
- 6 Right-click the **Traces** container, and then click **New > Trace**.
- 7 In the New Trace wizard, enter the following information:
 - The trace category that is closest to the Enterprise Vault subsystem that you want to trace. For example, you might choose "Search and Indexing issues" or "Restoring and Retrieval issues (Exchange)".
 - A title and optional description for the trace. If you log a call with Veritas Support, you may want to include the call number in the trace title. The title

appears in the trace list in the Administration Console and at the start of the trace log file.

- The length of time that you want to run the trace. Trace files can quickly grow large, so a few minutes is usually appropriate.
- A maximum size for the log file. The trace stops if the log file reaches this maximum size.
- The folder in which to store the log file.

After you have started the trace, you can view its properties by double-clicking the trace title in the Administration Console. The Trace Properties dialog box provides options with which you can open and copy the log, but they are unavailable until the trace is complete.

About the DTrace log

[Table 9-2](#) describes the columns in the log.

Table 9-2 Columns in DTrace log

Use this column	To do this
Sequence number	Determine whether any entries have been missed.
Time	Pinpoint slow processes.
Process ID	Identify the processes.
Process name	Identify the processes.
Thread ID	Follow multithread processes (such as the Archiving task).
Highest logging level	Determine the correct logging levels.
Function name	Determine the names of function and the results of those functions.

DTrace troubleshooting

In the unlikely event that you experience problems when you run DTrace, [Table 9-3](#) gives instructions on how to resolve them.

Table 9-3 Potential DTrace problems

Problem	What to do
Lines being skipped.	<p>The first figure on each trace line is the sequence number as it was captured. If there is insufficient CPU time available to process and write entries to the log file, DTrace may skip some lines. If you are tracing an agent task, try to lower the number of threads for the task and monitor a single thread only.</p> <p>If there are multiple tasks of the same type (for example, Archiving), stop all but one of them. DTrace does not differentiate between the different services.</p>
No output on the screen after monitor command, or no trace in log file.	<p>Ensure that you have selected the correct processes for DTracing. If you are running DTrace over Terminal Services or another remote control application that does not use the primary operating system console, note that DTrace events are written to the primary console and so may not appear when using Terminal Services. In Enterprise Vault, a message is posted in the log file to say that Terminal Services was used.</p>
Too much information in the log file.	<p>You can filter and trigger DTrace content based on specific words or events. If you need to focus on the root cause of a problem, you can also limit the number of processes and threads that you monitor.</p>

EVDominoExchangeMigration Tool

This chapter includes the following topics:

- [About the EVDominoExchangeMigration tool](#)
- [Client requirements for the EVDominoExchangeMigration tool](#)
- [Adding the EVDominoExchangeMigration tool to the Windows Server firewall exceptions list](#)
- [EVDominoExchangeMigration tool and Binary Tree](#)
- [Using Quest Notes Migrator for Exchange and the EVDominoExchangeMigration tool](#)
- [Requirements for other migration software with the EVDominoExchangeMigration tool](#)
- [Running the EVDominoExchangeMigration tool](#)

About the EVDominoExchangeMigration tool

The Enterprise Vault EVDominoExchangeMigration tool modifies shortcuts in Exchange Server mailboxes that have been migrated from Domino to Exchange Server.

EVDominoExchangeMigration does the following:

- Copies explicit mailbox permissions from the Exchange Server mailbox to the Domino archive. Typically these are just the permissions of the mailbox owner. No inherited permissions are copied and, in the case of mailboxes on Exchange 2007, no Active Directory permissions are copied to the Domino archive.

- Changes the message class of shortcuts to IPM.Note.EnterpriseVault.Shortcut.
- Corrects links in the shortcuts to items in the Domino archive.
- If the archived item has an attachment, adds the Outlook paperclip icon to the shortcut.

EVDominoExchangeMigration has been tested with Enterprise Vault shortcuts that had been migrated with the following:

- Binary Tree CMT Universal™ 2.7 (also known as CMT for Exchange™).
- Quest Notes Migrator for Exchange from Quest Software.

You can use a different migration tool, but you must ensure that the tool correctly maps the Enterprise Vault Notes document properties to the corresponding Enterprise Vault Exchange named properties.

Note: When you migrate items from Domino to Exchange Server, do not migrate any shortcuts that are in a state of archive-pending. Disable Domino archiving during the migration so that Enterprise Vault does not create new archive-pending shortcuts.

See [“Requirements for other migration software with the EVDominoExchangeMigration tool”](#) on page 57.

Client requirements for the EVDominoExchangeMigration tool

All client computers on which Outlook will be used to access items in the Enterprise Vault Domino archives must have the Enterprise Vault Outlook Add-In installed. You can install the Outlook Add-In before or after running EVDominoExchangeMigration.

Adding the EVDominoExchangeMigration tool to the Windows Server firewall exceptions list

By default, Windows Firewall blocks the EVDominoExchangeMigration tool. To allow the tool through Windows Firewall, you must add it to an exceptions list.

To add the EVDominoExchangeMigration tool to the Windows Server firewall exceptions list

- 1 In Control Panel, click **System and Security**, and then click **Windows Firewall**.
- 2 Click **Allow a program or feature through Windows Firewall**.

- 3 Click **Change settings**, and then click **Allow another program**.
- 4 Click **Browse**, and then browse to the Enterprise Vault program folder (typically, `C:\Program Files (x86)\Enterprise Vault`).
- 5 Click `EVDominoExchangeMigration.exe`, and then click **Open**.
- 6 Click **Add**, and then click **OK**.

EVDominoExchangeMigration tool and Binary Tree

Support for Enterprise Vault shortcuts is included in Binary Tree CMT Universal 2.7 and later.

CMT Universal automatically recognizes Enterprise Vault shortcuts, so no extra configuration is required when you use CMT Universal.

Using Quest Notes Migrator for Exchange and the EVDominoExchangeMigration tool

Before you migrate the users from Domino to Exchange Server, you must add the supplied custom attributes definitions to the Quest program folder. The settings in this file enable Quest to migrate Enterprise Vault shortcut attributes to Exchange Server mailboxes.

To define Quest custom attributes

- 1 Copy the supplied `example_customattrs.tsv` file from the Enterprise Vault program folder to the Quest Notes Migrator for Exchange program folder (for example `C:\Program Files (x86)\Quest Software Notes Migrator for Exchange`).
- 2 Rename the new copy of `example_customattrs.tsv` to `customattrs.tsv`.

You can now use Quest Notes Migrator to migrate mailboxes to Exchange Server. See the Quest Notes Migrator documentation for details of the process.

Warning: Do not run the Enterprise Vault mailbox archiving task on newly-migrated mailboxes. Doing so may archive the shortcuts that EVDominoExchangeMigration is needed to fix. Consider disabling the mailbox archiving task until EVDominoExchangeMigration has corrected the shortcuts.

Requirements for other migration software with the EVDominoExchangeMigration tool

EVDominoExchangeMigration has been tested with items that had been migrated using Binary Tree Universal and with Quest Notes Migrator for Exchange. If you want to use a different mailbox migration tool, you must ensure that the appropriate Enterprise Vault message attributes are mapped to their corresponding MAPI attributes.

[Table 10-1](#) lists the mappings required for message attributes.

Exchange named properties must all have a GUID of D0F41A15-9E91-D111-84E6-0000F877D428 and be of kind MNID_STRING.

Table 10-1 Enterprise Vault message attributes

Enterprise Vault Notes document property	Enterprise Vault Exchange document named property	Named property type
EV26C5E2CCF2B9267C. Archived	Archive ID	PT_STRING8
EV26C5E2CCF2B9267C. ArchivedDate	Archived Date	One of the following: <ul style="list-style-type: none"> ■ PT_SYSTIME ■ PT_STRING8 in the format <i>yyyyddmmhhmmss</i>. For example, 20071910141249 represents 2007/19/10 14:12.49.
EV26C5E2CCF2B9267C. SaveSetId	Saveset ID	PT_STRING8
EV26C5E2CCF2B9267C. RetentionCategory	Retention Category	PT_STRING8
EV26C5E2CCF2B9267C. HasAttachments	EVLotus_HasAttachments	PT_STRING8

Running the EVDominoExchangeMigration tool

This section describes how to run EVDominoExchangeMigration.

EVDominoExchangeMigration processes the shortcuts in a single mailbox. If you want to process the shortcuts in multiple mailboxes you must run EVDominoExchangeMigration once for each mailbox. If you have a large number of mailboxes to process the easiest method is to run EVDominoExchangeMigration from a script or batch file.

Syntax for EVDominoExchangeMigration tool

```
EVDominoExchangeMigration [-?] -ex ExchangeServer -sm SystemMailbox
-eu ExchangeSMTPAddress -du DominoUserName -po ExchangeMailboxPolicy
-lf LogFileFolder
```

where the parameters are as follows:

- ex The name of the Exchange Mailbox server that hosts the mailbox you want to process.
- eu The primary Exchange Server SMTP address of the user whose mailbox you want to process.
- du The Domino user name of the migrated user (for example, User1/MyOrgName) or the archive ID of the Enterprise Vault archive for the Domino user (for example 1C5D73ABD3B80474396FD566AB2A894031110000myServer.myCorp.com)
- po The Enterprise Vault Exchange Mailbox policy to apply. Must be one of Default, the name of a policy, or None.
 - Default: The Exchange Mailbox policy to apply when updating the shortcuts. If the user has been provisioned, this is the Mailbox Policy specified in the provisioning group. If the user has not been provisioned, this is the Default Exchange Mailbox Policy.
 - Policy Name: The name of the Exchange mailbox policy to use.
 - None: Do not apply a policy. This option does not correct links in shortcuts but does improve performance. Do not use this option if shortcuts contain links to the archived items.
- lf The absolute path of the folder that will contain the log files. The folder will be created if it does not exist. For example: C:\Migration\Logs. Note that only a single folder can be created automatically. In the example, the Logs folder would not be created unless C:\Migration already existed.

For example, the following command runs EVDominoExchangeMigration with the following settings:

```
Exchange Server provisioned mailbox    User12
Exchange Server                            myExchange
```

Primary SMTP address of User12	User12@myCorp.com
Exchange Mailbox policy	The mailbox policy from the user's provisioning group
Domino user name	User12/myCorp
Log file folder	C:\log files

```
EVDominoExchangeMigration -ex myExchange -eu User12@myCorp.com -po
default -du User12/myCorp -lf "C:\log files"
```

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

Log files for EVDominoExchangeMigration tool

EVDominoExchangeMigration creates the following log files:

- A log file for each mailbox that is processed. The file name is a combination of the SMTP address of the user, the date, and the time. For example, `user1@mycorp.com 2007-09-27 09-17-08.log`.
- A log file called `EVDominoExchangeMigrationSummary.log`, which contains a summary of all migrations. EVDominoExchangeMigration writes a one-line summary to this file for each mailbox that it processes. The information in the file is tab-separated, so you can easily open it with a spreadsheet program. EVDominoExchangeMigration never overwrites this log file, so you can use the same summary log file for multiple runs of EVDominoExchangeMigration.

Limitations of EVDominoExchangeMigration tool

[Table 10-2](#) describes some known limitations in the EVDominoExchangeMigration tool that you need to be aware of.

Table 10-2 EVDominoExchangeMigration tool limitations

Limitation	Description
Non-US-ASCII characters in Domino mailbox names may break scripted migrations.	<p>When there are many mailboxes to process, it can be convenient to run EVDominoExchangeMigration in a script. Note that, if the Domino mailbox name contains characters that are outside the US-ASCII character set, pasting the mailbox names into a Windows text editor is likely to result in failures because the characters are not interpreted correctly.</p> <p>There are various possible solutions to this problem, including the following:</p> <ul style="list-style-type: none"> ■ Create a Windows PowerShell script to process a list of mailbox names. ■ Use the MS-DOS Editor to create a batch file, as this lets you paste non-US-ASCII text. To do this, perform the following steps: <ul style="list-style-type: none"> ■ Open a Command Prompt window. ■ Type edit, and then press Enter. ■ Right-click the title bar of the Command Prompt window and then, on the shortcut menu, click Edit > Paste.
Appearance of shortcuts to Domino Calendar and To Do items.	<p>These items are retrieved as normal mail messages (IPM.Note) rather than as Calendar (IPM.Appointment) or To Do items (IPM.Task).</p> <p>We recommend that, if possible, you do not archive Domino Calendar and To Do items from Domino mailboxes but instead wait until they have been migrated to Microsoft Exchange and then archive them using the Microsoft Exchange Archiving task. They are then retrieved correctly.</p>
Message class restrictions	<p>EVDominoExchangeMigration does not process messages that have any of the following Exchange Server message classes:</p> <ul style="list-style-type: none"> ■ IPM.Appointment ■ IPM.Contact ■ IPM.Task ■ IPM.Stickynote

EVDuplicateCleaner

This chapter includes the following topics:

- [About EVDuplicateCleaner](#)
- [Configuring EVDuplicateCleaner](#)
- [Running EVDuplicateCleaner](#)
- [Fixing broken shortcuts after you have run EVDuplicateCleaner](#)

About EVDuplicateCleaner

In rare circumstances, a failure of one of the archiving tasks can cause Enterprise Vault to create duplicate savesets in mailbox archives. If you experience this issue, you can find and delete the duplicates by running the EVDuplicateCleaner command-line utility. After you have run the utility, only the last-archived instance is left from each set of duplicates.

EVDuplicateCleaner does not delete duplicate savesets if they are on legal hold.

To identify duplicates, EVDuplicateCleaner does the following:

- It checks multiple attributes of each saveset (ArchivePointIdentity, IdDateTime, IdUniqueNo, SisPartCount, and VaultIdentity).
- It examines the content fingerprint of each saveset.

Configuring EVDuplicateCleaner

Before you can run EVDuplicateCleaner, you must edit the configuration file that accompanies it.

To configure EVDuplicateCleaner

- 1 In Windows Explorer, browse to the Enterprise Vault program folder (typically `C:\Program Files\Enterprise Vault`).
- 2 Open the file `EVDuplicateCleaner.exe.config` in a plain-text editor such as Windows Notepad.
- 3 Set the following configuration values:

<code>DirDBSQLServer</code>	Mandatory. Specifies the name of the SQL Server computer that hosts the Enterprise Vault Directory database.
<code>MaxAllowedFailures</code>	Optional. Specifies the maximum number of errors that can occur before EVDuplicateCleaner stops processing. By default, an unlimited number of errors can occur.
<code>SavesetChunkSize</code>	Optional. Specifies the number of potential duplicates that EVDuplicateCleaner should fetch in each call to the Enterprise Vault Directory database. The default is 10000. By increasing this value, you can reduce the number of calls that EVDuplicateCleaner makes to the Directory database and thereby speed up the processing time. On the other hand, the larger the chunk size, the more memory you need for the fetched items.
<code>ErrorToLogFile</code>	Optional. Specifies whether to record errors in a log file. By default, EVDuplicateCleaner does log errors.
<code>MaxDuplicateSavesetsToReport</code>	Optional. Specifies the maximum number of duplicate savesets that EVDuplicateCleaner can process when it runs in report mode. The default is 100000.

TryToOverrideDumpster	<p>Optional. Specifies whether to delete the duplicate savesets immediately or retain them temporarily in the Enterprise Vault "dumpster", if you have enabled it. By default, EVDuplicateCleaner tries to delete the duplicate savesets immediately.</p> <p>The dumpster serves as a recycle bin in which Enterprise Vault retains deleted items for a specified period before it permanently deletes them. To enable the dumpster, check Enable recovery of user deleted items on the Site Properties: Archive Settings tab in the Administration Console.</p>
SkipLegacySavesets	<p>Optional. Specifies whether to ignore any legacy savesets that Enterprise Vault 2007 or earlier has created. By default, EVDuplicateCleaner does not ignore these savesets.</p> <p>You can speed up the processing time by choosing to ignore legacy savesets, if you are sure that you do not have any.</p>
SQLCommandTimeout	<p>Optional. Specifies the maximum number of seconds for which each SQL query that EVDuplicateCleaner issues can run before it times out. The default is 300. We recommend that you only increase this value if you experience 'Timeout expired' errors when you run EVDuplicateCleaner.</p>

- 4 Save and close the file.

Running EVDuplicateCleaner

If you have not already done so, configure EVDuplicateCleaner by setting the required values in the `EVDuplicateCleaner.exe.config` file.

Running EVDuplicateCleaner

- 1 Log on to the Enterprise Vault server as the Vault Service account.
- 2 Stop the Exchange Journaling task and Domino Journaling task.
- 3 Open a command prompt window with administrator privileges.
- 4 Change to the Enterprise Vault program folder (typically `C:\Program Files\Enterprise Vault`).
- 5 Type one of the following commands:

Fixing broken shortcuts after you have run EVDuplicateCleaner

- `EVDuplicateCleaner Execute vault_store_name [archive_entry_ID]`
This command instructs EVDuplicateCleaner to delete all the duplicate savesets that it finds.
- `EVDuplicateCleaner Report vault_store_name [archive_entry_ID]`
This command generates a log file that lists the duplicate savesets, but it does not delete them.
- `EVDuplicateCleaner Summary vault_store_name [archive_entry_ID]`
This command provides a count of the estimated number of duplicate savesets in each archive.

The `vault_store_name` parameter is mandatory. The `archive_entry_ID` parameter is optional and lets you process the nominated archive only. If you omit this parameter, EVDuplicateCleaner processes all the archives in the nominated vault store.

EVDuplicateCleaner generates two log files for each processing run:

- `EVDuplicateCleaner_timestamp.log` for legacy (pre-8.0) savesets
- `EVDuplicateCleanerv2_timestamp.log` for 8.0 and later savesets

In each log file, the "Estimated duplicate saveset counts" values show the number of savesets for which duplicates may exist; the "Estimated duplicate item counts" values show the potential number of duplicates. For example, when three savesets each have four duplicates, the "Estimated duplicate saveset counts" is 3 and the "Estimated duplicate item counts" is 12.

- 6 Restart the Exchange Journaling task and Domino Journaling task.

Fixing broken shortcuts after you have run EVDuplicateCleaner

After you run EVDuplicateCleaner, Exchange users may temporarily be unable to retrieve certain items from their mailbox archives. This issue arises when a mailbox shortcut is associated with a now-deleted instance of a duplicate saveset. The issue does not typically affect Domino mailboxes.

You can fix the broken shortcuts by adding a FixOrphanedShortcut entry to the registry on the Enterprise Vault server. After you add the entry, Enterprise Vault repairs each broken shortcut by associating it with the surviving instance of the duplicated saveset. If Enterprise Vault cannot find this instance, it deletes the shortcut.

To fix broken shortcuts after you have run EVDuplicateCleaner

- 1 Open the registry editor.
- 2 Locate and then click the following key in the registry:

```
HKEY_LOCAL_MACHINE
  \Software
    \Wow6432Node
      \KVS
        \Enterprise Vault
          \Agents
```

- 3 Create a DWORD entry that is called FixOrphanedShortcut, and give it a value of 1.
- 4 Run the Exchange Mailbox archiving task to process the shortcuts.

The task normally runs according to a schedule that you set up. However, if you want to run the task outside this schedule, you can use the Run Now facility in the Administration Console to run it immediately. See the *Administrator's Guide* for instructions.
- 5 After you have fixed the shortcuts, remove the FixOrphanedShortcut registry entry. This entry can have an adverse effect on archiving performance if left in place.

EVEARemovalUtility

This chapter includes the following topics:

- [About EVEARemovalUtility](#)
- [EVEARemovalUtility prerequisites](#)
- [Running EVEARemovalUtility](#)
- [EVEARemovalUtility syntax](#)
- [Format of the EVEARemovalUtility output and log files](#)
- [EVEARemovalUtility usage examples](#)

About EVEARemovalUtility

EVEARemovalUtility is a command-line utility that removes extended attributes from files.

Enterprise Vault cannot create placeholder shortcuts on NTFS file systems for files with extended attributes, such as files that were migrated from Novell file systems or files that were previously archived with applications such as EMC DiskXtender. This limitation is due to a Microsoft restriction. Placeholders use reparse points, which cannot contain extended attributes.

Enterprise Vault can archive files with extended attributes, but placeholder creation fails. Enterprise Vault includes a message similar to the following in the File System Archiving task report:

```
Failed to write a placeholder file because it has extended attributes.  
Use EVEARemovalUtility to remove the extended attributes.  
File Name: %1
```

Note: To obtain this message in the task report, a Windows file server must have the Enterprise Vault 10.0 or later FSA Agent installed.

You can use EVEARemovalUtility to remove the extended attributes from files. If placeholder creation failed previously, the removal of the extended attributes allows Enterprise Vault to create the placeholders on the next archiving run.

EVEARemovalUtility prerequisites

The user account under which you run EVEARemovalUtility requires one of the following:

- Local administrator rights on the file servers.
- Full access on the shares, and both Read Extended Attributes and Write Extended Attributes permissions on the files and folders.

If the computer has User Account Control (UAC) enabled, you must run the utility with Administrator privileges.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

You can run the utility on the Enterprise Vault server or another Windows computer, such as a Windows file server that contains the files you want to process. The computer must have the following software installed:

- .Net Framework 4.5.2
- Microsoft Visual C++ 2013 redistributable package

To run the utility on a Windows computer other than the Enterprise Vault server, copy the following files from the Enterprise Vault installation folder on the Enterprise Vault server to a suitable folder on the Windows computer. Do not overwrite any existing files:

- `EVEARemovalUtility.exe`
- `EVManagedLibrary.dll`
- `KVS.EnterpriseVault.Common.dll`
- `KVS.EnterpriseVault.FileServerArchiveCommon.dll`
- `KVS.EnterpriseVault.Runtime.dll`
- `EVRT.dll`

Running EVEARemovalUtility

You can run EVEARemovalUtility from the Enterprise Vault server, or from another Windows computer such as a Windows file server that holds the files you want to process.

See [“EVEARemovalUtility prerequisites”](#) on page 67.

Note: The utility permanently removes all extended attributes from the files that it processes. Before you run EVEARemovalUtility, we recommend that you take a full backup of the files that you want to process.

To run EVEARemovalUtility

- 1 Open a command prompt window and change directory to the folder that contains the EVEARemovalUtility files. On an Enterprise Vault server the files are in the Enterprise Vault installation folder.
- 2 Run EVEARemovalUtility with the required parameters.

We recommend that before you run the utility with the `-r` parameter, you run it with the `-l` parameter and the `-d` parameter, to create a log file that lists the attributes that the utility would remove.

See [“EVEARemovalUtility syntax”](#) on page 68.

EVEARemovalUtility syntax

The syntax of the EVEARemovalUtility command is as follows:

```
EVEARemovalUtility.exe path [-s | -f] [-l] [-d] [-r [-q]]
```

[Table 12-1](#) describes the available parameters.

Table 12-1 EVEARemovalUtility parameters

Parameter	Description
<i>path</i>	<p>One of the following:</p> <ul style="list-style-type: none"> ■ A UNC path for a single file to process, for example <code>\\fileserver1\share\file.txt</code> ■ A UNC path for a folder to process, for example <code>\\fileserver1\share</code> ■ A local path or UNC path for a log file that EVEARemovalUtility has previously generated. You must use the <code>-f</code> option in this case.

Table 12-1 EVEARemovalUtility parameters (*continued*)

Parameter	Description
-s	Process recursively the folder that is specified in <i>path</i> .
-f	Process the list of files in the EVEARemovalUtility log file that <i>path</i> specifies.
-l	Redirect the utility's output to a log file. EVEARemovalUtility creates the log file in the folder in which the utility is located. If you do not specify -l, EVEARemovalUtility displays its output in the command prompt window.
-d	Generate detailed output, which includes the names and values of the extended attributes for each file. This parameter has no effect if you specify the -r parameter.
-r	Remove extended attributes. If you omit this parameter, EVEARemovalUtility only lists information about the extended attributes.
-q	Run in quiet output mode. The output consists only of a summary which shows the number of processed files. This parameter has no effect unless you specify the -r parameter.

Format of the EVEARemovalUtility output and log files

The output of EVEARemovalUtility appears in the command prompt window, unless you specify the -l parameter to redirect the output to a log file.

The log file name format is `EVEARemovalUtility--timestamp.log`, where *timestamp* indicates when the log file was created. *timestamp* has the format `yyyymmddmmsscc`, where *cc* indicates hundredths of a second. For example, the log file `EVEARemovalUtility--20100907142304.log` was created at 14:23 and 04 hundredths of a second on 7th September 2010.

The following command generates a log file that lists the details of the extended attributes for the files in a folder and its subfolders:

```
EVEARemovalUtility.exe \\server1\e$\folder1 -d -s -l
```

Here is an example of the output from this command:

Extended Attribute Removal Utility.
 Veritas Enterprise Vault.
 Copyright (c) 2010. Veritas Technologies LLC.

List extended attributes from \\server1\e\$\folder1

```
-----
Filename  ExtAttrSTATE  Details
-----
##
\\server1\e$\folder1\file1.txt  PRESENT  <EA1-Value>, <EA2-Value2>
\\server1\e$\folder1\file2.txt  NOT PRESENT
\\server1\e$\folder1\file3.txt  PRESENT  <EA1-Value3>
\\server1\e$\folder1\file4.txt  PRESENT  <EA1-Value>
\\server1\e$\folder1\file5.txt  NOT PRESENT
\\server1\e$\folder1\file6.doc  NOT PRESENT
\\server1\e$\folder1\file7.txt  NOT PRESENT
\\server1\e$\folder1\file8.doc  NOT PRESENT
\\server1\e$\folder1\subfolder\file9.doc  PRESENT  <CS-12>, <AUTHOR-P1>
\\server1\e$\folder1\subfolder\file91.doc  NOT PRESENT
##
```

Summary

```
-----
Present  Not present  Start time  End time
-----
4  6  6-10-2010 At 20:51:22.137  6-10-2010 At 20:51:22.387
```

Total elapsed time : 0 hours 0 mins 0 seconds 249 msec

If you omit the `-d` parameter, the output omits the names and the values of the extended attributes.

If required, you can edit the contents of a log file before submitting it for processing with the `-f` parameter. For example, you may want to remove the extended attributes from all of the files that are listed in the example log file, except `file4.txt`. You can edit the log file to delete the line for `file4.txt`, and then submit the log file for processing.

Note: Do not to change the format of the lines that are bounded by the `##` characters, otherwise the utility may fail to read the file list correctly.

When you use the `-r` parameter to remove extended attributes and you also include the `-q` parameter, the command produces "quiet" output. The output then includes only summary information about the number of processed files.

EVEARemovalUtility usage examples

These example scenarios illustrate how you can use the EVEARemovalUtility to remove extended attributes:

- See [“EVEARemovalUtility example: processing a single file”](#) on page 71.
- See [“EVEARemovalUtility example: processing a folder and its subfolders”](#) on page 71.

EVEARemovalUtility example: processing a single file

Suppose that Enterprise Vault reports that it has failed to create a placeholder for a single file named `filex.txt` on file server `fs1`, share `e$`, folder `folder1`, because the file contains extended attributes.

- You run the following command from the Enterprise Vault server:

```
EVEARemovalUtility.exe \\fs1\e$\folder1\filex.txt -d
```

The output to the command prompt window lists the extended attributes for `filex.txt`.

- You decide that you want to remove the extended attributes. You run the following command to remove the extended attributes from `filex.txt`:

```
EVEARemovalUtility.exe \\fs1\e$\folder1\filex.txt -r
```

The output to the command prompt window indicates that the extended attributes for `filex.txt` have been removed.

EVEARemovalUtility example: processing a folder and its subfolders

Suppose that either of the following applies:

- Enterprise Vault reports that it has failed to create the placeholders for several files in the folder `folder1` and its subfolders on file server `fs1`, share `e$`, because the files contain extended attributes.
- Or you have migrated a folder structure from a UNIX system to a Windows file server and you want to remove extended attributes from the migrated files before you archive them with Enterprise Vault.

You might process the files with EVEARemovalUtility as follows:

- You enter the following command on the Enterprise Vault server to create a log file that lists details of the extended attributes for all the files on the relevant path, including subfolders:

```
EVEARemovalUtility.exe \\fs1\e$\folder1 -s -l -d
```

- You examine the log file `EVEARemovalUtility--timestamp.log`, and decide that you want to remove the extended attributes from all of the listed files that contain them.

- You enter the following command to remove the extended attributes:

```
EVEARemovalUtility.exe EVEARemovalUtility--timestamp.log -f -r
```

As the `-l` parameter is not specified, the results of the removal appear in the command prompt window.

Alternatively, you can run the following command to remove the extended attributes from all of the files in `folder1` and its subfolders. This command sends its output to a new log file:

```
EVEARemovalUtility.exe \\fs1\e$\folder1 -s -l -r
```

The following command has the same effect, but runs in quiet output mode. The command outputs to a log file only the summary information about the number of files it has processed:

```
EVEARemovalUtility.exe \\fs1\e$\folder1 -s -l -r -q
```

See [“Format of the EVEARemovalUtility output and log files”](#) on page 69.

EVFSASetRightsAndPermissions

This chapter includes the following topics:

- [About EVFSASetRightsAndPermissions](#)
- [Running EVFSASetRightsAndPermissions](#)

About EVFSASetRightsAndPermissions

On Windows file servers, unless the Vault Service account is a member of the local Administrators group, it requires a set of minimum permissions and privileges for File System Archiving. See the appendix “Permissions and privileges required for the Vault Service account on Windows file servers” in *Setting up File System Archiving*.

If you change the Vault Service account you must ensure that the new account is granted the required permissions and privileges. You can use the EVFSARightsAndPermissions utility to configure the permissions and privileges for the new account.

The EVFSARightsAndPermissions utility is installed on a file server when you install the FSA Agent.

The utility creates a log file named `EVFSASetRightsAndPermissions.log` in the Enterprise Vault program folder. The log file lists all the rights and permissions it has granted to the specified account, and indicates success or failure for each stage of the configuration.

Note: Ensure that your group policy permissions do not override the required local permissions for the Vault Service account.

Running EVFSASetRightsAndPermissions

You must run EVFSASetRightsAndPermissions using an account that is a member of the local Administrators group on the file server.

To run EVFSASetRightsAndPermissions

- 1 On the file server, log on as a user that is a member of the local Administrators group.
- 2 Open a command prompt window.
- 3 Navigate to the Enterprise Vault program folder (for example `C:\Program Files (x86)\Enterprise Vault`).
- 4 Type the following command:

```
EVFSASetRightsAndPermissions username
```

where *username* is the name of the Vault Service account.
- 5 The console output indicates the progress of the utility. If necessary, check the output log file `EVFSASetRightsAndPermissions.log` in the Enterprise Vault program folder.

EVrights

This chapter includes the following topics:

- [About EVrights](#)
- [EVrights syntax](#)

About EVrights

Use EVrights to grant rights to users and groups from a command line or batch file. You require Administrator privileges to set user rights.

EVrights syntax

`EVrights name right`

The name identifies the user or group whose rights you want to modify. Enclose the name in quotation marks if it contains space characters.

[Table 14-1](#) describes the rights that you can grant. These rights are case-sensitive and must be typed exactly as they appear.

Table 14-1 Available rights

Right	Description
SeAssignPrimaryTokenPrivilege	Replace a process level token.
SeAuditPrivilege	Generate security audits.
SeBackupPrivilege	Back up files and directories.
SeBatchLogonRight	Log on as a batch job.
SeChangeNotifyPrivilege	Bypass traverse checking.

Table 14-1 Available rights (*continued*)

Right	Description
SeCreatePagefilePrivilege	Create a page file.
SeCreatePermanentPrivilege	Create permanent shared objects.
SeCreateTokenPrivilege	Create a token object.
SeDebugPrivilege	Debug programs.
SeIncreaseBasePriorityPrivilege	Increase scheduling priority.
SeIncreaseQuotaPrivilege	Increase quotas.
SeInteractiveLogonRight	Log on locally.
SeLoadDriverPrivilege	Load and unload device drivers.
SeLockMemoryPrivilege	Lock pages in memory.
SeMachineAccountPrivilege	Add workstations to domain.
SeNetworkLogonRight	Access this computer from the network.
SeProfileSingleProcessPrivilege	Profile single process.
SeRemoteShutdownPrivilege	Force shutdown from a remote system.
SeRestorePrivilege	Restore files and directories.
SeSecurityPrivilege	Manage auditing and security log.
SeServiceLogonRight	Log on as a service.
SeShutdownPrivilege	Turn off the system.
SeSystemEnvironmentPrivilege	Modify firmware environment values.
SeSystemProfilePrivilege	Profile system performance.
SeSystemtimePrivilege	Change the system time.
SeTakeOwnershipPrivilege	Take ownership of files or other objects.
SeUnsolicitedInputPrivilege	Read unsolicited input from a terminal device.

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

EVservice

This chapter includes the following topics:

- [About EVservice](#)
- [EVservice prerequisites](#)
- [EVservice syntax](#)
- [EVservice list file format](#)

About EVservice

EVservice is a command-line utility that lets you start and stop Windows services and Enterprise Vault tasks on local or remote computers. EVservice can also pause and resume services and Enterprise Vault tasks that accept pause and resume requests.

Note the following:

- If you are running Enterprise Vault in a clustering environment, you can control tasks with EVservice but you cannot control services. To control services in a VCS cluster, use the `hares` command that is described in the *Veritas Cluster Server Administrator's Guide*.
- You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.
See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

EVservice prerequisites

If you intend to use EVservice to manage Enterprise Vault tasks on remote computers, ensure that the Enterprise Vault Administration Console is installed on the same computer as EVservice.

If you want to start or stop a service or Enterprise Vault task that is on a remote computer, the account that you use to run EVservice must be a member of the local administrator's group on the same computer as the service or task. If you add an account to the local administrator's group on the remote computer, you may find that you need to restart the computer before you can use EVservice.

EVservice syntax

Note that for those commands below that control services, you can specify any valid alias or name for the `computer` parameter. However, for those commands that control tasks, the specified `computer` name must correspond to the computer alias used in Enterprise Vault.

- `EVservice start|stop|pause|resume computer service [service...]`
Starts, stops, pauses, or resumes the specified services on the computer with the specified alias or name. If a service name contains spaces, enclose it in quotation marks. For example, the following command starts the Enterprise Vault Shopping service on computer GAMMA:

```
EVservice start GAMMA "Enterprise Vault Shopping Service"
```
- `EVservice start|stop|pause|resume computer task [task...]`
Starts, stops, pauses, or resumes the specified Enterprise Vault tasks on the computer with the specified name, which must correspond to the computer alias used in Enterprise Vault. If a task name contains spaces, enclose it in quotation marks. For example, the following command starts "Public Folder task for GAMMA" on computer OMEGA:

```
EVservice start OMEGA "Public Folder task for GAMMA"
```
- `EVservice start|stop|pause|resume computer listfile`
Starts, stops, pauses, or resumes the services and Enterprise Vault tasks that are listed in the named text file, which can be local or remote. For example, the following command starts the services and tasks that are listed in the file `evservices_and_tasks.txt`:

```
EVservice start GAMMA evservices_and_tasks.txt
```

The file can contain entries for many computers. However, the command acts on the services that are running on the computer that you specify on the command line.

- `EVservice start|stop|pause|resume listfile`
Starts, stops, pauses, or resumes all the services and Enterprise Vault tasks that are listed in the named text file.
EVservice starts the services and tasks in the order in which they are listed in the list file, and stops them in reverse order.

EVservice list file format

The format of the list file is as follows:

```
computer:service_or_task
```

EVservice ignores any line that does not contain a colon (:), so you can add comments if required. For example:

```
Enterprise Vault Service and Task Startup List (comment line)
GAMMA:Enterprise Vault Directory Service
GAMMA:Enterprise Vault Indexing Service
GAMMA:Enterprise Vault Shopping Service
GAMMA:Enterprise Vault Storage Service
GAMMA:Mailbox Archiving Task for EXCH1
DELTA:Mailbox Archiving Task for EXCH2
```

Note the following:

- In those lines where you specify the name of an Enterprise Vault task, the `computer` name must correspond to the computer alias used in Enterprise Vault.
- The easiest way to stop all tasks is to stop the Task Controller service. You can edit each task's properties to set its Startup type to Automatic, so that the tasks start automatically when you restart the Task Controller service. See the *Administrator's Guide* for more information.
- If you were to use the sample file above with the following command, the task on computer DELTA would be unaffected (because you have specified the computer GAMMA on the command line):

```
EVservice start GAMMA evservices_and_tasks.txt
```

EVSPShortcutManager

This chapter includes the following topics:

- [About EVSPShortcutManager](#)
- [Permissions required to run EVSPShortcutManager](#)
- [EVSPShortcutManager syntax](#)
- [EVSPShortcutManager examples](#)

About EVSPShortcutManager

EVSPShortcutManager is a command-line utility that enables you to manage the Enterprise Vault shortcuts that are in SharePoint.

You can use EVSPShortcutManager to do the following:

- Replace HTML shortcuts with new shortcuts that behave exactly like SharePoint documents. The new shortcuts use the same icons as the corresponding original documents.
- Recall archived items to replace all shortcuts in an entire site, collection, or library with the corresponding original documents.

Before Enterprise Vault 8.0 SP3, Enterprise Vault created HTML shortcuts in SharePoint. Enterprise Vault 8.0 SP3 introduced new SharePoint shortcuts that provide a seamless experience for users.

- Shortcuts can be edited and any changes are saved back to SharePoint.
- The shortcuts do not break SharePoint workflows. Previously, Enterprise Vault never replaced workflow items with shortcuts.
- Existing links to a document do not break when the document is archived.
- The shortcuts use the archived documents' original icons.

Permissions required to run EVSPShortcutManager

The account that you use to run EVSPShortcutManager must have the following roles:

- The local administrator role on the SharePoint server.
- The sysadmin server role on the SharePoint configuration database.

The account must also have one of the following roles:

- Site Collection Administrator. This role enables EVSPShortcutManager to process every site in the site collection.
- Web Application Administrator. This role enables EVSPShortcutManager to process every site in the web application. This role uses the Policy for Web Application to provide Full Control permission.
- Site Administrator. This role provides Full Control permission to the site. In this case the account must also have Full Control permission on the document libraries in the sites that EVSPShortcutManager processes.

Additionally, if you use the `-server` option, the account must have access to the SharePoint_Config configuration database.

EVSPShortcutManager syntax

`EVSPShortcutManager operation location -url "url" [options]`

Table 16-1 describes the parameters you can use with EVSPShortcutManager.exe.

Table 16-1 EVSPShortcutManager.exe parameters

Argument	Description
<i>operation</i>	<p>Specifies the action that you want to perform.</p> <p>The <i>operations</i> that you can specify are as follows:</p> <ul style="list-style-type: none"> ■ <code>-convert</code>. Use this option to replace HTML shortcuts with new shortcuts that behave exactly like SharePoint documents. ■ <code>-recall</code>. Use this option to replace shortcuts with the corresponding archived SharePoint documents. Before you use this option, run EVSPShortcutManager with the <code>-convert</code> option to make sure that all HTML shortcuts are replaced with new shortcuts.

Table 16-1 EVSPShortcutManager.exe parameters (*continued*)

Argument	Description
<i>location</i>	<ul style="list-style-type: none"> ■ <code>-server</code>. Process the entire SharePoint server. You do not need to supply a URL when you specify <code>-server</code>. ■ <code>-site</code> ■ <code>-library</code>
<i>url</i>	<p>The URL of the SharePoint site, collection, or library that you want to process. If there are spaces in the URL you must enclose the URL in quotes or use %20 to represent each space.</p> <p>You do not need to supply a URL when you specify <code>-server</code>.</p>
<i>options</i>	<ul style="list-style-type: none"> ■ <code>-report</code>. Run EVSPShortcutManager in report mode. In report mode EVSPShortcutManager does not process shortcuts but does create a log file that shows what EVSPShortcutManager would process if you ran it normally. ■ <code>-silent</code>. Use this option to ensure that EVSPShortcutManager never prompts for confirmation. Use <code>-silent</code> when you use EVSPShortcutManager in a script. ■ <code>-log</code>. Specify this option with a folder path to make EVSPShortcutManager create the log file in that folder. If you omit this option, EVSPShortcutManager creates a log file in the <code>SharePoint Logs\EVSPShortcutManager</code> subfolder of the Enterprise Vault installation folder. The log file name is <code>EVSPShortcutManageryyyyMMddHHmmss.xml</code> where <code>yyyymmddHHmmss</code> indicates the date and time when the log file was created. ■ <code>-norecurse</code>. Use this option with <code>-site</code> to avoid processing subsites. EVSPShortcutManager processes the subsites by default unless you specify this option. <p>You can combine options as needed. For example, you can use both <code>-report</code> and <code>-log</code> to run EVSPShortcutManager in report mode and to specify the log file location.</p>

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

EVSPShortcutManager examples

- To scan a SharePoint server for HTML shortcuts without converting any shortcuts and to place the log file in the default folder:

```
EVSPShortcutManager -convert -server -report
```

- To scan a SharePoint server for HTML shortcuts and place the log file in a folder named C:\MyLogs

```
EVSPShortcutManager -convert -server -report -log C:\MyLogs
```

- To convert HTML shortcuts to new shortcuts in a site library named "Financial Documents":

```
EVSPShortcutManager -convert -library -url "http://mySite/Financial Documents"
```

- To replace the shortcuts with corresponding archived documents at the site named "mySite" and not process subsites:

```
EVSPShortcutManager -recall -site -norecurse -url http://mySite
```

EVSVR

This chapter includes the following topics:

- [About EVSVR](#)
- [Starting EVSVR](#)
- [EVSVR commands](#)
- [EVSVR application states](#)
- [Creating an EVSVR operation file](#)
- [Editing an EVSVR operation file in which you have enabled checkpointing](#)
- [Running an EVSVR operation](#)
- [About the EVSVR operation settings](#)
- [Using the output from one EVSVR operation as input for another operation](#)
- [Viewing the EVSVR output log file](#)
- [Running EVSVR in interactive mode](#)
- [Improving EVSVR performance when processing CAB collections](#)

About EVSVR

EVSVR is a command-line utility with which you can report on, verify, and repair Enterprise Vault storage.

[Table 17-1](#) summarizes the types of operations that EVSVR can perform.

Table 17-1 EVSVR operation types

Operation type	Description
Report	<p>This operation provides a count or listing of the following:</p> <ul style="list-style-type: none"> ■ The items in vault store partitions. ■ The records in vault store databases and fingerprint databases. ■ Certain records in the Enterprise Vault Directory database. <p>For example, a report operation can provide the following:</p> <ul style="list-style-type: none"> ■ A count of all the files in the site's vault stores that were archived within the last two days. ■ The details of each saveset record in a vault store database. ■ The archive and archive folder information in the Directory database.
Verify	<p>This operation does one or more of the following:</p> <ul style="list-style-type: none"> ■ Verifies the vault store database and fingerprint database records against the vault store objects that they reference. ■ Verifies that vault store objects have valid records in the vault store databases and fingerprint databases. ■ Verifies the vault store database records against the equivalent fingerprint database records. ■ Verifies the vault store database records against the equivalent Directory database records. ■ Determines the number of collection records in the vault store databases that do not have creation dates.

Table 17-1 EVSVR operation types (*continued*)

Operation type	Description
Repair	<p>This operation does one or more of the following:</p> <ul style="list-style-type: none"> ■ Uses the vault store objects to repair the records within the vault store databases and between the vault store databases and fingerprint databases. ■ Blacklists any SIS parts that do not verify correctly. After you blacklist a SIS part, archiving a new item with the same SIS part causes Enterprise Vault to create a new SIS part file on disk. ■ Deletes the vault store and fingerprint database records that are associated with missing items. ■ Recreates any missing saveset and SIS part records in the vault store and fingerprint databases. ■ Sets a creation date for any collection record in a vault store database that does not have one. ■ Recreates any missing archive and archive folder information in the vault store databases when the corresponding information exists in the Directory database. ■ For Exchange Mailbox and File System archives, recreates any missing archive and archive folder information in the Directory database when the corresponding information exists in the vault store databases. ■ For Exchange Mailbox and File System archives, recreates any missing archive and archive folder information in the Directory database and vault store databases when the information is missing from them both, and EVSVR can obtain the required information from the target Exchange system or file system volume.

EVSVR can perform operations on CIFS, NTFS, and Centera partitions, partitions on streamer storage devices, and on both collected and uncollected items. Before you can perform an EVSVR operation, you must define it in an operation file.

See [“Creating an EVSVR operation file”](#) on page 92.

About the checkpointing facility in EVSVR

You can optionally enable *checkpointing* for any EVSVR operation. This facility causes EVSVR to create snapshots of the current state of an operation as it progresses. Then, if the operation is stopped or fails for any reason, you can continue it from the latest checkpoint instead of having to restart it from the beginning. You may find this facility useful if you need to run an operation on large volumes of data, which could otherwise be time-consuming to rerun from the start.

EVSVR checkpointing supports the following:

Operations	Some EVSVR operations combine multiple, single operations. A checkpointed operation continues from the operation that EVSVR was performing when it stopped.
Containers	EVSVR performs an operation over one Enterprise Vault site and one or more vault store groups, vault stores, and partitions (referred to as <i>containers</i>). A checkpointed operation continues from the container that EVSVR was processing when it stopped.
Steps	Some EVSVR operations process containers in multiple steps. A checkpointed operation continues from the step that EVSVR was performing when it stopped.
Phases and substeps	EVSVR performs some steps in an operation in multiple phases or substeps. A checkpointed operation continues from the phase or substep that EVSVR was performing when it stopped.

EVSVR does not checkpoint the position within scans of Enterprise Vault partitions and databases. So, for a single, one-step operation on a single container, you can enable checkpointing but it does not have any effect.

Note on performing EVSVR operations on CIFS and NTFS partitions

If you migrate archived data to secondary storage by using a migrator other than the Enterprise Vault migrator, you may find that running EVSVR leads to the temporary recall of large numbers of migrated CAB files. The recalled files occupy a large amount of partition space and can potentially cause a partition to become full. This issue does not arise if you use the Enterprise Vault migrator. Enterprise Vault deletes these temporary files according to how you set the **Recalled file cache period** property of the partition. This setting has a default value of seven days.

Before you run EVSVR, ensure that there is sufficient free space on the device on which the related Enterprise Vault partitions are located. To reduce the amount of time that Enterprise Vault retains the recalled files, you can lower the value of the **Recalled file cache period** property.

The collection process deletes the recalled files when the cache period has elapsed. You can trigger the collection process manually by using the **Run Now** option on the **Collections** tab of the partition properties.

Starting EVSVR

You must run EVSVR as the Vault Service account on an Enterprise Vault server. The server must be located in the Enterprise Vault site that contains the data that you want to process.

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

To start EVSVR

- 1 Log on to the Enterprise Vault server as the Vault Service account.
- 2 Do one of the following:
 - In Windows Explorer, navigate to the Enterprise Vault program folder (for example, `C:\Program Files (x86)\Enterprise Vault`) and double-click `evsvr.exe`.
 - Open a command prompt window and change to the Enterprise Vault program folder. Then type the following command:
`EVSVR`
EVSVR displays some startup information, which includes the following:
 - If the MAPI (Exchange) and Domino runtime components are not available, that this is the case. You must ensure that the appropriate runtime components are installed if you want to perform any EVSVR operation that requires the retrieval of savesets.
 - The name of the user account under which you are running EVSVR (that is, the Vault Service account).
 - The name of the Enterprise Vault site.
 - The version number of EVSVR.
- 3 Type a command at the `EVSVR>` prompt.

EVSVR commands

[Table 17-2](#) lists the commands that you can type at the `EVSVR>` prompt.

Table 17-2 EVSVR commands

Command	Effect
<code>continue</code>	<p>Continues execution of the current operation file from the latest checkpoint, if it is available. This command only has an effect if you have enabled checkpointing for the operation.</p> <p>A <code>continue</code> command is equivalent to a <code>start</code> command if you start an operation with checkpointing enabled for the first time.</p>
<code>edit</code>	<p>Opens the EVSVR Operations dialog box so that you can edit the currently loaded operation file or create a new one.</p> <p>See “Creating an EVSVR operation file” on page 92.</p>
<code>load [file]</code>	<p>Loads an operation file. If you do not specify a file, EVSVR prompts you to select one. You must load an operation file before you can run it.</p> <p>If an operation file is already loaded, EVSVR unloads it and loads the one that you specify.</p>
<code>unload</code>	Unloads the current operation file without performing any other actions.
<code>start</code>	Starts the execution of the current operation file from the beginning. If you have enabled checkpointing for the operation, this command resets the checkpointing information and starts the operation from the beginning.
<code>stop</code>	Stops the execution of the current operation file. EVSVR completes any actions that it is performing before it stops, and it generates a report file for the performed actions.
<code>pause</code>	Pauses the execution of the current operation file.
<code>resume</code>	Resumes the execution of the current operation file.
<code>restart</code>	Stops the execution of the current operation file and then starts it again from the beginning.
<code>status</code>	<p>Displays the current status of EVSVR, including its application state.</p> <p>See “EVSVR application states” on page 91.</p>
<code>cls</code>	Clears the EVSVR window.
<code>exit</code> or <code>quit</code>	Quits EVSVR.

Table 17-2 EVSVR commands (*continued*)

Command	Effect
<code>interactive</code>	<p>Runs EVSVR in interactive mode. This mode lets you perform a number of specialized activities, including the following:</p> <ul style="list-style-type: none"> ■ Retrieving the saveset and associated SIS parts of a specified archived item. ■ Retrieving a specified SIS part. ■ Extracting multiple savesets from an EMC Centera data blob. ■ Listing the locations where Enterprise Vault has stored all the parts of a specified saveset. <p>See “Running EVSVR in interactive mode” on page 134.</p>
<code>help</code> or <code>?</code>	Displays on-screen Help about the EVSVR commands.

EVSVR application states

[Table 17-3](#) lists the application states in which EVSVR can run.

Table 17-3 EVSVR application states

State	Description
Active	EVSVR is executing an operation file.
DialogueRunning	EVSVR is displaying the EVSVR Operations dialog box.
NotReady	No operation file is loaded. This state is the initial state if you start EVSVR without an argument list.
Paused	EVSVR has paused while it is executing an operation file.
Ready	An operation file is loaded.

The application state determines which EVSVR commands you can enter. For example, the `stop` command is only valid when the EVSVR state is Active or Paused. If you enter a command that is invalid for the current state, EVSVR displays an error message to indicate this fact.

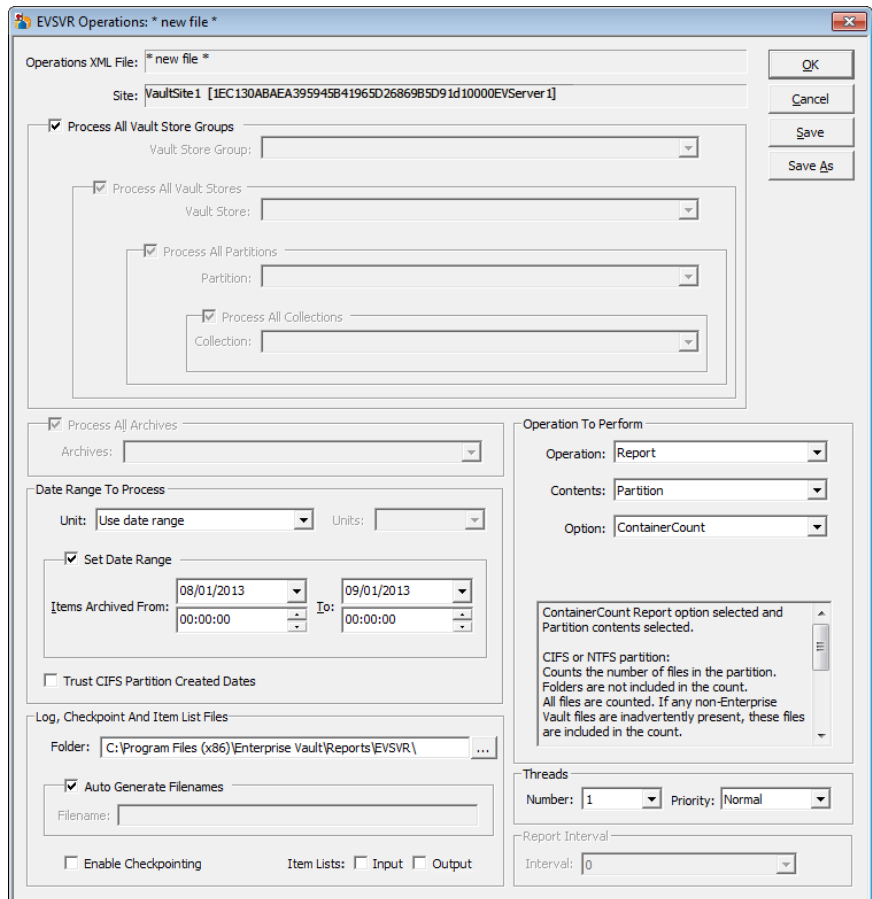
To determine the current state of EVSVR, type `status` at the `EVSVR>` prompt.

Creating an EVSVR operation file

You must create an operation file before you can perform an EVSVR operation. An operation file is an XML file that defines the actions that EVSVR is to perform, and on what data set.

You create an operation file by selecting the required options from the EVSVR Operations dialog box.

Figure 17-1 The EVSVR Operations dialog box



This dialog box lets you define the following:

- The storage data and Directory data to process. EVSVR processes the data that is associated with one of the following:

- All the partitions in all the vault stores in all the vault store groups in the Enterprise Vault site.
- All the partitions in all the vault stores in a single vault store group.
- All the partitions in a single vault store.
- A single partition.
- A specific archive to process. This applies only when EVSVR processes vault store databases or the archive information in the Directory database.
- The date range of archived items to process.
- The operation to perform.
- The location of the output log file. If you choose to enable the checkpointing or item list facilities, the name of the log file also determines the names of the checkpoint file and the folder in which EVSVR processes the item list files.
- The number of threads to use, and their priority level.

Note: Depending on the operation that you choose to perform, some of these options may not be available.

To create an operation file

- 1 At the `EVSVR>` prompt, type `edit` to open the EVSVR Operations dialog box.

Note the following:

- **Operations XML File** shows the name of the current operation file.
 - **Site** shows the name of the Enterprise Vault site for which to process the data. This is the site to which the Enterprise Vault server belongs. You cannot change the site.
- 2 Specify the storage data that you want to process. By default, the operation file specifies that EVSVR is to process the data for all partitions in all vault stores in all vault store groups in the Enterprise Vault site. However, you can minimize the amount of data that you process as follows:
 - To process a single vault store group, uncheck **Process All Vault Store Groups** and then select the required group.
 - To process a single vault store, uncheck **Process All Vault Stores** and then select the required vault store.
 - To process a single partition, uncheck **Process All Partitions** and then select the required partition.
 - 3 Select the required values for the other settings, as follows:

Process All Archives

By default, EVSVR processes all the archives in the selected storage data set. To select an individual archive, uncheck **Process All Archives** and then select an archive.

If there are a large number of archives, the dialog box displays a form so that you can filter by archive name.

Date Range To Process Do one of the following:

- Use the default setting, which does not impose a date range.
- Select a time unit in the **Unit** box, and then specify the number of units in the **Units** box. For example, if you select Hour and 2, EVSVR processes the items that were archived in the two hours before the time that you start the EVSVR operation.
- Select **Use date range** in the **Unit** box, and then check **Set Date Range** and specify a date range in the **Items Archived From** boxes.

When you set a date range, the option **Trust CIFS Partition Created Dates** becomes available. For operations that scan CIFS partitions, this option can increase the speed with which EVSVR scans the partitions. However, you must be confident that all the folders and files that you want to scan have accurate creation dates, because these dates play an important part in helping EVSVR to determine when certain, older items were archived.

- For each saveset (.dvs) file that Enterprise Vault 2007 or earlier has made, EVSVR uses the creation date to determine the date of the first archived item in the file. The last-modified date of the saveset file gives EVSVR the date of the last archived item that Enterprise Vault has added to the file as a sharer. The creation dates of saveset files may have changed if you have copied or moved them while restoring the partition from backup. On the other hand, if you trust the creation dates, and they fall outside the date range that you specify in EVSVR, then you can safely omit the files from the scan and so run it more quickly.
- For each saveset file that Enterprise Vault 8.0 or later has made, EVSVR establishes the archive date by looking at both the last-modified date of the file and the date in its folder path. These dates are preserved during backup and restore operations, so they provide a more robust way to determine each item's archive date.

Some EVSVR operations scan database records rather than the files in vault store partitions. For example, this is true of the ArchiveObjects Verify operation and DatabaseLinkages Verify operation. These operations ignore the **Trust CIFS Partition Created Dates** setting.

Whether you choose a date range depends on the severity of the issues that you want to address. If you want to repair a substantial number of items as part of a recovery procedure, it is important not to set a date range. This allows EVSVR to repair as many items as possible. On the other hand, setting a date range is desirable if you want to process a handful of items or a known range of items.

For example, suppose that a Repair operation has failed to repair a number of items. By repeating the operation against a date range that includes all the failed items, you may be able to identify the cause of the problem quickly. If you were to repeat the operation without specifying a date range, it could take days to complete.

For a non-critical operation, it is usually desirable to choose a small date range—especially if you select a data set with a large number of archived items. For example, this may be the case if you want to perform a daily Verify operation to validate the last week’s archived items only.

Operation To Perform Select an operation type (Report, Verify, or Repair), and then set the required options.
 See [“About the EVSVR operation settings”](#) on page 100.

Log, Checkpoint And
 Item List Files

Specify the following:

- The folder in which to save the output log file. By default, EVSVR saves the file in the `Reports\EVSVR` subfolder of the Enterprise Vault program folder. If the log file already exists, EVSVR appends the new information to it.

- The name of the log file. If you check **Auto Generate Filenames**, EVSVR uses the default file name, which is as follows:

`EVSVR_YYYYMMDDHHMMSS.Log`

Where `YYYYMMDDHHMMSS` specifies the date and time at which EVSVR created the log file.

- Whether to enable checkpointing. If you choose to do so, EVSVR stores the checkpoint information in an XML file that is in the same folder as the log file. The name of the checkpoint file matches that of the log file but includes the suffix `_Checkpoint`. For example, if you set the log file name to `EVSVR_Logfile.log`, the corresponding checkpoint file has the name

`EVSVR_Logfile_Checkpoint.xml`.

See [“About the checkpointing facility in EVSVR”](#) on page 87.

- Whether to process item list files. Some EVSVR operations let you output a list of items that have failed verification and need repairing. You can then input these item lists to a second EVSVR operation, which typically runs much faster than normal because it has less data to process. The name of the folder in which EVSVR outputs the item list files matches that of the log file but includes the suffix `_Items`. For example,

`EVSVR_Logfile_Items`.

See [“Using the output from one EVSVR operation as input for another operation”](#) on page 128.

Threads	<p>Specify the number of threads to use for the EVSVR operation. The maximum is 16.</p> <p>All the Verify and Repair operations can benefit from using multiple threads, but this is particularly the case with the DatabaseReferences Repair operation. Most Report operations always run with one thread only, even if you request more.</p> <p>Specify the thread priority as Normal, Low, or High.</p> <p>If you set the thread priority to High for the DatabaseReferences Repair operation, EVSVR automatically resets the priority level to Normal. This is designed to stop potential problems with resource scheduling and thread contention. Although intermittent, these problems can lead to errors when EVSVR tries to repair certain database references.</p>
---------	---

- 4 Click one of the following to save the specified values in an operation file:
 - **Save.** Saves the selected settings and their values in an operation file. If you previously saved the file, EVSVR overwrites the file. Otherwise, EVSVR prompts you for a file name.
 - **Save As.** Saves the selected parameters and their values in an operation file. EVSVR prompts you for a file name.
- 5 After you have defined the operation, click one of the following to exit from edit mode and return to the `EVSVR>` prompt:
 - **OK.** Exits and loads the last saved operation file. Any changes that you have made since your last save are lost.
 - **Cancel.** Exits without loading an operation file. Any changes that you have made since your last save are lost.

Editing an EVSVR operation file in which you have enabled checkpointing

When an operation for which you have enabled checkpointing is stopped, you can edit the settings in its operation file. However, if you change the selected operation type (Report, Verify, or Repair) or option, you cannot continue the execution of the operation file from the latest checkpoint; you must start the operation from the beginning. The EVSVR output log file reports this as follows:

```
Operation and Option mismatch:
Operation file: Operation: Repair, Option: Database Linkages
Checkpoint file: Operation: Repair, Option: Repair Fingerprint and Vault Store Database References
The operation will not be performed
```

However, you can change the other operation properties, such as the date range or number of threads, and then continue the operation from the latest checkpoint.

Running an EVSVR operation

After you have created an operation file, you can run it in any of the following ways:

- At the `EVSVR>` prompt, type `load` and then select the operation file that you want to load.
Type `start` to begin processing.
- In the EVSVR Operations dialog box, click **OK**. EVSVR closes the EVSVR Operations dialog box and loads the currently saved operation file, ready to run.
Type `start` to begin processing.
- At the MS-DOS command prompt, type the following command:

```
evsvr {-c|-r} operation_file_path
```

Where the parameters are as follows:

-c Loads the specified operation file, continues its execution from the latest checkpoint (if it is available), and then quits EVSVR. For example:

```
evsvr -c C:\op1.xml
```

-r Loads the specified operation file, starts its execution from the beginning, and then quits EVSVR. For example:

```
evsvr -r C:\op2.xml
```

operation_file_path Specifies the full path to the operation file. If the path contains spaces, enclose it in quotation marks. For example:

```
evsvr -r "C:\Operation Files\op3.xml"
```

You can add an EVSVR command to a batch file, if required.

Certain operations may take some time to complete, depending on factors such as the size of the data set, the date range, and the type of operation. You can use the `stop`, `pause`, `resume`, and `restart` commands to control a running operation, if required. Unless processing is interrupted, EVSVR continues processing until it has finished the operation.

About the EVSVR operation settings

You can select a Report, Verify, or Repair operation. All types of operation produce a log file that contains the results of the operation.

Report operations in EVSVR

The EVSVR Report operations provide a count or listing of the following:

- The items in vault store partitions.
- The records in vault store databases and fingerprint databases.
- Certain records in the Enterprise Vault Directory database.

The Option setting determines whether a report contains an item count or a list of items. It also determines the type of data that EVSVR counts or lists, if you select Partition as the Contents setting.

The Contents setting determines the type of data on which EVSVR reports. [Table 17-4](#) describes the settings from which you can select.

Table 17-4 Contents settings for Report operations

Contents setting	Action
Directory	Reports on the Archive records and Archive Folder records in the Directory database.
Directory and VaultStore	Reports on the Archive records and Archive Folder records in the Directory database, and the ArchivePoint records and Vault records in the vault store databases.
Fingerprint	Reports on fingerprint database records (SIS part records).

Table 17-4 Contents settings for Report operations (*continued*)

Contents setting	Action
Partition	<p>Reports on partition data (save sets and SIS parts, or Centera clips).</p> <p>Note the following:</p> <ul style="list-style-type: none"> ■ This operation is not supported on CIFS and NTFS partitions if you have enabled both the collection process and migration to secondary storage using a migrator other than the Enterprise Vault migrator. This is because non-Enterprise Vault migrators do not provide a way to scan the migrated data. ■ If you want to perform this operation on an EMC Centera partition, you must ensure that the Query capability is enabled for the Centera profile with which you connect to the Centera. EVSVR checks whether this capability is enabled and, if it is not, reports the fact in the EVSVR log file. <p>If the Query capability is disabled, use the Centera CLI or Centera Viewer to run the <code>Show Profile</code> command. This command lists the current capabilities of the Centera profile, which you can then enable or disable by running the <code>Update Profile</code> command.</p>
StorageQueue	<p>Reports on the files on the Enterprise Vault Storage Queue, if configured. In the properties of a vault store, you can choose whether to place safety copies of Exchange Server items on the Storage Queue rather than keeping them in their original locations.</p>
VaultStore	<p>Reports on vault store database records (save set information, ArchivePoint records and Vault records, and safety copies on the Enterprise Vault Storage Queue).</p>

EVSVR Directory report options

To obtain a report on the Archive records and ArchiveFolder records in the Directory database, select **Directory** as the **Contents** setting.

[Table 17-5](#) lists the available **Option** settings when you select **Directory** as the **Contents** setting.

Table 17-5 Option settings for Directory reports

Option setting	Action
ArchiveCount	For the selected vault store, counts the number of Archive records and ArchiveFolder records.
Archives	For the selected vault store, lists the Archive records and ArchiveFolder records.

EVSVR Directory and VaultStore report options

To obtain a report on the archive records in both the Directory database and the vault store databases, select **Directory and VaultStore** as the **Contents** setting.

[Table 17-6](#) lists the available **Option** settings when you select **Directory and VaultStore** as the **Contents** setting.

Table 17-6 Option settings for Directory and VaultStore reports

Option setting	Action
ArchiveCount	<p>For the selected vault store, counts the following:</p> <ul style="list-style-type: none"> ■ The number of Archive records and ArchiveFolder records in the Directory database. ■ The number of ArchivePoint records and Vault records in the vault store database. <p>These records catalog all the archives and archive folders in a vault store. They also provide information on the parent archive of each archive folder.</p>
Archives	<p>For the selected vault store, lists the following information:</p> <ul style="list-style-type: none"> ■ The Archive records and ArchiveFolder records in the Directory database. ■ The ArchivePoint records and Vault records in the vault store database. ■ The number of savesets in each archive and archive folder.

EVSVR Fingerprint report options

To obtain a report on fingerprint database records, select **Fingerprint** as the **Contents** setting.

[Table 17-7](#) lists the available **Option** settings when you select **Fingerprint** as the **Contents** setting.

Table 17-7 Option settings for Fingerprint reports

Option setting	Action
EVObjectCount	Counts the number of unreferenced, unshared, and shared SIS parts across all member tables.
EVObjects	Lists information for each SIS part record across all member tables, including the following: <ul style="list-style-type: none"> ■ SIS part ID ■ Archived date ■ Collection ID ■ Original size (bytes) ■ Stored size (bytes) ■ Reference count: The number of times that Enterprise Vault shares this SIS part ■ Converted content store size (bytes) ■ Converted content disposition (bytes) ■ Blacklisted reason, where applicable

EVSVR Partition report options

To obtain a report for the vault store partitions in the selected data silo, select **Partition** as the **Contents** setting.

[Table 17-8](#) lists the available **Option** settings when you select **Partition** as the **Contents** setting.

Table 17-8 Option settings for Partition reports

Option setting	CIFS or NTFS partition	Streamer partition	EMC Centera partition
ContainerCount	Counts the number of files in the partition. Folders are not included in the count. All files are counted. If any non-Enterprise Vault files are inadvertently present, these files are included in the count.	Counts the number of content streams, including those that applications other than Enterprise Vault have created.	Counts the number of clips, including those that applications other than Enterprise Vault have created.

Table 17-8 Option settings for Partition reports (*continued*)

Option setting	CIFS or NTFS partition	Streamer partition	EMC Centera partition
EVContainerCount	<p>Counts the number of Enterprise Vault files in the partition.</p> <p>If any non-Enterprise Vault files are inadvertently present, these files are excluded from the count.</p> <p>The count includes files with the following extensions:</p> <p>.ARCHCAB, .ARCHDVF, .ARCHDVFCC, .ARCHDVFSP, .ARCHDVS, .ARCHDVSCC, .ARCHDVSSP, .CAB, .DVF, .DVFCC, .DVFSP, .DVS, .DVSCC, .DVSSP</p>	<p>Counts the number of content streams that Enterprise Vault has created.</p>	<p>Counts the number of clips that Enterprise Vault has created.</p>
EVObjectCount	<p>Counts the number of Enterprise Vault savesets and SIS parts. These files have the following extensions:</p> <p>.CAB, .DVF, .DVFCC, .DVFSP, .DVS, .DVSCC, .DVSSP</p>	<p>Counts the number of content streams that Enterprise Vault has created for the current partition.</p>	
Containers	<p>Lists the full path of every file in a partition. Folders are not listed.</p> <p>All files are listed, including the files within .CAB files and the savesets within saveset files.</p> <p>If any non-Enterprise Vault files are inadvertently present, these files are included.</p>	<p>Lists information about the content streams that all applications have created.</p> <p>The report provides additional information on the content streams that Enterprise Vault has created.</p>	<p>Lists information about the clips that all applications have created.</p> <p>The report provides additional information on the clips that Enterprise Vault has created.</p>

Table 17-8 Option settings for Partition reports (*continued*)

Option setting	CIFS or NTFS partition	Streamer partition	EMC Centera partition
EVContainers	<p>Lists the full path of each Enterprise Vault file in the partition. Folders are not listed.</p> <p>The files within .CAB files and the savesets within saveset files are included.</p> <p>If any non-Enterprise Vault files are inadvertently present, these files are not included.</p> <p>The list includes files with the following extensions:</p> <p>.ARCHCAB, .ARCHDVF, .ARCHDVFCC, .ARCHDVFSP, .ARCHDVS, .ARCHDVSCC, .ARCHDVSSP, .CAB, .DVF, .DVFCC, .DVFSP, .DVS, .DVSCC, .DVSSP</p>	<p>Lists information about the content streams that Enterprise Vault has created.</p>	<p>Lists information about the clips that Enterprise Vault has created.</p>
EVOBJECTS	<p>Lists the full path of Enterprise Vault savesets and SIS parts. These files have the following extensions:</p>		
EVBVaultStoreObjects	<p>Lists the full path of Enterprise Vault savesets and SIS parts. These files have the following extensions:</p> <p>.CAB, .DVF, .DVFCC, .DVFSP, .DVS, .DVSCC, .DVSSP</p>	<p>Lists information about the content streams that Enterprise Vault has created for the current partition.</p>	<p>Lists information about the clips that Enterprise Vault has created.</p> <p>If the report covers more than one vault store, EVSVR lists the clips by vault store.</p> <p>For collection clips, the report includes information about the savesets in the clip.</p>

Note: The report provides a count or list of only those items that match the specified criteria. For example, a ContainerCount report on a CIFS vault store provides a count of the files that were archived within the specified date range, for each selected partition.

EVSVR StorageQueue report options

If you have configured a vault store to keep safety copies of Exchange Server items in a designated Storage Queue rather than in their original locations, you can obtain a report on the Storage Queue files by selecting **StorageQueue** as the **Contents** setting.

[Table 17-10](#) lists the available **Option** settings when you select **StorageQueue** as the **Contents** setting.

Table 17-9 Option settings for StorageQueue reports

Option setting	Action
QueuedItemsCount	Counts the Storage Queue (.EVSQ) files and large-file (.DVF) files, and the savesets in each Storage Queue file.
QueuedItems	Lists the Storage Queue (.EVSQ) files and large-file (.DVF) files, and the savesets in each Storage Queue file.

EVSVR VaultStore report options

To obtain a report on vault store database records, select **VaultStore** as the **Contents** setting.

[Table 17-10](#) lists the available **Option** settings when you select **VaultStore** as the **Contents** setting.

Table 17-10 Option settings for VaultStore reports

Option setting	Action
ArchiveCount	For each vault store, counts the number of ArchivePoint (Archive) records and Vault (ArchiveFolder) records in the vault store database.
Archives	Provides the following information for the ArchivePoint records and Vault records in the vault store database: <ul style="list-style-type: none"> ■ The Vault records that belong to each ArchivePoint. ■ The number of savesets that have been archived to each archive and archive folder. ■ The total number of ArchivePoint records and Vault records that have been allocated to each vault store.
EVObjectCount	Counts the number of saveset records.

Table 17-10 Option settings for VaultStore reports (*continued*)

Option setting	Action
EVObjects	Lists the following information for each saveset record: <ul style="list-style-type: none"> ■ Saveset ID ■ Archive ID ■ Archive date ■ Item size (kilobytes)
QueuedItemsCount	Counts the number of Storage Queue-related records, listed below.
QueuedItems	For a vault store that keeps safety copies of Exchange Server items in a designated Storage Queue rather than in their original locations, lists the following records in the vault store database: <ul style="list-style-type: none"> ■ StorageQueueBatch ■ StorageQueueSaveset ■ StorageQueueSavesetReingestLog

Verify operations in EVSVR

The EVSVR Verify operations let you check the consistency of the information in your vault store partitions, vault store databases, fingerprint databases, and Directory database.

A Verify operation has multiple **Option** settings from which you can select. The setting determines what data EVSVR verifies.

[Table 17-11](#) lists the available **Option** settings.

Table 17-11 Option settings for Verify operations

Option setting	Action
ArchiveObjects	Verifies that the vault store database records and fingerprint database records point to savesets and SIS parts in a partition: <ul style="list-style-type: none"> ■ Verifies that each saveset record points to a valid saveset. ■ Verifies that each SIS part record points to a valid SIS part. <p>You must select the required level of verification for this option. See “Verification levels for an EVSVR ArchiveObjects Verify operation” on page 111.</p>
Archives	Performs an ArchivesDirectory Verify operation, followed by an ArchivesVaultStore Verify operation.

Table 17-11 Option settings for Verify operations (*continued*)

Option setting	Action
ArchivesDirectory	<p>Verifies that the vault store database records have corresponding records in the Directory database:</p> <ul style="list-style-type: none"> ■ Verifies that each ArchivePoint record in the vault store database has a corresponding Archive record in the Directory database. ■ Verifies that each Vault record in the vault store database has a corresponding ArchiveFolder record in the Directory database.
ArchivesVaultStore	<p>Verifies that the Directory database records have corresponding records in the vault store databases:</p> <ul style="list-style-type: none"> ■ Verifies that each Archive record in the Directory database has a corresponding ArchivePoint record in the vault store database. ■ Verifies that each ArchiveFolder record in the Directory database has a corresponding Vault record in the vault store database.
Complete	<p>Performs a DatabaseLinkages Verify operation, followed by an ArchiveObjects Verify operation.</p> <p>EVSVR performs the ArchiveObjects Verify operation at the most detailed level (SavesetValid).</p>

Table 17-11 Option settings for Verify operations (*continued*)

Option setting	Action
DatabaseLinkages	<p>Verifies the linkages between the vault store databases and fingerprint databases:</p> <ul style="list-style-type: none"> ■ Verifies that for each archived item record in a vault store database, a SIS part record exists in the fingerprint database. You can select by archive and date range. ■ Verifies that the reference count for each SIS part record in the fingerprint database matches the total number of references in the vault store databases. You can select by date range but not by archive. ■ For each collection record, verifies that the number of referenced files in a CAB file or savesets in a clip matches the combined total of the following: <ul style="list-style-type: none"> ■ The number of savesets in the collection as recorded in the vault store database. ■ The number of SIS parts in the collection as recorded in the fingerprint database. <p>You can select by date range but not by archive.</p> ■ Reports on the number of unreferenced, unshared, and shared SIS parts.

Table 17-11 Option settings for Verify operations (*continued*)

Option setting	Action
DatabaseReferences	<p>Verifies that the savesets and SIS parts in a partition are referenced by database records:</p> <ul style="list-style-type: none"> ■ Verifies that each saveset that is located in a partition is pointed to by a saveset record in a vault store database. If the saveset is collected, also verifies that the collection record is complete. ■ Verifies that each SIS part that is located in a partition is pointed to by a fingerprint database record. If the SIS part is collected, also verifies that the collection record is complete. <p>Note the following:</p> <ul style="list-style-type: none"> ■ This operation is not supported on CIFS and NTFS partitions if you have enabled both the collection process and migration to secondary storage using a migrator other than the Enterprise Vault migrator. This is because non-Enterprise Vault migrators do not provide a way to scan the migrated data. ■ If you want to perform this operation on an EMC Centera partition, you must ensure that the Query capability is enabled for the Centera profile with which you connect to the Centera. If the Query capability is disabled, use the Centera CLI or Centera Viewer to run the <code>Show Profile</code> command. This command lists the current capabilities of the Centera profile, which you can then enable or disable by running the <code>Update Profile</code> command. ■ In rare cases, this operation may report the wrong results when items are archived to more than one Centera partition in the same vault store.
DetectCABCollectionId Mismatch	<p>Verifies that the vault store databases contain CAB file collection records whose collection identities match the file names of the associated CAB files. For example, this operation verifies that, when a collection record has a collection identity of 1234, the name of the associated CAB file is <code>Collection1234.cab</code>.</p> <p>If you find any instances of mismatches between the collection identities and the CAB file names for certain partitions in a vault store database, run the DatabaseReferences Repair operation for those partitions.</p> <p>See "Repair operations in EVSVR" on page 114.</p>

Table 17-11 Option settings for Verify operations (*continued*)

Option setting	Action
ItemCounts	Checks the Vault and ArchivePoint records in the vault store database and reports on any that have an incorrect count of archived items.
QueuedItems	Verifies that the Storage Queue files in a Storage Queue location are referenced by vault store database records.
StorageQueue	Verifies that the files in a Storage Queue location do not have any obvious errors, such as a file size of 0 bytes.
UndatedCollections	<p>Determines the number of collection records in the vault store databases that do not have specified creation dates. In Enterprise Vault 8.0 and later, all new collection records automatically have a specified creation date. However, this is not the case for collection records that an earlier version of Enterprise Vault has created. When the creation date is missing from any collection record, the DatabaseLinkages Verify operation ignores any date range that you specify and processes all of these collection records.</p> <p>To assign a creation date to any collection record that does not have one, run the UndatedCollections Repair operation.</p>

Verification levels for an EVSVR ArchiveObjects Verify operation

If you select the **ArchiveObjects** option for a Verify operation, you must also select a **Level** setting. This setting determines the level of verification that EVSVR performs.

The following table lists the **Level** settings and their effects. The table lists the first three levels in order of the level of verification, with the lowest level of verification listed first. For example, if you select the ObjectExtractsFromContainer level, the verification also includes the ObjectContainerExists and ObjectInContainer levels. As the level of verification increases, so does the time to perform the verification.

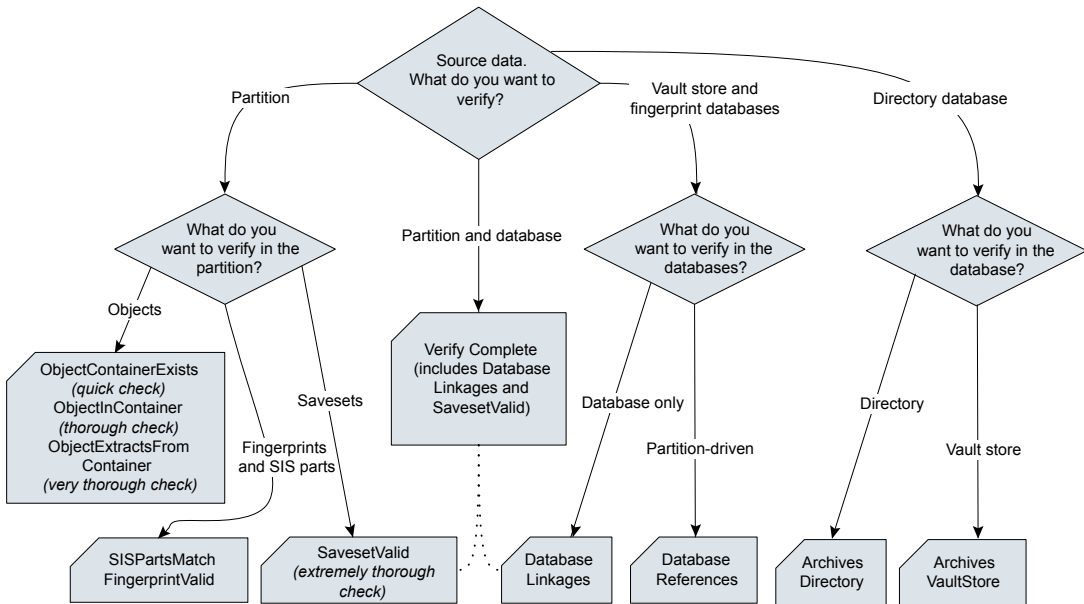
Table 17-12 Effects of the Level settings on an ArchiveObjects Verify operation

Level setting	CIFS partition without collections	CIFS partition with collections	Streamer partition	EMC Centera partition
ObjectContainerExists	Verifies that the saveset or SIS part file exists and has no obvious errors, such as a file size of 0 bytes.	Verifies that the CAB file exists and has no obvious errors, such as a file size of 0 bytes.	Verifies that the content streams containing savesets and SIS parts exist.	Verifies that the clip containing the saveset exists.
ObjectInContainer	Checks for converted content file, if appropriate.	Verifies that the CAB file contains the saveset or SIS part file, as defined by the CAB index.		
ObjectExtracts FromContainer		Verifies that the saveset or SIS part file can be extracted from the CAB file.		
SISPartsMatch	Verifies that the SIS part reference in the vault store database and the SIS part fingerprint in the fingerprint database match the SIS part map in a saveset file.		Not applicable.	
FingerprintValid	For each SIS part, recomputes the fingerprint and verifies that it matches the value in the fingerprint database. Decompresses compressed SIS parts and converted content files, where applicable.			
SavesetValid	Retrieves the saveset including all its SIS parts into an Enterprise Vault saveset, and perform a full verification.		Retrieves the saveset including all of its separately stored attachments and streams into an Enterprise Vault saveset, and performs a full verification.	

Choosing a suitable EVSVR Verify operation

Use [Figure 17-2](#) to help you choose a suitable operation when you want to verify your partitions and databases.

Figure 17-2 How to choose a suitable Verify operation



Example: Using EVSVR to verify the savesets in a vault store database

One common operation that you may want to perform with EVSVR is to verify the savesets in a vault store database.

To use EVSVR to verify the savesets in a vault store database

- 1 At the `EVSVR>` prompt, type `edit` to open the EVSVR Operations dialog box.
- 2 Choose the vault store group, vault store, or partition that you want to process. In most cases, you may want to process all the vault stores.
- 3 In the **Operation To Perform** list, select **Verify**.
 In the **Option** list, select **Complete**.
- 4 In the **Date Range To Process** box, specify the archived date of the items that you want to process. Alternatively, leave the date range blank to process all the items.
- 5 In the **Threads** box, keep the default thread number of 1.
- 6 Click **Save** to save the settings in an operation file.

- 7 Click **OK** to close the EVSVR Operations dialog box and load the new operation file.
- 8 At the `EVSVR>` prompt, type `start` to begin processing.
- 9 When EVSVR has finished processing, check the contents of the output log file.

Repair operations in EVSVR

If EVSVR reports any errors when you perform a Verify operation, you can correct them by performing a Repair operation. The function of the Repair operations is to recreate missing records in the vault store and fingerprint databases. In rare instances, a Repair operation creates new SIS parts on disk for items that have been shared many times.

It is very important to note the following points before you perform any Repair operation with EVSVR:

- Many of the Repair operations that are described below can cause data loss in some circumstances. Only the ArchivesVaultStore, BlacklistBadSISParts, and UndatedCollections operations cannot cause data loss.
 See [“Risk of data loss when you run certain EVSVR Repair operations”](#) on page 121.
 We strongly recommend that you contact Veritas Technical Support before you run any operation that can cause data loss.
- Only consider running a Repair operation if you encounter errors when you run a Verify operation.
- Before you run a Repair operation, make a backup copy of your databases and place the vault stores that you want to repair in backup mode. This is the case even if you have stopped the associated Storage service.

Caution: Starting the Storage service on a damaged system can damage it further. Do not start the Storage service before you have put the problematic vault stores in backup mode. Even then, only start the Storage service if it needs to be running.

A Repair operation has several Option settings from which you can select. [Table 17-13](#) describes the available settings.

Table 17-13 Option settings for Repair operations

Option setting	Action
Archives	<p>Combines the functions of multiple Repair operations: QueuedItems, ArchivesDirectory, and DatabaseReferences. In outline, the Archives operation does the following:</p> <ul style="list-style-type: none"> ■ Scans the files in a Storage Queue location and recreates or corrects the expected records in a vault store database. ■ Recreates any missing records in the fingerprint databases and vault store databases. ■ Recreates any missing Archive and ArchiveFolder records in the Directory database to make it consistent with the vault store databases. <p>This operation may be unable to recreate records if it cannot obtain the required information from the Directory database, vault store databases, savesets, target Exchange system (for Exchange Mailbox archives), or target file system volumes (for File System archives).</p> <p>Before you can run this operation, you must select the type of archive that you want to repair: Exchange Mailbox or File System. If the operation finds any items in the archive that do not match the selected type, it reports an error and stops processing.</p>
ArchivesDirectory	<p>Recreates any missing Archive and ArchiveFolder records in the Directory database to make it consistent with the vault store databases. To do this, the ArchivesDirectory operation does the following:</p> <ul style="list-style-type: none"> ■ Verifies that each ArchivePoint record in the vault store databases has a corresponding Archive record in the Directory database. If an Archive record is missing, the operation recreates it. ■ Verifies that each Vault record in the vault store databases has a corresponding ArchiveFolder record in the Directory database. If an ArchiveFolder record is missing, the operation recreates it. <p>Before you can run this operation, you must select the type of archive that you want to repair: Exchange Mailbox or File System. If the operation finds any items in the archive that do not match the selected type, it reports an error and stops processing.</p>

Table 17-13 Option settings for Repair operations (*continued*)

Option setting	Action
ArchivesVaultStore	<p>Recreates any missing ArchivePoint and Vault records in the vault store databases to make them consistent with the Directory database. To do this, the ArchivesVaultStore operation does the following:</p> <ul style="list-style-type: none"> ■ Verifies that each Archive record in the Directory database has a corresponding ArchivePoint record in the vault store databases. If an ArchivePoint record is missing, the operation recreates it. ■ Verifies that each ArchiveFolder record in the Directory database has a corresponding Vault record in the vault store databases. If a Vault record is missing, the operation recreates it. <p>You can also recreate missing ArchivePoint and Vault records in the vault store databases by running a DatabaseReferences Repair operation. However, after you run the DatabaseReferences operation, there can still be missing ArchivePoint and Vault records for archives and archive folders that do not contain savesets. In these circumstances, you must perform an ArchivesVaultStore Repair operation to recreate any missing records. Alternatively, you can do the following:</p> <ol style="list-style-type: none"> 1 Run an ArchivesVaultStore Repair operation to recreate the missing ArchivePoint and Vault records. 2 Run a DatabaseReferences Repair operation to recreate the missing saveset records and update the recreated ArchivePoint and Vault records.
BlacklistBadSISParts	<p>Blacklists any SIS part that does not verify correctly because it does not exist, has the wrong size, or does not match the value in the fingerprint database. After you blacklist a SIS part, archiving a new item with the same SIS part causes Enterprise Vault to create a new SIS part file on disk.</p>

Table 17-13 Option settings for Repair operations (*continued*)

Option setting	Action
DatabaseLinkages	<p>Does the following:</p> <ul style="list-style-type: none"> ■ Verifies and corrects the reference counts of savesets and SIS parts in the collection records in the vault store databases. ■ Recreates any missing information on the SIS parts used by savesets in the vault store databases. ■ Verifies the number of references to SIS parts in the fingerprint databases against the number of references in all vault store databases in the vault store group, and corrects any that are wrong. ■ Reports on the number of unreferenced, unshared, and shared SIS parts, after the repair operation has completed.

Table 17-13 Option settings for Repair operations (*continued*)

Option setting	Action
DatabaseReferences	<p>Recreates any missing records in the fingerprint databases and vault store databases. This option also updates any records that are found to be incorrect from the viewpoint of the partition.</p> <p>Note the following:</p> <ul style="list-style-type: none"> ■ This operation is not supported on CIFS and NTFS partitions if you have enabled both the collection process and migration to secondary storage using a migrator other than the Enterprise Vault migrator. This is because non-Enterprise Vault migrators do not provide a way to scan the migrated data. ■ If you want to perform this operation on an EMC Centera partition, you must ensure that the Query capability is enabled for the Centera profile with which you connect to the Centera. If the Query capability is disabled, use the Centera CLI or Centera Viewer to run the <code>Show Profile</code> command. This command lists the current capabilities of the Centera profile, which you can then enable or disable by running the <code>Update Profile</code> command. <p>The following additional settings are available when you choose to run a DatabaseReferences operation:</p> <ul style="list-style-type: none"> ■ Check Collection Counts. When selected, EVSVR checks the counts of referenced and unreferenced items in each CAB file and EMC Centera clip. If the number of unreferenced items is equal to the total number of items minus the number of referenced items—so, "unreferenced count = total count – referenced count"—then EVSVR does not recreate the database records for the unreferenced items because it assumes that they have been deleted. However, if you do not select Check Collection Counts, EVSVR considers all the missing database records as suitable for recreation. ■ Require Index Entries. When selected, EVSVR recreates missing saveset records for which the corresponding index entries exist, but it does not recreate any records that do not have index entries. <p>After you have performed a DatabaseReferences Repair operation, check that it was successful by reviewing its log file and performing a DatabaseReferences Verify operation. When you are satisfied that EVSVR has made the expected repairs, perform a DatabaseLinkages Repair operation on the same dataset.</p>

Table 17-13 Option settings for Repair operations (*continued*)

Option setting	Action
	<p>The DatabaseReferences operation processes all SIS parts before it processes anything else. This can lead to the situation where the operation recreates unused SIS parts that it finds in CAB files. After the operation has completed, you can resolve this issue as follows:</p> <ol style="list-style-type: none"> 1 Check the DatabaseReferences Repair log file for any errors that the operation encountered. Use the severity of any issues as a guide to what to do next. For example, you may need to restore missing or corrupt files in the partition from backup copies and then rerun the DatabaseReferences Repair operation. 2 After you have completed step 1 and judged the repair to be successful, run the DatabaseLinkages Repair operation. 3 After you have completed step 2 and judged the repair to be successful, run the Complete Verify operation to confirm this. 4 After you have completed step 3, verify that your environment is consistent and EVSVR has repaired everything that can be repaired. As a last resort, run the DeleteSurplusReferences Repair operation to remove any irreparable items and unused SIS parts.

Table 17-13 Option settings for Repair operations (*continued*)

Option setting	Action
DeleteSurplusReferences	<p>As a last resort, deletes the vault store and fingerprint database records that are associated with missing and irretrievably lost items. When a missing item consists of multiple parts, this option also deletes from disk any remaining parts that are associated with the item.</p> <p>You can also use this operation to remove unused SIS parts, but you must only do so when your environment is consistent.</p> <p>When you start a DeleteSurplusReferences operation, it first performs an internal DatabaseLinkages Verify operation. The DeleteSurplusReferences operation only starts to process when the DatabaseLinkages Verify operation reports that the environment is consistent and error-free.</p> <p>Before you perform a DeleteSurplusReferences operation, we recommend that you use the DatabaseReferences Repair operation to recreate any missing database references and ensure that the environment is consistent</p> <p>Note the following:</p> <ul style="list-style-type: none"> ■ The DeleteSurplusReferences operation does not take any action unless it can conclusively determine that the items in question are missing. For example, suppose that you have migrated archived data to secondary storage by using a non-Enterprise Vault migrator, such as Veritas NetBackup. If the migrator returns generic errors such as E_FAIL or E_UNEXPECTED, EVSVR does not take any action other than to report the errors. ■ When the DeleteSurplusReferences operation finds a CAB file or EMC Centera clip, it assumes that all the items that should exist within the CAB file or clip do exist.
ItemCounts	Repairs any Vault and ArchivePoint records in the vault store databases that have an incorrect count of archived items.
QueuedItems	Scans the files in a Storage Queue location and recreates or corrects the expected records in a vault store database.
StorageQueue	Scans the vault store database records for queued items and verifies that the expected files exist in the Storage Queue location. If any files are missing, EVSVR deletes the vault store database records and requests that the original items are rearchived.

Table 17-13 Option settings for Repair operations (*continued*)

Option setting	Action
UndatedCollections	Assigns a creation date to any collection record in a vault store database that does not have one. In Enterprise Vault 8.0 and later, all new collection records automatically have a specified creation date. However, this is not the case for any collection records that an earlier version of Enterprise Vault has created. When the creation date for a collection record is missing, EVSVR assigns the creation date of the associated CAB file or Centera clip to it.

The DatabaseReferences and DeleteSurplusReferences Repair operations do not work with savesets and SIS parts that you have migrated to secondary storage. The reason for this is that each operation needs to determine the locations of the migrated files from the vault store and fingerprint databases. As the information in these databases may be incorrect, the operation cannot proceed effectively.

If you want to perform a Repair operation on migrated files, we recommend that you first return them to their original store location.

Risk of data loss when you run certain EVSVR Repair operations

In some circumstances, data loss can occur when you run any of the following Repair operations:

- Archives
- ArchivesDirectory
- DatabaseLinkages
- DatabaseReferences
- DeleteSurplusReferences

We strongly recommend that you contact Veritas Technical Support before you run any of these operations.

The circumstances in which you can experience data loss include the following:

- The DatabaseReferences or Archives operation has failed to recreate all the missing records in the fingerprint databases and vault store databases. This issue typically arises when some of the Archive records and ArchiveFolder records in the Directory database are missing.
- You have specified an inappropriate date range for the DatabaseReferences or Archives operation. This can lead to the situation where the operation does not process some savesets and SIS parts that it ought to repair.

- You have selected the option **Require Index Entries** for the DatabaseReferences or Archives operation. However, some savesets do not have index information because one or more of the following conditions applies:
 - You have chosen to defer indexing (that is, Enterprise Vault does not index files as they are archived).
 - You have a backlog of index operations.
 - An index rebuild is in progress.
- You have run a DatabaseLinkages operation after a DatabaseReferences or Archives operation that failed to repair all the repairable items.
- You have run a DeleteSurplusReferences operation after a DatabaseReferences or Archives operation that failed to repair all the repairable items.
- You have not run a DatabaseLinkages operation after a DatabaseReferences or Archives operation that successfully completed.

For these reasons, it is important to ensure that the DatabaseReferences or Archives operation has repaired all the repairable items before you proceed with any other Repair operation.

For example, you can experience data loss when you run a DeleteSurplusReferences operation after a DatabaseReferences operation. This situation can arise if the DatabaseReferences operation has failed to repair all the saveset records in the vault store databases for some reason. When you subsequently run the DeleteSurplusReferences operation, certain SIS parts may incorrectly appear to be unused because the associated saveset records that should reference them were not repaired. For these SIS parts, the DeleteSurplusReferences operation then deletes both the corresponding records in the fingerprint database and any uncollected SIS part files on disk. For this reason, we recommend that you only run a DeleteSurplusReferences operation when you know that your environment is consistent and any missing items are irrecoverable.

Choosing a suitable EVSVR Repair operation

[Table 17-14](#) identifies the repair procedure to follow when you encounter specific errors during a Verify operation. For instructions on how to carry out each procedure, see [EVSVR repair procedures](#).

Table 17-14 How to select the appropriate repair procedure

If you run this Verify operation	And the log file reports this	Use this repair procedure
ArchiveObjects > ObjectContainerExists	"Verify failed count".	Procedure 1
ArchiveObjects > ObjectInContainer		Procedure 1
ArchiveObjects > ObjectExtractsFromContainer		Procedure 1
ArchiveObjects > SISPartsMatch		Procedure 2
ArchiveObjects > FingerprintValid		Procedure 2
ArchiveObjects > SavesetValid		Procedure 2
DatabaseReferences	Both "Savesets Unreferenced" and one of the following: <ul style="list-style-type: none"> ■ "Converted Content files unreferenced". ■ "Large files unreferenced". ■ "SISPart files unreferenced". 	Procedure 2
	Only "SISPart files unreferenced" or "Converted Content files unreferenced" or "Large files unreferenced".	Procedure 2
	Only "Savesets Unreferenced".	Procedure 2
DetectCABCollectionIdMismatch	"CAB Collection records with a Collection Identity mismatch: <i>number</i> ".	Procedure 2
DatabaseLinkages	Any error.	Procedure 3
UndatedCollections	"Undated Collection records: <i>number</i> ".	Run the UndatedCollections Repair operation.
ArchivesDirectory	Missing records.	Run the ArchivesDirectory Repair operation.

Table 17-14 How to select the appropriate repair procedure (*continued*)

If you run this Verify operation	And the log file reports this	Use this repair procedure
ArchivesVaultStore	Missing records.	Run the ArchivesVaultStore Repair operation.

[Table 17-15](#) identifies the repair procedure to use when you know of inconsistencies in a vault store database, fingerprint database, or partition.

Table 17-15 Suitable repair procedures for known inconsistencies in databases or partitions

Vault store database	Fingerprint database	Partition	Use this repair procedure
Consistent	Consistent	SIS part files are missing	Procedure 1
Inconsistent	Inconsistent	Consistent	Procedure 2
Consistent	Inconsistent	Consistent	Procedure 2
Inconsistent	Consistent	Consistent	Procedure 2
Inconsistent	Inconsistent	Not applicable	Procedure 3

EVSVR repair procedures

Caution: If you perform any of the following procedures, do not take your Enterprise Vault system out of backup mode until you have verified that the procedure has resolved the issue. Otherwise, you may damage your system.

Procedure 1

- 1 Using the information in the Verify log file to guide you, try to recover each missing file and corrupt file.
- 2 Rerun the Verify operation that you previously ran until you have resolved all the errors.
- 3 If you cannot recover all the SIS parts, run the BlacklistBadSISParts Repair operation to blacklist the fingerprint database records for the missing files.

Note: This is unnecessary if you have previously run an ArchiveObjects Verify operation with a verification level of SavesetValid. This operation has already blacklisted the database records for the missing SIS parts.

Procedure 2

- 1 Place the vault store groups that you want to repair in backup mode.
 If none of the Enterprise Vault services is running then, to place a vault store group in backup mode, you must start the Admin and Directory services only.
- 2 On each Enterprise Vault server, stop all Enterprise Vault services and related processes. Take care to ensure that storage-related processes such as `StorageServer.exe` and `StorageFileWatch.exe` have stopped.
- 3 Restart the following Enterprise Vault services only:
 - Enterprise Vault Admin service.
 - Enterprise Vault Directory service.
 - Enterprise Vault Indexing service (and all Indexing services that are associated with the vault store groups that you want to repair).
 - Storage service (optional). If the Storage service is running, users can retrieve items from the vault stores that you are repairing but Enterprise Vault does not delete existing items or archive new ones. When a vault store that you want to repair is not in backup mode, EVSVR reports the fact and does not perform the repair operation. In some cases, the Storage service must be running to process migrated files. EVSVR checks for these cases and, if the Storage service is not running but needs to be, EVSVR reports the fact and does not perform the operation.
- 4 Run a DatabaseReferences Repair operation.

Caution: SIS parts can be shared between different partitions in a single vault store and between partitions in different vault stores. Depending on how you have configured sharing, it is possible that the recreation of savesets in one vault store partition is dependent on SIS parts that belong to a partition of another vault store. These SIS part records must be available before you can recreate the savesets. So, the situation can arise where EVSVR cannot recreate some saveset records in a vault store database because they are dependent on SIS part records that you have yet to recreate in the fingerprint database.

To avoid this issue, use the sharing level that you have set for the vault store groups as a guide to what to repair. When the sharing level is "Share within group", you must repair the entire vault store group instead of repairing the vault stores and partitions one at a time. When the sharing level is "Share within vault store", you must repair the entire vault store instead of repairing the partitions one at a time. When the sharing level is "No sharing", or the partitions contain pre-8.0 savesets only, you can repair the partitions individually.

An additional consideration is the database that you need to repair. When this database is the vault store database, all the partitions belonging to that vault store are affected and require repair. However, if you need to repair a fingerprint database then, regardless of the sharing level that you have chosen, the entire vault store group is affected and requires repair

If any of the following conditions applies, you may want to uncheck the EVSVR operation setting **Require Index Entries**:

- You use deferred indexing (FSA).
- You have a backlog of index operations outstanding on any archives.
- An index rebuild is running.

The **Require Index Entries** operation setting controls whether EVSVR repairs database records based on the existence of index entries for the items.

- 5 If the DatabaseReferences Repair operation reports that it cannot recreate saveset records because of missing information in the Directory database, and the archive type is Exchange Mailbox or File System, perform an Archives Repair operation. This operation may be able to recreate the missing saveset records by obtaining the required information from the target Exchange system or file system volume.
- 6 Run a Complete Verify operation with EVSVR and investigate any errors. Depending on the nature of the errors, you may want to contact Enterprise Vault Support before you proceed.

- 7 Cancel all archive pending items in mailboxes and revert them to their normal state.
- 8 When the databases are in an acceptable state, start the remaining Enterprise Vault services and take the system out of backup mode.

Procedure 3

- 1 Place the vault store groups that you want to repair in backup mode.
 If none of the Enterprise Vault services is running then, to place a vault store group in backup mode, you must start the Admin and Directory services only. Do not start the Storage service.
- 2 On each Enterprise Vault server, stop all Enterprise Vault services and related processes. Take care to ensure that storage-related processes such as `StorageServer.exe` and `StorageFileWatch.exe` have stopped.
- 3 Restart the following Enterprise Vault services only:
 - Enterprise Vault Admin service
 - Enterprise Vault Directory service
 - Enterprise Vault Indexing service (and all Indexing services that are associated with the vault store groups that you want to repair).
 - Storage service (only if needed). If you need to start this service, the "initial database and partition checks" section of the EVSVR log file reports the fact.
- 4 Run a DatabaseLinkages Repair operation.
- 5 Run a Complete Verify operation, and investigate any errors.
 Depending on the nature of the errors, you may want to contact Enterprise Vault Support before you proceed.
- 6 If the vault store and fingerprint databases are still not consistent with each other or with the storage data in the affected partitions, run a DatabaseReferences Repair operation.
- 7 Cancel all archive pending items in mailboxes and revert them to their normal state.
- 8 When the databases are in an acceptable state, start the remaining Enterprise Vault services and take the system out of backup mode.

Using the output from one EVSVR operation as input for another operation

The Item Lists feature lets you use the output from one EVSVR operation as input for a second operation. There are two scenarios in which you may want to use EVSVR item lists:

- To use the output from a Verify operation as input for a Repair operation. For example, you can perform a Verify operation and output a list of items that have failed verification and need repairing. Then you can perform the appropriate Repair operation to process only the items in the item list.
- To use the output from a Verify operation as input for another Verify operation. For example, suppose that you perform a Verify operation and output a list of items that have failed verification and need repairing. If you cannot repair these items using EVSVR and must repair them by other means, such as restoring missing or corrupted files from backup, then you may want to rerun the Verify operation using the item list after you have done so. You can repeat the process with increasingly smaller input item lists until you have repaired all the items.

In both scenarios, the second operation is significantly faster than normal because it has less data to process. This is especially important in the case of Repair operations, which require you to place the target vault stores in backup mode and so block any modifications to them temporarily.

About EVSVR item list files

Each item list file is in XML format and provides information on the following:

- The date and time that the file was created.
- The EVSVR operation, option, and level that created the file.
- The Enterprise Vault site, vault store group, vault store, and partition on which EVSVR has performed the operation.
- Information on the items that the operation has found: IDs of savesets and SIS parts; file paths of savesets, SIS parts, and collection files; Centera Clip-Ids; streamer object identifiers; and EntryIds.

EVSVR creates separate item list files for each stage of an operation (option, container, step, and phrase or substep). The names of the item list files reflect the stages at which EVSVR created them. For example:

```
ArchiveObjects_16B887BC487590B48B73ACF1736E972801q10000EVSystem_Step-2_001
```

EVSVR stores the item list files in a folder whose name matches that of the corresponding log file. For example, if the name of the log file is

Using the output from one EVSVR operation as input for another operation

EVSVR_20140401113345.Log, EVSVR creates the item list files in a folder that is called EVSVR_20140401113345_Items.

As EVSVR controls the names and locations of the item list files, you do not need to select specific files when you use an output item list as the source for a follow-up operation.

EVSVR operations that support item list processing

Only certain combinations of Verify and Repair operations can process item lists; Report operations cannot output them or input them. In the **EVSVR Operations** dialog box, the **Input** and **Output** check boxes are disabled for operations and options that do not support item lists. Some operations can output item lists but cannot receive them as input, whereas for other operations the reverse is true; and some other operations support both input and output item lists.

When EVSVR opens an input item list file, it checks that a compatible operation created the file. For example, EVSVR does not let you use the output from a Verify ArchiveObjects operation as input for a Repair DatabaseReferences operation, as the two operations process different types of data. [Table 17-16](#) shows the combinations of operations that can process item lists.

Table 17-16 Compatible EVSVR operations for item list processing

Output operation	Input operation
Verify ArchiveObjects	Verify ArchiveObjects
Verify Archives	Either of the following: <ul style="list-style-type: none"> ■ Repair ArchivesDirectory ■ Repair ArchivesVaultStore
Verify ArchivesDirectory	Repair ArchivesDirectory
Verify ArchivesVaultStore	Repair ArchivesVaultStore
Verify Complete	Either of the following: <ul style="list-style-type: none"> ■ Repair DatabaseLinkages ■ Verify ArchiveObjects SavesetValid
Verify DatabaseLinkages	Repair DatabaseLinkages
Verify DatabaseReferences	Repair DatabaseReferences
Verify QueuedItems	Repair QueuedItems

Table 17-16 Compatible EVSVR operations for item list processing (*continued*)

Output operation	Input operation
Both of the following: <ul style="list-style-type: none"> ■ Verify ArchivesDirectory ■ Verify DatabaseReferences 	Repair Archives

All the verification levels for a Verify ArchiveObjects operation can output item lists. However, not all the verification levels are compatible. [Table 17-17](#) shows the verification levels from which each ArchiveObjects level can receive item lists as input.

Table 17-17 Compatible Verify ArchiveObjects levels

Level setting	Can input item lists that these levels have output
ObjectContainerExists	<ul style="list-style-type: none"> ■ ObjectContainerExists ■ ObjectExtractsFromContainer ■ ObjectInContainer
ObjectExtractsFromContainer	
ObjectInContainer	
FingerprintValid	<ul style="list-style-type: none"> ■ FingerprintValid
SavesetValid	<ul style="list-style-type: none"> ■ SavesetValid
SISPartsMatch	<ul style="list-style-type: none"> ■ SISPartsMatch

Viewing the EVSVR output log file

When EVSVR has finished processing, you can view the contents of the log file with a text editor. Alternatively, you send the log file to your Enterprise Vault Support representative.

The log file groups the information by vault store group, vault store, and partition. If EVSVR cannot find a vault store group, vault store, or partition, it reports this fact. This situation can occur if you have deleted a vault store group, vault store, or partition since you created the operation file.

[Figure 17-3](#) shows the start and end of an example log file.


```

opened existing Checkpoint file 'C:\EVSVRTest\EVSVR_20130604152430_Checkpoint.xml'
File:
  created: 2013-06-04 15:25:09
  opened: 2013-06-04 15:26:01
  updated: 2013-06-04 15:25:22
  closed: 2013-06-04 15:25:22

Continuing the operation from the last checkpoint
Control Step:
Number: 1, Name: Remove invalid collection records from the vault Store database, completed: 2013-06-04 15:25:22
Container:
Site:          VaultSite01
               16C25EAB12F4D514EACF7F56ED25C1AA61d10000xxxxsrv1.domain.local
Vault Store Group: Default Upgrade Group
               1476FAB3235BE544AA48448508A7A53721013300xxxxsrv1.domain.local
Vault Store:   VS0002
               13A1EEC7B4D67164D8031287156CFA7571210000xxxxsrv1.domain.local
Partition:    <All>
  
```

When EVSVR completes a checkpointed operation, the log file reports the fact. For example:

```
The Checkpointed operation has completed
```

If you issue a `continue` command for a checkpointed operation that has already completed, the log file provides the following information:

```

The checkpointed operation completed at 2013-06-04 15:26:02
The operation can be performed again by issuing a 'start' command
The operation will not be performed
  
```

About the item list information in the EVSVR log file

If you run an EVSVR operation that processes item list files, the log file provides the following additional information:

- The item list folder that the operation has created or opened.
- The output item list files that the operation has created.
- The output item list files that the operation has closed, and the number of items that it has added to the list.
- The input item list files that the operation has opened.
- The input item list files that the operation has closed, and the number of items that it has read and selected from the list.
- A count of the items that the operation cannot find. This situation can occur if the items are deleted between your adding them to an output item list file and using the file as input for a follow-up operation.

Additional log file information when you run certain EVSVR Repair operations

When you run a Repair operation to recreate any missing references in the vault store databases or fingerprint databases, a summary at the end of the log file provides a count of any references that EVSVR was unable to recreate

If you have tried to recreate the saveset references in the vault store databases, the log file provides the following additional information:

Saveset records not recreated	This is the sum of the five counts below.
No Directory Entry	A saveset reference was not recreated because, for the archive in which the saveset was originally archived, the Archive and Archive Folder records did not exist in the Enterprise Vault directory
No Index Entry	A saveset reference was not recreated because you selected the Require Index Entries option, and no index entry was found.
Missing SIS Parts	A saveset reference was not recreated because the required SIS part information was not available.
No open CIFS partition	A saveset reference was not recreated because it was necessary to duplicate a SIS part, and there was no open partition in which to do so. Open a partition in the vault store and repeat the Repair operation.
Errors	A saveset reference was not recreated because of other directory or database errors.

If you have tried to recreate the SIS part references in the fingerprint databases, the log file provides the following additional information:

SIS Part records not recreated	This is the sum of the four counts below.
Saveset SIS Part not available	The information that EVSVR needed to obtain from the vault store databases to recreate a SIS part was not available.
Error getting SIS Part information	Errors occurred when recreating the values required to recreate a SIS part reference.
Error creating Collection record	Database errors occurred when recreating a collection record in a vault store database for a SIS part file that exists in a .CAB collection file.

Error creating SIS Part record

Database errors occurred when recreating a SIS part reference in the fingerprint database.

For more information on specific savesets and SIS parts, and the errors that may have occurred, see the log file.

Running EVSVR in interactive mode

As well as performing EVSVR activities by creating and running an operation, you can perform a number of activities in interactive mode.

[Table 17-18](#) describes the commands that you can enter in interactive mode.

Table 17-18 Interactive mode commands

Command	Effect
DumpSaveset or DS	Retrieves the saveset and associated SIS parts of the specified archived item.
DumpSISPart or DP	Retrieves the specified SIS part.
ExtractSavesets or ES	Extracts multiple savesets from an EMC Centera data blob file or Enterprise Vault Storage Queue (.EVSQ) file.
GetNativeItem or GNI	Retrieves the native original item from a saveset file.
ListSavesetLocations or LS	Lists the locations where Enterprise Vault has stored all the parts of the specified saveset.
CLS	Clears the EVSVR window.
Help or ?	Displays on-screen Help about the EVSVR commands.
Exit or Quit	Quits EVSVR interactive mode.
? [<i>command_name</i>]	Displays detailed on-screen Help about the specified command.

To put EVSVR in interactive mode, start the utility and then type `interactive` at the `EVSVR>` prompt.

The following sections describe the syntax of the commands in detail.

DumpSaveset command

The DumpSaveset command retrieves the saveset and associated SIS parts of the specified archived item.

Syntax

```
DumpSaveset EntryId SavesetId [-o OutputFolder]
```

Where the parameters are as follows:

<i>EntryId</i>	Identifies the required vault store entry, archive entry, or archive folder entry. EVSVR uses this to determine the location of the saveset.
<i>SavesetId</i>	Specifies the required saveset ID or transaction ID.
<i>OutputFolder</i>	Specifies the path to the folder in which to store the retrieved files and log file. By default, this is the Reports\EVSVR subfolder of the Enterprise Vault program folder (for example C:\Program Files (x86)\Enterprise Vault).

Example

In the following example, the two parameters specify the vault store entry ID and saveset transaction ID of the required saveset:

```
ds 1995C3ACBB9472646AB0F3A0FDC7066B91210000testsrv1.domain.local
713C88D67D80E8046FFF279AE27D46B1
```

This command does not specify an output folder with the `-o` parameter, so DumpSaveset outputs the files to the default location, for example C:\Program Files (x86)\Enterprise Vault\Reports\EVSVR. All DumpSaveset files are output to a time-stamped folder under this output folder, such as EVSVR_DumpSaveset_20100714181917. So, in this example, the full output path is as follows:

```
C:\Program Files (x86)\Enterprise
Vault\Reports\EVSVR\EVSVR_DumpSaveset_20100714181917
```

Expected outputs

Except where noted, DumpSaveset outputs all the files and folders that are described below.

Table 17-19 Files and folders that are directly under the full output path

Output	Description
Log	<p>This is the log file. In the example above, its file name is <code>EVSVR_DumpSaveset_20100714181917.Log</code>.</p> <p>Always review the log file to determine how successful the operation was. The file shows any errors that occurred.</p> <p>See “Note on reviewing the messages in the EVSVR log files” on page 145.</p>
VSDbRecords.xml	This XML file contains the vault store database records for the saveset that DumpSaveset has retrieved.
Recombined folder	This folder contains the files that Enterprise Vault reconstructs after the entire saveset has been retrieved. See Table 17-20 .
Parts folder	This folder contains the files that comprise the retrieved saveset. See Table 17-21 .

Table 17-20 Contents of the Recombined folder

Output	Description
DVS	<p>This file contains all the data related to the retrieved saveset, except where this is a large-file saveset. DumpSaveset outputs large-file data to the <code>Parts</code> folder in the form of a DVF or DVFSP file, and it also outputs the data as a native item (see below). Sample file name:</p> <p><code>DS_201007078497509~201007071011490000~Z~611F6F215A2134E015849E23A4D6D601.DVS</code></p>
DocFile	<p>This file is an uncompressed structured storage version of the above recombined DVS file. You can inspect its contents with a structured storage viewer. Sample file name:</p> <p><code>DS_201007078497509~201007071011490000~Z~611F6F215A2134E015849E23A4D6D601.DocFile</code></p>
Native item	This is the original item that Enterprise Vault retrieved: a Domino message (DVNS) file, Exchange message (MSG) file, IMAP message (EML) file, or the original large file.

Table 17-21 Contents of the Parts folder

Output	Description
DVS/ARCHDVS (if CAB collected or migrated)	This file is either the entire saveset or, where sharing has been enabled, one part of a multipart saveset. Sample file name: DS_713C88D67D80E8046FFF279AE27D46B1.DVS
DocFile	This file is an uncompressed structured storage version of the above DVS file. You can inspect its contents with a structured storage viewer. Sample file name: DS_713C88D67D80E8046FFF279AE27D46B1.DocFile
DVSSP/ARCHDVSSP (if CAB collected or migrated)	Only output for multipart savesets where sharing has been enabled. The files are not generated for savesets that are stored on a Centera device. Sample file name: DS_713C88D67D80E8046FFF279AE27D46B1~2B~34D8CA20~00~1.DVSSP
DVSCC/ARCHDVSCC (if CAB collected or migrated)	Only output for multipart savesets where sharing has been enabled and converted content has been generated. The files are not generated for savesets that are stored on a Centera device. Sample file name: DS_713C88D67D80E8046FFF279AE27D46B1~2B~34D8CA20~00~1.DVSCC
DVFSP/ARCHDVFSP (if migrated)	Only output for large-file multipart savesets where sharing has been enabled. The files are not generated for savesets that are stored on a Centera device. Sample file name: DS_9111FB9F5230E0D6AB99C2014DC51611~CE~6E068DCC~00~1.DVFSP
DVF/ARCHDVF (if migrated)	Only output for large-file savesets where sharing has not been enabled. The files can also be generated for savesets that are stored on a Centera device. Sample file name: DS_713C88D67D80E8046FFF279AE27D46B1.DVF
DVFCC/ARCHDVFCC (if migrated)	Only output for large-file savesets where sharing has not been enabled and where converted content has been generated. The files are not generated for savesets that are stored on a Centera device. Sample file name: DS_713C88D67D80E8046FFF279AE27D46B1.DVFCC

Table 17-21 Contents of the Parts folder (*continued*)

Output	Description
CAB/ARCHCAB (if migrated)	Only output when parts of the retrieved saveset are stored on CIFS partitions that have been configured for collection using CAB files. DumpSaveset outputs a CAB file for each collected part of the saveset. The name of the CAB file has the form <code>DS_VaultStoreIdentity_CABfileName</code> . For example: <code>DS_VS8_Collection100.CAB</code>
CDF.xml	Only output for savesets that are stored on Centera devices. The XML file uses the Clip-Id related to the retrieved saveset as its file name. For example: <code>DS_8058S6H8CJLGLedF3SPTVDEKITTG4156M190N G0Q98CDMO8MC3SPT.CDF.xml</code>
MetaData.xml	Only output for savesets that have parts that are stored on streamer devices. DumpSaveset outputs an XML file for each part of the saveset. Sample file name: <code>DS_9111FB9F5230E0D6AB99C2014DC51611~CE~ 6E068DCC~00~1.DVSSP.MetaData.xml</code>

DumpSISPart command

The DumpSISPart command retrieves the specified SIS part.

Syntax

```
DumpSISPart EntryId SisPartId [-o OutputFolder]
```

Where the parameters are as follows:

<i>EntryId</i>	Identifies the required vault store entry, archive entry, or archive folder entry. EVSVR uses this to determine the location of the SIS part.
<i>SisPartId</i>	Identifies the SIS part.
<i>OutputFolder</i>	Specifies the path to the folder in which to store the retrieved files and log file. By default, this is the <code>Reports\EVSVR</code> subfolder of the Enterprise Vault program folder (for example <code>C:\Program Files (x86)\Enterprise Vault</code>).

Example

In the following example, the two parameters specify the vault store entry ID and SIS part ID of the required SIS part:

```
dp 1995C3ACBB9472646AB0F3A0FDC7066B91210000testsrv1.domain.local
714003019523969A1D9431D0592CCE41~91~BAC3E35A~00~1
```

This command does not specify an output folder with the `-o` parameter, so `DumpSISPart` outputs the files to the default location, for example `C:\Program Files (x86)\Enterprise Vault\Reports\EVSVR`. All `DumpSISPart` files are output to a time-stamped folder under this output folder, such as `EVSVR_DumpSISPart_20100715114342`. So, in this example, the full output path is as follows:

```
C:\Program Files (x86)\Enterprise
Vault\Reports\EVSVR\EVSVR_DumpSISPart_20100715114342
```

Expected outputs

Except where noted, `DumpSISPart` outputs all the files that are described below.

Table 17-22 Files and folders that are directly under the full output path

Output	Description
Log	<p>This is the log file. In the example above, its file name is <code>EVSVR_DumpSISPart_20100715114342.Log</code>.</p> <p>Always review the log file to determine how successful the operation was. The file shows any errors that occurred.</p> <p>See “Note on reviewing the messages in the EVSVR log files” on page 145.</p>
xml	<p>The XML files contain the vault store database records for the vault stores that reference the SIS part. <code>DumpSISPart</code> generates one XML file for each vault store in the vault store group in which the SIS part resides. Only the XML files for vault stores that reference the SIS part contain information; the others contain an empty <code>EnterpriseVault</code> XML element. Sample file names:</p> <pre>VSDBRecords - VS0101.xml VSDBRecords - VS0102Collected.xml</pre>
DVSSP/ARCHDVSSP (if CAB collected or migrated)	<p>Only output for non-large-file SIS parts. Sample file name:</p> <pre>DP_714003019523969A1D9431D0592CCE41~91~ BAC3E35A~00~1.DVSSP</pre>

Table 17-22 Files and folders that are directly under the full output path
(continued)

Output	Description
DVSCC/ARCHDVSCC (if CAB collected or migrated)	<p>Only output for non-large-file SIS parts where converted content has been generated. Sample file name:</p> <pre>DP_714003019523969A1D9431D0592CCE41~91~BAC3E35A~00~1.DVSCC</pre>
DVFSP/ARCHDVFSP (if migrated)	<p>Only output for large-file SIS parts. Sample file name:</p> <pre>DP_714003019523969A1D9431D0592CCE41~91~BAC3E35A~00~1.DVFSP</pre>
DVFCC/ARCHDVFCC (if migrated)	<p>Only output for large-file SIS parts where converted content has been generated. Sample file name:</p> <pre>DP_714003019523969A1D9431D0592CCE41~91~BAC3E35A~00~1.DVFCC</pre>
decompressed	<p>If the SIS part or SIS part converted content file that DumpSISPart has generated was compressed, the command also generates a decompressed version. Sample file names:</p> <pre>DP_714003019523969A1D9431D0592CCE41~91~BAC3E35A~00~1.DVSSP.decompressed</pre> <pre>DP_714003019523969A1D9431D0592CCE41~91~BAC3E35A~00~1.DVSCC.decompressed</pre>
CAB/ARCHCAB (if migrated)	<p>Only output when the SIS parts are stored on CIFS partitions that have been configured for collection using CAB files. A CAB file is expected when the SIS part has been collected. The name of the CAB file has the form <code>DP_CABfileName</code>. For example:</p> <pre>DP_Collection1.CAB</pre>
MetaData.xml	<p>Only output for SIS parts that are stored on streamer devices. DumpSISPart generates one XML file for the SIS part and another for the SIS part converted content that is stored on the streamer device. Sample file names:</p> <pre>DP_9111FB9F5230E0D6AB99C2014DC51611~CE~6E068DCC~00~1.DVSSP.MetaData.xml</pre> <pre>DP_9111FB9F5230E0D6AB99C2014DC51611~CE~6E068DCC~00~1.DVSCC.MetaData.xml</pre>

ExtractSavesets command

The ExtractSavesets command extracts one or more savesets from the following types of files:

- EMC Centera data blob files
- Enterprise Vault Storage Queue (.EVSQ) files

These types of files are called *appended savesets* files because they contain multiple savesets that are appended together.

Syntax

```
ExtractSavesets AppendedSavesetsFile [-o OutputFolder] [-n
FileNameTemplate] [-f Offset -s Size]
```

Where the parameters are as follows:

<i>AppendedSavesets File</i>	Specifies the full path to the file that contains the savesets.
<i>OutputFolder</i>	Specifies the path to the folder in which to store the retrieved files and log file. By default, this is the Reports\EVSVR subfolder of the Enterprise Vault program folder (for example C:\Program Files (x86)\Enterprise Vault).
<i>FileNameTemplate</i>	Specifies the file naming convention to use for the extracted savesets. If you do not specify a file name template, EVSVR applies the name of the input file to the savesets, but without the path or extension. If you do not specify an offset and size, EVSVR extracts all the savesets and sequentially names them <i>FileNameTemplate_nnn.DVS</i> . If you do specify the size and offset, EVSVR extracts the <i>size</i> bytes starting from <i>offset</i> into one saveset file that is named <i>FileNameTemplate.DVS</i> .
<i>Offset</i>	Specifies the offset in bytes from the start of the input file from which to start extracting the required saveset. If you specify an offset parameter, you must also specify a size parameter.
<i>Size</i>	Specifies the size in bytes of the data to extract from the input file. If you specify a size parameter, you must also specify an offset parameter.

Example

In the following example, the two parameters specify the path to an EMC Centera data blob file and the folder in which to extract its contents:

```
es "C:\Centera
Blobs\2RGPDMAIG8D51eAMOCBFS25BBK2G415357TU510G996D0BM2P8330.Blob126"
-o c:\MyOutputFolder
```

If the output folder does not exist, ExtractSaveSets creates it. All ExtractSaveSets files are output to a time-stamped folder under this output folder, such as EVSVR_ExtractSaveSets_20100715131545. So, in this example, the full output path is as follows:

```
C:\MyOutputFolder\EVSVR_ExtractSaveSets_20100715131545\
```

The sample command does not include a `-f` parameter to specify the offset or a `-s` parameter to specify the size, so it extracts all the saveSets in the blob file.

Expected outputs

Except where noted, ExtractSaveSets outputs all the files and folders that are described below.

Table 17-23 Files and folders that are directly under the full output path

Output	Description
Log	This is the log file. In the example above, its file name is EVSVR_ExtractSaveSets_20100715131545.Log. Always review the log file to determine how successful the operation was. The file shows any errors that occurred. See “Note on reviewing the messages in the EVSVR log files” on page 145.
Extracted SaveSets folder	This folder contains the files that ExtractSaveSets has extracted from the input file. See Table 17-24 .

Table 17-24 Contents of the Extracted SaveSets folder

Output	Description
DVS	Given the specified input parameters, if the appended saveSet file contains DVS data, the command extracts all the DVS files from it. The name of each DVS file has the form <i>AppendedSaveSetName_IndexNumber.DVS</i> . For example: 2RGPDMAIG8D51eAMOCBFS25BBK2G415357TU510G996D0BM2P8330_001.DVS

Table 17-24 Contents of the Extracted Savesets folder (*continued*)

Output	Description
DocFile	<p>This file is an uncompressed structured storage version of the above extracted DVS file. You can inspect its contents with a structured storage viewer. Sample file name:</p> <pre>2RGPDMAIG8D51eAMOCBFS25BBK2G415357TU510 G996D0BM2P833O_001.DocFile</pre>

GetNativeItem command

The `GetNativeItem` command opens the specified saveset file and retrieves the original native item. The command also saves the saveset as a `DocFile`, which is an uncompressed saveset file that you can read with a structured storage viewer.

The command cannot recombine SIS parts or retrieve separately stored large files. If the specified saveset file does not contain the native item, the command creates an empty file.

Syntax

```
GetNativeItem SavesetFile [-o OutputFolder]
```

Where the parameters are as follows:

- SavesetFile* Specifies the path to the required saveset file. If you omit the file name extension, `GetNativeItem` assumes that it is `.DVS`.
- OutputFolder* Specifies the path to the folder in which to store the native item and log file. By default, this is the `Reports\EVSVR` subfolder of the Enterprise Vault program folder (for example, `C:\Program Files (x86)\Enterprise Vault`).

Example

In the following example, the two parameters specify the path to the required saveset and the folder in which to output the results:

```
gni c:\MySavesets\713C88D67D80E8046FFF279AE27D46B1.DVS -o
c:\MyOutputFolder
```

Expected outputs

Table 17-25 Files that are directly under the full output path

Output	Description
Native item	This is the original item that Enterprise Vault retrieved: a Domino message (DVNS) file, Exchange message (MSG) file, or IMAP message (EML) file.
DocFile	This file is an uncompressed structured storage version of the DVS file. You can inspect its contents with a structured storage viewer. Sample file name: GNI_713C88D67D80E8046FFF279AE27D46B1.DocFile
Log	This is the log file. In the example above, its file name is EVSVR_GetNativeItem_20150127112935.Log. Always review the log file to determine how successful the operation was. The file shows any errors that occurred. See “Note on reviewing the messages in the EVSVR log files” on page 145.

ListSavesetLocations command

The ListSavesetLocations command lists the locations where Enterprise Vault has stored all the parts of the specified saveset.

Syntax

```
ListSavesetLocations EntryId SavesetId [-o OutputFolder]
```

Where the parameters are as follows:

<i>EntryId</i>	Identifies the required vault store entry, archive entry, or archive folder entry. EVSVR uses this to determine the location of the saveset.
<i>SavesetId</i>	Specifies the required saveset ID or transaction ID.
<i>OutputFolder</i>	Specifies the path to the folder in which to store the retrieved files and log file. By default, this is the Reports\EVSVR subfolder of the Enterprise Vault program folder (for example C:\Program Files (x86)\Enterprise Vault).

Example

In the following example, the parameters specify the vault store entry ID and saveset transaction ID of the required saveset, and the folder in which to output the results:

```
ls 1995C3ACBB9472646AB0F3A0FDC7066B91210000testsrv1.domain.local
713C88D67D80E8046FFF279AE27D46B1 -o c:\MyOutputFolder
```

If the output folder does not exist, ListSavesetLocations creates it. All ListSavesetLocations files are output to a time-stamped folder under this output folder, such as EVSVR_ListSavesetLocations_20100715112935. So, in this example, the full output path is as follows:

```
C:\MyOutputFolder\EVSVR_ListSavesetLocations_20100715112935
```

Expected outputs

Table 17-26 Files that are directly under the full output path

Output	Description
Log	<p>This is the log file. In the example above, its file name is EVSVR_ListSavesetLocations_20100715112935.Log.</p> <p>Always review the log file to determine how successful the operation was. The file shows any errors that occurred.</p> <p>See “Note on reviewing the messages in the EVSVR log files” on page 145.</p>

Note on reviewing the messages in the EVSVR log files

When an interactive mode operation has finished, it displays a message to indicate whether it was successful. If the operation failed for any reason, check the log file for more information.

Note that the underlying Enterprise Vault components may record a message in the event log in certain cases when errors are encountered, but the operation may still be considered a success. The event log messages that the Enterprise Vault code generates when EVSVR calls it are redirected to the log file, and they do not appear in the event log. So, it is important to review the log file to determine if any errors occurred. For example, the file may contain an event log-related message like the following, even though the overall status of the operation was "Completed operation with success":

```
2010-07-14 19:13:00 Event Output: Failed to recall a Saveset from
its Collection.
```

Reason: Failed to extract the file from the CAB file. The file name is not in the CAB file index.

Improving EVSVR performance when processing CAB collections

When the following EVSVR operations process CAB collection files, they can cause high CPU usage and take a long time to complete:

- DatabaseLinkages Verify
- DatabaseLinkages Repair
- DatabaseReferences Repair

If you experience this problem, you can markedly improve performance by creating an index for each fingerprint database that you want to verify or repair. Then, after you have run the EVSVR operation, you can either remove the index or leave it in place for when you next run the operation

Note: Creating an index for a fingerprint database can marginally reduce archiving performance and increase the size of the database. However, you may consider these to be acceptable drawbacks if you run EVSVR regularly.

To improve EVSVR performance when processing CAB collections

- 1 On the SQL Server computer, start SQL Server Management Studio.
- 2 In the left pane of the SQL Server Management Studio window, expand the tree until the required fingerprint database is visible.
- 3 Click the fingerprint database, and then click **New Query**.
- 4 Do one of the following:
 - To create an index, enter the following query and then click **Execute**:

```
DECLARE @RC int
DECLARE @Create bit
DECLARE @ByteRangeStart tinyint
DECLARE @ByteRangeEnd tinyint
DECLARE @debug bit
SET @Create = 1
SET @ByteRangeStart = 0
SET @ByteRangeEnd = 255
SET @debug = 0 /* Set to 1 to view debug information */
```

```
EXECUTE @RC = [dbo].[Factory_EVSVR_Index_01]
@Create, @ByteRangeStart, @ByteRangeEnd, @debug
```

- To remove an existing index, enter the following query and then click **Execute**:

```
DECLARE @RC int
DECLARE @Create bit
DECLARE @ByteRangeStart tinyint
DECLARE @ByteRangeEnd tinyint
DECLARE @debug bit
SET @Create = 0
SET @ByteRangeStart = 0
SET @ByteRangeEnd = 255
SET @debug = 0 /* Set to 1 to view debug information */
EXECUTE @RC = [dbo].[Factory_EVSVR_Index_01]
@Create, @ByteRangeStart, @ByteRangeEnd, @debug
```

FSARunNow

This chapter includes the following topics:

- [About FSARunNow](#)
- [Running FSARunNow](#)
- [FSARunNow syntax](#)
- [FSARunNow examples](#)

About FSARunNow

FSARunNow is a utility with which you can initiate File System Archiving tasks on demand, using the command-line interface. It provides more options than the **Run Now** facility in the Administration Console.

You can use the FSARunNow utility to do any of the following:

- Initiate archiving. You can specify a File System Archiving task. Alternatively you can archive from a specific file server or file server volume by quoting the appropriate Entry Id from the Directory database.
- Initiate the synchronization of file server archive permissions with folder permissions.
- Initiate the pruning of earlier versions of archived files until only the required number of versions remains. The File System Archiving task performs pruning according to the version and age-based pruning settings on the **Pruning** tab of the task's properties.
- For files archived from EMC Celerra/VNX devices, initiate the deletion of archived files whose placeholders have been deleted.

Running FSARunNow

Note that you can create a batch file that contains the required FSARunNow commands and use Windows Task Scheduler to run the file when required.

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

To run FSARunNow

- 1 Log on to any Enterprise Vault server using the Vault Service account.

Caution: You must log on to the Enterprise Vault server locally. You cannot run FSARunNow if you log on remotely.

- 2 Open a command prompt window.
- 3 Navigate to the Enterprise Vault program folder (for example `C:\Program Files (x86)\Enterprise Vault`).
- 4 Run FSARunNow with the required options.
See [“FSARunNow syntax”](#) on page 149.

FSARunNow syntax

Type the command in one of the following forms:

- To initiate archiving for a specified file system archiving task, file server, or file server volume:

```
FSARunNow Archive TaskName | TaskEntryId | FileServerEntryId  
[VolumeEntryId] [Report | Normal] [ShortcutsOnly]
```

- To initiate the synchronization of file server archive permissions with folder permissions:

```
FSARunNow Synchronize TaskName | TaskEntryId | FileServerEntryId
```

- To initiate the pruning of earlier versions of archived files:

```
FSARunNow Prune TaskName | TaskEntryId | FileServerEntryId  
[Report | Normal]
```

- To initiate the deletion of files, archived from EMC Celerra/VNX devices, whose placeholders have been deleted:

```
FSARunNow CelerraDelOnDel TaskName | TaskEntryId |
FileServerEntryId [Report | Normal]
```

where the parameters are as follows:

TaskName	<p>Specifies the name of the task you want to process. You can determine the TaskName as follows:</p> <ol style="list-style-type: none"> 1 In the left pane of the Administration Console, expand Enterprise Vault Servers. 2 Expand the name of the computer that runs the task you want to process. 3 Click Tasks. <p>The right pane shows the tasks on that computer.</p> <p>Note: If the task name contains spaces, enclose it in quotation marks.</p>
TaskEntryId	<p>Specifies the TaskEntryId of the task you want to process. You can determine the TaskEntryId as follows:</p> <ol style="list-style-type: none"> 1 Start SQL Server Management Studio. 2 In the tree view at the left, select Databases > EnterpriseVaultDirectory. 3 In the toolbar, click New Query. 4 In the Query window, type the following: <pre style="margin-left: 20px;">select * from task</pre> 5 Press F5 to execute the query. 6 Scan the query results for the TaskEntryId of the task to process. <p>To specify a TaskEntryId, you must use the ID for the task that has the appropriate suffix. Tasknames in the query results include the following suffixes:</p> <ul style="list-style-type: none"> ■ For Archive – <i>TaskName</i> ■ For Synchronize – <i>TaskName_Synchronization</i> ■ For Prune – <i>TaskName_Pruning</i> ■ For CelerraDelOnDel – <i>TaskName_CelerraDelOnDel</i> <p>For instance, to specify the TaskEntryId for a Prune run where the task name is <i>FSA_Task1</i>, use the TaskEntryId corresponding to <i>FSA_Task1_Pruning</i>.</p>

FileServerEntryId

Specifies the FileServerEntryId of the computer whose archives you want to process. You can determine the FileServerEntryId as follows:

- 1 Start SQL Server Management Studio.
- 2 In the tree view at the left, select **Databases > EnterpriseVaultDirectory**.
- 3 In the toolbar, click **New Query**.
- 4 In the **Query** window, type the following:

```
select * from fileserverentry
```
- 5 Press F5 to execute the query.
- 6 Scan the query results for the FileServerEntryId of the computer to process.

VolumeEntryId

Specifies the VolumeEntryId of the computer whose archives you want to process. You can determine the VolumeEntryId as follows:

- 1 Start SQL Server Management Studio.
- 2 In the tree view at the left, select **Databases > EnterpriseVaultDirectory**.
- 3 In the toolbar, click **New Query**.
- 4 In the **Query** window, type the following:

```
select * from fileservervolumeentry
```
- 5 Press F5 to execute the query.
- 6 Scan the query results for the VolumeEntryId of the computer to process.

Report	<p>Runs the File System Archiving task or tasks in report mode. Each task generates a report that outlines the changes that the task would make if it ran in normal mode, but no changes are made.</p> <p>Note that by default all FSARunNow options run in Report mode except the Synchronize option, which does not use this parameter.</p> <p>File System Archiving tasks generate reports in the following folders:</p> <ul style="list-style-type: none"> ■ Archiving reports: The <code>Reports\FSA</code> subfolder of the Enterprise Vault installation folder. ■ Pruning reports and Celerra/VNX deletion of archived files reports: The <code>Reports</code> subfolder of the Enterprise Vault installation folder. <p>For more information about these reports, see "About File System Archiving Task reports" in <i>Setting up File System Archiving</i>.</p>
Normal	<p>Runs the File System Archiving task or tasks in normal mode. Each task performs the requested actions, and also generates a report that outlines the changes it made.</p>
ShortcutsOnly	<p>Restricts the archiving task so that it only creates shortcuts. If you use this option, the task does not perform archiving.</p>

FSARunNow examples

The following are examples of how to run FSARunNow.

- To perform an archive run in Report mode:

```
FSARunNow Archive "File System Archiving Task1"
```
- To perform a synchronizing run for a specified file server:

```
FSARunNow Synchronize  

1D6D9206BFDBFB846B2E0F8135A1989331d100002example.server.local
```
- To perform a pruning run for a specified file server in Report mode:

```
FSARunNow prune  

1AD6297BC643DCC40A924CAB74D0BCDCE141000server.example.net
```
- To run a File System Archiving task to delete archived files on a Celerra/VNX whose placeholders have been deleted:

```
FSARunNow CelerraDelOnDel FSATask1 normal
```

FSAUndelete

This chapter includes the following topics:

- [About FSAUndelete](#)
- [Running FSAUndelete](#)
- [FSAUndelete syntax](#)
- [FSAUndelete examples](#)

About FSAUndelete

FSAUndelete cancels the permanent deletion of the archived items for specified placeholders, or for all of the placeholders in a specified folder of a file server.

FSAUndelete is typically for use when all of the following conditions apply:

- You have set the option **Enable recovery of user deleted items** on the **Archive Settings** tab of the Site Properties in the Administration Console. This option provides a "soft delete" mechanism. When a user deletes an item, Enterprise Vault retains the archived item for a specified number of days, before it permanently deletes the archived item.
- You use placeholders, and you use an archiving policy with the setting "Delete archived file when placeholder is deleted".
- You restore placeholders to a file server, for example from a backup.

In this scenario, some of the archived files that are associated with the restored placeholders may be due for permanent deletion. A restored placeholder works only until Enterprise Vault permanently deletes the archived file. By using FSAUndelete you can cancel the permanent deletion of the archived files, without the need to restore all of the files in the archive.

FSAUndelete provides options to do the following:

- Undelete an archived file that corresponds to a specified placeholder.
- Undelete the archived files that correspond to all of the placeholders in a specified folder. You can optionally choose to include all of the subfolders of the specified folder.

FSAUndelete generates a report on the command line that lists the files that it has undeleted, and any failures such as orphaned placeholders or specified placeholders that it did not find.

Running FSAUndelete

Run FSAUndelete when you want to cancel the permanent deletion of archived files that are associated with file server placeholders.

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

To run FSAUndelete

- 1 Identify the computer on which to run FSAUndelete:
 - For a Windows file server you can run FSAUndelete on either of the following computers:
 - On the Enterprise Vault server that runs the File System Archiving task for the file server volume. The volume must be configured as a file server target volume in the Administration Console.
 - On the file server that contains the placeholders that you want to process.
 - For a non-Windows file server, run FSAUndelete on the Enterprise Vault server that runs the File System Archiving task for the file server volume. The volume must be configured as a file server target volume in the Administration Console.
- 2 Log on as required to the Enterprise Vault server or the Windows file server, either with the Vault Service account, or an account that meets the following requirements:
 - If a Windows file server is the target for the undelete operation: An account that is a member of the Enterprise Vault Placeholder Application role. The account must also be a member of both the Print Operators group and the Distributed COM Users group on the Windows file server.

- If a NetApp file server is the target for the undelete operation: An account that is a member of the Enterprise Vault Placeholder Application role: The account must also be a member of the Administrators group on the NetApp file server.
- If an EMC Celerra device is the target for the undelete operation: An account that is a member of both the Enterprise Vault Placeholder Application role and the Enterprise Vault File Server Administrator role. The account must also be a member of the Administrators group on the Celerra device.

You can assign roles using Enterprise Vault's RBA PowerShell cmdlets.

See "Roles-based administration" in the *Administrator's Guide*.

- 3 Open a command prompt window and change to the Enterprise Vault installation folder, for example `C:\Program Files (x86)\Enterprise Vault`.
- 4 Run the `FSAUndelete` command with the required options.

See "[FSAUndelete syntax](#)" on page 155.

FSAUndelete syntax

Use one of the following options with `FSAUndelete`. Include a path in quotes if it contains spaces.

- To undelete a file that is associated with a single placeholder:

```
FSAUndelete placeholder_path
```

Where *placeholder_path* is the local path or the UNC path of the placeholder. `FSAUndelete` does not support wildcard characters.

You can use this option in a script, for example to undelete the archived file for each placeholder in a log of restored backup files.

- To undelete the files that are associated with all the placeholders in a specified folder, but not in any subfolders:

```
FSAUndelete folder_path
```

Where *folder_path* is the local path or the UNC path of the folder.

- To undelete the files that are associated with all the placeholders in a specified folder, and recursively in all subfolders:

```
FSAUndelete folder_path -r
```

Where *folder_path* is the local path or the UNC path of the folder.

FSAUndelete examples

The following examples assume that you run FSAUndelete from the Enterprise Vault server:

- Undelete the archived file for the placeholder with the UNC path
\\myserver\myfiles\file1:
`FSAUndelete \\myserver\myfiles\file1`
- Undelete the archived files for the placeholders in the folder with the UNC path
\\myserver\myfiles\, but do not process any subfolders:
`FSAUndelete \\myserver\myfiles\`
- Undelete the archived files for all the placeholders in the folder with the UNC
path \\myserver\myfiles\, and in any subfolders:
`FSAUndelete \\myserver\myfiles\ -r`

The following examples assume that you run FSAUndelete on a Windows file server for which you want to process the placeholders. You can therefore specify local paths to placeholders and folders.

- Undelete the archived file for the placeholder C:\myfiles\file 9:
`FSAUndelete "C:\myfiles\file 9"`
- Undelete the archived file for all of the placeholders in the folder C:\myfiles\
and its subfolders:
`FSAUndelete C:\myfiles\ -r`

FSAUtility

This chapter includes the following topics:

- [About FSAUtility](#)
- [Running FSAUtility](#)
- [About using FSAUtility with EMC Celerra/VNX placeholders](#)
- [FSAUtility options](#)

About FSAUtility

FSAUtility is a command-line utility with which you can do the following:

- Recreate archive points on the original path.
- Recreate the placeholders for archived files in their original location.
- Move placeholders from one location to another location, and move the corresponding archived files.
- Migrate placeholders from a source path to a destination path without any movement of the archived data.
- Delete orphaned placeholders for which no corresponding item exists in the archive.
- Restore all archived files, or archived files of the specified file types, to their original location or a new location.
- Recall the archived files that correspond to placeholders that are present in a folder.

The utility works with archive points and placeholders on Windows file servers, NetApp Filers, and EMC Celerra/VNX devices.

For more information on migrating and consolidating file servers that have content that has been archived with Enterprise Vault, see the following article on the Veritas Support website:

<http://www.veritas.com/docs/TECH147519>

Running FSAUtility

Before you run FSAUtility, note the following:

- We recommend that you do not run more than one instance of FSAUtility at a time. Issues can arise when you specify the same source or target for multiple, concurrent instances of the utility.
- We recommend that before you run FSAUtility you stop any File System Archiving tasks that process the target file server. This action ensures that no manual or scheduled archiving occurs on the file server while FSAUtility is processing files, which ensures better performance and prevents inconsistent behavior. For example, if Enterprise Vault archives a volume while a file recall to that volume is in progress, Enterprise Vault may convert the recalled files to placeholders.
- You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.
See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.
- An EMC restriction prevents FSAUtility from processing files or folders on a Celerra/VNX device if the target path exceeds 1024 characters. On the Enterprise Vault server an event log message states that the input string was not in a correct format.
- FSAUtility has two methods for identifying Celerra/VNX placeholders. If you want to use FSAUtility with placeholders on Celerra/VNX volumes, make sure that you use the method that is appropriate for your Celerra/VNX configuration. See [“About using FSAUtility with EMC Celerra/VNX placeholders”](#) on page 159.

To run FSAUtility

- 1 Log on to any Enterprise Vault server with the Vault Service account. If you use the utility to process a Windows file server, the account must also have local administrator permissions on the file server.
- 2 Open a command prompt window.

- 3 Navigate to the Enterprise Vault program folder, for example `C:\Program Files (x86)\Enterprise Vault`.
- 4 Run FSAUtility with the required options.
 See “[FSAUtility options](#)” on page 163.

About using FSAUtility with EMC Celerra/VNX placeholders

Read this section if you want to use FSAUtility with placeholders on EMC Celerra/VNX volumes.

FSAUtility can identify a placeholder on an EMC Celerra/VNX device by using either a Windows API call or a Celerra/VNX HTTP API call. [Table 20-1](#) lists the API call that FSAUtility uses by default with each of its placeholder-related options.

Table 20-1 Default API call for detecting Celerra/VNX placeholders

FSAUtility option	FSAUtility parameter	Default API call for detecting Celerra/VNX placeholders
Bulk recall of files corresponding to placeholders	<code>FSAUtility -b</code>	Windows API call
Recreate placeholders	<code>FSAUtility -c</code>	Windows API call
Move placeholders and their corresponding files	<code>FSAUtility -m</code>	Windows API call
Delete orphaned placeholders	<code>FSAUtility -o</code>	Windows API call
Migrate placeholders	<code>FSAUtility -pm</code>	Celerra/VNX API call

Note that by default `FSAUtility` with the `-pm` parameter uses the Celerra/VNX API call. This default helps to ensure that placeholder migration always succeeds, regardless of the Celerra/VNX configuration. If you use `FSAUtility -pm` with any supported Celerra/VNX configuration, you may want to change the setting for this option to use the more performance-efficient Windows API call.

With the other placeholder-related parameters (`-b`, `-c`, `-m`, and `-o`), FSAUtility uses the efficient Windows API call by default.

You can configure the API call that FSAUtility uses for each placeholder-related option by editing the `FSAUtility.exe.config` file.

Configuring which API call FSAUtility uses to identify Celerra/VNX placeholders

The `FSAUtility.exe.config` file controls which API call FSAUtility uses to identify Celerra/VNX placeholders. For each placeholder-related FSAUtility option, the file contains an entry to specify which API call to use.

In the supplied `FSAUtility.exe.config` file these entries are all commented out, so FSAUtility uses its default API call for each option. That is, FSAUtility uses the Celerra/VNX API call for placeholder migration, and the Windows API call for the other placeholder-related options.

You can edit the `FSAUtility.exe.config` to set which API call FSAUtility uses for an option.

To configure which API call FSAUtility uses to identify Celerra/VNX placeholders

- 1 On the Enterprise Vault server on which you want to run FSAUtility, navigate to the Enterprise Vault installation folder, for example `C:\Program Files (x86)\Enterprise Vault`.
- 2 Open the `FSAUtility.exe.config` file with a text editor such as Notepad.
- 3 Find the section of the file for the FSAUtility option whose API call you want to set:
 - **<PHMigration>** for the `FSAUtility -pm` option.
 - **<MovePlaceholder>** for the `FSAUtility -m` option.
 - **<BulkRecall>** for the `FSAUtility -b` option.
 - **<RecreatePlaceholder>** for the `FSAUtility -c` option.
 - **<OrphanPlaceholder>** for the `FSAUtility -o` option.
- 4 Remove the comment characters from the start and end of the section.
- 5 Edit the value of the **CheckCelerraOfflineAttribute** key to the required setting:
 - A value of **0** sets the option to use the Windows API call.
 - A value of **1** sets the option to use the Celerra/VNX API call.

If the **CheckCelerraOfflineAttribute** key is omitted or commented out for any option, FSAUtility uses its default API call for that option.
- 6 Repeat steps 3 to 5 for each FSAUtility option for which you want to configure the API call.
- 7 Save the changes to `FSAUtility.exe.config` file.

Example FSAUtility.exe.config file settings

The following examples show a `FSAUtility.exe.config` file that has been edited to produce various results.

Example 1

The file sets the `PHMigration` option (for `FSAUtility -pm`) to use a Windows API call rather the default Celerra/VNX API call. No other values are defined, so FSAUtility uses the Windows API call for all of its placeholder-related options.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="PHMigration"
      type="System.Configuration.DictionarySectionHandler"/>
    <section name="BulkRecall"
      type="System.Configuration.DictionarySectionHandler"/>
    <section name="MovePlaceHolder"
      type="System.Configuration.DictionarySectionHandler"/>
    <section name="RecreatePlaceHolder"
      type="System.Configuration.DictionarySectionHandler"/>
    <section name="OrphanPlaceHolder"
      type="System.Configuration.DictionarySectionHandler"/>
  </configSections>
  <PHMigration>
    <add key="CheckCelerraOfflineAttribute" value = "0"/>
  </PHMigration>
  <MovePlaceHolder>
    <!--><add key="CheckCelerraOfflineAttribute" value = "0"/><!-->
  </MovePlaceHolder>
  <BulkRecall>
    <!--><add key="CheckCelerraOfflineAttribute" value = "0"/><!-->
  </BulkRecall>
  <RecreatePlaceHolder>
    <!--><add key="CheckCelerraOfflineAttribute" value = "0"/><!-->
  </RecreatePlaceHolder>
  <OrphanPlaceHolder>
    <!--><add key="CheckCelerraOfflineAttribute" value = "0"/><!-->
  </OrphanPlaceHolder>
  <runtime>
    <generatePublisherEvidence enabled="false"/>
  </runtime>
</configuration>
```

Example 2

This configuration produces the same result as [Example 1](#). Each placeholder-related option is set to use the Windows API call.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="PHMigration"
      type="System.Configuration.DictionarySectionHandler"/>
    <section name="BulkRecall"
      type="System.Configuration.DictionarySectionHandler"/>
    <section name="MovePlaceholder"
      type="System.Configuration.DictionarySectionHandler"/>
    <section name="RecreatePlaceholder"
      type="System.Configuration.DictionarySectionHandler"/>
    <section name="OrphanPlaceholder"
      type="System.Configuration.DictionarySectionHandler"/>
  </configSections>
  <PHMigration>
    <add key="CheckCelerraOfflineAttribute" value = "0"/>
  </PHMigration>
  <MovePlaceholder>
    <add key="CheckCelerraOfflineAttribute" value = "0"/>
  </MovePlaceholder>
  <BulkRecall>
    <add key="CheckCelerraOfflineAttribute" value = "0"/>
  </BulkRecall>
  <RecreatePlaceholder>
    <add key="CheckCelerraOfflineAttribute" value = "0"/>
  </RecreatePlaceholder>
  <OrphanPlaceholder>
    <add key="CheckCelerraOfflineAttribute" value = "0"/>
  </OrphanPlaceholder>
  <runtime>
    <generatePublisherEvidence enabled="false"/>
  </runtime>
</configuration>
```

Example 3

In this example, all of the placeholder-related options use the Celerra/VNX API call.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
```

```

<configSections>
  <section name="PHMigration"
    type="System.Configuration.DictionarySectionHandler"/>
  <section name="BulkRecall"
    type="System.Configuration.DictionarySectionHandler"/>
  <section name="MovePlaceHolder"
    type="System.Configuration.DictionarySectionHandler"/>
  <section name="RecreatePlaceHolder"
    type="System.Configuration.DictionarySectionHandler"/>
  <section name="OrphanPlaceHolder"
    type="System.Configuration.DictionarySectionHandler"/>
</configSections>
<PHMigration>
  <add key="CheckCelerraOfflineAttribute" value = "1"/>
</PHMigration>
<MovePlaceHolder>
  <add key="CheckCelerraOfflineAttribute" value = "1"/>
</MovePlaceHolder>
<BulkRecall>
  <add key="CheckCelerraOfflineAttribute" value = "1"/>
</BulkRecall>
<RecreatePlaceHolder>
  <add key="CheckCelerraOfflineAttribute" value = "1"/>
</RecreatePlaceHolder>
<OrphanPlaceHolder>
  <add key="CheckCelerraOfflineAttribute" value = "1"/>
</OrphanPlaceHolder>
<runtime>
  <generatePublisherEvidence enabled="false"/>
</runtime>
</configuration>

```

FSAUtility options

Table [Table 20-2](#) lists the actions you can perform with FSAUtility.

Table 20-2 FSAUtility options

Action	FSAUtility parameter	More information
Recreate archive points on the original path.	FSAUtility -a	See "Recreating archive points" on page 164.

Table 20-2 FSAUtility options (*continued*)

Action	FSAUtility parameter	More information
Recreate the placeholders for archived files in their original location.	FSAUtility -c	See “Recreating placeholders” on page 165.
Move placeholders and the corresponding archived files. The archive point that applies to the destination folder determines the destination archive.	FSAUtility -m	See “Moving placeholders and corresponding files” on page 167.
Migrate placeholders from a source path to a destination path without moving the archived data.	FSAUtility -pm	See “Migrating placeholders” on page 169.
Delete orphaned placeholders for which no corresponding item exists in the archive.	FSAUtility -o	See “Deleting orphaned placeholders” on page 174.
Restore all archived files, or archived files of the specified file types, to their original location or a new location.	FSAUtility -t	See “Restoring archived files” on page 175.
Recall the archived files that correspond to placeholders that are present in a folder.	FSAUtility -b	See “Recalling files corresponding to placeholders” on page 177.
Fix folder points.	FSAUtility -fp	Use this option only when directed by Veritas Support.

Recreating archive points

You can use FSAUtility with the `-a` parameter to recreate the archive points on the original path for a target volume.

Syntax

```
FSAUtility -a -s UNC_path [-l log_level] [-r]
```

Where:

- `-s UNC_path` specifies the UNC path to the target volume.

- `-l log_level` specifies whether to log both successful operations and failed operations (0) or failed operations only (1). By default, FSAUtility logs failed operations only.
- `-r` specifies report-only mode. FSAUtility generates a text report that outlines the activities that it would perform if you were to run it in normal mode, but without performing those activities. The report is named `EV_FILESYSTEM_UTILITY_REPORT_DateTime.txt`, and it is generated in the folder `installpath\Reports\FSAUtility`.
If you run the command in normal mode (without `-r`), FSAUtility generates an XML report of the actions it has taken, named `EV_FILESYSTEM_UTILITY_REPORT_DateTime.xml`.

When FSAUtility recreates an archive point, it examines the relevant records in the Directory database to determine which archive is associated with the folder path. If more than one archive is associated with the folder path, FSAUtility does as follows:

- It assigns the archive ID of the oldest non-empty archive to the archive point.
- It records in its XML report or in the report-only mode's text report, the archive IDs of the multiple archives that were found to be associated with the folder path.

Examples

The following command reports on the archive points that FSAUtility would recreate for the volume `\\myserver\users`:

```
FSAUtility -a -s \\myserver\users -r
```

The following command recreates the archive points for the volume `\\myserver\users`, recording both the successful operations and failed operations in the XML report:

```
FSAUtility -a -s \\myserver\users -l 0
```

Recreating placeholders

You can use FSAUtility with the `-c` parameter to recreate the placeholders for archived files in their original location. This facility may prove useful if you need to restore a file server to its original state or synchronize the file server with the Enterprise Vault archive. If multiple versions of the same file exist in the archive, the utility creates a placeholder for the latest version only.

Note: Before you use this option with Celerra/VNX placeholders, ensure that FSAUtility is configured to use a suitable method for identifying the placeholders. See [“About using FSAUtility with EMC Celerra/VNX placeholders”](#) on page 159.

Syntax

```
FSAUtility -c -s UNC_path [-D mm-dd-yyyy] [-f] [-l log_level] [-r]
```

Where:

- `-s UNC_path` specifies the path to the required folder, volume, or file server.
- `-D mm-dd-yyyy` specifies the date after which items must be archived before you can recreate the placeholders for them.
- `-f` forces FSAUtility to recreate the placeholders when placeholders or files of the same name already exist. The utility first deletes the existing placeholders or files and then creates the new ones.
- `-l log_level` specifies whether to log both successful operations and failed operations (0) or failed operations only (1). By default, FSAUtility logs failed operations only.
- `-r` specifies report mode. FSAUtility generates a report that outlines the activities that it would perform if you were to run it in normal mode, but without performing those activities.

FSAUtility generates a report named

`EV_FILESYSTEM_UTILITY_REPORT_DateTime.txt`, in the folder
`installpath\Reports\FSAUtility`.

If you run `-c` in normal mode, FSAUtility generates a report named

`EV_FILESYSTEM_UTILITY_REPORT_DateTime.xml`.

Examples

The following command recreates the placeholders for the folder `\\myserver\users` and generates a log file that lists both successful operations and failed operations. The command runs in report mode.

```
FSAUtility -c -s \\myserver\users -l 0 -r
```

The following command recreates the placeholders for those files that were archived after July 10 2005 from the folder `\\myserver\users\user1`. If any files or placeholders of the same name already exist, the command overwrites them with new placeholders.

```
FSAUtility -c -f -s \\myserver\users\user1 -D 07-10-2005 -l 0
```

Notes

- FSAUtility does not support "hard link" files (directory references to files). You cannot recreate any existing placeholders for hard link files. When you perform a recreate operation, FSAUtility recalls any placeholders that are hard link files.
- When you recreate placeholders with FSAUtility, you may receive the message "Internal Error Moving Placeholders: Archive ID null for folder folder_path" if you subsequently try to move them to another location. To stop this message from appearing, recreate the archive points and then archive the source folder before you try to move the placeholders.
- Due to a NetApp restriction, FSAUtility does not recreate placeholders if the path to the folder exceeds 512 characters.

Moving placeholders and corresponding files

You can use FSAUtility with the `-m` parameter to move placeholders and the corresponding archived files. The archive point that applies to the destination folder determines the destination archive. The destination archive can be in a different vault store.

The command moves placeholders in subfolders of the source folder, unless the subfolder has an archive point. The subfolders are created under the destination folder if they do not exist there.

If no archive point exists on the path to the destination folder, the command exits without proceeding.

If the archive point on the destination folder has no archive ID or an invalid archive ID, FSAUtility checks the Directory database records to determine whether the folder path is associated with any archive IDs:

- If no archive ID is associated with the folder path, FSAUtility creates an archive and assigns the archive ID to the archive point.
- If one archive ID is associated with the folder path, FSAUtility assigns that archive ID to the archive point.
- If more than one archive ID is associated with the folder path, FSAUtility does as follows:
 - It assigns the archive ID of the oldest existing archive to the archive point.
 - It generates a warning event with event ID 41484 in the Enterprise Vault event log. The event lists the archive IDs of the multiple archives for the folder path, and indicates that the oldest archive will be used for archiving.

Note that after FSAUtility has assigned an archive ID to the archive point, no further warnings are issued about the existence of multiple archives for the folder path.

If there is an archive point on the source folder and there are no archive points on any subfolders, then the source folder archive point is deleted if all the placeholders are moved successfully.

Note: Before you use this option with Celerra/VNX placeholders, ensure that FSAUtility is configured to use a suitable method for identifying the placeholders. See [“About using FSAUtility with EMC Celerra/VNX placeholders”](#) on page 159.

Syntax

```
FSAUtility -m -s UNC_path -d UNC_path [-l log_level] [-r]
```

Where:

- *-s UNC_path* specifies the path to the source folder.
- *-d UNC_path* specifies the path to the destination folder.
- *-l log_level* specifies whether to log both successful operations and failed operations (0) or failed operations only (1). By default, FSAUtility logs failed operations only.
- *-r* specifies report mode. FSAUtility generates a report that outlines the activities that it would perform if you were to run it in normal mode, but without performing those activities.

FSAUtility generates a report named

EV_FILESYSTEM_UTILITY_REPORT_DateTime.txt, in the folder
installpath\Reports\FSAUtility.

If you run *-m* in normal mode, FSAUtility generates a report named

EV_FILESYSTEM_UTILITY_REPORT_DateTime.xml.

Examples

The following command moves the placeholders from the first folder to the second folder. It also moves the archived files to the corresponding archive location. The log file lists failed operations only.

```
FSAUtility -m -s \\myserver\users\user1 -d \\sample\share\user1
```

Notes

- You cannot move placeholders from the root folder of a volume. You can move placeholders from the subfolders of the root folder.

- FSAUtility does not delete a source folder from which you have moved placeholders after it has completed the operation. The folder may contain other, unarchived files that it would be inappropriate to delete.
- If you halt an FSAUtility operation to move placeholders before it has finished, then when you next start the utility it prompts you to resume the operation.
- The volume policy or folder policy that applies to the destination location determines whether Enterprise Vault deletes archived files when their placeholders are deleted. See “Deleting archived files on placeholder deletion” in *Setting up File System Archiving*.
- If the source vault store or destination vault store is in backup mode when you try to move placeholders, the utility exits without proceeding.
- FSAUtility does not support “hard link” files (directory references to files). You cannot move any existing placeholders for hard link files. When you perform a move operation, FSAUtility moves any placeholders that are hard link files.
- If any of the following becomes unavailable while you move placeholders, FSAUtility does not try to process any outstanding placeholders:
 - Enterprise Vault Directory Service
 - Enterprise Vault File Placeholder Service
 - Enterprise Vault Storage Service
 - The network connection between Enterprise Vault and the file serverInstead, the utility exits after recording an error message in the event log, DTrace log, and FSAUtility log file.

See also

See [“Migrating placeholders”](#) on page 169.

Migrating placeholders

Note: Before you migrate placeholders, make sure that you have a backup of the Directory database, the vault store databases, and the folder hierarchy under the source path. Back up the folder hierarchy under the destination path also, if it contains archived files.

You can use FSAUtility with the `-pm` parameter to migrate placeholders and archive points from a source folder structure to a destination folder structure, for example on another volume or file server.

This option moves the placeholders and archive points, but it does not move any files in the archives. The migrated placeholders retain their links to the archived files in their original locations. This option therefore provides a faster solution for moving placeholders than the FSAUtility move (-m) option. Use the -m option if you want to move archives, consolidate vault stores, or align archives with file servers.

This option always migrates placeholders in subfolders recursively, provided that they reside under a valid archive point. The option creates the destination subfolders, if necessary.

Before performing a placeholder migration, FSAUtility checks for any conflicts between the archive points at the source location and the destination location. It then performs the following actions, in the order listed:

- Moves the placeholders. FSAUtility creates the placeholders on the destination location and then deletes the placeholders at the source location. The migration retains the placeholder file's security descriptor, which contains information about the ownership and NTFS permissions for the file. The migration also retains any alternate data streams associated with the placeholder file.
- Moves the archive points to the destination location.
- Updates the Directory database with the new folder paths.

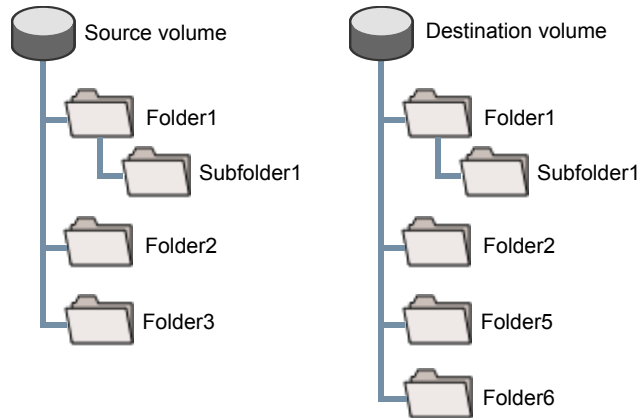
Note the following requirements for placeholder migration:

- The source volume and destination volume must both be specified as FSA targets in the Administration Console.
- The same Enterprise Vault server must manage the source volume's vault store and the destination volume's vault store. If FSAUtility cannot confirm that the same Storage service computer manages both vault stores, it quits with an explanatory message.
- If the destination volume is on a NetApp file server, you must run the FSAUtility command from an Enterprise Vault server that is registered with the destination file server's FPolicy. For example, to migrate placeholders from `NetAppFiler1\volumeA` to `NetAppFiler2\volumeB`, you must run FSAUtility from an Enterprise Vault server that is registered with `NetAppFiler2`.
- If any folders with archive points are missing from the source folder structure, FSAUtility does not proceed with the migration.
- You cannot migrate placeholders to a subfolder of the source folder.

Note that FSAUtility does not proceed with a migration if the destination path already contains an archive for a folder that matches the folder hierarchy on the source path. This restriction prevents a split archive, where the files with the migrated placeholders are in a different vault store from the other archived files. You must specify a destination path that has not been archived from, or one that contains no

folders with archived files in the same folder structure as the source path. For example, consider the example source folder structure and destination folder structure shown in [Figure 20-1](#):

Figure 20-1 FSAUtility placeholder migration: example folder structures



FSAUtility does not proceed with the migration if either of the following applies:

- An archive point with archived files exists at the root level, for both the source volume and the destination volume.
- An archive point with archived files exists in both of the folder structures, on any of the following folders:
 - Folder1
 - Subfolder1
 - Folder2

FSAUtility can migrate the placeholders if there is no clash of archive points that have archived files. For example, the migration is not prevented if either of the following applies:

- On the destination folder structure, only Folder5 and Folder 6 have archive points with archived files.
- Folder2 in the source folder structure has an archive point with archived files, but Folder2 in the destination folder structure does not.

FSAUtility records the following events in the event log:

- The start of a placeholder migration
- Whether a migration completes without errors, or with errors.

Additionally, during a placeholder migration FSAUtility displays appropriate messages on the console, and records detailed entries including errors in the Dtrace logs and in the `Reports\FSAUtility\EV_FILESYSTEM_UTILITY_LOG_DateTime.xml` file.

If a placeholder migration fails, do not archive files on the destination path. Otherwise the archived data for that path may become split across multiple archives. Retry the placeholder migration to see whether it can complete successfully.

Note: Before you use this option with Celerra/VNX placeholders, ensure that FSAUtility is configured to use a suitable method for identifying the placeholders.

See [“About using FSAUtility with EMC Celerra/VNX placeholders”](#) on page 159.

Syntax

```
FSAUtility -pm -s UNC_path -d UNC_path [-cs] [-csf] [-f] [-l  
log_level] [-i]
```

Where:

- `-s UNC_path` specifies the path to the source folder. The path must specify the volume in the format that you used when you added the volume target in the Administration Console.
- `-d UNC_path` specifies the path to the destination folder. This path to the folder must already exist. The path must specify the volume in the format that you used when you added the volume target in the Administration Console.
- `-cs` copies folder security descriptors to new folders at the destination. Security descriptors of existing folders at the destination are not overwritten. This option cannot be used with `-csf`. If you do not specify `-cs` or `-csf`, no folder security descriptors are copied.
- `-csf` copies folder security descriptors from source folders to destination folders, overwriting the security descriptors of destination folders that already exist. This option cannot be used with `-cs`. If you do not specify `-cs` or `-csf`, no folder security descriptors are copied.
- `-f` forces the migration of placeholders when placeholders or files of the same name already exist at the destination. The utility first deletes the existing placeholders or files at the destination and then creates the new ones.
- `-l log_level` specifies whether to log both successful operations and failed operations (0), or failed operations only (1). By default, FSAUtility logs failed operations only.

- `-i` causes FSAUtility to ignore any errors that occur when it moves the placeholders, such as:
 - Failure to determine whether a file is a placeholder.
 - Failure to create placeholders at the destination location, for example as a result of permission issues or insufficient disk space.
 - Failure to delete placeholders from the source location.

If you omit the `-i` parameter and any placeholder move errors occur, FSAUtility logs the errors and stops when it has finished attempting to move all the placeholders. It does not go on to move the archive points or update the Directory database. In this case you may need to rerun `FSAUtility -pm` when you have fixed the causes of the placeholder move failures.

If you specify the `-i` parameter and any placeholder move errors occur, FSAUtility logs the errors but it continues with the remaining steps of the migration: it goes on to move the archive points and update the Directory database. Errors that occur during the archive point migration or the database update are not ignored. FSAUtility continues to log all errors in the log file

```
Reports\FSAUtility\EV_FILESYSTEM_UTILITY_LOG_DateTime.xml.
```

If you specify `-i` and any placeholder move errors occur, you can correct these errors when the command has completed, if you want. For example, you can:

- Delete placeholders at the source location.
- Recreate unmigrated placeholders at the destination location, using the FSAUtility `-c` option.

See [“Recreating placeholders”](#) on page 165.

Recreating unmigrated placeholders using these methods does not retain the security descriptors of the original placeholders, and it does not recreate any alternate data streams that were associated with the placeholders.

Note: We recommend that on the first run of a placeholder migration you omit the `-i` parameter. If the migration fails and the report indicates that the failure was due to errors when moving some placeholders, you can rerun the command with the `-i` parameter if you want FSAUtility to ignore those errors.

Examples

The following command migrates the placeholders along with the archive points from the first folder structure to the second folder structure. The command copies the security descriptors for newly-created folders from the source folders. If any files or placeholders of the same name already exist, the command overwrites them

with new placeholders. FSAUtility does not ignore errors when it moves the placeholders.

```
FSAUtility -pm -s \\myserver\users\user1 -d \\server2\share\user1  
-cs -f
```

Notes

- You cannot run `-pm` in report mode.
- After a placeholder migration, other FSAUtility options do not work on the destination folder until File System Archiving task has processed the folder at least once.
- The migration migrates any placeholders in the source folder tree, including placeholders that were cut and pasted into it. However, if the archived files associated with cut and pasted placeholders are not available in the source tree's archives before the migration occurs, the files will not be in the archives afterwards.
- If a source folder has an archive point and the names of the source folder and destination folder differ, then after a placeholder migration the archive's name does not change to match the destination folder until the File System Archiving task has processed the volume.
- If a vault store already contains an archive with the same name as a destination folder name, then after the migration you see a second archive with the same name. There is no consolidation of the archives.

See also

See [“Moving placeholders and corresponding files”](#) on page 167.

Deleting orphaned placeholders

You can use FSAUtility with the `-o` parameter to delete orphaned placeholders for which no corresponding item exists in the archive. It may also be useful after you delete an entire vault store, vault store partition, or archive.

Note: Before you use this option with Celerra/VNX placeholders, ensure that FSAUtility is configured to use a suitable method for identifying the placeholders.

See [“About using FSAUtility with EMC Celerra/VNX placeholders”](#) on page 159.

Syntax

```
FSAUtility -o -s UNC_path [-l log_level] [-r]
```

Where:

- `-s UNC_path` specifies the path to the required folder, volume, or file server.
- `-l log_level` specifies whether to log both successful operations and failed operations (0) or failed operations only (1). By default, FSAUtility logs failed operations only.
- `-r` specifies report mode. FSAUtility generates a report that outlines the activities that it would perform if you were to run it in normal mode, but without performing those activities.

FSAUtility generates a report named

`EV_FILESYSTEM_UTILITY_REPORT_DateTime.txt`, in the folder
`installpath\Reports\FSAUtility`.

If you run `-o` in normal mode, FSAUtility generates a report named

`EV_FILESYSTEM_UTILITY_REPORT_DateTime.xml`.

Examples

The following command deletes the orphaned placeholders from an entire file server.

```
FSAUtility -o -s \\myserver
```

Restoring archived files

You can use FSAUtility with the `-t` parameter to restore some or all of the archived files to their original location or a new location.

To make sure that any file recalls do not exceed the recall limits, create an Enterprise Vault Backup Operators group in Active Directory, and include in that group the account that is to perform the restore.

By default, FSAUtility restores the files in asynchronous mode. You can choose instead to restore files synchronously if you want. With a synchronous restore you can set a timeout for file restores, and see the progress of each file restore operation. To restore files synchronously and to set a timeout for file restores, create a DWORD entry called `FileDownloadTimeOut` under the following registry key on the Enterprise Vault server:

```
HKEY_LOCAL_MACHINE
\SOFTWARE
\Wow6432Node
\KVS
\Enterprise Vault
\FSARestore
```

If `FileDownloadTimeOut` is set to 0, FSAUtility restores files asynchronously. Any value greater than 0 denotes the timeout, in seconds, for each file recall.

Syntax

```
FSAUtility -t -s UNC_path [-D mm-dd-yyyy] -d UNC_path [-e ext_list] [-f] [-l log_level] [-r]
```

Where:

- `-s UNC_path` specifies the path to the required folder, volume, or file server.
- `-D mm-dd-yyyy` specifies the date after which items must be archived before you can restore them.
- `-d UNC_path` specifies the path to the destination folder.
- `[-e ext_list]` specifies the file types to restore as a comma-separated list of file name extensions. For example:
`*.xls,*.doc,*.txt`
By default, the utility restores all file types.
- `-f` forces FSAUtility to restore the files when placeholders or files of the same name already exist. The utility first deletes the existing placeholders or files and then restores the files.
- `-l log_level` specifies whether to log both successful operations and failed operations (0) or failed operations only (1). By default, FSAUtility logs failed operations only.
- `-r` specifies report mode. FSAUtility generates a report that outlines the activities that it would perform if you were to run it in normal mode, but without performing those activities.

FSAUtility generates a report named

`EV_FILESYSTEM_UTILITY_REPORT_DateTime.txt`, in the folder

`installpath\Reports\FSAUtility`.

If you run `-t` in normal mode, FSAUtility generates a report named

`EV_FILESYSTEM_UTILITY_REPORT_DateTime.xml`.

Examples

The following command restores the Word and Excel files in the folder `\\myserver\users`. It also generates a log file that lists both successful operations and failed operations.

```
FSAUtility -t -s \\myserver\users -e *.doc,*.xls -l 0
```

The following command restores the Word and Excel files for an entire file server.

```
FSAUtility -t -s \\myserver -e *.doc,*.xls -l 0
```

The following command restores all the files that were archived after September 26 2006 on the entire file server.

```
FSAUtility -t -s \\myserver -D 09-26-2006
```

The following command restores the files that were archived after January 2 2002 from \\myserver\users to \\newserver\users.

```
FSAUtility -t -s \\myserver\users -d \\newserver\users -D 01-02-2002  
-l 0
```

Notes

- FSAUtility does not support "hard link" files (directory references to files). You cannot restore an archived file if that file has the same name as a hard link file in the destination folder. When you perform a restore operation, FSAUtility recalls any placeholders that are hard link files.
- When you recall files to an EMC Celerra/VNX device, FSAUtility applies only the folder permissions to the files. If there are placeholders with file-specific permissions, the file permissions are lost and you must reapply them manually.
- Due to a NetApp restriction, FSAUtility does not restore archived files if the path to the folder exceeds 512 characters.

See also

See ["Recalling files corresponding to placeholders"](#) on page 177.

Recalling files corresponding to placeholders

You can use FSAUtility with the `-b` parameter to recall files corresponding to placeholders present in a folder. This facility recalls the placeholders in a given source folder, irrespective of the volume and archive in which the files are located. It also works with placeholders that you have copied into the source folder from another folder.

You can choose to recall files recursively from the subfolders of the source folder, if required.

Note: Before you use this option with Celerra/VNX placeholders, ensure that FSAUtility is configured to use a suitable method for identifying the placeholders.

See ["About using FSAUtility with EMC Celerra/VNX placeholders"](#) on page 159.

By default, FSAUtility recalls the files in asynchronous mode. You can choose instead to recall files synchronously if you want. With a synchronous recall you can set a timeout for file recalls, and see the progress of each file recall operation. To

recall files synchronously and to set a timeout for file recalls, create a DWORD registry entry called `FileDownloadTimeOut` under the following key on the Enterprise Vault server:

```
HKEY_LOCAL_MACHINE
  \SOFTWARE
    \Wow6432Node
      \KVS
        \Enterprise Vault
          \FSARestore
```

If `FileDownloadTimeOut` is set to 0, FSAUtility recalls files asynchronously. Any value greater than 0 denotes the timeout, in seconds, for each file recall.

Syntax

```
FSAUtility -b -s UNC_path [-D mm-dd-yyyy] [-e ext_list] [-recurse]
[-r]
```

Where:

- `-s UNC_path` specifies the path to the required folder, volume, or file server.
- `-D mm-dd-yyyy` specifies the date after which items must be archived before you can recall them.
- `[-e ext_list]` specifies the file types to recall as a comma-separated list of file name extensions. For example:
`*.xls,*.doc,*.txt`
By default, the utility recalls all file types.
- `-recurse` recalls files recursively from subfolders. If not specified, the utility recalls files only from the parent folder.
- `-r` specifies report mode. FSAUtility generates a report that outlines the activities that it would perform if you were to run it in normal mode, but without performing those activities.

FSAUtility generates a report named

```
EV_FILESYSTEM_UTILITY_REPORT_DateTime.txt, in the folder
installpath\Reports\FSAUtility.
```

If you run `-b` in normal mode, FSAUtility generates a report named

```
EV_FILESYSTEM_UTILITY_REPORT_DateTime.xml.
```

Examples

The following command recalls the Word and Excel files that have placeholders in the folder `\\myserver\users`. It also recalls files within subfolders, if any.

```
FSAUtility -b -s \\myserver\users -e *.doc,*.xls -recurse
```

The following command recalls all the files that have placeholders that were archived after May 26 2009 in the folder \\myserver\users. It only recalls files from the parent folder.

```
FSAUtility -b -s \\myserver\users -D 05-26-2009
```

Notes

- If you halt an FSAUtility operation to recall placeholders before it has finished then, when you next start the utility, it prompts you to resume the operation.
- When you recall files to an EMC Celerra/VNX device, FSAUtility applies only the folder permissions to the files. If there are placeholders with file-specific permissions, the file permissions are lost and you must reapply them manually.
- Due to a NetApp restriction, FSAUtility does not recall files if the path to the folder exceeds 512 characters.

See also

See [“Restoring archived files”](#) on page 175.

NTFS to Centera Migration

This chapter includes the following topics:

- [About NTFS to Centera Migration](#)
- [Managing migrator jobs using NTFS to Centera Migration](#)
- [Creating migrator jobs using NTFS to Centera Migration](#)
- [Deleting active jobs using NTFS to Centera Migration](#)
- [Deleting source files after migration using NTFS to Centera Migration](#)
- [NTFS to Centera Migration log files](#)

About NTFS to Centera Migration

The NTFS to Centera Migration utility copies Enterprise Vault savesets from an NTFS source partition to an EMC Centera destination partition. The source partition and destination partition are always in the same vault store, so performing a migration does not affect existing archives and indexes. The source partition files are not deleted.

To start a migration, you create a "migrator job". All jobs run continuously until completed. If the Storage Service is restarted, the migrator jobs restart automatically.

Managing migrator jobs using NTFS to Centera Migration

To manage migrator jobs you use a command-line utility, `NTFSCenteraMigrator.Exe`.

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

To manage migrator jobs using NTFS to Centera Migration

- 1 Open a Command Prompt window.
- 2 Change to the Enterprise Vault program folder (for example C:\Program Files (x86)\Enterprise Vault).
- 3 Type the following command:

```
NTFSCenteraMigrator
```

The command presents you with the following options:

0 = Exit Closes the NTFSCenteraMigrator management program without affecting any existing jobs.

1 = List jobs Lists each of the current NTFS to Centera Migrator jobs, as follows:

```
Job Id: NCM_20031203164814
Storage Service computer: SSCOMPUTER
Vault Store:
Name: MigratorTest
Description: Migrator Test
Source Partition:
Name: MigratorTest Ptn20
Description: Partition of Vault Store
MigratorTest
Destination Partition:
Name: MigratorTest Ptn21
Description: Partition of Vault Store
MigratorTest
Share archived items: Enabled
Start date range: 1999-11-25
End date range: 2003-12-31
Threads: 15
Threads priority: Below Normal
Saveset sharing: Partition property
Log file: <Default>.
```

2 = Create new job Displays a series of prompts with which you can specify the details of a new NTFS to Centera Migrator job.

See [“Creating migrator jobs using NTFS to Centera Migration”](#) on page 182.

3 = Delete existing job Deletes an unfinished job.

- 4 Select the option you require.
- 5 When the migration process has finished, delete the source partition files.

Creating migrator jobs using NTFS to Centera Migration

You start a new NTFS to Centera migration by creating a new migrator job.

To create a migrator job using NTFS to Centera Migration

- 1 Run `NTFSCenteraMigrator`.
- 2 Select option 2, `Create new job`.
- 3 Type the number of the vault store to use as the source for the migration.
- 4 Type the number of the source partition to migrate.
- 5 Type the number of the Centera partition to use as the destination partition.
- 6 When the utility prompts you to type the start date and end date of a range, press `Enter` without specifying a date, or type the year, month, and day. (Use four digits to specify the year; for example, 2006.) If you do not specify either date, the utility migrates all the savesets in the partition.
- 7 When the utility prompts you for the number of worker threads to use, type a number between 1 and 25. The default is 15.

The number of threads affects the rate at which items can be stored in the Centera. Higher numbers increase the storage rate but use more resources on the Storage Service computer.

- 8 Enter the worker thread priority to use. This priority can be either of the following:
 - **Below Normal.** Windows gives priority to other threads, so migrator activities have lower priority than applications on the computer.

Setting the number of worker threads to 15 or more and selecting **Below Normal** should give good performance when the computer is not busy with other tasks.

- **Normal.** Windows gives equal priority to migrator activities and other applications.
- 9** Enter the saveset sharing option to use. This option can be one of the following:
- **0 — Use Partition property.** Use the same setting as for the destination partition.
 - **Force off.** Saveset sharing is disabled. This increases performance at the expense of space.
 - **Force on.** Saveset sharing is enabled. This maximizes the storage but reduces the migration performance.
- 10** When the utility prompts you for the name and location of the log file, either type the full path to the file or press Enter to use the default name and location. For example, you could try the path `E:\Reports\Migration001.log`. Any folder that you specify must already exist.

By default, the NTFS to Centera Migrator creates a log file for each job in the Enterprise Vault `Reports` subfolder (for example `C:\Program Files (x86)\Enterprise Vault\Reports`). If you do not specify a log file name, the name that is used is `NCM_DateAndTime.log`, where `DateAndTime` indicates the date and time that the job was created.

See [“NTFS to Centera Migration log files”](#) on page 185.

- 11** Choose whether to remove all references to a saveset if the saveset file no longer exists in the source partition.
- 12** If a saveset has two or more sharers, choose whether to remove the unselected sharers and compact the saveset before storing it.
- If you choose not to remove unselected sharers, the utility stores the complete saveset in the Centera clip, including multiple sharers, if present. This results in larger savesets on the Centera and hence more occupied space. The required sharer is selected when the saveset is stored and retrieved.
- 13** Choose the required error handling options.

Error wait time	<p>Specifies the number of seconds for which the utility waits before retrying the operation, if an error occurs. The default is 10.</p> <p>Note that the utility does not perform a retry for the following error conditions:</p> <ul style="list-style-type: none">■ STORAGE_E_EXTRACT_CAB_HR: Error extracting Saveset file from Cab file■ STORAGE_E_SAVESET_DECOMPRESSION: Error decompressing Saveset■ STORAGE_E_SAVESETNOTVALID: Invalid Saveset <p>For these error conditions, the utility immediately abandons processing of the saveset. However, it tries to process the saveset again when the Storage Service is restarted. (Restarting this service restarts the migration job.)</p>
Error count	<p>Specifies the maximum number of times that the utility retries processing a saveset. The default is five.</p> <p>If the utility fails to process the saveset after the maximum number of retries, it performs one of the following actions:</p> <ul style="list-style-type: none">■ If the error appears irrecoverable, the utility abandons processing of the saveset. However, it tries to process the saveset again when the Storage Service is restarted.■ If the error is potentially recoverable, such as a network problem, the utility pauses the thread for the error pause time (see below), and then tries to process the saveset again.
Error pause time	<p>Specifies the number of minutes for which to pause the thread before trying to process a saveset again, if the utility fails to process the saveset after the maximum number of tries, but the error is potentially recoverable. The default is five.</p>

- 14 Restart the Storage Service that manages the vault store. The new job starts when the Storage Service has restarted.

Deleting active jobs using NTFS to Centera Migration

The NTFS to Centera Migration utility automatically deletes jobs when they have completed. However, you can manually delete any jobs that are still in progress.

To delete an active job using NTFS to Centera Migration

- 1 Start NTFSCenteraMigrator.
- 2 Select option 3, `Delete existing job`.
 NTFSCenteraMigrator lists the active jobs.
- 3 Type the number of the job that you want to delete.
 The job is now marked for deletion and no longer appears in the list of jobs.
- 4 Restart the Storage Service that manages the vault store.

Deleting source files after migration using NTFS to Centera Migration

NTFSCenteraMigrator does not delete the source files after they have been migrated to Centera. Data in the source folders may be shared with other partitions and you must not delete the data while there are still references to it. You must not delete the source files unless it is safe to do so.

If you have moved all NTFS partitions to Centera then you can delete the source data.

To delete the source files after migration using NTFS to Centera Migration

- 1 In the Administration Console, expand **Vault Store Groups**.
 Expand the vault store that contains the partition you want to delete.
- 2 Right-click the partition and, on the shortcut menu, click **Delete**. The Administration Console prompts you to confirm that you want to delete the partition.
- 3 Click **Yes**.
- 4 If the Administration Console lets you delete the partition then you can use Windows Explorer to delete the partition's files.
 If the Administration Console does not let you delete the partition then it is not safe to delete the partition's files.

NTFS to Centera Migration log files

The NTFS to Centera Migration utility creates a log file for each job. The utility prompts you for the name and location of the file to create.

The log file is locked while the job is running.

The following is an example of a log file.

```
2005-12-02 13:08:53 NTFS to Centera Migrator Log file created for
Job NCM_20031202130732
2005-12-02 13:08:53
2005-12-02 13:08:53 Starting migration from Test Ptn16 to Test Ptn17
in Test
2005-12-02 13:08:53 Savesets in NTFS partition: 368
2005-12-02 13:09:25 Migration stopped
2005-12-02 13:09:25 Savesets migrated: 368, Rate: 42735
Savesets/hour
2005-12-02 13:09:25 Savesets in NTFS partition: 0
2005-12-02 13:09:25 Migration completed - job entry has been deleted
```

Permission Browser

This chapter includes the following topics:

- [About Permission Browser](#)
- [Running Permission Browser](#)

About Permission Browser

Permission Browser lets you view the security identifiers (SIDs) and access permissions for the archives and archive folders in an Enterprise Vault directory database.

Running Permission Browser

Follow the instructions below to open Permission Browser and select the archives and folders that interest you.

To run Permission Browser

- 1 In Windows Explorer, browse to the Enterprise Vault program folder (for example `C:\Program Files (x86)\Enterprise Vault`).
- 2 Double-click `PermissionBrowser.exe`.
- 3 Ensure that the **ODBC DSN Name** field shows the name of the Enterprise Vault directory database for whose archives you want to determine the access permissions.
- 4 Check **Use NTLM** to log on to the database using your current Windows credentials, or uncheck it to submit a SQL user name and password.
- 5 In the **Select an archive** and **Select a folder** boxes, click items to display their security descriptors in the box at the right.

Policy Manager (EVPM)

This chapter includes the following topics:

- [About Policy Manager](#)
- [Policy Manager syntax](#)
- [Saving a Policy Manager initialization file as a Unicode file](#)
- [Policy Manager initialization file syntax](#)
- [Sections and keynames in Policy Manager initialization file](#)
- [Policy Manager initialization file examples](#)
- [About using the Provisioning API to run Policy Manager scripts](#)

About Policy Manager

Enterprise Vault Policy Manager provides a scripted way to modify and control Exchange mailboxes and archives so that they conform to your Enterprise Vault archiving policies. You can apply settings to individual mailboxes in a much more specific manner than you can when you use the Administration Console.

Additionally, you can use Policy Manager to migrate the contents of PST files and NSF files to Enterprise Vault.

Note: You cannot use Policy Manager to modify or control Domino mail files or archives.

The program runs from a command prompt window and uses an initialization file of settings to apply to mailboxes or archives, or to control the migration of PST and NSF files.

To ensure the correct permissions, run Policy Manager while you are logged on as the vault service account.

You cannot use Policy Manager to change permissions to Domino archives.

Policy Manager is installed in the Enterprise Vault program folder (for example `C:\Program Files (x86)\Enterprise Vault`). Its file name is `EVPM.EXE`.

Policy Manager syntax

```
EVPM [-?] {[-e Exchange_server] [-m mailbox_alias] | [-d]} [-f  
input_file]
```

where:

<code>-?</code>	Displays help information on the utility.
<code>-e <i>Exchange_server</i></code>	Specifies the name of the Exchange Server computer. When you run EVPM with this parameter, it ignores any Domino related settings in the initialization file. For Exchange Server 2010 and later, you must specify the fully qualified domain name of the Exchange Server computer. It might be necessary to provide a fully qualified domain name if your Exchange Server and the Enterprise Vault server are in separate Active Directory forests.
<code>-m <i>mailbox_alias</i></code>	Specifies the name of the Enterprise Vault system mailbox.
<code>-d</code>	Run Domino tasks. When you run EVPM with this parameter, it ignores any Exchange related settings in the initialization file.
<code>-f <i>input_file</i></code>	Specifies the name and location of the initialization file.

For example:

- `EVPM -e ExchSvr1.evexample.local -m EVSvceMbx -f c:\ExchSvr1.ini`
This command processes the settings in `c:\ExchSvr1.ini`, against Exchange Server `ExchSvr1.evexample.local`, using the Exchange system mailbox `EVSvceMbx`.
- `EVPM -d -f c:\DominoSvr1.ini`
This command processes the NSF migration settings in `c:\DominoSvr1.ini`.

If you run Policy Manager without any parameters, it prompts you for them. After the first time you run Policy Manager, it offers the values you set last time as the

default when it prompts. You can press Enter to accept the default, or enter a new value.

Note: You must run this utility with Administrator privileges if the computer has User Account Control (UAC) enabled.

See [“Running the Enterprise Vault command-line utilities with administrator privileges”](#) on page 16.

Saving a Policy Manager initialization file as a Unicode file

A Policy Manager initialization file must be a Unicode file. You can use Windows Notepad to create such files.

To save a Policy Manager initialization file as a Unicode file

- 1 On the **Tools** menu in Notepad, click **Save As**.
- 2 Type a name for the file.
- 3 Next to **Encoding**, select **Unicode** from the list.
- 4 Click **Save**.

Policy Manager initialization file syntax

The Policy Manager initialization file is a standard Windows INI file that contains sections, keynames, and values, as follows:

```
[SectionName]
KeyName1=Value1
KeyName2=Value2
...
```

Note the following:

- The section names and keynames are not case-sensitive.
- If a keyname can have multiple values, separate them with commas and make sure that they are all on the same line.
- You need only specify mandatory keynames and those optional keynames whose value you want to set. Ignore the other keynames.
- A line that starts with a semicolon (;) is a comment. The semicolon must be the first non-whitespace character on the line.

See [“Policy Manager initialization file examples”](#) on page 236.

Sections and keynames in Policy Manager initialization file

[Table 23-1](#) lists the sections that the initialization file can contain.

Table 23-1 Sections of Policy Manager initialization file

Section	Description
Directory	Must be the first section in the file. See “ [Directory] section of the Policy Manager initialization file” on page 192.
Archive	Lets you modify the properties of one or more archives. See “[Archive] section of the Policy Manager initialization file” on page 193.
ArchivePermissions	Lets you change the permissions on one or all archives. See “[ArchivePermissions] section of the Policy Manager initialization file” on page 195.
Filter	Lets you specify a group of settings to apply to folders within mailboxes. See “[Filter] section of the Policy Manager initialization file” on page 196.
Mailbox	Lets you change settings for one or more mailboxes. See “[Mailbox] section of the Policy Manager initialization file” on page 202.
Folder	Lets you modify the properties of individual folders or complete mailboxes. See “[Folder] section of the Policy Manager initialization file” on page 205.
PublicFolder	Lets you modify the properties of public folders. See “[PublicFolder] section in the Policy Manager initialization file” on page 210.

Table 23-1 Sections of Policy Manager initialization file (*continued*)

Section	Description
PSTdefaults	Mandatory section when you migrate the contents of PST files to Enterprise Vault. See “[PSTdefaults] section in the Policy Manager initialization file” on page 212.
PST	Lets you migrate the contents of PST files to Enterprise Vault. See “[PST] section in the Policy Manager initialization file” on page 217.
PSTcheckpoint	Policy Manager generates this section automatically. See “[PSTcheckpoint] section in the Policy Manager initialization file” on page 223.
NSFDefaults	Mandatory section when you to migrate the contents of NSF files to Enterprise Vault. See “[NSFDefaults] section in the Policy Manager initialization file” on page 224.
NSF	Mandatory section when you migrate the contents of NSF files to Enterprise Vault. See “[NSF] section in the Policy Manager initialization file” on page 229.
NSFCheckPoint	Policy Manager generates this section automatically. See “[NSFCheckPoint] section in the Policy Manager initialization file” on page 234.

[Directory] section of the Policy Manager initialization file

This section is mandatory and must be the first section in the file.

DirectoryComputerName

Mandatory. Specifies the computer that hosts the Enterprise Vault directory service.

SiteName

Mandatory. For Exchange mailbox tasks and PST migrations, this keyname specifies the name or ID of the Enterprise Vault site that manages the archives or the Exchange mailboxes you want to modify or migrate.

For NSF migrations, this keyname specifies the name or ID of the Enterprise Vault site that manages the archives into which you want to migrate NSF file content.

StorageSvcComputerName

Optional. For NSF migrations, this keyname specifies the server that runs the storage service. EVP runs the NSF migrator server on the computer you specify, to validate the NSF files. If you do not set a value for this keyname, EVP runs the NSF migrator server on any Enterprise Vault server that has a storage service and has the Notes client installed.

[Archive] section of the Policy Manager initialization file

Include this section if you want to modify the properties of one or more archives.

ArchiveName

Mandatory. Identifies the archive to process.

Possible values:

- Archive name
- Archive ID

If the archive does not exist, Policy Manager creates a shared archive. (If you want to create mailbox archives, enable the mailboxes.)

BillingOwner

Mandatory. Specifies a Windows account for billing purposes.

DeleteExpiredItems

Optional. Specifies whether Enterprise Vault can automatically delete items from the archive when their retention periods expire. If not specified, existing archives are not modified.

Possible values:

- true (default, for new archives only)
- false

You can place the archive on legal hold by setting this keyname to false and the [DeleteProtected](#) keyname to true.

DeleteProtected

Optional. Specifies whether to allow users to delete items manually from the archive. If you choose to prevent this then, in addition, the archive cannot be moved or deleted. If not specified, existing archives are not modified.

Possible values:

- true
- false (default, for new archives only)

You can place the archive on legal hold by setting this keyname to true and the [DeleteExpiredItems](#) keyname to false.

Description

Optional. Sets the description that the user sees when selecting an archive in which to search. The description is also shown in the Administration Console.

If you do not specify a description, existing archives are unchanged, and the text that is used for new archives is "Created by the Policy Manager".

IndexingLevel

Optional. Specifies how detailed an index Enterprise Vault is to create for the archive.

If you omit `IndexingLevel`, the site default setting is used for new archives. Existing archives are not modified.

Possible values:

- Brief
- Full

IndexAttachmentSnippet

Optional. Specifies whether preview text is displayed for attachments in the search results list. Enabling this option increases the size of an index.

Note: This option is for a future release; you cannot display the previews in the current version of Enterprise Vault.

Possible values:

- true
- false (default)

IndexSnippetLength

Optional. Specifies the amount of preview text (number of characters) that is displayed in the search results list. The size of an index increases when you increase the preview length.

If you omit `IndexSnippetLength`, the site default setting is used for new archives. Existing archives are not modified.

Possible values:

- 128 (default)
- 1000

VaultStoreName

Mandatory. The name of the vault store in which the archive exists or is to be created.

[ArchivePermissions] section of the Policy Manager initialization file

Include this section if you want to make changes to the permissions on one or all archives.

ArchiveName

Mandatory. Identifies the archive to which the permission settings are applied.

If there are multiple folders with the same name and you specify a name, Policy Manager modifies only the first one that it finds. In this case, you must use archive IDs to specify the archives.

Possible values:

- The name of an archive
- An archive ID
- ALL (permissions are applied to all journal, shared, and mailbox archives in the specified vault site)
- ALL_JOURNAL (permissions are applied to all journal archives)
- ALL_SHARED (permissions are applied to all shared archives)
- ALL_MAILBOX (permissions are applied to all mailbox archives)

DenyAccess

Optional. Removes the access to the specified archive. If DenyAccess is specified with GrantAccess, DenyAccess is used and GrantAccess is ignored. You can have many occurrences of DenyAccess within the same [ArchivePermissions] section.

Possible values:

- A list of the permissions, followed by a comma and then a comma-delimited list of groups or accounts that are denied the specified access. Permissions can be any of `read`, `write`, and `delete`, followed by a comma. For example to deny ourdomain\smith read and write access:

```
DenyAccess = read write, ourdomain\smith
```

GrantAccess

Optional. Grants to the specified Windows accounts the specified access to the archive.

The new values supplement any existing access rights. You can have many occurrences of GrantAccess within the same [ArchivePermissions] section.

Possible values:

- A list of permissions, followed by a comma and then a comma-delimited list of groups or accounts that are granted the specified permissions. Permissions can be any of `read`, `write`, and `delete`, followed by a comma. For example, to grant read and write access to ourdomain\smith:

```
GrantAccess = read write, ourdomain\smith
```

Zap

Optional. Clears all permissions on the archive. If you specify Zap, GrantAccess and DenyAccess are ignored.

Possible values:

- `true`
- `false` (default)

[Filter] section of the Policy Manager initialization file

Include this section to specify a group of settings to apply to folders within mailboxes. You then apply this setting by specifying the filter name in the [Folder] section.

Note: The [Filter] section must be specified before the [Folder] section in the initialization file.

ALargeItemThresholdPeriod

Optional. This setting is equivalent to the number that you select for **Never archive items younger than** on the Archiving Rules tab of the Exchange Mailbox Policy dialog box.

If you use this setting, you must specify `Name = mailboxroot` in the [Folder] section that references the filter.

If you specify `ALargeItemThresholdPeriod`, you must also set values for all the following:

- `UseInactivityPeriod` (must be set to `true`)

- APrioritizeLargeltems
- APriorizeltemsOver
- ALargeltemThresholdUnits

Possible values:

- A positive integer

ALargeltemThresholdUnits

Optional. This setting is equivalent to the units entry for **Never archive items younger than** on the Archiving Rules tab of the Exchange Mailbox Policy dialog box.

If you use this setting, you must specify Name = mailboxroot in the [Folder] section that references the filter.

If you specify ALargeltemThresholdUnits, you must also set values for all the following:

- UseInactivityPeriod (must be set to true)
- APrioritizeLargeltems
- APriorizeltemsOver
- ALargeltemThresholdPeriod

Possible values:

- Days
- Weeks
- Months
- Years

APriorizeltemsOver

Optional. This setting is equivalent to the size that you select for **Start with items larger than** on the Archiving Rules tab of the Exchange Mailbox Policy dialog box.

If you use this setting, you must specify Name = mailboxroot in the [Folder] section that references the filter.

If you specify APriorizeltemsOver, you must also set values for all the following:

- UseInactivityPeriod (must be set to true)
- APrioritizeLargeltems
- ALargeltemThresholdUnits
- ALargeltemThresholdPeriod

Possible values:

- An integer that specifies the size of items in KB to which you want to give priority

APrioritizeLargerItems

Optional. This setting is equivalent to the **Start with items larger than** option on the Archiving Rules tab of the Exchange Mailbox Policy dialog box.

If you use this setting, you must specify Name = mailboxroot in the [Folder] section that references the filter.

If you specify APrioritizeLargerItems, you must also set values for all the following:

- UseInactivityPeriod (must be set to true)
- APrioritizeItemsOver
- ALargerItemThresholdUnits
- ALargerItemThresholdPeriod

Possible values:

- true
- false

CreateShortcut

Mandatory. Specifies whether Enterprise Vault is to create shortcuts to items that are archived from the folder to which this filter is applied.

Possible values:

- true
- false

DeleteOriginal

Mandatory. Specifies whether Enterprise Vault is to delete the original items when it archives from the folder to which this filter is applied.

Possible values:

- true
- false

InactivityPeriod

Optional, but mandatory when you set UseInactivityPeriod to true. InactivityPeriod is valid only when you specify UseInactivityPeriod. You must also specify InactivityUnits to indicate how long an item can remain unmodified before it is eligible

for archiving. This is the same as the **Archive items when they are older than** setting in the Archiving Rules tab of the Exchange Mailbox Policy dialog box.

Possible values:

- An integer between 0 and 500

InactivityUnits

Optional, but mandatory when you set `UseInactivityPeriod` to true. Valid only when you specify `UseInactivityPeriod`. When you use this setting, you must specify it with `InactivityPeriod` to indicate how long an item can remain unmodified before it is eligible for archiving. This is the same as the **Archive items when they are older than** setting in the Archiving Rules tab of the Exchange Mailbox Policy dialog box.

Possible values:

- Days
- Weeks
- Months
- Years

Name

Mandatory. Identifies the filter. This name applies only within this initialization file. You refer to this filter section by name in any [Folder] section in the initialization file.

PercentageQuota

Optional, but mandatory when you set `UsePercentageQuota` to true. This setting applies only when using quota-based archiving. Enterprise Vault archives from the mailbox until this percentage of mailbox storage limit is free.

If you use this setting, you must specify `Name = mailboxroot` in the [Folder] section that references the filter.

`PercentageQuota` is not valid for public folders.

Possible values:

- An integer between 0 and 99

QMinimumAgeThresholdPeriod

Optional. This setting is equivalent to the value that you select for **Never archive items younger than** on the Archiving Rules tab of the Exchange Mailbox Policy dialog box.

If you use this setting, you must specify Name = mailboxroot in the [Folder] section that references the filter.

If you specify QMinimumAgeThresholdPeriod, you must also set values for the following:

- UsePercentageQuota (must be set to true)
- QMinimumAgeThresholdUnits.

QMinimumAgeThresholdPeriod is not valid for public folders.

Possible values:

- An integer

QMinimumAgeThresholdUnits

Optional. This setting is equivalent to the units that you select for **Never archive items younger than** on the Archiving Rules tab of the Exchange Mailbox Policy dialog box.

If you use this setting, you must specify Name = mailboxroot in the [Folder] section that references the filter.

If you specify QMinimumAgeThresholdUnits, you must also set values for the following:

- UsePercentageQuota (must be set to true)
- QMinimumAgeThresholdPeriod.

QMinimumAgeThresholdUnits is not valid for public folders.

Possible values:

- Days
- Weeks
- Months
- Years

QPrioritizeItemsOver

Optional. This setting is equivalent to the **Start with items larger than** size entry on the Archiving Rules tab of the Exchange Mailbox Policy dialog box.

If you use this setting, you must specify Name = mailboxroot in the [Folder] section that references the filter.

If you specify QPrioritizeItemsOver, you must also set values for the following:

- UsePercentageQuota (must be set to true)

- QPrioritizeLargerItems

QPrioritizeItemsOver is not valid for public folders.

Possible values:

- An integer that specifies the size of items in KB to which you want to give priority.

QPrioritizeLargerItems

Optional. This setting is equivalent to the **Start with items larger than** checkbox on the Archiving Rules tab of the Exchange Mailbox Policy dialog box.

If you use this setting, you must specify Name = mailboxroot in the [Folder] section that references the filter.

If you specify QPrioritizeLargerItems, you must also set values for the following:

- UsePercentageQuota (must be set to true)
- QPrioritizeItemsOver

QPrioritizeLargerItems is not valid for public folders.

Possible values:

- true
- false

UnreadMail

Mandatory. Specifies whether Enterprise Vault archives unread mail items from the folder to which you apply this filter.

Possible values:

- true
- false

UseInactivityPeriod

Mandatory, unless Filtername in the [Folder] section is set to SystemDefault or DoNotArchive.

When you use UseInactivityPeriod and UsePercentageQuota, you must set at least one of them to true.

UseInactivityPeriod specifies whether to use age-based archiving.

If you use this setting, you must specify Name = mailboxroot in the [Folder] section that references the filter.

Possible values:

- true (use age-based archiving)

- false (do not use age-based archiving)

UsePercentageQuota

Optional. When you use UseInactivityPeriod and UsePercentageQuota, you must set at least one of them to true.

UsePercentageQuota specifies whether to use quota-based archiving.

If you use this setting, you must specify Name = mailboxroot in the [Folder] section that references the filter.

If you set UsePercentageQuota to true, you must also set a value for PercentageQuota.

UsePercentageQuota is not valid for public folders.

Possible values:

- true (use quota-based archiving)
- false (do not use quota-based archiving)

[Mailbox] section of the Policy Manager initialization file

Include this section if you want Policy Manager to change settings for one or more mailboxes.

DistinguishedName

Optional. Identifies a mailbox.

To apply attributes to all non-system mailboxes on the Exchange server, create a [Mailbox] section and set DistinguishedName to All.

A single [Mailbox] section can contain multiple DistinguishedName keywords, LDAPQuery keywords, or a mixture of the two.

You can run Exchange Mailbox Tasks in report mode to obtain a list of all the mailboxes. You can then copy distinguished names from the report to the initialization file.

The distinguished name value required is the legacyExchangeDN property for the mailbox in Active Directory. For example:

/o=Org1/ou=Admin Group/cn=Recipients/cn=jones

You can also view the legacyExchangeDN property using an Active Directory editor, such as the LDP (ldp.exe) tool, or Active Directory Service Interfaces (ADSI) Edit.

LDAPquery

Optional. Lets you select mailboxes by using LDAP attributes. The value uses standard LDAP query syntax:

```
LDAPquery = StandardQuery
```

A simple query looks like the following:

```
LDAPquery = (attribute operator value)
```

where:

- *attribute* is the LDAP attribute, such as "department".
- *operator* is a valid LDAP operator. This operator is normally one of the following:

&	logical and
	logical or
!	logical not
=	equal to

When an operator follows an attribute, there must be no space between the operator and the attribute. For example, "company=" is correct, whereas "company =" is not.

You can use the asterisk wildcard (*) in string values. For example, to select all mailboxes with a surname that starts with the letter J:

```
LDAPquery = sn= j*
```

Notes:

- If you specify an incorrect LDAP attribute, Policy Manager does not find the mailbox and so does not make any changes.
- The following are useful attributes:

```
cn [common name]  
sn [surname]  
company  
department  
displayName  
extensionAttribute1  
extensionAttribute2  
extensionAttribute3  
extensionAttribute4  
extensionAttribute5  
extensionAttribute6
```

```
extensionAttribute7  
extensionAttribute8  
extensionAttribute9  
extensionAttribute10  
extensionAttribute11  
extensionAttribute12  
extensionAttribute13  
extensionAttribute14  
extensionAttribute15  
memberof
```

Some example queries are as follows:

- To select mailboxes with LDAP attribute "department" equal to "research":
`LDAPquery = department= research`
- To select mailboxes with LDAP attribute "department" equal to "research" and "Extension-Attribute-1" set to "10000":
`LDAPquery = (& (department= research) (extensionAttribute1= 10000))`
- To select mailboxes belonging to the users in the IT Guys security group in the Texas organizational unit:
`LDAPquery = (memberof= CN=IT Guys,OU=texas,DC=evdemo,DC=local)`

ProvisioningGroup

Optional. Lets you select mailboxes that have been provisioned by a specific provisioning target group.

For example, to select all the mailboxes that have been provisioned by a group called "VIPs":

```
ProvisioningGroup=VIPs
```

Note: In this example, EVPM selects only the mailboxes that have actually been provisioned by the provisioning target group. Other users may be eligible under the same group, but not provisioned because they have already been provisioned by a higher priority group. You must also run the provisioning task before running EVPM scripts that use the ProvisioningGroup setting, to ensure that provisioning is up to date.

ResetArchiveFolderPerm

Optional. Lets you reset the permissions on archive folders to the user's default permissions.

When it migrates the contents of a PST file to an archive, Enterprise Vault assigns the same access permissions to the imported PST folders as it does to their parent folder. The access permissions on the PST file itself are not transferred to the newly-created folders. This is in line with standard Exchange policy, but it may give rise to a possible security issue: Any user who has read permissions to the parent folder in the Exchange mailbox can potentially access the migrated items in the PST import folders. You can address this issue by resetting the permissions on the archive folders and thereby stopping unqualified users from viewing the contents of PST import folders.

Possible values:

- 1. (Reset the archive permissions on all folders to the user's default permissions.)
- 2. (As for 1, but also performs a mailbox synchronization when Policy Manager has reset the archive folder permissions.)

[Folder] section of the Policy Manager initialization file

Include this section if you want to modify the properties of individual folders or complete mailboxes.

ArchiveName

Optional. Identifies the archive in which items from the folder are archived. The default is the value that is set on the mailbox root.

Possible values:

- An archive name or archive ID

DisassociateArchiveFromMailbox

Optional. Disassociates a mailbox from its related archive. Use `DisassociateArchiveFromMailbox` in conjunction with `Zap`.

If you zap a mailbox and disassociate it from its archive, Enterprise Vault creates a new archive for the mailbox when it is later enabled instead of relinking the mailbox to its old archive.

`DisassociateArchiveFromMailbox` is valid only if:

- `Name=mailboxroot`
- `zap=true`

Possible values:

- `true`
- `false`

Enabled

Optional. Specifies that the mailbox is enabled or disabled. If not specified, the mailbox setting remains unchanged. Can be applied to the mailbox root folder only.

If you enable a mailbox that was once enabled but subsequently disabled, Policy Manager automatically reconnects it to the existing mailbox archive.

Possible values:

- true
- false

ExchangePermissions

Optional. Specifies the folder permissions that you want to add, change, or remove.

You can specify one of the following:

- Author
- Contributor
- Editor
- NoneditingAuthor
- Owner
- PublishingAuthor
- PublishingEditor
- Reviewer

When you specify users, you can use either of the following forms:

- The user's display name from the Global Address List (GAL). For example, "Sue Smith".
- The mailbox Distinguished Name. For example, "/o=Org1/ou=Admin Group/cn=Recipients/cn=smith". Use this format if there are likely to be duplicate display names in the GAL.

The distinguished name value required is the legacyExchangeDN property for the mailbox in Active Directory.

See "[Mailbox] section of the Policy Manager initialization file" on page 202.

Possible values:

- To grant access to a folder, use either of the following forms:

```
ExchangePermissions = ADD; UserA:RoleA;UserB:RoleB;...  
ExchangePermissions = +; UserA:RoleA;UserB:RoleB;...
```

where `UserA` is the first user and `RoleA` is the permission that you want to add.

- To remove permissions, use either of the following forms:

```
ExchangePermissions = DEL; UserA;UserB;...
ExchangePermissions = -; UserA;UserB;...
```

where `UserA` is the first user to remove and `UserB` is the second user to remove.

- To replace the permissions of users who already have access to the folder:

```
ExchangePermissions = UserA:RoleA;UserB:RoleB;...
```

where `UserA` is the first user and `RoleA` is the permission that you want to add or modify.

Filtername

Optional. Specifies the name of one of the standard filters, or the name of a filter that you have defined within the initialization file. The filter defines the settings that you want Policy Manager to apply to mailboxes.

Possible values:

- `SystemDefault`. (Default. Use the default Enterprise Vault site settings, as defined in the Administration Console.)
- `DoNotArchive`. (Do not archive from the folder to which the filter is applied.)
- Name of filter. (A filter that is defined within the initialization file.)
- `Parent`. (Use the settings that are configured for the parent folder.)

MailboxDN

Optional. Specifies a mailbox and restricts the [Folder] section so that it applies only to the specified mailbox.

The distinguished name value required is the `legacyExchangeDN` property for the mailbox in Active Directory.

Name

Mandatory. If the specified folder hierarchy does not exist, Policy Manager creates it and sets the specified properties.

Possible values:

- `mailboxroot` (specifies the root folder).
- folder path. You do not need to specify a path for the following special folders that Outlook creates: `Inbox`, `Outbox`, `SentItems`, `DeletedItems`, `Drafts`, `Calendar`, `Contacts`, `Journal`, `Notes`, and `Tasks`. In these cases, specify only the folder

name without the leading backslash. These names work for all languages. For example, you can specify "Inbox" on a Japanese system.

Examples:

- To create a folder that is called "xyz" in the root folder:

```
Name = \xyz
```

- To specify the Deleted Items folder:

```
Name = DeletedItems
```

NonDeletable

Optional. Specifies whether Outlook and OWA users can delete, move, or copy the folder and all subfolders.

Possible values:

- true
- false

Caution: For information on known problems with this setting, see article [TECH55275](#) on the Veritas Enterprise Support site.

OverrideArchiveLocks

Optional. Overrides all Administration Console lock settings. This setting forces Policy Manager to modify folder settings even if the Administration Console has Force Users To Use Site Settings For Archiving set on the Mailbox Actions property page.

Note: The default is for Policy Manager to obey all lock settings. If you want to override lock settings, include `OverrideArchiveLocks` and set the value to true.

Possible values:

- true
- false (default)

RetentionCategory

Optional. Specifies the retention category to use when you archive from the folder. If not specified, the site default retention category is used.

SiteName

Optional. Can be applied to the mailbox root folder only.

Suspended

Optional. Specifies whether the mailbox is suspended. If not specified, the default of false is applied. Can be applied only to the mailbox root folder.

Possible values:

- true
- false (default)

URL

Optional. Specifies the URL of the web page that is displayed when a user opens the folder in Outlook. For example, you can use this feature to create folders with links to Enterprise Vault Search.

VaultStoreName

Optional. Identifies the vault store to use when you create a new archive. If the mailbox is already enabled or disabled, VaultStoreName is ignored. If VaultStoreName is not specified, Policy Manager uses the default vault store.

VaultStoreName is valid only if:

- Name=mailboxroot
- Enabled=true
- ArchiveName is not specified
- The mailbox has never been enabled

Possible values:

- The name or ID of the vault store to use

Zap

Optional, but mandatory when you set DisassociateArchiveFromMailbox to true. Removes all Enterprise Vault properties from the folder. If you apply this setting to the mailbox root, it makes the mailbox appear as though it has never been enabled for archiving. If Zap is specified, it overrides all other [Folder] keynames.

Possible values:

- true
- false (default)

[PublicFolder] section in the Policy Manager initialization file

Include this section if you want to modify the properties of public folders. This section is optional.

ApplyToSubfolders

Optional. Causes Policy Manager to modify all subfolders beneath the folder that is specified in Name, regardless of which Exchange Public Folder Task processes those public folders.

ExchangePermissions

Optional. Specifies the folder permissions that you want to add, change, or remove.

You can specify one of the following:

- Author
- Contributor
- Editor
- NoneditingAuthor
- Owner
- PublishingAuthor
- PublishingEditor
- Reviewer

When you specify users, you can use either of the following forms:

- The user's display name from the Global Address List (GAL). For example, "Sue Smith".
- The mailbox Distinguished Name. The distinguished name value required is the legacyExchangeDN property for the mailbox in Active Directory. For example, "/o=Org1/ou=Admin Group/cn=Recipients/cn=smith". Use this format if there are likely to be duplicate display names in the GAL.

Possible values:

- To grant access to a folder, use either of the following forms:

```
ExchangePermissions = ADD; UserA:RoleA;UserB:RoleB;...  
ExchangePermissions = +; UserA:RoleA;UserB:RoleB;...
```

where `UserA` is the first user and `RoleA` is the permission that you want to add.

- To remove permissions, use either of the following forms:

```
ExchangePermissions = DEL; UserA;UserB;...  
ExchangePermissions = -; UserA;UserB;...
```

where `UserA` is the first user to remove and `UserB` is the second user to remove.

- To replace the permissions of users who already have access to the folder:

```
ExchangePermissions = UserA:RoleA;UserB:RoleB;...
```

where `UserA` is the first user and `RoleA` is the permission that you want to add or modify.

See [“Policy Manager initialization file examples”](#) on page 236.

Filtername

Optional. Specifies the name of one of the standard filters, or the name of a filter that you have defined in the initialization file. The filter defines the settings for Policy Manager to apply to public folders.

Possible values:

- `SystemDefault`. (Default. Use the default public folder settings, as defined in the Administration Console.)
- `DoNotArchive`. (Do not archive from the folder to which the filter is applied.)
- Name of filter. (A filter that you have defined within the initialization file.)

OverrideArchiveLocks

Optional. Overrides all Administration Console lock settings. The default is for Policy Manager to obey all lock settings. Since you almost always want to override lock settings, you probably want to include `OverrideArchiveLocks` and set the value to `true`.

Possible values:

- `true`
- `false` (default)

Name

Mandatory.

RetentionCategory

Mandatory. Specifies the retention category to apply to the folder. The retention category must already exist.

[PSTdefaults] section in the Policy Manager initialization file

This section is mandatory when you use Policy Manager to migrate the contents of PST files to Enterprise Vault.

This section specifies the default settings that apply to all PST migrations. You can override these default settings for individual PST files by specifying the appropriate option in the [PST] section for that file.

ArchiveNonExpiredCallItems

Optional. Controls whether Policy Manager migrates the unexpired calendar items. If you choose to migrate unexpired calendar items, users must restore the items before they can modify them.

Possible values:

- True
- False (default)

CancelMbxAutoArchive

Optional. Controls whether Policy Manager turns off Outlook AutoArchiving for all the folders in the target mailboxes. This stops Outlook from automatically archiving items to PST files.

- true
- false (default)

CompactPST

Optional. Controls whether the PST file is compacted after successful migration of its contents.

If you intend to use this PST compaction feature at the end of migrations, you may need some spare disk capacity to provide room for the compaction to occur. You may require as much as the size of the largest PST file, plus approximately 5% of its size.

Possible values:

- true
- false (default)

ConcurrentMigrations

Optional. Specifies the maximum number of concurrent PST migrations. This setting takes effect only if MigrationMode is set to Process.

Possible values:

- An integer in the range 1 to 25. The default is 10.

DeletePST

Optional. Controls whether the PST file is deleted after the successful migration of its contents.

Possible values:

- true
- false (default)

IncludeDeletedItems

Optional. Controls whether the PST Deleted Items folder is migrated.

Possible values:

- true
- false (default)

MailboxFolder

Optional. Identifies the top-level mailbox folder in which Policy Manager places shortcuts to migrated items. If the folder does not exist, Policy Manager creates it. Beneath this folder, PST Migrator duplicates the original folder structure and places shortcuts in the appropriate folders.

If not specified in either the [PST] or [PSTDefaults] sections, the original folder structure is recreated at the top level of the mailbox.

Possible values:

- A folder name. For example, PST items.

MergePSTFolders

Optional. Controls the placement of migrated folders in the target mailbox. When set to true, migrating more than one PST file for the same user causes Policy Manager to merge the identically-named folders.

When set to false, Policy Manager appends a number to the folder names, if necessary, and thereby keeps the folders separate. For example, if two folders at the same level are called "MyFolder", Policy Manager creates "MyFolder" and "MyFolder 1".

Possible values:

- true (default)
- false

Examples:

If MergePSTFolders is set to false and you migrate three PST files that have the display name "Personal Folders", and all contain top-level folders "Inbox" and "Sent Items", then you get a structure like this:

```
PST Migration (specified by MailboxFolder)
  Personal Folders
    Inbox
    Sent Items
  Personal Folders 1
    Inbox
    Sent Items
  Personal Folders 2
    Inbox
    Sent Items
```

MigrationMode

Mandatory. Specifies the modes in which to run.

The options are as follows.

- **Report mode.** Policy Manager checks each listed PST file to determine whether it is possible to migrate the file contents.
Policy Manager creates a new initialization file that shows any problems with the listed PST files, such as files that are inaccessible or password-protected. The new initialization file has the same name as the original, with a number added to make it unique. For example, if the original script was called `PSTMigration.ini` then the new script would be called `PSTMigration_1.ini`. Policy Manager also creates a log file with the same name as the original initialization file and a file type of `.log`. For example, if the original script was called `PSTMigration.ini` then the log would be called `PSTMigration.log`.
- **Process mode.** Policy Manager processes PST files and migrates the contents to the appropriate archives. Policy Manager migrates the file contents and writes a log file with the same name as the initialization file and a file type of `.log`.
If any PST files fail the migration process, Policy Manager writes a new initialization file with which you can process the failed files. Those files that were successfully processed are commented out in the new initialization file.

Possible values:

- Report
- Process

PSTLanguage

Mandatory for Outlook 97 to Outlook 2002 PST files. Not required for Outlook 2003 or later PST files. Specifies the Windows codepage that was used when the PSTs were created. You must specify the language here, in the [PSTdefaults] section, or, for individual PST files, in the [PST] section.

Note the following if the language used was not Western European:

- If the wrong codepage is used, limitations in Exchange Server mean that the folder names may be corrupted. However, there will be no problems with items within the folders.
- If a folder name is corrupted, you may experience the following problems:
 - The corrupt folder name is used if a user ever chooses to restore an item to its original folder.
 - A user who wants to search for an item, and who enters the original location, must enter the corrupt folder name.

To avoid these problems, specify the language that was used when the PSTs were created.

- The language that you specify here must be available on the Storage Service computer that archives the contents of the PST files.

Possible values:

- Arabic
- Baltic
- Central European
- Cyrillic
- Greek
- Hebrew
- Japanese
- Korean
- Simplified Chinese
- Thai
- Traditional Chinese
- Turkish
- Vietnamese
- Western European (default)

ServerComputerName

Optional. Identifies the computer that is running the Storage Service. If you omit `ServerComputerName`, Policy Manager uses the name of the computer on which it is running.

Possible values:

- A computer identification, which can be its LanMan name, DNS name, or IP address.

Examples:

- LanMan: SERVER2
- DNS: server2.Veritas.com
- IP address: 18.94.12.3

SetPSTHidden

Optional. Controls whether the PST file is set as hidden after successful migration of its contents. If you have set your desktop so that it does not show hidden files, this hides PST files that you have migrated successfully. This option is provided for compatibility with the PST Migrator wizard and is not likely to be used in scripted migrations.

Possible values:

- true
- false (default)

SetPSTReadOnly

Optional. Controls whether the PST file is set to be read-only after the successful migration of its contents. This prevents users from opening the files with Outlook.

Possible values:

- true
- false (default)

ShortcutMode

Optional. Defines the PST migration mode, which determines how Policy Manager treats the contents of the PST at the end of the migration.

Possible values:

- `PSTShortcuts` (default). Create shortcuts to the migrated items and leave the shortcuts in the PST files.

- MailboxShortcuts. Create shortcuts to the migrated items and put the shortcuts into the designated Exchange mailbox. Also copies to the mailbox any items that were excluded from archiving.
- NoShortcuts. Do not create any shortcuts to migrated items. Any items that were excluded from archiving remain in the PST files.

[PST] section in the Policy Manager initialization file

Include this section if you want to migrate the contents of PST files to Enterprise Vault.

The settings you provide in this section override any default settings that you may have defined in the [PSTdefaults] section.

ArchiveName

Optional. Specifies the name or archive ID of the archive to which Policy Manager migrates the items in the PST files.

Notes:

- You can make Policy Manager automatically determine the correct archive to use, in which case you do not need to specify ArchiveName.
- Policy Manager uses the first archive that has a matching name. If you have archives with duplicate names, the result may not be what you want. To avoid this problem, use the archive ID, which you can copy from the Advanced tab of the archive's properties in the Administration Console.

Possible values:

- The name of the archive to process
- The archive ID of the archive to process

ArchiveNonExpiredCallItems

Optional. Controls whether Policy Manager migrates the unexpired calendar items. If you choose to migrate unexpired calendar items, users must restore the items before they can modify them.

Possible values:

- True
- False (default)

CancelMbxAutoArchive

Optional. Controls whether Policy Manager turns off Outlook AutoArchiving for all the folders in the target mailboxes. This stops Outlook from automatically archiving items to PST files.

- true
- false (default)

CompactPST

Optional. Controls whether the PST file is compacted after successful migration of its contents.

If you intend to use this PST compaction feature at the end of migrations, you may need some spare disk capacity to provide room for the compaction to take place. This capacity is typically the size of the largest PST file plus approximately 5% of that size.

Possible values:

- true
- false (default)

DeletePST

Optional. Controls whether the PST file is deleted after the successful migration of its contents.

Possible values:

- true
- false (default)

DoNotProcess

Optional. Indicates whether Policy Manager is to ignore this file when it processes PST files. In report mode, Policy Manager ignores this setting and checks the status of every PST file listed.

In the new initialization file that Policy Manager creates after a report mode run, [PST] sections that have caused errors contain the entry `DoNotProcess = True`.

Possible values:

- true
- false (default)

FileName

Optional. Specifies the path to the PST file that you want to process.

Examples:

```
\\central\share\test1.pst  
e:\PSTfiles\test2.pst
```

IncludeDeletedItems

Optional. Controls whether the PST Deleted Items folder is migrated.

Possible values:

- true
- false (default)

JobStatus

Optional. Do not use. Policy Manager inserts JobStatus when you run in process mode. JobStatus indicates whether the file was successfully processed.

Possible values:

- Processed. The file has been successfully processed. Its [PST] section is commented out to prevent reprocessing.
- Unprocessed. Policy Manager cannot begin processing this file.
- Incomplete. Policy Manager was processing this file when a failure occurred that stopped all processing, such as a power cut.
- Partially_Processed. Some items in the PST file cannot be processed. All these items have been placed in a folder that is called PST Migration Failed Items in the PST file. Policy Manager cannot migrate these items.
- Failed. The file cannot be processed for some reason. For example, the Storage Service may not be running, or the user may have opened the file.

MailboxDN

Optional. Specifies the distinguished name of the mailbox in which to place shortcuts to the items that have been migrated. The distinguished name value required is the legacyExchangeDN property for the mailbox in Active Directory.

The easiest way to determine a number of MailboxDN values is to run the Exchange Mailbox Task in report mode. For instructions on how to use report mode to test archiving, see the Administration Console help file. The output file then contains the MailboxDN of each mailbox on that Exchange Server computer.

Possible values:

- A distinguished name, such as the following:

```
/o=acme/ou=developer/cn=Recipients/cn=smithj
```

MailboxFolder

Optional. Identifies the top-level mailbox folder in which Policy Manager places shortcuts to migrated items. If the folder does not exist, Policy Manager creates it. Beneath this folder, PST Migrator duplicates the original folder structure and places shortcuts in the appropriate folders.

If not specified in either the [PST] or [PSTDefaults] sections, the original folder structure is recreated at the top level of the mailbox.

Possible values:

- A folder name. For example, PST items.

MergePSTFolders

Optional. Controls the placement of migrated folders in the target mailbox. When set to true, migrating more than one PST file for the same user causes Policy Manager to merge the identically-named folders.

When set to false, Policy Manager appends a number to the folder names, if necessary, and thereby keeps the folders separate. For example, if two folders at the same level are called "MyFolder", Policy Manager creates "MyFolder" and "MyFolder 1".

Possible values:

- true (default)
- false

Examples:

If MergePSTFolders is set to false and you migrate three PST files that have the display name "Personal Folders", and all contain top-level folders "Inbox" and "Sent Items", then you get a structure like the following:

```
PST Migration (specified by MailboxFolder)
```

```
Personal Folders
  Inbox
  Sent Items
Personal Folders 1
  Inbox
  Sent Items
Personal Folders 2
```

Inbox
 Sent Items

PSTLanguage

Mandatory for Outlook 97 to Outlook 2002 PST files. Not required for Outlook 2003 or later PST files. Specifies the Windows codepage that was used when the PSTs were created. You must specify the language here, in the [PSTdefaults] section, or, for individual PST files, in the [PST] section.

Note the following if the language used was not Western European:

- If the wrong codepage is used, limitations in Exchange Server mean that the folder names may be corrupted. However, there are no problems with items within the folders.
- If a folder name is corrupted, you may experience the following problems:
 - The corrupt folder name is used if a user ever chooses to restore an item to its original folder.
 - A user who wants to search for an item, and who enters the original location, must enter the corrupt folder name.

To avoid these problems, specify the language that was used when the PSTs were created.

- The language that you specify here must be available on the Storage Service computer that archives the contents of the PST files.

Possible values:

- | | |
|--------------------|------------------------------|
| ■ Arabic | ■ Korean |
| ■ Baltic | ■ Simplified Chinese |
| ■ Central European | ■ Thai |
| ■ Cyrillic | ■ Traditional Chinese |
| ■ Greek | ■ Turkish |
| ■ Hebrew | ■ Vietnamese |
| ■ Japanese | ■ Western European (default) |

RetentionCategory

Optional. Specifies the name or ID of the retention category to apply to the migrated PST items.

Although RetentionCategory is optional, Policy Manager must be able to obtain a retention category from somewhere.

Policy Manager takes the first retention category it finds in the following:

- The file's RetentionCategory setting in the [PST] section.

- If MailboxDN is specified in the [PST] section, the default retention category for that mailbox.
- If ArchiveName is specified in the [PST] section, the default retention category for the mailbox that is associated with that archive.

Possible values:

- A retention category name
- A retention category ID

ServerComputerName

Optional. Identifies the computer that is running the Storage Service. If you omit ServerComputerName, Policy Manager uses the name of the computer on which it is running.

Possible values:

A computer identification, which can be its LanMan name, DNS name, or IP address.

Examples:

- LanMan: SERVER2
- DNS: server2.Veritas.com
- IP address: 18.94.12.3

ShortcutMode

Optional. Defines the PST migration mode, which determines how Policy Manager treats the contents of the PST at the end of the migration.

Possible values:

PSTShortcuts (default)	Create shortcuts to the migrated items and leave the shortcuts in the PST files.
MailboxShortcuts	Create shortcuts to the migrated items and put the shortcuts into the designated Exchange mailbox. Also copies to the mailbox any items that were excluded from archiving.
NoShortcuts	Do not create any shortcuts to migrated items. Any items that were excluded from archiving remain in the PST files.

SetPSTHidden

Optional. Controls whether the PST file is set as hidden after successful migration of its contents. If you have set your desktop so that it does not show hidden files, this hides the PST files that you have migrated successfully. This option is provided

for compatibility with the PST Migrator wizard and is not likely to be used in scripted migrations.

Possible values:

- true
- false (default)

SetPSTReadOnly

Optional. Controls whether the PST file is set to be read-only after the successful migration of its contents. This prevents users from opening the files with Outlook.

Possible values:

- true
- false (default)

[PSTcheckpoint] section in the Policy Manager initialization file

Do not include this section, which Policy Manager generates automatically.

Created

Specifies the creation date and time of the new initialization file generated by Policy Manager.

Generation

Provides a number that indicates the restart sequence number. This number is incremented each time you run the initialization file. It is also appended to the name of the initialization file to make the name of the new initialization file.

For example, suppose that your original initialization file is called `migrate-these.ini`. If you run Policy Manager with this file, you produce a new file that is called `migrate-these_1.ini` and that contains details of any problems. You can fix the problems that are indicated in this new file and then run it as before.

Source

Specifies the path and file name of the original Policy Manager initialization file.

PSTFailedCount

Shows the total number of PST files that are listed in this initialization file and that cannot be migrated. Each of these migrated files also has a JobStatus entry of Failed.

PSTIncompleteCount

Generated by a process mode run. Shows the number of PST files that were being processed when Policy Manager was interrupted. This number is never more than one.

Each of these migrated files also has a JobStatus entry of Incomplete.

PSTNotReadyCount

Generated by a report mode run. A problem with this PST file has prevented processing. Policy Manager has added a `DONOTPROCESS = TRUE` line to the [PST] section.

PSTPartialCount

Generated by a process mode run. Shows the number of PSTs that contain one or more items that cannot be migrated. All these items have been placed in a folder that is called PST Migration Failed Items in the PST file.

Each of these migrated files also has a JobStatus entry of Partially_Processed.

PSTProcessedCount

Generated by a process mode run. Shows the number of PST files that were successfully migrated on the previous run of the script. These files are still listed in the restart script, but their sections are commented out.

Each of these migrated files also has a JobStatus entry of Processed.

PSTUnprocessedCount

Generated by a process mode run. Shows the number of PST files that were listed in this file and that were ignored in the last run.

Each of these migrated files also has a JobStatus entry of Unprocessed.

PSTWarningCount

Generated by a report mode run. Shows the number of marked PST files whose marked settings are being overridden in the initialization file. You can find these files by searching for "Report_Status: Warning".

[NSFDefaults] section in the Policy Manager initialization file

This section is mandatory when you use Policy Manager to migrate the contents of NSF files to Enterprise Vault.

Use this section to specify the default settings that apply to NSF migrations. You can override these default settings for individual NSF files in the [NSF] section of the initialization file.

See “[NSF] section in the Policy Manager initialization file” on page 229.

If you do not specify a value for an optional keyname in the [NSFDefaults] section, Policy Manager uses the value that is marked as "default" as the default setting.

ArchiveNonExpiredCallItems

Optional. Controls whether Policy Manager migrates the unexpired calendar items that are contained in the NSF files. If you choose to migrate unexpired calendar items, users must restore the items before they can modify them.

Possible values:

- True
- False (default)

CompactNSF

Optional. Controls whether the NSF files are compacted after successful migration.

Possible values:

- True (default)
- False

ConcurrentMigrations

Optional. Sets the maximum number of concurrent NSF migrations. This setting takes effect only when MigrationMode is set to Process.

Possible values:

- An integer in the range 1 (default) to 5

DeleteNSF

Optional. Controls whether the NSF files are deleted after successful migration.

Possible values:

- True
- False (default)

IgnoreInsufficientMailFileAccess

Optional. By default, EVP does not process an NSF file if the Domino archiving user does not have sufficient access set the ACL of the corresponding mail file. Set this keyname to True to override this default behavior.

Possible values:

- True

- False (default)

IgnoreNoManagerAccess

Optional. By default, EVPM does not process an NSF file if the Domino archiving user does not have manager access set in the ACL of the corresponding mail file. Set this keyname to True to override this default behavior.

Possible values:

- True
- False (default)

IgnoreNonExistentMailFile

Optional. By default, EVPM does not migrate the contents of NSF files whose associated mail file is not available. Set this keyname to True to override this default behavior.

Possible values:

- True
- False (default)

IgnoreNonStandardTemplate

Optional. By default, EVPM does not process an NSF file that is based on a non-standard template. The list of standard templates is determined by a registry string value which is called `DominoMailTemplates` under the following registry key on all the storage servers:

```
HKEY_LOCAL_MACHINE
  \SOFTWARE
    \Wow6432Node
      \KVS
        \Enterprise Vault
          \Agents
```

Set this keyname to True to override this default behavior and migrate the contents of NSF files that are based on non-standard templates.

Possible values:

- True
- False (default)

IncludeTrash

Optional. Controls whether Policy Manager migrates the deleted items from the Trash folders in the NSF files.

Possible values:

- True
- False (default)

MailFileFolder

Optional. Sets the name of the migration target folder. Policy Manager creates this folder beneath the **Folders** view in each user's mail file, if it does not exist already. Policy Manager then places shortcuts and migrated content in this folder.

Possible values:

- A folder name. For example, **NSF items**. If you do not specify a folder name, Policy Manager uses the default name **Notes Archive**.

MergeNSFFolders

Optional. For users who have multiple NSF files, MergeNSFFolders controls whether the NSF files' folder structures are merged or kept separate in the users' mail files.

Possible values:

- True (default). Merge the folder structures that are contained in multiple NSF files. For example, two NSF files that belong to one user, both contain a folder called Personal. Policy Manager places the shortcuts to the contents of these folders in a merged Personal folder in the user's mail file.
- False. Keep separate the folder structures from multiple NSF files. In the user's mail file, a new folder is created for each NSF file, and the shortcuts to its contents are placed in the folders.

MigrationMode

Mandatory. Controls whether Policy Manager runs in report mode or in process mode.

Possible values:

- Report. Policy Manager checks each NSF file listed in the [NSF] sections of the initialization file, to determine whether it can migrate the file's contents. Policy Manager creates a new initialization file, which contains a count of all the files that are not ready for migration. In the new initialization file, any NSF file which cannot be migrated has the entry DoNotProcess=True added to its [NSF] section. This setting prevents Policy Manager from attempting to process the file when it is next run in process mode.

The new initialization file has the same name as the original, with a number appended to make it unique. For example, if the original file was called `NSFMigration.ini`, the new file is called `NSFMigration_1.ini`.

- Process. Policy Manager migrates items from the NSF files that are listed in the [NSF] section, and generates summary and detailed reports. Policy Manager also writes a new initialization file. You can use the new file to migrate any failed files when you have corrected the problems that prevented their migration. Each NSF file has a JobStatus entry added to its [NSF] section of the new initialization file. For example, the files that were successfully migrated have JobStatus=Processed added to the [NSF] section. Policy Manager does not attempt to migrate these files again when you use the new initialization file for the next migration run.

The new initialization file has the same name as the original, with a number appended to make it unique. For example, if the original file was called `NSFMigration.ini`, the new file is called `NSFMigration_1.ini`.

RetentionCategory

Mandatory. Specifies the name of the default retention category that is applied to items during migration.

Possible values:

- A retention category name
- A retention category ID

SetNSFHidden

Optional. Controls whether Policy Manager sets the hidden attribute on NSF files after successful migration. This option is provided for compatibility with the NSF migrator wizard and is not likely to be used in scripted migrations.

Possible values:

- True
- False (default)

SetNSFReadOnly

Optional. Controls whether Policy Manager sets the read-only attribute on NSF files after successful migration. This setting prevents users from adding new items to the NSF files after migration.

Possible values:

- True
- False (default)

ShortcutMode

Optional. Controls what Policy Manager does with the contents of the NSF files after migration.

Possible values:

- MailFileShortcuts (default). Creates shortcuts to the migrated items and puts them in the users' mail files.
- NSFShortcuts. Creates shortcuts to the migrated items and leaves the shortcuts in the NSF files.
- NoShortcuts. Does not create any shortcuts to migrated items. Any items that were excluded from archiving remain in the NSF files.

[NSF] section in the Policy Manager initialization file

The initialization file must contain one [NSF] section for each NSF file you migrate. Each [NSF] section must contain at least a FileName setting to specify the name and location of the NSF file. You can also make further migration settings in the [NSF] section to override the default settings that are specified in the [NSFDefaults] section.

See “[NSFDefaults] section in the Policy Manager initialization file” on page 224.

ArchiveName

Optional. Specifies the name or the ID of the archive to which Policy Manager migrates the items from the current the NSF file.

Note: In the [NSF] section, you can set either the ArchiveName or the UserCN. You cannot set both. See the details for the UserCN setting.

This keyname is optional because Policy Manager can automatically match archives to NSF files. However, it always uses the first archive that has a matching name. If there are archives with duplicate names, items can be migrated to the wrong archives. To avoid this issue, use ArchiveName to specify the ID of an archive for each NSF file.

You can find the ID of an archive on the **Advanced** tab of the archive's properties page in the administration console.

Possible values:

- The ID of the target archive
- The name of the target archive

ArchiveNonExpiredCallItems

Optional. Controls whether Policy Manager migrates unexpired calendar items from the current NSF file. If you choose to migrate unexpired calendar items, users must restore the items before they can modify them.

Possible values:

- True
- False

CompactNSF

Optional. Controls whether the current NSF file is compacted after successful migration.

Possible values:

- True
- False

DeleteNSF

Optional. Controls whether the current NSF file is deleted after successful migration.

Possible values:

- True
- False

DoNotProcess

Optional. When Policy Manager runs in report mode (MigrationMode=Report), it writes a new initialization file. In the new file, it sets DoNotProcess to True for any NSF file on which it encounters errors. This setting prevents Policy Manager from processing the NSF file when you run it again in process mode (MigrationMode=Process), using the new initialization file.

Policy Manager ignores this setting when it runs in report mode.

Possible values:

- True
- False (default)

FileName

Mandatory. Specifies the path and the file name of each NSF file.

Note: You should use UNC paths to specify the locations of the NSF files. The NSF migrator server that processes the NSF files might be on a different computer from the one on which you run EVPM. Additionally, the NSF migrator server might run under a different user context from the one under which you run EVPM. In both these cases, only full UNC paths provide a reliable way for the NSF migrator server to access the files.

Examples:

- `\\Server1\home\JohnDoe\quarter1.nsf`
- `E:\data\backup.nsf`

IgnoreInsufficientMailFileAccess

Optional. By default, EVPM does not process an NSF file if the Domino archiving user does not have sufficient access set the ACL of the corresponding mail file. Set this keyname to True to override this default behavior for the current NSF file.

Possible values:

- True
- False (default)

IgnoreNoManagerAccess

Optional. By default, EVPM does not process an NSF file if the Domino archiving user does not have manager access set in the ACL of the corresponding mail file. Set this keyname to True to override this default behavior for the current NSF file.

Possible values:

- True
- False (default)

IgnoreNonExistentMailFile

Optional. By default, EVPM does not migrate the contents of NSF files whose associated mail file is not available. Set this keyname to True to override this default behavior for the current NSF file.

Possible values:

- True
- False (default)

IgnoreNonStandardTemplate

Optional. By default, EVPM does not process an NSF file that is based on a non-standard template. The list of standard templates is determined by a registry string value which is called DominoMailTemplates under the following registry key on all the storage servers:

```
HKEY_LOCAL_MACHINE
  \SOFTWARE
    \Wow6432Node
      \KVS
        \Enterprise Vault
          \Agents
```

Set this keyname to True to override this default behavior and migrate the contents of the current NSF files if it is not based on non-standard templates.

Possible values:

- True
- False (default)

IncludeTrash

Optional. Controls whether Policy Manager migrates the deleted items from the Trash folder in the current NSF file.

Possible values:

- True
- False

JobStatus

Policy Manager writes a JobStatus in each [NSF] section of the new initialization file when it runs in process mode. This value indicates the status of each NSF file after the last process run. See also the details for MigrationMode.

Possible values:

- Failed. The NSF file failed migration.
- Partially_Processed. The NSF file contains items that Policy Manager was unable to migrate.
- Processed. Policy Manager migrated the NSF file successfully.
- Unprocessed. Policy Manager ignored the NSF file.

MailFileFolder

Optional. Sets the name of the migration target folder. Policy Manager creates this folder beneath the **Folders** view in the user's mail file, if it does not exist already. Policy Manager then places shortcuts and migrated content in this folder.

Possible values:

- A folder name. For example, **NSF items**. If you do not specify a folder name, Policy Manager uses the default name that the setting in the [NSFDefaults] section of the initialization file determines.

MergeNSFFolders

Optional. For a user who has multiple NSF files, MergeNSFFolders controls whether the folder structures they contain are merged or kept separate in the user's mail file.

Possible values:

- True. Merge the folder structures that are contained in multiple NSF files. For example, two NSF files that belong to one user, both contain a folder that is called Personal. The shortcuts to the contents of these folders are placed in a merged Personal folder in the user's mail file.
- False. Keep separate the folder structures from multiple NSF files. Beneath the Folders view in the user's mail file, a new folder is created for each NSF file. The shortcuts to the contents of these NSF files are placed in the corresponding folders.

RetentionCategory

Optional. Specifies the name of the retention category that is applied to items from the current NSF file during migration.

Possible values:

- A retention category name
- A retention category ID

SetNSFHidden

Optional. Controls whether Policy Manager sets the hidden attribute on the current NSF file after successful migration. This option is provided for compatibility with the NSF migrator wizard and is not likely to be used in scripted migrations.

Possible values:

- True
- False

SetNSFReadOnly

Optional. Controls whether Policy Manager sets the read-only attribute on the current NSF file after successful migration. This prevents the user from adding new items to the NSF file after migration.

Possible values:

- True
- False

ShortcutMode

Optional. Controls what Policy Manager does with the contents of the current NSF file after migration.

Possible values:

- MailFileShortcuts. Create shortcuts to the migrated items and put them in the user's mail file. Also copies to the mail file any items that were excluded from archiving.
- NSFShortcuts. Create shortcuts to the migrated items and leave the shortcuts in the NSF file.
- NoShortcuts. Do not create any shortcuts to migrated items. Any items that were excluded from archiving remain in the NSF file.

UserCN

Optional. Specifies the canonical name (CN) of the user whose archive and mail file are the targets for the migration of the current NSF file.

Note: In the [NSF] section, you can set either the ArchiveName or the UserCN. You cannot set both. See also the details for ArchiveName

Possible values:

- Canonical form of the user name in the user's person record. For example for user John Doe/Acme, the canonical name form is cn=John Doe/o=Acme

[NSFCheckPoint] section in the Policy Manager initialization file

Do not include this section, which Policy Manager generates automatically.

Policy Manager creates an [NSFCheckPoint] section when it writes a new initialization file. This section contains information about the new initialization file, and statistics about the run of Policy Manager that created the file.

In some cases the values that Policy Manager writes to the new initialization file depend on the setting of MigrationMode on the [NSFDefaults] section.

Created

Shows the creation date and time of the new initialization file.

Generation

Shows the number that was appended to the name of the new initialization file that Policy Manager generates. This number is incremented each time you run Policy Manager.

Source

Shows the path and the file name of the original initialization file.

NSFFailedCount

This value is generated when Policy Manager runs in Process mode.

Shows the number of NSF files that are listed in this initialization file, but cannot be migrated. For each NSF file that cannot be migrated, Policy Manager writes JobStatus = Failed in the relevant [NSF] section of the new initialization file.

NSFNotReadyCount

This value is generated when Policy Manager runs in Report mode.

Shows the number of NSF files that are listed in this initialization file, but are not ready. For each NSF file that is not ready, Policy Manager writes DoNotProcess = True in the relevant [NSF] section of the new initialization file.

NSFPartialCount

This value is generated when Policy Manager runs in Process mode.

Shows the number of NSF files that are listed in the initialization file, and contain one or more items that cannot be migrated. All these items have been placed in a folder that is called NSF Migration Failed Items in the NSF file. If Policy Manager is interrupted, NSFPartialCount also includes the number of NSF files that were being processed when the interruption took place.

For each NSF file that is partially processed, Policy Manager writes JobStatus = Partially_Processed in the relevant [NSF] section of the new initialization file.

NSFProcessedCount

This value is generated when Policy Manager runs in Process mode.

Shows the number of NSF files that are listed in the initialization file, and were successfully migrated on the previous Policy Manager run. These files are still listed

in the initialization file. However, for each NSF file that is processed, Policy Manager writes `JobStatus = Processed` in the relevant [NSF] section of the new initialization file. This setting prevents Policy Manager from processing the files again when you use the new initialization file.

NSFUnprocessedCount

This value is generated when Policy Manager runs in Process mode.

Shows the number of NSF files that were listed in this file but ignored in the last Policy Manager run. Policy Manager ignores any NSF files with the following settings:

- `JobStatus = Processed`
- `DoNotProcess = True`

For each NSF file that is ignored because `DoNotProcess` is set to `True`, Policy Manager writes `JobStatus = Unprocessed` in the relevant [NSF] section of the new initialization file.

Policy Manager initialization file examples

The following sections provide examples of what to include in an initialization file.

Policy Manager initialization file example 1

This initialization file does the following:

- Enables a mailbox.
- Creates a default archive for the mailbox.
- Applies the system default filter and retention category to the mailbox.

```
[Directory]
DirectoryComputerName= myserver
SiteName = MattSite
[Mailbox]
DistinguishedName = /o=Org1/ou=Admin Group/cn=Recipients/cn=jones
[Folder]
Name = mailboxroot
Enabled = true
```

Policy Manager initialization file example 2

This initialization file does the following:

- Defines a filter that archives all items older than one month.

- Creates a "Personal Archive" folder in all mailboxes and applies the filter to the folder.
- Applies the Personal retention category to the new Personal Archive folder.

```
[Directory]
directorycomputername = myserver
sitename = MattSite
[Filter]
name = filter1
CreateShortcut = true
DeleteOriginal = true
unreadMAIL = false
UseInactivityPeriod = true
InactivityUnits = months
InactivityPeriod = 1
[Mailbox]
distinguishedname = all
[Folder]
name = \personal archive
filtername = filter1
retentioncategory = personal
```

Policy Manager initialization file example 3

This initialization file does the following:

- Defines a filter that archives all read items older than three weeks.
- Creates an archive that is called "shared finance archive", with smithj as the billing account and a description of "Shared archive for all finance users".
- Grants all members of the group enterprise\financeusers write access to the new archive.
- Enables all users in department finance, and sets the system default filter at the root of each mailbox and the Business retention category.
- Creates a folder that is called "finance archive folder" and applies the newly-created archive and the Business retention category to it.

```
[Directory]
directorycomputername = myserver
sitename = MattSite
[Filter]
name = filter1
CreateShortcut = true
```

```
DeleteOriginal = true
unreadMAIL = false
UseInactivityPeriod = true
InactivityUnits = weeks
InactivityPeriod = 3
[Archive]
ArchiveName = Shared Finance Archive
description = Shared archive for all finance users
billingOwner = enterprise\smithj
[ArchivePermissions]
ArchiveName = Shared Finance Archive
GrantAccess = write, enterprise\financeusers
[Mailbox]
ldapquery = (department= finance)
[Folder]
name = mailboxroot
enabled = true
suspended = false
filtername = systemdefault
RetentionCategory = business
[Folder]
name = \Finance Archive Folder
filtername = filter1
retentioncategory = Business
ArchiveName = Shared Finance Archive
```

Policy Manager initialization file example 4: PST migration

This initialization file does the following:

- Defines the default PST migration settings that apply to all the PST files. These settings are not overridden in any of the [PST] sections in the initialization file.
- Lists three PST files whose contents are to be migrated to Enterprise Vault. No destination mailboxes are specified because their owners have opened all the PST files, and so they have been marked.

The default settings make Policy Manager do the following:

- Migrates all the PST file contents to the appropriate mailboxes, including items that are in the Deleted Items folder.
- Place shortcuts to migrated items into the owning mailboxes. The shortcuts all go into a folder that is called "PST Migrations".
- After successful migration, compact PST files and make them read-only.

- **Cancel Outlook AutoArchive.** This stops Outlook from automatically archiving items to PST files.

```
[Directory]
directorycomputername = myserver
sitename = vs1
[PSTdefaults]
;
; Default option settings applicable to all PST migrations
;
PSTLanguage=Western European
servercomputername = myserver.kvsinc.com
MailboxFolder = PST Migrations
MigrationMode = PROCESS
IncludeDeletedItems = true
SetPSTHidden = false
SetPSTReadOnly = true
CompactPST = true
DeletePST = false
CancelMbxAutoArchive = true
;
; Individual PST migration settings
;
[PST]
fileName = \\myserver\share\test1.pst
[PST]
fileName = \\myserver\share\test2.pst
[PST]
fileName = \\myserver\share\test3.pst
```

Policy Manager initialization file example 5: NSF migration

The [NSFDefaults] section in this initialization file does the following:

- Turns on process mode
- Allows two concurrent migrations
- Sets Business as the default retention category
- Turns on the migration of Trash items
- Specifies that the read-only attributes on NSF files are set after successful migration

The subsequent [NSF] sections specify the locations and the names of individual NSF files. Some of these settings override the default migration settings.

```
[Directory]
DirectoryComputerName = DominoServer
sitename = EV1

; Default option settings applicable to all NSF migrations

[NSFDefaults]
MigrationMode = Process
ConcurrentMigrations = 2
RetentionCategory = Business
IncludeTrash = True
SetNSFReadOnly = True

; Individual NSF migration settings

[NSF]
FileName = \\FileServer\e$\Users\UserA\Archive.nsf
DeleteNSF = True
IncludeTrash = False

[NSF]
FileName = \\FileServer\e$\Users\UserB\Q1.nsf
ArchiveName = User B/Veritas
SetNSFReadOnly = False

[NSF]
FileName = \\FileServer\e$\Users\UserC\Personal.nsf
UserCN = CN=John Doe/O=Veritas
RetentionCategory = Personal
```

Policy Manager initialization file example 6: folder permissions

This initialization file does the following:

- Applies the initial permissions to a new folder.
- Modifies the existing user permissions on a folder.
- Removes the existing user permissions from a folder.
- Applies some permissions to the public folder.

```
[DIRECTORY]
DIRECTORYCOMPUTERNAME = OURSERVER
SITENAME = CC_Site1
[mailbox]
```

```

DISTINGUISHEDNAME = /O=ACME/OU=DEVELOPER/CN=RECIPIENTS/CN=SUES
;
;-----
; 1. Apply initial permissions to a new folder
;
[Folder]
Name = \New Folder
MailboxDN = /O=ACME/OU=DEVELOPER/CN=RECIPIENTS/CN=SUES
;
; User specified as Mailbox DN
;
ExchangePermissions
=/O=ACME/OU=DEVELOPER/CN=RECIPIENTS/CN=SUES:OWNER
;
; Add additional user specified by GAL user name
;
ExchangePermissions = Charles Parker:Contributor; John Gillespie:
Reviewer
;-----
; 2. Modify existing user permissions on an existing folder
;
[Folder]
Name = \Existing Folder
MailboxDN = /O=ACME/OU=DEVELOPER/CN=RECIPIENTS/CN=SUES
;
; Modify existing user
;
ExchangePermissions = +; John Gillespie:Editor
;-----
; 3. Remove existing user permissions on an existing folder
;
[Folder]
Name = \Existing Folder
MailboxDN = /O=ACME/OU=DEVELOPER/CN=RECIPIENTS/CN=SUES
;
; Remove existing users
;
ExchangePermissions = -; Charles Parker; John Gillespie
;-----;
; 4. Apply permissions to public folder
;
[PUBLICFOLDER]
Name = \Our Public Folder

```

```
ExchangePermissions =Charles Parker:reviewer  
APPLYTOSUBFOLDERS = false
```

About using the Provisioning API to run Policy Manager scripts

The Provisioning API allows application service providers (ASPs) to automatically enable or disable mailboxes for new customers. For example, you could set up a web page that lets users sign up for the site, which in turn automatically enables mailboxes for them.

Provisioning API scripting properties for Policy Manager scripts

The API uses a scriptable object to allow enabling and disabling of mailboxes. You can set the following properties on the object before enabling or disabling a mailbox:

Required properties:

- Directory
- SiteId
- ExchangeServer
- SystemMailbox (this mailbox must exist)

Either of the following properties is required. They are mutually exclusive, so setting one clears the other:

- MailboxDN (This must be the legacyExchangeDN property for the mailbox in Active Directory)
- LDAPQuery (allows enabling and disabling of multiple mailboxes at the same time)

If the following optional properties are not set, the script uses default settings:

- VaultStore
- RetentionCategory
- IndexingService
- Timeout (the time allowed for the script to run before it is aborted)

If you supply this standard set of properties, the code generates a script and runs it.

Methods are available on the object to enable and disable a mailbox. These methods use the settings above to generate a script to enable or disable a mailbox or set of mailboxes matching the DN or LDAP query.

Example provisioning API Policy Manager script

```
\ '
' Enable a mailbox
'
Dim Enabler
Set Enabler = CreateObject("EnterpriseVault.ExchangeArchivePoint")
Enabler.Directory = "MACHINE1"
Enabler.Site = "site1" '(Entry Id or Site Name)
Enabler.ExchangeServer = "DITTO" '(Entry Id or Exchange Name)
Enabler.SystemMailbox = "EnterpriseVault-DITTO"
Enabler.MailboxDN = "/o=Eng2000/ou=First Administrative
Group/cn=Recipients/cn=Bruiser"
Enabler.VaultStore = "VaultStoreMain" '(Entry Id or Vault Store
Name)
Enabler.RetentionCategory = "Business" '(Entry Id or Retention
Category Name)
Enabler.IndexingService = "MACHINE1"
Enabler.Enable

'
' Disable a mailbox
'
Dim Enabler
Set Enabler = CreateObject("EnterpriseVault.ExchangeArchivePoint")
Enabler.Directory = "MACHINE1"
Enabler.Site = "site1" '(Entry Id or Site Name)
Enabler.ExchangeServer = "DITTO" '(Entry Id or Exchange Name)
Enabler.SystemMailbox = "EnterpriseVault-DITTO"
Enabler.MailboxDN = "/o=Eng2000/ou=First Administrative
Group/cn=Recipients/cn=Bruiser"
Enabler.Disable
```

After the script has been run, the read-only properties ReportText and LastScript are available to return information on the script.

Provisioning API Advanced settings for Policy Manager scripts

The basic scripting object covers the simple case in which a user wishes to enable or disable a mailbox using some basic settings. More advanced settings let you apply per-folder settings.

SetScript methods for provisioning API advanced settings for Policy Manager scripts

The SetScript methods let you provide a template as either a text string or a file. The API uses the template and replaces the values in it by a combination of properties set on the object and the values from the array passed into the following methods:

```
SetScriptText(Text, ArryOfParameters)
SetScriptFile(Filename, ArryOfParameters)
```

The SetScript methods allow a custom string or file to be passed in and used as a template. The array of parameters lets you use a list of substitutions on the template, if required.

Sample script for provisioning API advanced settings for Policy Manager

```
Script1.ini
[Directory]
DirectoryComputerName= #DIRECTORY#
SiteName = #SITE#
[Mailbox]
DistinguishedName = #MAILBOX#
[Folder]
Name = mailboxroot
Enabled = #1#
```

The special values #DIRECTORY#, #SITE#, and #MAILBOX# are automatically replaced by the properties Directory, Site, and MailboxDN set on the object.

Table 23-2 Special values

Special value	Object property name
#DIRECTORY#	Directory
#INDEXINGSERVICE#	IndexingService
#LDAPQUERY#	LDAPQuery

Table 23-2 Special values (*continued*)

Special value	Object property name
#MAILBOX#	MailboxDN
#RETENTIONCATEGORY#	RetentionCategory
#SITE#	Site
#VAULTSTORE#	VaultStore

The value #1# is replaced by the first item in the ArrayOfParameters array passed into the SetScriptFile or SetScriptText method. If more items are added to the array, the values #2#, #3#, and so on are replaced.

Example of enabling a mailbox using a script file with provisioning API advanced settings for Policy Manager

```
Dim ArrayOfParameters(0)
ArrayOfParameters(0) = "true"

Dim Enabler
Set Enabler = CreateObject("EnterpriseVault.ExchangeArchivePoint")

Enabler.Directory = "MACHINE1"
Enabler.Site = "site1" '(Entry Id or Site Name)
Enabler.ExchangeServer = "DITTO" '(Entry Id or Exchange Name)
Enabler.SystemMailbox = "EnterpriseVault-DITTO"
Enabler.MailboxDN = "/o=Eng2000/ou=First Administrative
Group/cn=Recipients/cn=Bruiser"
Enabler.SetScriptFile ("C:\MyScripts\Script1.ini", ArrayOfParameters)

Enabler.ExecuteScript ' runs the EVP script against the script1.ini
file after making the substitutions in the strings.
```

Provisioning API Interface methods for Policy Manager scripts

The full set of methods follows.

Disable method for Provisioning API Interface for Policy Manager scripts

The Disable method takes no arguments. The Directory, Siteld, ExchangeServer, SystemMailbox, and MailboxDN/LDAPQuery properties must be set before this method is called.

```
HRESULT Disable()
```

Enable method for Provisioning API Interface for Policy Manager scripts

The Enable method takes no arguments. The Directory, Siteld, ExchangeServer, SystemMailbox, and MailboxDN/LDAPQuery properties must be set before this method is called.

```
HRESULT Enable()
```

ExecuteScript method for Provisioning API Interface for Policy Manager scripts

The ExecuteScript method takes no arguments. Instead, it uses text or a file as specified with the SetScriptFile or SetScriptText method and runs that script. The Directory, Siteld, ExchangeServer, SystemMailbox, and MailboxDN/LDAPQuery properties must be set before this method is called.

```
HRESULT ExecuteScript()
```

SetScriptFile method for Provisioning API Interface for Policy Manager scripts

The SetScriptFile method specifies the file name of a Policy Manager script that you want to run.

```
HRESULT SetScriptFile(BSTR newVal, VARIANT vArrayOfParams)
```

Table 23-3 Arguments on the SetScriptFile method

Argument	Description
newVal	A string containing the file name of the Policy Manager script to run.
VARIANT vArrayOfParams	An array of variants used to perform substitutions.

SetScriptText method for Provisioning API Interface for Policy Manager scripts

The SetScriptText method specifies a Policy Manager script to run.

```
HRESULT SetScriptText(BSTR newVal, VARIANT vArrayOfParams)
```

Table 23-4 Arguments on the SetScriptText method

Argument	Description
newVal	A string containing the Policy Manager script to run.
VARIANT vArrayOfParams	An array of variants used to perform substitutions.

Provisioning API error handling for Policy Manager scripts

When setting object properties, HRESULT errors are returned if the property is invalid. If Policy Manager returns an error when calling EnableScript, DisableScript, or ExecuteScript, you can use the two properties available to help with tracing problems with the Provisioning API.

These properties are as follows:

ReportText Returns the report text from the previous run.

LastScript Returns the script from the previous run.

[Table 23-5](#) describes the standard set of errors that the API returns.

Table 23-5 Provisioning API error codes

Error code	Error type	Message text
0xC004C000	PROV_DIRECTORY_INVALID	The Directory Service name is invalid or the Directory Service is not running.
0xC004C001	PROV_MUST_SET_DIRECTORY_FIRST	The Directory property must be set first.
0xC004C002L	PROV_COULD_NOT_CREATE_DIRECTORYCONNECTION	Could not create the Enterprise Vault Directory Connection object.
0xC004C003	PROV_ENTRYID_INVALID	The Entry Id is not valid.

Table 23-5 Provisioning API error codes (continued)

Error code	Error type	Message text
0xC004C004	PROV_INVALID_TABLE_ID	Invalid table ID.
0xC004C005	PROV_ERROR_INSERTING_PARAMETERS	An error occurred replacing the script parameters.
0xC004C006	PROV_INVALID_ARG_PARAMETER	One of the arguments supplied in the arguments array could not be converted to a string.
0xC004C007	PROV_MUST_SET_SITE_FIRST	The Site property must be set before this property.
0xC004C008	PROV_NAME_INVALID	Invalid property value.
0xC004C009	PROV_INDEXING_SVC_NOT_FOUND	The Indexing Service could not be found.
0xC004C00A	PROV_NOT_ENOUGH_PROPERTIES_SET	The following properties must be set before Enable/Disable can be called: %n%nDirectory, Site, ExchangeServer, SystemMailbox, (MailboxDN or LDAPQuery).
0xC004C00B	PROV_FAILED_CREATE_STDIN_PIPE	Failed to create a StdIn pipe.
0xC004C00C	PROV_FAILED_CREATE_STDOUTERR_PIPE	Failed to create the StdOut pipe.
0xC004C00D	PROV_FAILED_DUPLICATE_HANDLE	Failed to duplicate the std handle.
0xC004C00E	PROV_FAILED_CLOSE_TEMP_HANDLE	Failed to close the temporary handle.
0xC004C00F	PROV_NO_PASSWORD_FOR_USER	The password for the Logon details was not set.
0xC004C010	PROV_CREATE_PROCESS_FAILED	Failed to create the policy manager process.
0xC004C011	PROV_CREATE_PROCESS_AS_USER_FAILED	Failed to create the policy manager process under the specified account.

Table 23-5 Provisioning API error codes (*continued*)

Error code	Error type	Message text
0xC004C012	PROV_LOGON_USER_FAILED	Could not log the user on for the policy manager process.
0xC004C013	PROV_WAIT_SINGLE_OBJECT_FAILED	Failed to wait for the process to complete.
0xC004C014	PROV_GETEXITPROCESS_FAILED	Could not get the exit code from the policy manager process.
0xC004C015	PROV_FAILED_GET_TEMP_PATH	Could not get the temp file path.
0xC004C016	PROV_FAILED_GET_TEMP_FILE_NAME	Could not get the temp file name.
0xC004C017	PROV_FAILED_CREATE_INI_FILE	Could not create the provisioning initialization file.
0xC004C018	PROV_WRITE_WRITE_INI_FILE	Could not write the provisioning initialization file.
0xC004C019	PROV_FAILED_CLOSE_INI_FILE	Could not close the provisioning initialization file.
0xC004C01A	PROV_FAILED_COCREATE_POLICYINVOKER	Failed to connect to the Admin Service.
0xC004C01B	PROV_PARAMS_NOT_ARRAY	The second argument must be an array.
0xC004C01C	PROV_SCRIPT_FILE_NOT_FOUND	The Script file could not be found.
0xC004C01D	PROV_INPUT_FILE_NOT_UNICODE	Script file is not unicode.
0xC004C01E	PROV_FAILED_OPEN_REGISTRY	Could not open the Enterprise Vault Registry Key.
0xC004C01F	PROV_FAILED_READ_REGISTRY	Could not read the Installation directory from the registry.
0xC004C020	PROV_FAILED_EXECUTE	The script returned errors, check the report for details.
0xC004C021	PROV_SCRIPT_TIMED_OUT	The script timed out.

Table 23-5 Provisioning API error codes (*continued*)

Error code	Error type	Message text
0xC004C022	PROV_FAILED_READ_LOGON_DETAILS	Failed to read the Logon Details.

ResetEVClient

This chapter includes the following topics:

- [About ResetEVClient](#)
- [ResetEVClient syntax](#)

About ResetEVClient

The ResetEVClient utility fixes a number of problems with the Enterprise Vault Outlook Add-In. To do this, the utility does the following:

- Deletes the Outlook data files `frmcache.dat`, and `frmdata64.dat`. The following table describes the function of these files.

`frmcache.dat` Stores the forms for the 32-bit version of Outlook 2010 and later.

`frmdata64.dat` Stores the forms for the 64-bit version of Outlook 2010 and later.

The utility cannot delete these files while Outlook is running.

- Empties the user's Temporary Internet Files folder. Users who cannot view archived items with any of the Enterprise Vault web applications may find that emptying this folder fixes their problem.
- Reregisters the Enterprise Vault client DLLs `desktopclientcache.dll` and `valkyrie.dll`.
- Adds Virtual Vault information to `mapisvc.inf`. If `mapisvc.inf` does not exist, ResetEVClient creates it.
- Checks the registry for the list of add-ins that Outlook has flagged for disabling. If this list includes the Enterprise Vault client DLL, `valkyrie.dll`, ResetEVClient removes it from the list.

ResetEVClient is an exception to the rule that you must always run Enterprise Vault command-line utilities with Administrator privileges. For ResetEVClient to do the following, you must start it as the user who has the problems with the Outlook Add-In:

- Delete the correct `.dat` files
- Empty the user's Temporary Internet Files folder

If necessary, ResetEVClient then prompts for the name and password of an account with Administrator privileges before it performs the remaining actions. If the current user has Administrator privileges or the computer does not have User Account Control (UAC) enabled, ResetEVClient does not prompt for these credentials.

ResetEVClient syntax

ResetEVClient

Vault Store Usage Reporter

This chapter includes the following topics:

- [About Vault Store Usage Reporter](#)
- [Starting Vault Store Usage Reporter](#)
- [Setting up a shortcut link to Vault Store Usage Reporter](#)
- [Understanding the usage summary from Vault Store Usage Reporter](#)
- [Checking that the IIS authentication method is correctly set for Vault Store Usage Reporter](#)

About Vault Store Usage Reporter

Vault Store Usage Reporter is a browser-based application that lets you obtain reports on current vault store usage. For a selected vault store, you can determine usage by archive or billing account.

You can use your web browser to view the reports or download them as tab-separated value files, suitable for use in your own analysis tools. Note that the reports may take some time to generate, depending on the size of the vault stores and the performance of your system.

Starting Vault Store Usage Reporter

You can start Vault Store Usage Reporter from either a web browser or the Enterprise Vault Administration Console.

To start Vault Store Usage Reporter from a web browser

- 1 Log on as an administrator of Enterprise Vault.
If you want to see billing account details, the account you use must also have permissions within the Windows domain.
- 2 Open your web browser.
- 3 Enter the Vault Store Usage Reporter address like this:

```
http://server/EnterpriseVault/usage.asp
```

For example:

```
http://vaultserver.company.com/EnterpriseVault/usage.asp
```

To start Vault Store Usage Reporter from the Administration Console

- ◆ In the left pane of the Administration Console, right-click the **Vault Store Groups** container or a vault store and then click **Reporting**.

Note: If you have configured Enterprise Vault Reporting, Vault Store Usage Reporter is only available from the shortcut menu of a vault store.

Setting up a shortcut link to Vault Store Usage Reporter

By adding a Vault Store Usage Reporter link to the left pane of the Administration Console, you can quickly access usage reports from the console.

To set up a shortcut link to Vault Store Usage Reporter

- 1 Open the Administration Console.
- 2 On the **File** menu, click **Add/Remove Snap-in**.
- 3 On the **Standalone** tab of the Add/Remove Snap-in dialog box, click **Add**.
- 4 In the list of available standalone snap-ins, click **Link to Web Address** and then click **Add**.
- 5 In the first page of the Link to Web Address wizard, type the address of Vault Store Usage Reporter, and then click **Next**. The address is as follows:

```
http://server/EnterpriseVault/usage.asp
```

- 6 Type a name for the new link, such as "Usage Reporter", and then click **Finish**.

- 7 Click **Close** to close the Add Standalone Snap-in dialog box.
 - 8 Click **OK** to close the Add/Remove Snap-in dialog box.
- The new link appears in the left pane of the Administration Console.

Understanding the usage summary from Vault Store Usage Reporter

[Table 25-1](#) describes the information that the usage summary provides.

Table 25-1 Columns in the usage report.

Column	Description
Vault Store	Identifies the vault stores. Click the name of a vault store to view more detailed reports on it.
Save Report By	Provides some links with which you can save the reports in tab-separated files. You can choose to sort the data by archive name or billing account.
Active Archives	Shows the number of archives in the vault store that contain archived items.
Total Items	Shows the total number of archived items in each vault store.
Total Size (MB)	Shows the total size before archiving of all the items that are stored in the archive.
Awaiting Backup	Shows the number of archived items in the vault store that have not been backed up. This entry applies only if the vault store is configured to keep safety copies until after backup or replication.
SQL Server	Identifies the SQL Server that hosts the vault store.

The report also provides the following additional information:

- The total number of vault stores
- The total number of active archives in all vault stores
- The total number of items in all vault stores
- The total size of items in all vault stores
- The average size of the archives in the vault store
- The total number of items that are awaiting backup

Checking that the IIS authentication method is correctly set for Vault Store Usage Reporter

If you receive the message "Access Denied" when you try to run Vault Store Usage Reporter, check that the IIS authentication method is correctly set.

To check the authentication method for Vault Store Usage Reporter in IIS

- 1 Open Internet Information Services (IIS) Manager.
- 2 Expand the tree in the left pane until the **EnterpriseVault** virtual directory is visible.
- 3 Click the **EnterpriseVault** virtual directory in the left pane.
- 4 Switch to Content View to display the contents of the **EnterpriseVault** virtual directory.
- 5 For the files `listvaults.asp` and `usage.asp`, perform the following steps in the order listed:
 - Click the required file in Content View.
 - Switch to Features View.
 - In Features View, double-click **Authentication**.
 - Ensure that only **Basic Authentication** is enabled.
If you need to disable an authentication method, right-click it and then select **Disable**.
- 6 After you have checked and amended the authentication method for both files, restart IIS.

Index

A

- Archive points
 - editing 20
- ArchivePoints 14
 - syntax 20
- ArchivePoints utility 20
 - example commands 24
- Audit Viewer 14, 25
 - changing settings 27
 - copying search results 27
 - running a report 25

C

- Centera
 - NTFS to Centera migration 180
- CenteraPing 14, 34
 - syntax 34

D

- Domino Archive Exporter 15, 36
 - example 38
 - syntax 36
- Domino Profile Document Tool 15, 39
 - examples 40
 - syntax 39
- Domino retention folder 41
- Domino retention plan 41
 - defining 42
 - XML file 41
- Domino Retention Plan Tool 15, 41
 - permissions 42
 - syntax 45
- DTrace 15, 47
 - commands 48
 - log 52
 - starting and stopping 48
 - troubleshooting 52
- Duplicate savesets, deleting 61

E

- Enterprise Vault utilities
 - running 16
- Enterprise Vault utilities 14
- EVDominoExchangeMigration 15
- EVDominoExchangeMigration tool 54
 - Binary Tree 56
 - client requirements 55
 - firewall requirements 55
 - limitations 59
 - log files 59
 - other migration software 57
 - Quest Notes Migrator for Exchange 56
 - running 57
 - syntax 58
- EVDominoRetentionPlans.exe 41
- EVDuplicateCleaner
 - configuring 61
 - fixing broken shortcuts after use 64
 - introduction to 61
 - running 63
- EVEARemovalUtility 15
 - about 66
- EVFSASetRightsAndPermissions 15, 73
- EVrights 15, 75
 - syntax 75
- EVservice 15, 78
 - list file format 80
 - syntax 79
- EVSPShortcutManager 81
 - examples 84
 - syntax 82
- EVSVR 15
 - and checkpointing 87, 98–99, 131
 - and possible data loss 121
 - application states 91
 - choosing a Repair operation 122
 - choosing a Verify operation 112
 - commands 89
 - creating an operation file 92
 - Directory report 101–102

EVSVR (*continued*)

- DumpSaveset command 135
- DumpSISPart command 138
- editing an operation file 98
- ExtractSavesets command 141
- Fingerprint database report 102
- GetNativeItem 143
- improving performance when processing CAB collections 146
- interactive mode 134
- introduction 85
- item list files 128, 132
- ListSavesetLocations command 144
- output log file 130–132
- Partition data report 103
- performing operations on CIFS and NTFS partitions 88
- Repair operations 114
- repair procedures 124
- Report operations 100
- reviewing log file messages 131, 145
- running an operation 99
- starting 89
- StorageQueue report 106
- Vault store report 106
- verification levels for an ArchiveObjects Verify operation 111
- Verify operations 107

Extended attributes

- removal 66

F**File System Archiving**

- FSAUtility 157
- migrating and consolidating file servers 158

FSARunNow 15, 148

- examples 152
- running 149
- syntax 149

FSAUndelete 15, 153

- examples 156
- running 154
- syntax 155

FSAUtility 16, 157

- and EMC Celerra/VNX placeholders 159
- deleting orphaned placeholders 174
- list of options 163
- migrating placeholders 169
- moving placeholders 167

FSAUtility (*continued*)

- prerequisites 158
- recalling files 177
- recreating archive points 164
- recreating placeholders 165
- restoring archived files 175
- running 158

N**NTFS to Centera Migration** 16, 180**NTFSCenteraMigrator** utility

- creating jobs 182
- deleting jobs 184
- deleting source files 185
- log files 185
- managing jobs 180

P**Permission Browser** 16, 187

- running 187

Placeholders

- undeletion of 153

Policy Manager 16, 188

- initialization file [Archive] section 193
- initialization file [ArchivePermissions] section 195
- initialization file [Directory] section 192
- initialization file [Filter] section 196
- initialization file [Folder] section 205
- initialization file [Mailbox] section 202
- initialization file [NSF] section 229
- initialization file [NSFCheckPoint] section 234
- initialization file [NSFDefaults] section 224
- initialization file [PST] section 217
- initialization file [PSTcheckpoint] section 223
- initialization file [PSTdefaults] section 212
- initialization file [PublicFolder] section 210
- initialization file example 236–240
- initialization file format 190
- initialization file format syntax 190
- initialization file keynames 191
- initialization file sections 191
- Provisioning API 242
- Provisioning API advanced settings 244
- Provisioning API error handling 247
- Provisioning API example script 243
- Provisioning API interface disable method 246
- Provisioning API interface enable method 246

Policy Manager (*continued*)

- Provisioning API interface ExecuteScript method 246
- Provisioning API interface methods 245
- Provisioning API interface Set ScriptText method 247
- Provisioning API interface SetScriptFile method 246
- Provisioning API sample enabling mailbox script 245
- Provisioning API sample script 244
- Provisioning API scripting properties 242
- Provisioning API SetScript 244
- syntax 189

R

- ResetEVClient 16, 251
- syntax 252

S

- Savesets, deleting duplicates 61
- Security identifiers (SIDs) 187

T

- Troubleshooting
- tracing 51

U

- Undelete placeholders 153
- Utilities 14
- running 16

V

- Vault Store Usage Reporter 16

X

- XML file
- for Domino retention plan 41