

# **Client Commands and Utilities Reference**

## **Dynamic File Services 2.0**

November 11, 2011

## Legal Notices

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the [Novell International Trade Services Web page \(http://www.novell.com/info/exports/\)](http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2009–2011 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc.  
1800 South Novell Place  
Provo, UT 84606  
U.S.A.  
[www.novell.com](http://www.novell.com)

*Online Documentation:* To access the latest online documentation for this and other Novell products, see the [Novell Documentation Web page \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

## Novell Trademarks

For Novell trademarks, see [the Novell Trademark and Service Mark list \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html).

## Third-Party Materials

All third-party trademarks are the property of their respective owners.

This product includes DotNetZip open source software that is developed by the DotNetZip Library open source project. For information, see [DotNetZip Library \(http://dotnetzip.codeplex.com/\)](http://dotnetzip.codeplex.com/) on CodePlex.com.

This product includes Glacial ListView open source software that is developed by Glacial Components Software and the C# ListView open source project. For information, see [C# ListView \(http://www.codeproject.com/KB/list/aa\\_listview.aspx\)](http://www.codeproject.com/KB/list/aa_listview.aspx) on CodeProject.com.

This product includes Hammock open source software that is developed by the Hammock open source project. For information, see [Hammock \(http://sourceforge.net/projects/hammock/\)](http://sourceforge.net/projects/hammock/) on SourceForge.net.

This product includes the Newtonsoft.Json.dll library in the Json.NET open source software that is developed by the Json.NET open source project. For information, see [Json.NET \(http://json.codeplex.com\)](http://json.codeplex.com) on CodePlex.com.

This product includes log4net open source software that is developed as part of the Apache Logging Services open source project. For information, see [log4net \(http://logging.apache.org/log4net/\)](http://logging.apache.org/log4net/) on Apache.org.

This product includes Plossum open source software that is developed by the Plossum open source project. For information, see [Plossum \(http://sourceforge.net/projects/plossum\)](http://sourceforge.net/projects/plossum) on SourceForge.net.

This product includes TweetSharp open source software that is developed by Aplitize and the TweetSharp open source project. For information, see [TweetSharp \(http://nuget.org/List/Packages/TweetSharp\)](http://nuget.org/List/Packages/TweetSharp) on Nuget.org.

This product includes ZedGraph open source software that is developed by the ZedGraph open source project. For information, see [ZedGraph.org \(http://zedgraph.org\)](http://zedgraph.org).

---

# Contents

<b>About This Guide</b>	<b>7</b>
<b>1 Overview of the Dynamic File Services Client Commands and Utilities</b>	<b>9</b>
1.1 Client CLI and Utilities	9
1.2 Command Line Syntax	10
<b>2 What's New for the Dynamic File Services Client CLI and Utilities</b>	<b>11</b>
2.1 What's New for Dynamic File Services 2.0	11
2.1.1 Registration Actions	11
2.1.2 Pair Actions	11
2.1.3 Policy Actions	12
2.1.4 Schedule Actions	14
<b>3 Using Client Commands for Pair and Policy Management</b>	<b>15</b>
3.1 DswCli.exe Syntax Overview	16
3.2 Help Command	20
3.3 Authentication Parameters	20
3.3.1 Setting Authentication Parameters in Commands	21
3.3.2 Setting Authentication Parameters as Environment Variables	23
3.4 Common Parameters	25
3.4.1 Description	25
3.4.2 Direction to Move Files	26
3.4.3 Name	26
3.4.4 Pair ID	26
3.4.5 Policy ID	27
3.4.6 Policy ID List	27
3.4.7 Schedule ID	27
3.4.8 Schedule Type	28
3.5 Common Options	28
3.5.1 Debug	28
3.5.2 Output to a File	28
3.5.3 Silent	28
3.6 System Actions	29
3.6.1 Display Active Directory Computers	29
3.6.2 Display Active Directory Shares	29
3.6.3 Display the File Types Information	29
3.6.4 Display Local Drives	30
3.6.5 Display Network Shares	30
3.6.6 Display Server System Information	30
3.6.7 Display or Set the Logging Level for the Service or Standard Policy	30
3.6.8 Query for File Types or Extensions (for Novell Support Use Only)	32
3.7 Registration Actions	32
3.7.1 Display the Registration Status	33
3.7.2 Register a Product License Key	33
3.8 Pair Actions	33
3.8.1 Pair Parameters	34
3.8.2 Add a Pair	35
3.8.3 Unlink a Pair	38

3.8.4	Add Exclude/Include Paths to a Pair	38
3.8.5	Delete Exclude/Include Paths from a Pair	39
3.8.6	Associate a Pair and Policy	40
3.8.7	Disassociate a Pair and Policy	41
3.8.8	Associate a Review Schedule to a Retention Pair	41
3.8.9	Disassociate a Review Schedule from a Retention Pair	42
3.8.10	List Pairs	42
3.8.11	List Details for a Pair	42
3.8.12	Execute a List of Policies on a Pair	43
3.8.13	Move a List of Files or Folders in a Pair	43
3.9	Policy Actions	45
3.9.1	Policy Parameters	46
3.9.2	Add a Policy	46
3.9.3	Modify a Policy	52
3.9.4	Delete a Policy	53
3.9.5	List All Policies	53
3.9.6	List Details for a Policy	53
3.9.7	Associate a Policy Schedule to a Policy	54
3.9.8	Disassociate a Policy Schedule from a Policy	54
3.9.9	Associate a Policy and a Pair	54
3.9.10	Disassociate a Policy and a Pair	54
3.9.11	List Policies Associated with a Pair	55
3.10	Schedule Actions	55
3.10.1	Schedule Parameters	55
3.10.2	Add a Review Schedule	56
3.10.3	Add a Policy Schedule	59
3.10.4	Delete a Schedule	65
3.10.5	Modify a Review Schedule	65
3.10.6	Modify a Policy Schedule	67
3.10.7	List All Schedules	68
3.10.8	List Details for a Schedule	68
3.10.9	Associate a Review Schedule to a Retention Pair	68
3.10.10	Disassociate a Review Schedule from a Retention Pair	68
3.10.11	Associate a Policy Schedule to a Policy	69
3.10.12	Disassociate a Policy Schedule from a Policy	69
3.11	Using CLI Commands in Scripts	69
3.12	Example Scripts Using CLI Commands	70
3.12.1	Creating a Pair and Policy	70
3.12.2	Running the Policies Manually	71
3.12.3	Deleting the Policies	71
3.12.4	Deleting the Pair Relationship	71

## **4 Dynamic File Services File System Inventory Utility 73**

4.1	DswInventory	73
4.1.1	Description	73
4.1.2	Syntax	74
4.2	Preparing a Command File for a File System Inventory	74
4.2.1	Inventory Path	74
4.2.2	Report Types and Filenames	75
4.2.3	Action List Filename	75
4.2.4	Inventory Options	75
4.3	Example of a General Inventory	77
4.3.1	Create a Command File	77
4.3.2	Run the File System Inventory Utility	77
4.3.3	View the Summary Report	77
4.3.4	View the Detailed Report	78
4.4	Example of a Filtered Inventory	81
4.4.1	Create a Command File	82

4.4.2	Run the File System Inventory Utility . . . . .	82
4.4.3	View the Summary Report . . . . .	82
4.4.4	View the Detailed Report . . . . .	82
4.5	Additional Information . . . . .	84
<b>5</b>	<b>Dynamic File Services Synchronize Pair Utility</b>	<b>85</b>
5.1	DswSyncPair . . . . .	85
5.1.1	Description . . . . .	85
5.1.2	Syntax . . . . .	85
5.1.3	Using the Utility . . . . .	86
5.1.4	Check Action Options . . . . .	86
5.1.5	Control Options . . . . .	87
5.1.6	Report Options . . . . .	87
5.1.7	Examples . . . . .	88
5.2	Additional Information . . . . .	89
<b>6</b>	<b>Dynamic File Services Configuration Dump Utility</b>	<b>91</b>
6.1	DswDump . . . . .	91
6.1.1	Description . . . . .	91
6.1.2	Syntax . . . . .	92
6.1.3	Options . . . . .	92
6.1.4	Examples . . . . .	93
6.2	Config.txt Output . . . . .	93



---

# About This Guide

This guide provides essential information for the client command line interface (CLI) and utilities for Novell Dynamic File Services (DynamicFS) 2.0.

- ♦ [Chapter 1, “Overview of the Dynamic File Services Client Commands and Utilities,”](#) on page 9
- ♦ [Chapter 2, “What’s New for the Dynamic File Services Client CLI and Utilities,”](#) on page 11
- ♦ [Chapter 3, “Using Client Commands for Pair and Policy Management,”](#) on page 15
- ♦ [Chapter 4, “Dynamic File Services File System Inventory Utility,”](#) on page 73
- ♦ [Chapter 5, “Dynamic File Services Synchronize Pair Utility,”](#) on page 85
- ♦ [Chapter 6, “Dynamic File Services Configuration Dump Utility,”](#) on page 91

## Audience

This guide is designed to help storage solutions administrators use commands and utilities to create and manage Dynamic File Services pairs and policies.

## Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

## Documentation Updates

For the most recent version of the *Novell Dynamic File Services 2.0 Client Commands and Utilities Guide*, visit the [Dynamic File Services 2.0 Documentation Web site](http://www.novell.com/documentation/dynamic_file_services/) ([http://www.novell.com/documentation/dynamic\\_file\\_services/](http://www.novell.com/documentation/dynamic_file_services/)).

## Additional Documentation

Additional documentation is available on the [Dynamic File Services 2.0 Documentation Web site](http://www.novell.com/documentation/dynamic_file_services/) ([http://www.novell.com/documentation/dynamic\\_file\\_services/](http://www.novell.com/documentation/dynamic_file_services/)):

- ♦ *Readme*
- ♦ *Installation Guide*
- ♦ *Administration Guide*



---

# 1 Overview of the Dynamic File Services Client Commands and Utilities

Novell Dynamic File Services (DynamicFS) 2.0 provides a client command line interface (CLI) and utilities for the Windows environment. The commands allow you to create and manage pairs and policies. The CLI is a text interface that also allows scripting of the commands instead of using the graphical user interface (GUI) tools.

For information about using the GUI tools to create and manage Dynamic File Services, see the [Dynamic File Services 2.0 Administration Guide](#).

- ♦ [Section 1.1, “Client CLI and Utilities,”](#) on page 9
- ♦ [Section 1.2, “Command Line Syntax,”](#) on page 10

## 1.1 Client CLI and Utilities

The following table describes the commands and utilities that you can use to create and manage Dynamic File Services pairs and policies.

Component	Description
CLI	<p>The DynamicFS CLI application allows you to create and manage pairs and policies on the server by issuing commands in the Windows Command Prompt console. The application runs only when you issue a command.</p> <p>For information, see <a href="#">Chapter 3, “Using Client Commands for Pair and Policy Management,”</a> on page 15.</p> <p>The application is <code>DswCli.exe</code>.</p>
Configuration Dump	<p>The DynamicFS Configuration Dump utility aggregates information about the current DynamicFS server settings for pairs, policies, and logs, and puts it an output file in XML format. This tool is available to help with troubleshooting when working with Novell Support.</p> <p>For information, see <a href="#">Chapter 6, “Dynamic File Services Configuration Dump Utility,”</a> on page 91.</p> <p>The application is <code>DswDump.exe</code>.</p>

Component	Description
File System Inventory	<p>The DynamicFS File System Inventory utility automatically runs a Pair History Scan on a pair each day at 4:00 a.m. by default. It scans the pairs to gather metadata information about the data stored on the primary and secondary locations, such as the file sizes, number of files, and file types.</p> <p>The time and frequency of pair history scanning is configurable. For information, see <a href="#">“Scheduling the Pair History Scan”</a> in the <i>Dynamic File Services 2.0 Administration Guide</i>.</p> <p>The DynamicFS Service controls when the File System Inventory runs. The application is <code>DswInventory.exe</code>.</p>
Synchronize Pair	<p>The DynamicFS Synchronize Pair utility is used to detect duplicate files in the pair structure or to detect folders with attribute or ACL permission differences. It can generate reports in CSV and XML format.</p> <p>The primary and secondary locations are rarely out of synchronization. Such conditions might occur, for example, after recovering files in the two locations from backup media.</p> <p>For information, see <a href="#">Chapter 5, “Dynamic File Services Synchronize Pair Utility,”</a> on page 85.</p> <p>The application is <code>DswSyncPair.exe</code>.</p>

## 1.2 Command Line Syntax

The following notation is used in the Dynamic File Services command line syntax:

Notation	Description
Text without brackets or braces	Command, action, parameter, or option that you must type as shown.
<i>&lt;Italicized text inside angle brackets&gt;</i>	Variable that you must replace with a value.
<i>Italicized text</i>	Values in DynamicFS commands are read as character strings and should be surrounded by quotation marks.
[Text inside square brackets]	Optional parameters or options.
{Text inside braces}	A set of mutually exclusive items separated by a vertical bar ( ). Choose only one.

---

# 2 What's New for the Dynamic File Services Client CLI and Utilities

This section describes the new features and changes for the client command line interface (CLI) commands and utilities in each release of Novell Dynamic File Services.

- ♦ [Section 2.1, "What's New for Dynamic File Services 2.0," on page 11](#)

## 2.1 What's New for Dynamic File Services 2.0

In addition to bug fixes, Novell Dynamic File Services 2.0 provides the following new features and changes for the client commands and utilities since Dynamic File Services 1.6:

- ♦ [Section 2.1.1, "Registration Actions," on page 11](#)
- ♦ [Section 2.1.2, "Pair Actions," on page 11](#)
- ♦ [Section 2.1.3, "Policy Actions," on page 12](#)
- ♦ [Section 2.1.4, "Schedule Actions," on page 14](#)

### 2.1.1 Registration Actions

Registration is required to create multiple pairs and policies on the Dynamic File Services server. The registration action allows you to register a product key and check its status. See [Section 3.7, "Registration Actions," on page 32](#).

### 2.1.2 Pair Actions

Dynamic File Services 2.0 now supports two types of pairs: standard and retention. The following changes and enhancements were made to provide this support in the command line interface:

- ♦ ["Standard Pair" on page 12](#)
- ♦ ["Retention Pair" on page 12](#)
- ♦ ["Upgrade Changes for Existing Pairs" on page 12](#)
- ♦ ["Pair Type Option" on page 12](#)
- ♦ ["Exclude/Include Option" on page 12](#)

## Standard Pair

The standard pair supports the merged view of two paths for users. Only local primary paths are supported for standard pairs. The secondary path can be a local path, a remote share on a network filer (like NetApp or EMC), or a remote share on a supported Windows platform. Standard is the default pair type.

## Retention Pair

The retention pair provides a retention repository on the secondary path. Users see files only on the primary path. Primary path and secondary path for a retention pair can be any combination of a local path, a remote share on a network filer (like NetApp or EMC), or a remote share on a supported Windows platform.

You can perform the following actions on a retention pair:

- ♦ Create a retention pair. See [Section 3.8.2, “Add a Pair,” on page 35](#).
- ♦ Unlink a retention pair. See [Section 3.8.3, “Unlink a Pair,” on page 38](#).

A review schedule indicates when review notifications are sent to designated reviewers. This works only if the Notifications Service is configured, and if reviewers are configured to receive notifications about the retention review events. You can associate one review schedule with a retention pair.

- ♦ Associate a review schedule to a retention pair. See [Section 3.8.8, “Associate a Review Schedule to a Retention Pair,” on page 41](#).
- ♦ Disassociate a review schedule from a retention pair. See [Section 3.8.9, “Disassociate a Review Schedule from a Retention Pair,” on page 42](#).

## Upgrade Changes for Existing Pairs

If you upgrade to version 2.0 and have existing pairs, each pair’s definition is modified to add the pair type “standard” in the pair database.

## Pair Type Option

The `-pairType` option allows you to specify the type of pair to create. See [“-pairType” on page 36](#).

## Exclude/Include Option

The `-exclude` and `-include` options for a pair are not supported for pairs with remote primary paths. Only retention pairs allow remote primary paths. See [Section 3.8.4, “Add Exclude/Include Paths to a Pair,” on page 38](#).

### 2.1.3 Policy Actions

The following changes and enhancements were made to policy actions:

- ♦ [“Policy Schedules” on page 13](#)
- ♦ [“Upgrade Changes for Existing Policies” on page 13](#)
- ♦ [“Renaming a Policy” on page 13](#)

- ♦ [“Policy Action Options” on page 13](#)
- ♦ [“Filter Options” on page 13](#)

## Policy Schedules

Dynamic File Services 2.0 now manages a policy separately from its policy. For information, see [Section 2.1.4, “Schedule Actions,” on page 14](#).

## Upgrade Changes for Existing Policies

Schedules are now managed separately from policies. If you upgrade to version 2.0 and have existing policies, each policy’s schedule is extracted from the policy, and saved as a schedule with the same name as the policy. You can associate one policy schedule with a policy.

## Renaming a Policy

You can now modify a policy name. See [Section 3.9.3, “Modify a Policy,” on page 52](#).

## Policy Action Options

The following changes and enhancements were made for policy actions:

Policy Action Option	Change Description
-execute	This option replaces the <code>-run</code> option. See <a href="#">“Execute a List of Policies on a Pair” on page 43</a> .
-run	This filter option is now an alias for the <code>-execute</code> option. See <a href="#">“Execute a List of Policies on a Pair” on page 43</a> .

## Filter Options

The following changes and enhancements were made for policy filter options:

Filter Option	Change Description
-execute	This option is new for policies. It replaces the <code>-run</code> option. See <a href="#">“Execute a List of Policies on a Pair” on page 43</a> .
-fileType	This filter option was expanded to use the perceived types defined in the <code>Dynamic File Services\DswFileTypes.cfg</code> file in addition to the MIME and perceived types found in the server’s Windows Registry. See <a href="#">“-fileType” on page 50</a> .
-groups	This filter option is new for policy filters. It allows you to move files based on a group identity. See <a href="#">“-groups” on page 51</a> .
-run	This filter option is replaced by the <code>-execute</code> option. It is still supported as an alias. See <a href="#">“Execute a List of Policies on a Pair” on page 43</a> .

## 2.1.4 Schedule Actions

Schedules are now managed separately from policies. For the command line, you can create a schedule for policies and retention reviews. See [Section 3.10, “Schedule Actions,”](#) on page 55.

- ◆ [Add a review schedule](#)
- ◆ [Add a policy schedule](#)
- ◆ [Delete a schedule](#)
- ◆ [Modify a review schedule](#), including its name
- ◆ [Modify a policy schedule](#), including its name
- ◆ [List schedules](#)
- ◆ [List details of a schedule](#)
- ◆ [Associate a review schedule to a retention pair](#)
- ◆ [Disassociate a review schedule from a retention pair](#)
- ◆ [Associate a policy schedule to a policy](#)
- ◆ [Disassociate a policy schedule from a policy](#)

The following enhancements were added for policy schedules:

Filter Option	Change Description
-custom	This filter option is new for policy schedules. See <a href="#">Section 3.10.3, “Add a Policy Schedule,”</a> on page 59.
-quarterly	This filter option is new for policy schedules. See <a href="#">Section 3.10.3, “Add a Policy Schedule,”</a> on page 59.
last	When specifying the schedule dates option, use the “last” option to specify that the policy runs on the last day of every month. The policy runs on day 28, 29, 30, or 31, according to how many days are in the month.

The notification review schedule for retention pairs supports only monthly, quarterly, yearly, and custom frequencies.

---

# 3 Using Client Commands for Pair and Policy Management

The Novell Dynamic File Services (DynamicFS) client command line interface (`DswCli.exe`) provides the ability to manage pairs and policies at the command line or in a script. The Dynamic File Service must be running on the server you want to manage in order to use the utility.

The Dynamic File Services client CLI utility provides the following basic functionality of the GUI interface:

- ◆ Register a product license key
- ◆ Add and remove standard pairs and retention pairs.
- ◆ Add, modify, and remove policies.
- ◆ Add, modify, and remove review schedules and policy schedules.
- ◆ Associate and disassociate a pair with a policy.
- ◆ Associate and disassociate a policy schedule with a policy.
- ◆ Associate and disassociate a review schedule with a retention pair.
- ◆ Execute a list of policies for a pair.
- ◆ Move a list of files or folders for a pair.
- ◆ Set log levels for the Standard Policy log and Service log.
- ◆ Perform system queries.

The utility is found in the `C:\Program Files\Dynamic File Services` directory (or the directory where you installed Dynamic File Services).

See the following sections for command actions and options:

- ◆ [Section 3.1, “DswCli.exe Syntax Overview,” on page 16](#)
- ◆ [Section 3.2, “Help Command,” on page 20](#)
- ◆ [Section 3.3, “Authentication Parameters,” on page 20](#)
- ◆ [Section 3.4, “Common Parameters,” on page 25](#)
- ◆ [Section 3.5, “Common Options,” on page 28](#)
- ◆ [Section 3.6, “System Actions,” on page 29](#)
- ◆ [Section 3.7, “Registration Actions,” on page 32](#)
- ◆ [Section 3.8, “Pair Actions,” on page 33](#)
- ◆ [Section 3.9, “Policy Actions,” on page 45](#)
- ◆ [Section 3.10, “Schedule Actions,” on page 55](#)
- ◆ [Section 3.11, “Using CLI Commands in Scripts,” on page 69](#)
- ◆ [Section 3.12, “Example Scripts Using CLI Commands,” on page 70](#)

## 3.1 DswCli.exe Syntax Overview

### AUTHENTICATION PARAMETERS

#### Authentication Parameters for Each Command

```
[-servername [-port [-username [-password]]]]
```

### PAIR ACTIONS

#### Add a Pair

```
DswCli.exe [authentication_parameters] -PAIR
-ADD
-name=<pairname>
-primarypath=<path>
-secondaryPath=<path>
[-description=<"text">]
[-pairType=<"standard"|"retention">]
If pairType is not specified,the default is "standard".
```

#### Unlink a Pair

```
DswCli.exe [authentication_parameters] -PAIR
-DELETE
-pairid=<pairname|GUID>
```

#### Add Exclude/Include Paths for a Pair

```
DswCli.exe [authentication_parameters] -PAIR
-ADDEXCLUDEINCLUDEPATHS
-pairid=<pairname|GUID>
-flags=<Exclude|Include|None>
-paths=<path1[;path2;path3...]>
```

#### Delete Exclude/Include Paths from a Pair

```
DswCli.exe [authentication_parameters] -PAIR
-DELETEEXCLUDEINCLUDEPATHS
-pairid=<pairname|GUID>
-flags=<Exclude|Include|None>
-paths=<path1[;path2;path3...]>
```

#### Associate a Policy and Pair

```
DswCli.exe [authentication_parameters] -PAIR
-ASSOCIATE
-pairid=<pairname|GUID>
-policyid=<policyname|GUID>
```

#### Disassociate a Policy and Pair

```
DswCli.exe [authentication_parameters] -PAIR
-DISASSOCIATE
-pairid=<pairname|GUID>
-policyid=<policyname|GUID>
```

#### Associate a Review Schedule to a Retention Pair

```
DswCli.exe [authentication_parameters] -PAIR
-ASSOCIATE
-pairid=<pairname|GUID>
-scheduleid=<schedulename|GUID>
```

## Disassociate a Review Schedule from a Retention Pair

```
DswCli.exe [authentication_parameters] -PAIR
-DISASSOCIATE
-pairid=<pairname|GUID>
-scheduleid=<schedulename|GUID>
```

## Execute a List of Policies for a Pair

```
DswCli.exe [authentication_parameters] -PAIR
-EXECUTE
-pairid=<pairname|GUID>
-policyidlist=<<policyname|GUID>[, <policyname|GUID>, ...]>
```

## Move a List of Files or Folders on a Pair

```
DswCli.exe [authentication_parameters] -PAIR
-MOVE
-pairId=<pairname|GUID>
[-folderList="filename"]
[-fileList="filename"]
<-primaryToSecondary | -secondaryToPrimary>
```

## List All Pairs

```
DswCli.exe [authentication_parameters] -PAIR
```

## List Details for a Pair

```
DswCli.exe [authentication_parameters] -PAIR
-DETAIL
-pairid=<pairname|GUID>
```

## List Details for the Review Schedule Associated with a Retention Pair

```
DswCli.exe [authentication_parameters] -PAIR
-SCHEDULE
-DETAIL
-pairid=<pairname|GUID>
```

## POLICY ACTIONS

### Add a Policy

```
DswCli.exe [authentication_parameters] -POLICY
-ADD
-name=<policyname>
[-description=<"text">]
<-primaryToSecondary| -secondaryToPrimary>
[-fileExtension=<SEE FORMAT>]
[-fileOwners=<SEE FORMAT>]
[-groups=<SEE FORMAT>]
[-lastAccessed=<SEE FORMAT>]
[-lastModified=<SEE FORMAT>]
[-fileTypes=<SEE FORMAT>]
[-fileSize=<SEE FORMAT>]
```

## Modify a Policy

```
DswCli.exe [authentication_parameters] -policy
  -modify
  -policyId=<"policyname" | "GUID">
  [-name=<newpolicyname>]
  [-description"text"]
  <-primaryToSecondary|-secondaryToPrimary>
  [-fileExtension=<SEE FORMAT>]
  [-fileOwners=<SEE FORMAT>]
  [-groups=<SEE FORMAT>]
  [-lastAccessed=<SEE FORMAT>]
  [-lastModified=<SEE FORMAT>]
  [-fileTypes=<SEE FORMAT>]
  [-fileSize=<SEE FORMAT>]
```

## Delete a Policy

```
DswCli.exe [authentication_parameters] -POLICY
  -DELETE
  -policyid=<policyname | GUID>
```

## List Details for a Policy

```
DswCli.exe [authentication_parameters] -POLICY
  -DETAIL
  -policyid=<policyname | GUID>
```

## Associate a Policy Schedule to a Policy

```
DswCli.exe [authentication_parameters] -POLICY
  -ASSOCIATE
  -policyid=<policyname | GUID>
  -scheduleid=<schedulename | GUID>
```

## Disassociate a Policy Schedule from a Policy

```
DswCli.exe [authentication_parameters] -POLICY
  -DISASSOCIATE
  -policyid=<policyname | GUID>
  -scheduleid=<schedulename | GUID>
```

## SCHEDULE ACTIONS

### Add a Review Schedule

```
DswCli.exe [authentication_parameters] -SCHEDULE
  -ADD
  -name=<schedulename>
  -scheduleType="review"
  [-description=<"text">]
  [-monthly=<SEE FORMAT>]
  [-quarterly=<SEE FORMAT>]
  [-yearly=<SEE FORMAT>]
  [-custom=<SEE FORMAT>]
```

## Add a Policy Schedule

```
DswCli.exe [authentication_parameters] -SCHEDULE
-ADD
-name=<schedulename>
-scheduleType="policy"
[-description=<"text">]
[-time=<SEE FORMAT>] Note: -time not allowed with -hourly
[-hourly=<SEE FORMAT>]
[-daily=<SEE FORMAT>]
[-weekly=<SEE FORMAT>]
[-monthly=<SEE FORMAT>]
[-quarterly=<SEE FORMAT>]
[-yearly=<SEE FORMAT>]
[-custom=<SEE FORMAT>]
```

## Delete a Review Schedule or a Policy Schedule

```
DswCli.exe [authentication_parameters] -SCHEDULE
-DELETE
-scheduleID=<schedulename|GUID>
```

## List All Schedules

```
DswCli.exe [authentication_parameters] -SCHEDULE
```

## List All Review Schedules

```
DswCli.exe [authentication_parameters] -SCHEDULE
-scheduleType="review"
```

## List All Policy Schedules

```
DswCli.exe [authentication_parameters] -SCHEDULE
-scheduleType="policy"
```

## List Details of a Review Schedule or Policy Schedule

```
DswCli.exe [authentication_parameters] -SCHEDULE
-DETAIL
-scheduleID=<schedulename|GUID>
```

## Modify a Review Schedule

```
DswCli.exe [authentication_parameters] -SCHEDULE
-MODIFY
-scheduleID=<schedulename|GUID>
[-name=<new_schedulename>]
[-description=<"text">]
[-monthly=<SEE FORMAT>]
[-quarterly=<SEE FORMAT>]
[-yearly=<SEE FORMAT>]
[-custom=<SEE FORMAT>]
```

## Modify a Policy Schedule

```
DswCli.exe [authentication_parameters] -SCHEDULE
-MODIFY
-scheduleID=<schedulename|GUID>
[-name=<new_schedulename>]
[-description=<"text">]
[-time=<SEE FORMAT>] Note: -time not allowed with -hourly
[-hourly=<SEE FORMAT>]
[-daily=<SEE FORMAT>]
[-weekly=<SEE FORMAT>]
[-monthly=<SEE FORMAT>]
[-quarterly=<SEE FORMAT>]
[-yearly=<SEE FORMAT>]
[-custom=<SEE FORMAT>]
```

## REGISTRATION ACTIONS

### Register a License Key

```
DswCli.exe [authentication_parameters] -REGISTRATION  
-regFilePath=<"path">
```

### Display the Registration Status

```
DswCli.exe [authentication_parameters] -REGISTRATION
```

## 3.2 Help Command

### -help, -h

Displays basic information about the syntax for Dynamic File Services CLI options, parameters, and actions. It also identifies the software version of Dynamic File Services that is running.

Entering `DswCli.exe` without a parameter lists all of the help without paging.

#### Syntax

At the Command Prompt console, go to the directory where you installed Dynamic File Services, then enter one of the following:

```
dswcli.exe -help
```

```
dswcli.exe -h
```

```
dswcli.exe (displays all online help without paging)
```

## 3.3 Authentication Parameters

You must specify the server name and login credentials for the server where you want to manage Dynamic File Services pairs and policies. This allows you to connect to the service running on the DynamicFS server and execute the command. The authentication parameters include the server name, username, password, and port number for the server that you want to manage.

---

**IMPORTANT:** Login credentials are not required if you run a `DswCli` command while you are logged in as the Administrator user or a user in the `Dynamic File Services` group on the server. Credentials are required if you are managing the server remotely from another computer, or if you are on the server you want to manage but are logged in as a user without the privileges needed to manage Dynamic File Services.

---

You can specify the server name and login credentials with each command, or you can set environment variables for the parameters while you are managing the server.

- ♦ [Section 3.3.1, “Setting Authentication Parameters in Commands,” on page 21](#)
- ♦ [Section 3.3.2, “Setting Authentication Parameters as Environment Variables,” on page 23](#)

## 3.3.1 Setting Authentication Parameters in Commands

Use the options in this section to specify the authentication parameters in each Dynamic File Services command for a server. With this method, you must provide the `-servername`, `-username`, and `-password` options for every command. You need to specify the port number for the Dynamic File Service port on the target server only if the server has been previously configured to use a port other than the default (8999).

- ◆ [“Syntax” on page 21](#)
- ◆ [“Authentication Parameters” on page 21](#)
- ◆ [“Authentication Examples” on page 22](#)

### Syntax

Specifies the authentication parameters as needed in the pair or policy command.

```
DswCli.exe [-servername [-port [-username [-password]]]] {-pair [pair_options] |  
-policy [policy_options]}
```

### Authentication Parameters

This section describes the following authentication parameters:

- ◆ `-servername`
- ◆ `-port`
- ◆ `-username`
- ◆ `-password`

#### **-servername**

Specifies the IP address or DNS name of the server where you want to create or manage the pair. If you are issuing commands locally on the DynamicFS server, you can also use the loopback address (127.0.0.1) instead of its assigned IP address, or you can use `localhost` as the server name instead of its full DNS name. The IPv4 format is supported for the IP address. DNS names are case sensitive.

#### **Syntax**

```
-servername={ "ip_address" | "DNS_name" }
```

#### **Examples**

```
-servername="192.168.1.1"  
-servername="server1.site1.example.com"  
-servername="localhost"
```

#### **-port**

Instead of using the configured port number, use the supplied value.

You need to specify this parameter only if you have set up the target server to use a port other than the default Dynamic File Service port (8999) for remote management communications.

If this parameter is not used and the port is not configured as an environment variable, the default port is automatically used for DynamicFS communications.

## Syntax

```
-port=<"portnumber">
```

## Examples

```
-port="1234"  
-port=8999
```

### **-username, -u**

Specifies the username of a user who is a member of the `Dynamic File Services` group (or is the Administrator user) for the DynamicFS server that you want to manage.

## Syntax

```
-username=<"admin_user_name">
```

```
-u=<"admin_user_name">
```

## Examples

```
-username="Administrator"
```

```
-u="Administrator"
```

```
-u="john"
```

### **-password, -p**

Specifies the password for the user whose username you supplied.

## Syntax

```
-password=<"admin_user_password">
```

```
-p=<"admin_user_password">
```

## Examples

```
-password="novell"
```

```
-p="novell"
```

## Authentication Examples

### **Example: Using the IP Address of the Server**

```
DswCli.exe -servername=localhost -u=Administrator -p=novell  
-pair
```

This command connects to the server via the IP address 192.168.1.1, logs in with the credentials of Administrator and novell, then lists all pairs on the specified server. You can also provide the credentials of a user who is a member of the `Dynamic File Services` group. It assumes that the default port is 8999.

### **Example: Using the DNS Address of the Server**

```
DswCli.exe -servername=localhost -u=Administrator -p=novell  
-port="1234"  
-policy
```

This command connects to the server via the DNS name of `server1.site1.example.com`, logs in with the credentials of Administrator and novell, then lists all policies on the specified server. You can also provide the credentials of a user who is a member of the `Dynamic File`

Services group. It specifies that the TCP connection should be made with port 1234 on the target server, because you have previously configured the Dynamic File Service port on that server to use port 1234 instead of the default port 8999.

### 3.3.2 Setting Authentication Parameters as Environment Variables

Dynamic File Services allows you to use the Windows `set` command in the Command Prompt console to specify environment variables for the authentication parameters (server name, username, password, and port) for the server that you want to manage with commands. The environment variables are set for only one server at a time. The environment variable settings persist if you issue the `set` command again with different values, or until you close the Command Prompt console.

After the environment variables are set for a given server, the parameter values are automatically applied for every `DswCli.exe` command where you do not provide authentication parameters. For example, the following commands apply only to the server that has been previously set up with the environment variables:

```
DswCli.exe -pair  
DswCli.exe -policy
```

You can change the environment variables settings by issuing the `set` commands again with the new values.

---

**IMPORTANT:** You can remove the environment variable settings by closing the Command Prompt console.

---

If it is used without parameters, the `set` command displays the current environment variable settings.

The Dynamic File Services environment variables for the `set` command are:

---

Environment Variable	Description
DswParm1=<servername>	Specifies the <i>servername</i> by providing the DNS address or the IP address of the server you want to manage.
	<b>Examples</b>
	The following command sets the environment variable for <i>servername</i> to 192.168.1.1:
	<pre>set DswParm1=192.168.1.1</pre>
	The following command displays the current setting for the DswParm1 environment variable:
	<pre>set DswParm1</pre>

---

Environment Variable	Description
DswParm2=<username>	<p>Specifies the <i>username</i> of the user identity you want to use to manage the server. The user must be a member of the <code>Dynamic File Services</code> group (or the Administrator user) on the server.</p> <p><b>Examples</b></p> <p>The following command sets the environment variable for <i>username</i> to Administrator. You can also provide the credentials of a user who is a member of the <code>Dynamic File Services</code> group.</p> <pre>set DswParm2=Administrator</pre> <p>The following command displays the current setting for the DswParm2 environment variable:</p> <pre>set DswParm2</pre>
DswParm3=<password>	<p>Specifies the <i>password</i> of the user identity you specified with DswParm2.</p> <p><b>Examples</b></p> <p>The following command sets the environment variable for <i>password</i> to novell:</p> <pre>set DswParm3=novell</pre> <p>The following command displays the current setting for the DswParm3 environment variable:</p> <pre>set DswParm3</pre>
DswParm4=<portnumber>	<p>Specifies the <i>portnumber</i> to use for DynamicFS communications with the server you want to manage.</p> <p>You need to specify this parameter only if you have set up the target server to use a port other than the default Dynamic File Service port for remote management communications. The default port value is automatically assumed if this parameter is not otherwise specified.</p> <p><b>Examples</b></p> <p>The following command sets the environment variable for <i>portnumber</i> to 1234:</p> <pre>set DswParm4=1234</pre> <p>The following command displays the current setting for the DswParm4 environment variable:</p> <pre>set DswParm4</pre>

To use environmental variables while executing Dynamic File Services commands:

- 1 On a Windows server or workstation where Dynamic File Services is installed, open the Command Prompt console.
- 2 At the command prompt, specify the server name and login credentials for the server that you want to manage by entering the following commands as needed:

```
set DswParm1=192.168.1.1
set DswParm2=Administrator
set DswParm3=password
set DswParm4=1234
```

Make sure to change the values to the credentials needed to log in to the server you want to manage.

- 3 At the command prompt, enter the `DswCli.exe` commands to manage pairs and policies on the specified server.

You can specify the commands without providing credentials.

- 4 Repeat [Step 3](#) for all of the `DswCli.exe` commands you want to issue for that server.
- 5 When you are done, repeat [Step 2](#) to [Step 4](#) for each server you want to manage.
- 6 Close the Command Prompt console to remove the credentials as environmental variables.

## 3.4 Common Parameters

The parameters in this section can be used by Dynamic File Services pair or policy actions.

- ♦ [Section 3.4.1, "Description," on page 25](#)
- ♦ [Section 3.4.2, "Direction to Move Files," on page 26](#)
- ♦ [Section 3.4.3, "Name," on page 26](#)
- ♦ [Section 3.4.4, "Pair ID," on page 26](#)
- ♦ [Section 3.4.5, "Policy ID," on page 27](#)
- ♦ [Section 3.4.6, "Policy ID List," on page 27](#)
- ♦ [Section 3.4.7, "Schedule ID," on page 27](#)
- ♦ [Section 3.4.8, "Schedule Type," on page 28](#)

### 3.4.1 Description

#### **-description**

Specifies a textual description of a pair, a policy, or a schedule that you are creating with the `-pair -add action`, the `-policy -add action`, or the `-schedule -add action`.

#### **Syntax**

```
-description=<"text">
```

#### **Examples**

```
-description="This is a description of myPair"
```

```
-description="Policy to move graphics files to the secondary"
```

```
-description="This is a description of a schedule"
```

## 3.4.2 Direction to Move Files

### **-primaryToSecondary**

Specifies that files will move from the primary to the secondary location when an action is performed. If a direction option is not specified, the default direction is primary to secondary.

**Restriction:** This option cannot be used with the `-secondaryToPrimary` option.

### **-secondaryToPrimary**

Specifies that files will move from the secondary to the primary location when an action is performed. If a direction option is not specified, the default direction is primary to secondary.

The `-secondaryToPrimary` option is invalid for retention pairs.

**Restriction:** This option cannot be used with the `-primaryToSecondary` option.

## 3.4.3 Name

### **-name**

Specifies a name for a pair, a policy, or a schedule that you are creating with the `-pair -add action`, the `-policy -add action`, or the `-schedule -add action`.

#### **Syntax**

```
-name=<"text">
```

#### **Pair Examples**

```
-name="myPair"
```

```
-name="ProjectA Pair"
```

#### **Policy Examples**

```
-name="myPolicy"
```

```
-name="Move music and videos"
```

#### **Schedule Examples**

```
-name="quarterly_1st"
```

```
-name="custom_financial"
```

## 3.4.4 Pair ID

### **-pairId**

Specifies the pair identifier. You can provide the pair name, or provide the GUID (globally unique identification) of the pair. A GUID is automatically assigned by Dynamic File Services when you create a pair.

#### **Syntax**

```
-pairId=<"pairname" | "GUID">
```

#### **Example**

```
-pairId="myPair"
```

```
-pairID="42e0064d-0b2c-4bb3-9825-bfa82999720e"
```

## 3.4.5 Policy ID

### **-policyId**

Specifies the policy identifier. You can provide the policy name, or provide the GUID of the policy. A GUID is automatically assigned by Dynamic File Services when you create a policy.

#### **Syntax**

```
-policyId=<"polycyname" | "GUID">
```

#### **Examples**

```
-policyId="myPolicy"
```

```
-policyId="My JPG and BMP Policy"
```

```
-policyId="My Last Modified GT 6 Months Policy"
```

```
-policyId="4b5b5820-da6c-4c07-b9da-07e3b83ebe02"
```

## 3.4.6 Policy ID List

### **-policyIdList**

Specifies a comma-separated list of `policyId` parameters. For each policy in the list, you can provide the policy name or the GUID. Separate entries with a comma and no spaces. Place quotation marks around the list.

#### **Syntax**

```
-policyIdList=<"<polycyname | GUID>[ , <polycyname | GUID> , ... ]">
```

#### **Examples**

```
-policyIdList="myPolicy,myPolicy100,4b5b5820-da6c-4c07-b9da-07e3b83ebe02"
```

```
-policyIdList="My JPG and BMP Policy"
```

```
-policyIdList="My Last Modified GT 6 Months Policy,myPolicy"
```

## 3.4.7 Schedule ID

### **-scheduleId**

Specifies the schedule identifier. You can provide the schedule name, or provide the GUID of the schedule. A GUID is automatically assigned by Dynamic File Services when you create a schedule.

#### **Syntax**

```
-scheduleId=<"schedulename" | "GUID">
```

#### **Examples**

```
-scheduleId="quarterly_1st"
```

```
-scheduleId="custom_financial"
```

## 3.4.8 Schedule Type

### **-scheduleType**

Specifies whether the type of schedule is a review schedule or a policy schedule.

#### **Syntax**

```
-scheduleType=<"review" | "policy">
```

#### **Examples**

```
-scheduleType="review"
```

```
-scheduleType="policy"
```

## 3.5 Common Options

Common options can be used in combination with any of the system, pair, or policy actions.

- ♦ [Section 3.5.1, “Debug,” on page 28](#)
- ♦ [Section 3.5.2, “Output to a File,” on page 28](#)
- ♦ [Section 3.5.3, “Silent,” on page 28](#)

### 3.5.1 Debug

#### **-debug**

Displays debug messages.

#### **Syntax**

```
-debug
```

### 3.5.2 Output to a File

#### **-of**

Dumps certain requests and replies to a file.

#### **Syntax**

```
-of
```

### 3.5.3 Silent

#### **-silent**

Use this option when you do not want to display errors or help on execution, such as in scripts for batch files where the batch file checks for return codes. This option returns 0 if the command is successful. It returns 1 if the command fails.

If a command fails, you can enter the command at the Command Prompt console without the `-silent` option to view the error messages.

#### **Syntax**

```
-silent
```

## 3.6 System Actions

You can use the system action options to display information about a specified Dynamic File Services server. For information about authentication parameters, see [Section 3.3, “Authentication Parameters,”](#) on page 20.

- ◆ [Section 3.6.1, “Display Active Directory Computers,”](#) on page 29
- ◆ [Section 3.6.2, “Display Active Directory Shares,”](#) on page 29
- ◆ [Section 3.6.3, “Display the File Types Information,”](#) on page 29
- ◆ [Section 3.6.4, “Display Local Drives,”](#) on page 30
- ◆ [Section 3.6.5, “Display Network Shares,”](#) on page 30
- ◆ [Section 3.6.6, “Display Server System Information,”](#) on page 30
- ◆ [Section 3.6.7, “Display or Set the Logging Level for the Service or Standard Policy,”](#) on page 30
- ◆ [Section 3.6.8, “Query for File Types or Extensions \(for Novell Support Use Only\),”](#) on page 32

### 3.6.1 Display Active Directory Computers

#### **-adcomputers**

Displays a list of the computers in the Active Directory domain.

#### **Syntax**

```
DswCli.exe [authentication_parameters] -adcomputers
```

### 3.6.2 Display Active Directory Shares

#### **-adshares**

Displays the shares that are published in Active Directory.

#### **Syntax**

```
DswCli.exe [authentication_parameters] -adshares
```

### 3.6.3 Display the File Types Information

#### **-fileTypesInfo, -types**

Displays a list of file types found on the target server. File types are the MIME content types or perceived types of file extensions that are associated in the Windows Registry with applications installed on the server. It also lists the perceived file types that are defined in the `..\Dynamic File Services\DswFileTypes.cfg` file. You can customize the types defined in the `DswFileTypes.cfg` file. For information, see [“Setting Perceived Types in the DswFileTypes.cfg File”](#) in the *Dynamic File Services 2.0 Administration Guide*.

#### **Syntax**

```
DswCli.exe [authentication_parameters] -fileTypesInfo
```

```
DswCli.exe [authentication_parameters] -types
```

## 3.6.4 Display Local Drives

### **-localdrives**

Displays local drive information for the specified server.

#### **Syntax**

```
DswCli.exe [authentication_parameters] -localdrives
```

## 3.6.5 Display Network Shares

### **-shares**

Displays network share information for the specified server.

#### **Syntax**

```
DswCli.exe [authentication_parameters] -shares
```

## 3.6.6 Display Server System Information

### **-system**

Displays system information for the specified server.

#### **Syntax**

```
DswCli.exe [authentication_parameters] -system
```

## 3.6.7 Display or Set the Logging Level for the Service or Standard Policy

### **-logfile [filename] [-logfilelevel]**

Novell Dynamic File Services uses Apache log4net to log events. Log level settings determine the type of events that are logged. You can use the log level commands to display or set the log levels for the Dynamic File Service (`DswMcpCore.log`) log file and the Standard Policy (`DswStandardPolicy.log`) log file.

#### **Syntax**

```
DswCli [authentication_parameters]
    -logfile=< * | SERVICE | STANDARDPOLICY | filename>
    [-logfilelevel=<ALL | DEBUG | INFO | WARN | ERROR | FATAL | OFF>]
```

The two log level options are used together to set the log level for the specified log file. If you use the `-logfile` option without the `-logfilelevel` option, the current setting for the specified log file is displayed.

## Log File Name Options

You must specify whether you want to apply the `-loglevel` setting to the Service, Standard Policy, or both the Service and Standard Policy.

Logfilename Option	Description
*	Apply the specified <code>-loglevel</code> setting to both the Service ( <code>DswMcpCore.config.xml</code> ) log file and the Standard Policy ( <code>DswStandardPolicy.config.xml</code> ) log file.
SERVICE SVC	Apply the specified <code>-loglevel</code> setting to the Service ( <code>DswMcpCore.config.xml</code> ) log file.
STANDARDPOLICY SPOL	Apply the specified <code>-loglevel</code> setting to the Standard Policy ( <code>DswStandardPolicy.config.xml</code> ) log file.
<code>"filename"</code>	Apply the specified <code>-loglevel</code> setting to the specified file. Valid filenames are <code>DswMcpCore.config.xml</code> or <code>DswStandardPolicy.config.xml</code> .  This option allows you to specify the names of the Service and Standard Policy log files as an alternative to using the SERVICE or STANDARDPOLICY options.

### Logfilename Examples

```
-logfilename=*
```

```
-logfilename=SERVICE
```

## Log Level Options

The log level options determine which events are logged for the Dynamic File Service log file or Standard Policy log file. Use this option in combination with the `-logfilename` option to modify the log level setting for the specified log file.

Each level includes the events specified, plus the events of the levels below it. The log4net software supports the following logging levels in order of increasing priority:

Log Level Option	Description
All	Record all events in the specified log file. (This is the same output as the DEBUG level.)
DEBUG	Record debug, information, warning, error, and fatal events in the specified log file.
INFO	Record information, warning, error, and fatal events in the specified log file.
WARN	(Default) Record warning, error, and fatal events in the specified log file.
ERROR	Record error and fatal events in the specified log file.
FATAL	Record fatal events in the specified log file.
OFF	No events are recorded in the specified log file.

### LogLevel Examples

```
-loglevel=INFO
```

```
-loglevel=FATAL
```

### Logging Level Examples

Get the current value of the Service log file (DswMcpCore.config.xml):

```
DswCli [authentication_parameters] -logfile=SERVICE
```

Set the log level to ALL for the Service log file (DswMcpCore.config.xml):

```
DswCli [authentication_parameters] -logfile=SERVICE -loglevel=ALL
```

## 3.6.8 Query for File Types or Extensions (for Novell Support Use Only)

Dynamic File Services provides the following query commands for use only by Novell Support:

### -fileTypeExtensionQuery, -extQuery

Returns a list of file types associated with the specified extensions.

#### Example

```
DswCli.exe [authentication_parameters] -fileTypeExtensionQuery=".jpg,.mp3"
```

### -fileTypeQuery, -typeQuery

Returns a list of file extensions associated with the specified file type. File types are the MIME content types or perceived types of file extensions that are associated with applications installed on the server.

#### Example

```
DswCli.exe [authentication_parameters] -fileTypeQuery="image"
```

## 3.7 Registration Actions

### -registration

Performs operations on a registration key file. The `-registration` action allows you to register a license key on the server where you have installed the Service component of Dynamic File Services. When it is used without providing a registration key file path, it checks to see if a key has already been registered.

You can get a license key from [Novell Customer Center \(http://www.novell.com/customercenter/\)](http://www.novell.com/customercenter/). Save the key as a `.txt` or `.html` file on the server.

#### Syntax

```
DswCli.exe [authentication_parameters] -registration  
[-regFilePath=<"path">]
```

For information about how to provide the servername and login credentials that are needed to connect to the server that you want to manage, see [Section 3.3, "Authentication Parameters," on page 20](#).

- ◆ [Section 3.7.1, "Display the Registration Status," on page 33](#)
- ◆ [Section 3.7.2, "Register a Product License Key," on page 33](#)

## 3.7.1 Display the Registration Status

### (no path)

When the `-registration` action is used without providing a key file path, it displays whether a license key has been registered on the Dynamic File Services server.

### Syntax

```
DswCli.exe [authentication_parameters] -registration
```

### Example

The following `-registration` command displays the registration status.

```
DswCli.exe -servername=localhost -u=Administrator -p=novell  
-registration
```

## 3.7.2 Register a Product License Key

### -regFilePath

The `-regFilePath` option specifies the path and filename of the license key file for Dynamic File Services. The key file must have been previously obtained from the [Novell Customer Center](http://www.novell.com/customercenter/) (<http://www.novell.com/customercenter/>) and saved to a location on the server.

### Syntax

```
DswCli.exe [authentication_parameters] -registration  
-regFilePath=<"path">
```

### Example

The following command reads the specified license key file and registers the license key with DynamicFS:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell  
-registration  
-regFilePath="C:\dir1\myregistration.txt"
```

## 3.8 Pair Actions

### -pair

Performs operations on a pair. When it is used without other pair action options, all pairs on the server are listed.

### Syntax

```
DswCli.exe [authentication_parameters] -pair  
[pair_action]  
[-pairId]  
[pair_option]
```

Each of the following pair actions are described below:

```
-add
-remove
-addExcludeIncludePaths
-removeExcludeIncludePaths
-associate
-disassociate
-detail
-execute
-move
```

For information about how to provide the servername and login credentials that are needed to connect to the server that you want to manage, see [Section 3.3, “Authentication Parameters,”](#) on page 20.

- ◆ [Section 3.8.1, “Pair Parameters,”](#) on page 34
- ◆ [Section 3.8.2, “Add a Pair,”](#) on page 35
- ◆ [Section 3.8.3, “Unlink a Pair,”](#) on page 38
- ◆ [Section 3.8.4, “Add Exclude/Include Paths to a Pair,”](#) on page 38
- ◆ [Section 3.8.5, “Delete Exclude/Include Paths from a Pair,”](#) on page 39
- ◆ [Section 3.8.6, “Associate a Pair and Policy,”](#) on page 40
- ◆ [Section 3.8.7, “Disassociate a Pair and Policy,”](#) on page 41
- ◆ [Section 3.8.8, “Associate a Review Schedule to a Retention Pair,”](#) on page 41
- ◆ [Section 3.8.9, “Disassociate a Review Schedule from a Retention Pair,”](#) on page 42
- ◆ [Section 3.8.10, “List Pairs,”](#) on page 42
- ◆ [Section 3.8.11, “List Details for a Pair,”](#) on page 42
- ◆ [Section 3.8.12, “Execute a List of Policies on a Pair,”](#) on page 43
- ◆ [Section 3.8.13, “Move a List of Files or Folders in a Pair,”](#) on page 43

## 3.8.1 Pair Parameters

This section describes the following common pair parameters:

- ◆ [-description](#)
- ◆ [-pairId](#)

### **-description**

Specifies a textual description of the pair. The description is optional.

#### **Syntax**

```
-description="text"
```

#### **Example**

```
-description="Moves graphics files to the secondary location."
```

### **-pairId**

Specifies the pair identifier. You can provide the pair name, or provide the GUID of the pair. A GUID is automatically assigned by Dynamic File Services when you create a pair.

#### **Syntax**

```
-pairId=<"pairname" | "GUID">
```

## Example

```
-pairId="myPair"  
-pairID="42e0064d-0b2c-4bb3-9825-bfa82999720e"
```

## 3.8.2 Add a Pair

### -add

The `-add` action creates the pair with the desired name and stores the configuration as an XML file in the `C:\ProgramData\Dynamic File Services\Pairs` folder.

---

**IMPORTANT:** Before you issue a command to create a pair, make sure your system meets the requirements in “[Pair Requirements](#)” in the *Dynamic File Services 2.0 Administration Guide*.

---

### Syntax

```
DswCli.exe [authentication_parameters] -pair  
-add  
-name=<"pairname">  
-primarypath=<"path">  
-secondaryPath=<"path">  
[-description=<"text">]  
[-pairType=<"standard"|"retention">]
```

You must specify a name for the pair, a primary path, and a secondary path. You can optionally specify a description of the pair. If the `pairType` option is not specified, the default is "standard".

For a standard pair, the primary path is a local path. The secondary path can be a local path or the UNC (Universal Naming Convention) path of a remote share. The standard pair provides a merged view of data in the two locations. You must create a network share for the primary path in order to provide a merged view of the data for users.

---

**IMPORTANT:** Use the Microsoft Network Sharing tool to create a network share on the primary path of a standard pair. Users connect to the network share by mapping a local drive letter on their workstations to the network share. All user access takes place through the share.

---

For a retention pair, the primary path and secondary path can be a local path or the UNC path of a remote share. A retention pair type does not have a merged view.

If you specify a UNC path for remote shares, the share must already exist before you run the command. To set up a share in an Active Directory environment, you must do the following:

- ◆ Create the remote network share.
- ◆ Publish the share in Active Directory.
- ◆ Add the `Dynamic File Services Storage Rights` group to the remote share and give the group all permissions.

After you create a pair, you should verify that the pair setup is correct before granting users access to the pair or running policies on the pair.

Data is not moved between the primary path and secondary path until you associate the policy with one or more policies, or until you move a specified list of files or folders by using the `move` command. For information about creating policies, see [Section 3.9.2, “Add a Policy,” on page 46](#). For information about moving specified files or folders, see [Section 3.8.13, “Move a List of Files or Folders in a Pair,” on page 43](#).

## Add Pair Parameters

### **-name**

Specifies the pair name. The name must be unique on the server you are managing.

#### **Syntax**

```
-name=<"pairname">
```

#### **Example**

```
-name="myPair"
```

### **-primaryPath**

Specifies the path of the primary location in the pair. The drive or path must already exist; the command does not create it for you.

For a retention pair type, the primary path can also be a UNC path to a remote share. The parser requires that you use 3 backslashes (\\\) instead of two preceding the server name, such as \\remoteservername\share.

#### **Syntax**

```
-primaryPath=<"path">
```

#### **Examples**

```
-primaryPath="C:\users"
```

```
-primaryPath="M:"
```

```
-primaryPath="K:\engineering\proj1"
```

```
-primaryPath="\\remoteserver\share"
```

### **-secondaryPath**

Specifies the path of the secondary location in the pair. The drive or path must already exist; the command does not create it for you.

The secondary path can also be a UNC path to a remote share. The parser requires that you use 3 backslashes (\\\) instead of two preceding the server name, such as \\remoteserver\share.

#### **Syntax**

```
-secondaryPath=<"path">
```

#### **Examples**

```
-secondaryPath="L:\users\media"
```

```
-secondaryPath="N:\m_sh"
```

```
-secondaryPath="Z:\project1"
```

```
-secondaryPath="\\remoteserver\share"
```

## Add Pair Options

### **-pairType**

Specifies whether the pair is a standard pair or retention pair. If this option is not specified, the default pair type is "standard".

## Syntax

```
-pairType=<"standard"|"retention">
```

## Example

```
-pairType="standard"
```

## Add Pair Examples

### Example: Create a Pair

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-pair
-add
-name="myPair"
-primaryPath="e:\PrimaryPath"
-secondaryPath="f:\SecondaryPath"
-pairType="standard"
```

This command uses the user credentials of the Administrator user to log you in to the server where you are running the command. You can also provide the credentials of a user who is a member of the `Dynamic File Services` group. It creates a standard pair named `myPair` on the server. The pair's primary path is the `e:\PrimaryPath` directory. The pair's secondary path is the `f:\SecondaryPath` directory. No files are moved until you create a policy for the pair, and associate the policy to the pair. Users access a share on the primary path to see a merged view of the data.

### Example: Create a Pair with a Description

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-pair
-add
-name="ProjectA"
-primaryPath="e:\PrimaryPath"
-secondaryPath="f:\SecondaryPath"
-description="Project A management files"
```

This command uses the user credentials of the Administrator user to log you in to the server where you are running the command. You can also provide the credentials of a user who is a member of the `Dynamic File Services` group. It creates a standard pair named `ProjectA`. It uses the optional description field to provide more information about the pair. The pair's primary path is the `e:\PrimaryPath` directory. The pair's secondary path is the `f:\SecondaryPath` directory. No files are moved until you create a policy for the pair, and associate the policy to the pair. The default pair type of "standard" is used. Users access a share on the primary path to see a merged view of the data.

### Example: Create a Retention Pair

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-pair
-add
-name="retentionPair"
-primaryPath="e:\PrimaryPath"
-secondaryPath="s:\SecondaryPath"
-pairType="retention"
```

This command uses the credentials of the Administrator user to log you in to the server where you are running the command. You can also provide the credentials of a user who is a member of the `Dynamic File Services` group. It creates a retention pair named `retentionPair` on the server. The pair's primary path is the `e:\PrimaryPath` directory. The pair's secondary path is the `s:\SecondaryPath` directory. No files are moved until you create a policy for the pair, and associate the policy to the pair. A merged view is not available for retention pairs. Users can view and access only the data on the primary pair.

## 3.8.3 Unlink a Pair

### -delete

The `-delete` action removes the specified pair from the pair database. All links between the two storage locations are removed. Data is not destroyed. The data is not moved; that is, the data remains in the location where it was stored when the delete command was executed. The associations between the pair and any policies are removed. After the pair is deleted, the users who are logged in to the network share on the primary location can see and access the data only on the primary location.

You must specify the `pairId` for the pair that you want to delete.

### Syntax

The delete option unlinks the two paths in a pair. The files are not deleted.

```
DswCli.exe [authentication_parameters] -pair
           -delete
           -pairId=<"pairname" | "GUID">
```

### Example

The following `-pair` command deletes the pair named `myPair` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
           -pair
           -delete
           -pairId="myPair"
```

## 3.8.4 Add Exclude/Include Paths to a Pair

### -addExcludeIncludePaths

Adds exclude or include folder paths to a pair. It requires the `-pairId`, `-paths`, and `-flags` parameters. A given pair can include folders or exclude folders, but you cannot do both.

---

**IMPORTANT:** This option is not supported for pairs with remote primary paths.

---

### Syntax

```
DswCli.exe [authentication_parameters] -pair
           -addExcludeIncludePaths
           -pairID
           -flags
           -paths
```

### Exclude/Include Paths Options

#### -flags

Specifies the flag to exclude or include the folders specified by the `-paths` option. A given pair can include folders or exclude folders, but you cannot do both.

#### Syntax

```
-flags=<"exclude" | "include">
```

#### Examples

```
-flags="exclude" -paths="path1"
```

```
-flags="include" -paths="path1;path2;path3"
```

### **-paths**

Specifies one or more subdirectory paths that are to be included or excluded from policies run on the pair. Separate multiple paths with a semicolon. Exclude/include should be set on subfolders and not the root of the primary share. Use this option in combination with the `-flags` option.

The primary path must reside on a device that is attached to the Dynamic File Services server. Remote primary paths are not supported.

#### **Syntax**

```
-paths=<"dirpath[;dirpath;...]">
```

#### **Examples**

```
-flags="exclude" -paths="path1"
```

```
-flags="include" -paths="path1;path2;path3"
```

```
-flags="exclude" -paths="C:\primary\subdir1;C:\primary\subdir2"
```

#### **Add Exclude/Include Paths Examples**

```
DswCli.exe -servername=localhost -u=Administrator -p=novell  
-pair  
-addExcludeIncludePaths  
-pairId="myPair"  
-flags="exclude"  
-paths="path1"
```

```
DswCli.exe -servername=localhost -u=Administrator -p=novell  
-pair  
-addExcludeIncludePaths  
-pairId="myPair"  
-flags="include"  
-paths="path1;path2;path3"
```

## **3.8.5 Delete Exclude/Include Paths from a Pair**

### **-deleteExcludeIncludePaths**

Removes exclude or include folder paths from a pair. It requires the `-pairId`, `-paths`, and `-flags` options.

#### **Syntax**

```
DswCli.exe [authentication_parameters] -pair  
-deleteExcludeIncludePaths  
-pairID  
-flags  
-paths
```

#### **Exclude/Include Paths Options**

##### **-flags**

Specifies the flag to exclude or include the folders specified by the `-paths` option. A given pair can include folders or exclude folders, but you cannot do both.

#### **Syntax**

```
-flags=<"exclude" | "include">
```

## Examples

```
-flags="exclude" -paths="path1"  
-flags="include" -paths="path1;path2;path3"
```

### -paths

Specifies one or more directory paths to be included or excluded from policies run on the pair. Separate multiple paths with a semicolon. Use this option in combination with the `-flags` option.

### Syntax

```
-paths=<"dirpath[;dirpath;...]">
```

## Examples

```
-flags="exclude" -paths="path1"  
-flags="include" -paths="path1;path2;path3"  
-flags="exclude" -paths="C:\primary\subdir1;C:\primary\subdir2"
```

## Delete Exclude/Include Paths Examples

```
DswCli.exe -servername=localhost -u=Administrator -p=novell  
-pair  
-deleteExcludeIncludePaths  
-pairId="myPair"  
-flags="exclude"  
-paths="path1"  
  
DswCli.exe -servername=localhost -u=Administrator -p=novell  
-pair  
-deleteExcludeIncludePaths  
-pairId="myPair"  
-flags="include"  
-paths="path1;path2;path3"
```

## 3.8.6 Associate a Pair and Policy

### -associate

Associates a policy with a retention pair or standard pair. Associates a review schedule with a retention pair. Associates a policy schedule with a policy.

When the `-policyid` option is used with the `-associate` action and a `-pairid` parameter, it links the specified pair and policy. Policies can be associated with retention pairs and standard pairs.

When the `-scheduleid` option for a review schedule is used with the `-associate` action and a `-pairid` parameter for a retention pair, it links the specified review schedule and retention pair.

When the `-scheduleid` option for a policy schedule is used with the `-associate` action and a `-policyid` parameter for a policy, it links the specified policy schedule and policy.

### Syntax

```
DswCli.exe [authentication_parameters] -pair -associate  
-pairId=<"pairname"|"GUID" >  
-policyId=<"policyname"|"GUID">
```

### Example

The following `-pair` command associates the pair named `myPair` with the policy named `myPolicy` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
  -pair
  -associate
  -pairId="myPair"
  -policyId="myPolicy"
```

## 3.8.7 Disassociate a Pair and Policy

### **-disassociate**

Disassociates a policy from a retention pair or standard pair. Disassociates a review schedule from a retention pair. Disassociates a policy schedule with a policy.

When the `-policyid` option is used with the `-disassociate` action and a `-pairid` parameter, it unlinks the specified pair and policy.

When the `-scheduleid` option for a review schedule is used with the `-disassociate` action and a `-pairid` parameter for a retention pair, it unlinks the specified review schedule and retention pair.

When the `-scheduleid` option for a policy schedule is used with the `-disassociate` action and a `-policyid` parameter for a policy, it unlinks the specified policy schedule and policy.

### Syntax

```
DswCli.exe [authentication_parameters] -pair
  -disassociate
  -pairId=<"pairname" | "GUID">
  -policyId=<"policyname" | "GUID">
```

### Example

The following `-pair` command removes the association between the pair named `myPair` and the policy named `myPolicy` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
  -pair
  -disassociate
  -pairId="myPair"
  -policyId="myPolicy"
```

## 3.8.8 Associate a Review Schedule to a Retention Pair

### **-associate**

When the `-scheduleid` option is used with the `-associate` action, it links a specified retention pair to a specified schedule. You must specify the `pairId` parameter for the pair that you want to associate, and the `scheduleId` parameter for the schedule.

### Syntax

```
DswCli.exe [authentication_parameters] -pair
  -associate
  -pairId=<"pairname" | "GUID">
  -scheduleId=<"schemulename" | "GUID">
```

### Example

The following `-pair` command associates the pair named `myRetentionPair` with the schedule named `quarterly_1st` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
  -pair
  -associate
  -pairId="myPair"
  -scheduleId="quarterly_1st"
```

## 3.8.9 Disassociate a Review Schedule from a Retention Pair

### **-disassociate**

When the `-scheduleid` option is used with the `-disassociate` action, it removes the association between a specified retention pair and schedule. You must specify the `pairId` parameter for the pair. You must specify the `scheduleId` parameter for the schedule that you no longer want to be associated with the pair.

### Syntax

```
DswCli.exe [authentication_parameters] -pair
  -disassociate
  -pairId=<"pairname" | "GUID">
  -scheduleId=<"schemulename" | "GUID">
```

### Example

The following `-pair` command removes the association between the retention pair named `myRetentionPair` and the schedule named `quarterly_1st` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
  -pair
  -disassociate
  -pairId="myPair"
  -scheduleId="quarterly_1st"
```

## 3.8.10 List Pairs

### (no action options)

When the `-pair` action option is used without any other parameters or options, all pairs on the server are listed.

### Syntax

```
DswCli.exe [authentication_parameters]] -pair
```

### Example

The following `-pair` command displays a list of all pairs on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
  -pair
```

## 3.8.11 List Details for a Pair

### **-detail**

The `-detail` action provides a details for a pair, including its pair type, paths, and associated policies. For retention pairs, the associated schedule is also identified. You must specify the `pairId` parameter for the pair.

### Syntax

```
DswCli.exe [authentication_parameters] -pair
           -detail
           -pairId=<"pairname" | "GUID">
```

### Example

The following `-pair` command lists details for the pair named `myPair` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
           -pair
           -detail
           -pairId="myPair"
```

## 3.8.12 Execute a List of Policies on a Pair

### **-execute, -run**

The `-execute` action causes all policies defined in the specified policy ID list to be executed for the specified pair. You must specify the `pairId` parameter for the pair. Use the [-policyIdList parameter](#) to list one or more policies to run. If the policy list is not specified, all policies run that are associated with the specified pair.

The `-run` option is supported as an alias.

### Syntax

```
DswCli.exe [authentication_parameters] -pair
           -execute -pairId=<"pairname" | "GUID">
           [-policyIdList=<"policy1[,policy2,...]">]
```

### Example

The following `-pair` command runs the DynamicFS policies named `myPolicy` and `myPolicy100` for the pair named `myPair` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
           -pair
           -execute
           -pairId="myPair"
           -policyIdList="myPolicy,myPolicy100"
```

## 3.8.13 Move a List of Files or Folders in a Pair

### **-move**

Moves a specified list of files or a list of folders from primary to secondary. You can specify either a list of files, a list of folders, or both lists in a single move command.

### Syntax

```
DswCli.exe [authentication_parameters] -pair
           -pairId=<"pairname" | "GUID">
           -move < -folderList | -fileList | -folderList -fileList >
           -primaryToSecondary | -secondaryToPrimary
```

### Description

The `-move` option requires the following options be specified:

```
-pair
-pairId=<pairname|GUID>
-primaryToSecondary | -secondaryToPrimary
```

-folderList, -folderList, or both of these options

The `-pairId` option identifies the pair where the folders are located.

The `-primaryToSecondary` or `-secondaryToPrimary` option specifies which direction to move the folders and their content. The `-secondaryToPrimary` option is valid on a standard pair, but not on a retention pair.

In the `-fileList` option, the `ListOfFilesToMove.txt` file contains a list of files to move.

In the `-folderList` option, the `ListOfFoldersToMove.txt` file contains a list of folders to move. The folders and all of their contents are moved.

## List Options

### -fileList

Specifies the path to a text file that contains a list of the files that are to be moved. This parameter is used in combination with the `-move` option.

#### Syntax

```
-fileList=<"path\ListOfFilesToMove.txt">
```

Each entry in the file provides path and filename of the file to be moved. Each file appears on a different line in the text file. The file path is relative to the pair's root directory.

For example, the following lines are sample text content for the `ListOfFilesToMove.txt` file:

```
\file.jpg  
\dir1\anotherfile.avi  
\dir2\dir3\file3.bmp
```

#### Examples

```
-fileList="M:\primary\dir1\ListOfFilesToMove.txt"  
-fileList="C:\dir1\movelist.txt"
```

### -folderList

Specifies the path to a text file that contains a list of the files that are to be moved. This parameter is used in combination with the `-move` option.

#### Syntax

```
-folderList=<"path\ListOfFoldersToMove.txt">
```

Each entry in the file provides path of the folder to be moved. Each folder path appears on a different line in the text file. The folder path is relative to the pair's root directory.

For example, the following lines are sample text content for the `ListOfFoldersToMove.txt` file:

```
\home\user1\music  
\dir1\archive  
\dir2\dir3\reports
```

#### Examples

```
-folderList="M:\primary\dir1\ListOfFoldersToMove.txt"  
-folderList="C:\dir1\movelist.txt"
```

## Move Files or Folders Examples

### Example: Move a List of Folders

The following `-pair -move` command moves the folders that are specified in the `ListOfFoldersToMove.txt` file from the primary location to the secondary location of the `myPair` pair. All of the contents in each folder are also moved.

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
           -pair -pairId="myPair"
           -move
           -folderList="c:\dir1\ListOfFoldersToMove.txt
           -primaryToSecondary
```

### Example: Move a List of Files and a List of Folders

The following `-pair -move` command moves the folders that are specified in the `ListOfFoldersToMove.txt` file from the primary location to the secondary location of the `myPair` pair. All of the contents in each folder are also moved. The command also moves the individual files that are separately specified in the `ListOfFilesToMove.txt` file.

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
           -pair -pairId="myPair"
           -move
           -folderList="c:\dir1\ListOfFoldersToMove.txt
           -fileList="c:\dir1\ListOfFilesToMove.txt
           -primaryToSecondary
```

## 3.9 Policy Actions

### **-policy**

Performs operations on a policy. When it is used without policy action options, all policies on the server are listed.

#### **Syntax**

```
DswCli.exe [authentication_parameters] -policy
           [-add | -delete | -detail]
           [policy_parameter]
           [policy_option]
```

For information about how to provide the `servername` and login credentials that are needed to connect to the server that you want to manage, see [Section 3.3, "Authentication Parameters,"](#) on page 20.

- ♦ [Section 3.9.1, "Policy Parameters,"](#) on page 46
- ♦ [Section 3.9.2, "Add a Policy,"](#) on page 46
- ♦ [Section 3.9.3, "Modify a Policy,"](#) on page 52
- ♦ [Section 3.9.4, "Delete a Policy,"](#) on page 53
- ♦ [Section 3.9.5, "List All Policies,"](#) on page 53
- ♦ [Section 3.9.6, "List Details for a Policy,"](#) on page 53
- ♦ [Section 3.9.7, "Associate a Policy Schedule to a Policy,"](#) on page 54
- ♦ [Section 3.9.8, "Disassociate a Policy Schedule from a Policy,"](#) on page 54
- ♦ [Section 3.9.9, "Associate a Policy and a Pair,"](#) on page 54

- ♦ [Section 3.9.10, “Disassociate a Policy and a Pair,” on page 54](#)
- ♦ [Section 3.9.11, “List Policies Associated with a Pair,” on page 55](#)

## 3.9.1 Policy Parameters

This section describes the following common policy parameters:

- ♦ [-description](#)
- ♦ [-policyId](#)
- ♦ [-policyIdList](#)

### **-description**

Specifies a textual description of the policy. The description is optional.

#### **Syntax**

```
-description=<"text">
```

#### **Example**

```
-description="Move graphics to secondary weekly"
```

### **-policyId**

Specifies the policy identifier. You can provide the policy name, or provide the GUID of the policy.

#### **Syntax**

```
-policyId=<"polycyname" | "GUID">
```

#### **Examples**

```
-policyId="myPolicy"
```

```
-policyId="My JPG and BMP Policy"
```

```
-policyId="My Last Modified GT 6 Months Policy"
```

### **-policyIdList**

Specifies a comma-separated list of `policyId` parameters. For each policy, you can provide the policy name, or provide the GUID of the policy.

If the policy list is not specified, all policies that are associated with the specified pair are run.

#### **Syntax**

```
-policyIdList=<"<polycyname | GUID>[ , <polycyname | GUID> , ... ]">
```

#### **Example**

```
-policyIdList="myPolicy,myPolicy100"
```

## 3.9.2 Add a Policy

### **-add**

The `-add` action creates the policy with the specified name and stores the configuration as an XML file in the `C:\ProgramData\Dynamic File Services\Policies` folder.

## Syntax

```
DswCli.exe [authentication_parameters] -policy
    -add
    -name=<polycyname>
    [-description"text" ]
    directionOption
    filterOptions
    frequencyOptions
```

After you add a policy, you must associate it with one or more pairs before you can run it on the pairs. For information, see [Section 3.8.6, "Associate a Pair and Policy," on page 40](#)

You can run a policy manually, or associate it with a policy schedule to run it automatically. For information, see [Section 3.10.3, "Add a Policy Schedule," on page 59](#) and [Section 3.9.7, "Associate a Policy Schedule to a Policy," on page 54](#).

### Add Policy Parameter

#### **-name**

Specifies a name for the policy. The name must be unique on the server you are managing.

#### **Syntax**

```
-name=<"polycyname">
```

#### **Example**

```
-name="myPolicy"
```

### Add Policy Direction Option

Choose one of the direction options in combination with the `-add` action.

- ♦ [-primaryToSecondary](#)
- ♦ [-secondaryToPrimary](#)

#### **-primaryToSecondary**

Specifies the direction to move files. If a direction option is not specified, the default direction is primary to secondary.

**Restriction:** This option cannot be used with the `-secondaryToPrimary` option.

#### **-secondaryToPrimary**

Specifies the direction to move files. If a direction option is not specified, the default direction is primary to secondary.

The `-secondaryToPrimary` option is valid on a standard pair, but not on a retention pair.

**Restriction:** This option cannot be used with the `-primaryToSecondary` option.

### Add Policy Filter Options

Specify the following as filter options in combination with the `-add` action. Filter options are applied in combination to determine which files you want to move. All of the specified filter options must be met in order for the file to be moved.

You can specify one or more filter options in the same policy. Only one filter option of each type can be used in the same policy.

Filters set in the same policy are enforced as AND conditions. A file must meet all filter conditions to be moved. For example, if you specify a filter with the file extension option for files ending in \*.jpg and \*.gif, then any file with either of the specified extensions is moved. If you specify a second filter with a file size option for files with a file size greater than 5 MB, only the \*.jpg and \*.gif files that have a file size greater than 5 MB are moved.

Filters set in different policies that run at the same time are enforced as OR conditions. A file that meets the conditions in any one of the policies is moved. In the example above, if each of the filters was set in two separate policies and both policies run at the same time, then a file is moved if it ends in \*.jpg or \*.gif, or the file is moved if it is greater than 5 MB with any file extension.

The following filter options are available:

- ◆ -fileSize
- ◆ -lastAccessed
- ◆ -lastModified
- ◆ -fileExtension
- ◆ -fileType
- ◆ -fileOwner
- ◆ -groups

#### **-fileSize**

Specifies the file size of files to filter for movement.

##### **Syntax**

```
-fileSize=<"ccn[... ]uu">
```

Where:

cc = gt or lt conditional

n[... ] = any length numeric value

uu = units of size

Valid uu values are:

b=bytes

kb=kilobytes

mb=megabytes

gb=gigabytes

##### **Examples**

Move all files that are greater than 1 GB in size:

```
-fileSize="gt1gb"
```

Move all files that are less than 100 KB in size:

```
-fileSize="lt100kb"
```

#### **-lastAccessed**

Moves files that meet the specified condition based on the last accessed time.

##### **Syntax**

```
-lastAccessed=<"ccn[... ]u">
```

Where

cc = gt or lt conditional

n[ . . . ] = any length numeric value

u = units of time

Valid u values are:

d = days

w = weeks

m = months

y = years

### Examples

Move all files that have a last accessed time greater than 10 days old:

```
-lastAccessed="gt10d"
```

Move all files that have a last accessed time less than 5 weeks old:

```
-lastAccessed="lt5w"
```

### **-lastModified**

Moves files that meet the specified condition based on the last modified time.

#### Syntax

```
-lastModified=<"ccn[ . . . ]u">
```

Where

cc = gt or lt conditional

n[ . . . ] = any length numeric value

u = units of time

Valid u values are:

d = days

w = weeks

m = months

y = years

### Examples

Move all files that have a modified time greater than 10 days old:

```
-lastModified="gt10d"
```

Move all files that have a modified time less than 5 weeks old:

```
-lastModified="lt5w"
```

### **-fileExtension**

Moves files that have the specified file extensions. An asterisk (\*) can be used as a wildcard character. Separate multiple entries with a comma.

#### Syntax

```
-fileExtension=<"extensionList">
```

The `-fileextension` option has been aliased to `-filePattern`.

```
-filePattern="patternList"
```

### Examples

Move all .jpg files and .gif files:

```
-filePattern="*.jpg,*.gif"
```

```
-fileExtension="*.jpg,*.gif"
```

Move all files that begin with gw that have the file extension .mail:

```
-filepattern="gw*.mail"
```

Move all .txt files:

```
-filePattern="*.txt"
```

### -fileType

Moves files based on the file type. Separate multiple entries with a comma.

File types are the MIME content types or perceived types of file extensions that are associated with applications installed on the server. These types appear in the server's Windows Registry.

You can also use the perceived file types that are defined in the `Dynamic File Services\DswFileTypes.cfg` file. The file lists well-known file extensions and their perceived types. Each line in the `DswFileTypes.cfg` contains a file extension and its perceived type. For example:

```
.doc/document
```

You can customize the list by using a text editor to add or remove file extension entries in the proper format. The file types and extensions can be listed in any order.

### Syntax

```
-fileType=<"typeList">
```

Valid types are:

```
application  
audio  
compressed  
image  
message  
model  
system  
text  
video
```

### Examples

Move video and audio files:

```
-fileType="video,audio"
```

### -fileOwner

Move files based on identity.

You can specify one or more valid usernames in order to move files based on file ownership. For usernames, files are moved if they are owned by any of the specified users.

---

**IMPORTANT:** The CLI does not verify the usernames that you add. Make sure to enter valid usernames. Invalid usernames are ignored when the policy runs.

---

### Syntax

```
-fileOwner=<"userList">
```

### Example

```
-fileowner="user1,user2"
```

### -groups

Move files based on group identity.

You can specify one or more valid group names in order to move files based on file ownership. For usernames, files are moved if they are owned by any users listed as members of the group when the policy runs.

---

**IMPORTANT:** The CLI does not verify the group names that you add. Make sure to enter valid group names. Invalid group names are ignored when the policy runs.

---

### Syntax

```
-groups=<"groupList">
```

### Example

```
-groups="group1,group2"
```

## Add Policy Examples

Each of the following `-policy` commands creates a policy with one type of filter. The direction option is not set in these policies, so data is moved in the default direction of primary to secondary.

### Example: Policy to Filter by File Extension

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-policy
-add
-name="My Pictures Policy"
-fileExtension="*.jpg,*.bmp,*.gif"
```

### Example: Policy to Filter by File Size

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-policy
-add
-name="My Size GT 1GB Policy"
-fileSize="gt1gb"
```

### Example: Policy to Filter by Last Modified Date

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-policy
-add
-name="My Last Modified GT 10 Days Policy"
-lastModified="gt10d"
```

## 3.9.3 Modify a Policy

### **-modify**

The `-modify` action modifies a specified policy and stores the configuration changes in the XML file in the `C:\ProgramData\Dynamic File Services\Policies` folder.

### **Syntax**

```
DswCli.exe [authentication_parameters] -policy
    -modify
    -policyId=<"polycyname" | "GUID">
    [-name=<"polycyname>]
    [-description" text"]
    directionOption
    filterOptions
    frequencyOptions
```

For information about the options, see [Section 3.9.2, "Add a Policy,"](#) on page 46.

For modifying a policy, the required parameters are policy and policyID. Name, description, direction, and filter options can be changed. Policy GUIDs cannot be changed.

The changes apply to all pairs that are associated with the policy.

### **Modify Policy Examples**

Each of the following `-policy` commands modifies a policy that has one type of filter.

#### **Example: Policy to Filter by File Extension**

Modify a policy to add `.mp3` files to the list of file extensions. Change the policy name from "My Pictures Policy" to "My Pictures and Music Policy":

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
    -policy
    -modify
    -policyID="My Pictures Policy"
    -name="My Pictures and Music Policy"
    -fileExtension="*.jpg,*.bmp,*.gif"
```

#### **Example: Policy to Filter by File Size**

Modify a policy that moves files bigger than 1 GB to move only files bigger than 2 GB. Change the name to reflect the new setting:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
    -policy
    -modify
    -policyID="My Size GT 1GB Policy"
    -name="BIG FILES GT 2GB Policy"
    -fileSize="gt2gb"
```

#### **Example: Policy to Filter by Last Modified Date**

Modify a policy that moves files based on last modified date greater than 10 days to one greater than 10 months:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
    -policy
    -add
    -policyID="My Last Modified GT 10 Days Policy"
    -name="My Last Modified GT 10 Months Policy"
    -lastModified="gt10m"
```

## 3.9.4 Delete a Policy

### **-delete**

The `-delete` action removes the specified policy from the database. All links to any pairs are removed.

### **Syntax**

```
DswCli.exe [authentication_parameters] -policy
           -delete
           -policyId=<"polycyname" | "GUID">
```

### **Example**

The following `-policy` command deletes the DynamicFS policy named `myPolicy` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
           -policy
           -delete
           -policyId="myPolicy"
```

## 3.9.5 List All Policies

### **(no action options)**

When the `-policy` action option is used without any other parameters or options, all policies on the server are listed.

### **Syntax**

```
DswCli.exe [authentication_parameters] -policy
```

### **Example**

The following `-policy` command displays a list of all policies on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
           -policy
```

## 3.9.6 List Details for a Policy

### **-detail**

The `-detail` action provides a details for a specified policy, including the direction that files are moved, the filter options, the policy schedule, and its associated pairs. You must specify the `policyId` parameter for the policy.

### **Syntax**

```
DswCli.exe [authentication_parameters] -policy
           -detail
           -policyId=<"polycyname" | "GUID">
```

### **Example**

The following `-policy` command lists details for the DynamicFS policy named `myPolicy` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
           -policy
           -detail
           -policyId="myPolicy"
```

## 3.9.7 Associate a Policy Schedule to a Policy

### -associate

When the `-scheduleId` option is used with the `-associate` action for a policy, it links a specified policy schedule to a specified policy. You must specify the `policyId` parameter for the policy that you want to associate, and the `scheduleId` parameter for the schedule.

### Syntax

```
DswCli.exe [authentication_parameters] -policy
-associate
-policyId=<"policyname" | "GUID">
-scheduleId=<"schedulename" | "GUID">
```

### Example

The following `-policy` command associates the policy named `myPolicy` with the policy schedule named `Weekends` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-policy
-associate
-policyId="myPolicy"
-scheduleId="Weekends"
```

## 3.9.8 Disassociate a Policy Schedule from a Policy

### -disassociate

When the `-scheduleId` option is used with the `-disassociate` action for a policy, it removes the association between a specified policy and schedule. You must specify the `policyId` parameter for the policy. You must specify the `scheduleId` parameter for the schedule that you no longer want to be associated with the policy.

### Syntax

```
DswCli.exe [authentication_parameters] -policy
-disassociate
-policyId=<"policyname" | "GUID">
-scheduleId=<"schedulename" | "GUID">
```

### Example

The following `-policy` command removes the association between the policy named `myPolicy` and the schedule named `Weekends` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-policy
-disassociate
-policyId="myPolicy"
-scheduleId="Weekends"
```

## 3.9.9 Associate a Policy and a Pair

See [Section 3.8.6, "Associate a Pair and Policy,"](#) on page 40.

## 3.9.10 Disassociate a Policy and a Pair

See [Section 3.8.7, "Disassociate a Pair and Policy,"](#) on page 41.

## 3.9.11 List Policies Associated with a Pair

See [Section 3.8.11, “List Details for a Pair,”](#) on page 42.

## 3.10 Schedule Actions

### **-schedule**

Performs operations on a schedule. When it is used without other schedule action options, all schedules on the server are listed.

#### **Syntax**

```
DswCli.exe [authentication_parameters] -schedule  
          [-add | -delete | -modify | -detail]  
          [-scheduleId]  
          [schedule_option]
```

For information about how to provide the servername and login credentials that are needed to connect to the server that you want to manage, see [Section 3.3, “Authentication Parameters,”](#) on page 20.

- ◆ [Section 3.10.1, “Schedule Parameters,”](#) on page 55
- ◆ [Section 3.10.2, “Add a Review Schedule,”](#) on page 56
- ◆ [Section 3.10.3, “Add a Policy Schedule,”](#) on page 59
- ◆ [Section 3.10.4, “Delete a Schedule,”](#) on page 65
- ◆ [Section 3.10.5, “Modify a Review Schedule,”](#) on page 65
- ◆ [Section 3.10.6, “Modify a Policy Schedule,”](#) on page 67
- ◆ [Section 3.10.7, “List All Schedules,”](#) on page 68
- ◆ [Section 3.10.8, “List Details for a Schedule,”](#) on page 68
- ◆ [Section 3.10.9, “Associate a Review Schedule to a Retention Pair,”](#) on page 68
- ◆ [Section 3.10.10, “Disassociate a Review Schedule from a Retention Pair,”](#) on page 68
- ◆ [Section 3.10.11, “Associate a Policy Schedule to a Policy,”](#) on page 69
- ◆ [Section 3.10.12, “Disassociate a Policy Schedule from a Policy,”](#) on page 69

### 3.10.1 Schedule Parameters

#### **-description**

Specifies a textual description of the schedule. The description is optional.

#### **Syntax**

```
-description=<"text">
```

#### **Example**

```
-description="Custom schedule for the Finance department"
```

#### **-scheduleId**

Specifies the schedule identifier. You can provide the schedule name, or provide the GUID of the schedule. A GUID is automatically assigned by Dynamic File Services when you create a schedule.

### Syntax

```
-scheduleId=<"schedulename" | "GUID">
```

### Examples

```
-scheduleId="quarterly_1st"  
-scheduleId="custom_financial"
```

### -scheduleType

Specifies whether the type of schedule is a review schedule or a policy schedule.

### Syntax

```
-scheduleType=<"review" | "policy">
```

### Examples

```
-scheduleType="review"  
-scheduleType="policy"
```

## 3.10.2 Add a Review Schedule

### -add

The -add action with the “review” schedule type creates a review schedule with the specified name and stores the configuration as an XML file in the C:\ProgramData\Dynamic File Services\Schedules folder.

### Syntax

```
DswCli.exe [authentication_parameters] -SCHEDULE  
-ADD  
-name=<schedulename>  
-scheduleType="review"  
[-description=<"text">]  
[-monthly=<SEE FORMAT>]  
[-quarterly=<SEE FORMAT>]  
[-yearly=<SEE FORMAT>]  
[-custom=<SEE FORMAT>]
```

After you add a review schedule, you can associate it with one or more retention pairs. For information, see [Section 3.8.8, “Associate a Review Schedule to a Retention Pair,”](#) on [page 41](#).

### Add Review Schedule Parameter

#### -name

Specifies a name for the schedule. The name must be unique on the server you are managing.

#### Syntax

```
-name=<"schedulename">
```

#### Example

```
-name="quarterly_1st"
```

## Add Review Schedule Dates Options

Specify one of the following date options for the schedule. The dates indicate when you want to perform reviews. Notifications are sent by default at 12:10 a.m. (0010 hours) on that date. The default time is controlled by the setting in the `..\Dynamic File Services\DswCore.xml` file.

Use the "last" option to specify that the policy runs on the last day of every month. The policy runs on day 28, 29, 30, or 31, according to how many days are in the month.

### **-custom**

Specifies the custom dates that you want to review the retained data in a retention pair.

#### **Syntax**

```
-Custom=<"dd:nn:<yy|yyyy>[ ,dd:nn:<yy|yyyy> , . . . ]">
```

Where:

dd = calendar day of the month (01 to 31)

nn = month (01 to 12)

yy or yyyy = year (e.g., 11 or 2011 is the year 2011)

#### **Example**

The following example sends review notifications for a retention pair so that reviews can be held on March 30, 2011, April 15, 2011, and June 30, 2011:

```
-custom="30:03:2011,15:04:11,30:06:11"
```

### **-monthly**

Specifies a monthly review frequency. You must specify the month and calendar day to begin the review cycle. Reviews are planned on the same calendar day in each month. The cycle repeats with no end time.

#### **Syntax**

```
-monthly=<"<dd|'last':nn">
```

Where:

dd = calendar day of the month (01 to 31)

last = the last calendar day of the month (e.g., 28, 29, 30, or 31 as appropriate)

nn = month (01 to 12)

#### **Example**

The following example sends review notifications for a retention pair so that reviews can be held on the last day of every month, beginning in June of the current year. Reviews are planned for June 30, July 31, August 31, September 30, October 31, November 30, December 31, January 31, February 28 (or 29 in a leap year), March 31, April 30, and May 31. The cycle repeats with no end time.

```
-monthly="last:06"
```

### **-quarterly**

Specifies a quarterly review frequency. You must specify the month and calendar day to begin the review cycle. Reviews are planned on the same calendar day every three months. The cycle repeats with no end time.

### Syntax

```
-quarterly=<"<dd|'last'>:nn">
```

Where:

dd = calendar day of the month (01 to 31)

last = the last calendar day of the month (e.g., 28, 29, 30, or 31 as appropriate)

nn = month (01 to 12)

### Example

The following example sends review notifications for a retention pair so that reviews can be held on the last day of every quarter, beginning in June of the current year. Reviews are planned for June 30, September 30, December 31, and March 31. The cycle repeats with no end time.

```
-quarterly="last:06"
```

### -yearly

Specifies a yearly frequency. You must specify the month and calendar day to begin the review cycle. Reviews are planned on the same calendar day in each year. The cycle repeats with no end time.

### Syntax

```
-yearly=<"<dd|'last'>:nn">
```

Where:

dd = calendar day of the month (01 to 31)

last = the last calendar day of the month (e.g., 28, 29, 30, or 31 as appropriate)

nn = month (01 to 12)

### Example

The following example sends review notifications for a retention pair so that reviews can be held on January 15 of each year, beginning in January of the current year. Reviews are planned for January 15. The cycle repeats with no end time.

```
-yearly="15:01"
```

## Add Review Schedule Examples

Each of the following `-schedule` commands creates a review schedule with the frequency specified. A schedule must be separately associated with a retention pair.

### Example: Add a Schedule to Review Retained Data on Custom Dates

The following example sends review notifications for a retention pair so that reviews can be held on March 30, 2011, April 15, 2011, June 30, 2011, and January 2, 2012:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell  
-schedule  
-add  
-name="custom_financial"  
-description="Custom schedule for the Finance department"  
-custom="30:03:2011,15:04:11,30:06:11,02:01:12"
```

### Example: Add a Schedule to Review Retained Data Quarterly

The following example sends review notifications for a retention pair so that reviews can be held quarterly, beginning on July 1 of the current year. Reviews are planned for July 1, October 1, January 1, and April 1 of each year.

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-schedule
-add
-name="quarterly_1st"
-quarterly="01:07"
```

## 3.10.3 Add a Policy Schedule

### -add

The -add action with the “policy” schedule type creates a policy schedule with the specified name and stores the configuration as an XML file in the C:\ProgramData\Dynamic File Services\Schedules folder.

### Syntax

```
DswCli.exe [authentication_parameters] -SCHEDULE
-ADD
-name=<schedulename>
-scheduleType="policy"
[-description=<"text">]
[-time=<SEE FORMAT>]
[-hourly=<SEE FORMAT>]
[-daily=<SEE FORMAT>]
[-weekly=<SEE FORMAT>]
[-monthly=<SEE FORMAT>]
[-quarterly=<SEE FORMAT>]
[-yearly=<SEE FORMAT>]
[-custom=<SEE FORMAT>]
```

After you add a policy schedule, you can associate it with one or more policies. For information, see [Section 3.9.7, “Associate a Policy Schedule to a Policy,” on page 54](#).

### Add Policy Schedule Parameter

#### -name

Specifies a name for the schedule. The name must be unique on the server you are managing.

#### Syntax

```
-name=<"schedulename">
```

#### Example

```
-name="quarterly_1st"
```

### Add Policy Schedule Frequency Options

Specify only one frequency option in combination with the -add action. The specified frequency option determines when the is enforced. The policy is enforced for all pairs associated with the policy.

You can specify the start and stop time by using the -time option or by specifying the times as part of the frequency parameter.

The following frequency options are available:

- ♦ [time](#)

- ♦ -manual
- ♦ -hourly
- ♦ -daily
- ♦ -weekly
- ♦ -monthly
- ♦ -quarterly
- ♦ -yearly
- ♦ -custom

#### **-time**

If the `-time` option is not used, or if the time is not specified by a frequency option, a default start time of 00:00 (midnight) is assumed and the scan runs until complete.

If the start time is provided and the stop time is omitted, the scan starts at the given time and runs until complete.

#### **Syntax**

```
-time=<"hh[:mm[-hh:mm]]">
```

hh[:mm[-hh:mm]] specifies the scan start and stop time. Twenty-four-hour time is required with valid values of 00:00 to 23:45 (that is, midnight to 11:45 p.m.). The first two units specify hours from 00 to 23. The second two units specify minutes in 15-minute increments of 00, 15, 30, or 45.

**Restrictions:** This option cannot be combined with the following frequency options:

```
-manual
-hourly
```

This option cannot be used in combination with other frequency options where the time is set explicitly as a parameter for the frequency.

#### **-manual**

Scan frequency is performed manually.

**Restrictions:** This option cannot be combined with the following frequency options:

```
-time
-hourly
-daily
-weekly
-monthly
-quarterly
-yearly
-custom
```

#### **-hourly**

Scan frequency is performed hourly.

**Restrictions:** This option cannot be combined with the following frequency options:

```
-time
-manual
-daily
```

- weekly
- monthly
- quarterly
- yearly
- custom

### **-daily**

Scan frequency is performed daily.

If the start and stop time are omitted, a default start time of 00:00 (midnight) is assumed and the scan runs until complete.

If the start time is provided and the stop time is omitted, the scan starts at the given time and runs until complete.

#### **Syntax**

```
-daily[="hh:mm[-hh:mm]" ]
```

hh:mm[-hh:mm] specifies the scan start and stop time, respectively. Twenty-four-hour time is required with valid values of 00:00 to 23:45 (that is, midnight to 11:45 p.m.). The first two units specify hours from 00 to 23. The second two units specify minutes in 15-minute increments of 00, 15, 30, or 45.

**Restrictions:** This option cannot be combined with the following frequency options:

- manual
- hourly
- weekly
- monthly
- quarterly
- yearly
- custom

#### **Example**

The scan runs daily from 1:00 p.m. to 3:00 p.m.:

```
-daily="13:00-15:00"
```

### **-weekly**

Scan frequency is performed weekly.

If the day of the week and the start and stop times are omitted, the scan starts on Sunday at 00:00 (12:00 midnight), and the scan runs until complete.

If the start and stop time are omitted, a default start time of 00:00 (midnight) is assumed, and the scan runs until complete.

If the start time is provided and the stop time is omitted, the scan starts at the given time and runs until complete.

#### **Syntax**

```
-weekly[="ww|[@hh:mm[-hh:mm]]"]
```

Where

ww specifies the day of the week. Valid values are:

01 = Sunday

02 = Monday  
03 = Tuesday  
04 = Wednesday  
05 = Thursday  
06 = Friday  
07 = Saturday

@hh:mm[ -hh:mm] specifies the scan start and stop time. Twenty-four-hour time is required with valid values of 00:00 to 23:45 (that is, midnight to 11:45 p.m.). The first two units specify hours from 00 to 23. The second two units specify minutes in 15-minute increments of 00, 15, 30, or 45.

**Restrictions:** This option cannot be combined with the following frequency options:

-manual  
-hourly  
-daily  
-monthly  
-quarterly  
-yearly  
-custom

### Examples

The scan runs every Monday from 1:00 p.m. until complete:

```
-weekly="02@13:00"
```

The scan runs every Monday from 1:00 a.m. to 5:00 a.m.:

```
-weekly="02@01:00-05:00"
```

### -monthly

Scan frequency is performed monthly.

If the day of the month and the start and stop times are omitted, the scan starts on first day of the month at 00:00 (midnight) and runs until complete.

If the start and stop time are omitted, a default start time of 00:00 is assumed and the scan runs until complete.

If the start time is provided and the stop time is omitted, the scan starts at the given time and runs until complete.

### Syntax

```
-monthly[="<dd | 'last'>[@hh:mm-hh:mm]]"
```

dd specifies the day of the month numerically. Valid values are from 01 to 31.

last specifies the last day of the calendar month (that is, 28, 29, 30, or 31 as appropriate).

@hh:mm[ -hh:mm] specifies the scan start and stop time. Twenty-four-hour time is required with valid values of 00:00 to 23:45 (that is, midnight to 11:45 p.m.). The first two units specify hours from 00 to 23. The second two units specify minutes in 15-minute increments of 00, 15, 30, or 45.

**Restrictions:** This option cannot be combined with the following frequency options:

-manual

- hourly
- daily
- weekly
- quarterly
- yearly
- custom

### Example

The scan runs on the 25th day of the month from 1:00 p.m. until complete:

```
-monthly="25@13:00"
```

### -quarterly

Scan frequency is performed quarterly. You must specify the month and calendar day to begin the run. Reviews are planned on the same calendar day every three months. The cycle repeats with no end time.

### Syntax

```
-quarterly=<"<dd|'last'>:nn">
```

Where:

dd = calendar day of the month (01 to 31)

last = the last calendar day of the month (that is, 28, 29, 30, or 31 as appropriate)

nn = month (01 to 12)

**Restrictions:** This option cannot be combined with the following frequency options:

- manual
- hourly
- daily
- weekly
- monthly
- yearly
- custom

### Example

Run the policy on the last day of every quarter, beginning in June of the current year. The scans occur on June 30, September 30, December 31, and March 31. The cycle repeats with no end time.

```
-quarterly="last:06"
```

### -yearly

Scan frequency is performed yearly.

The day of the month and the month fields are required to be specified. There are no defaults specified.

If the start and stop times are omitted, a default start time of 00:00 is assumed and the scan runs until complete.

If start time is provided and the stop time is omitted, the scan starts at the given time and runs until complete.

## Syntax

```
-yearly=<"dd:nn[@hh:mm[-hh:mm]]">
```

dd specifies the day of the month numerically. Valid values are from 01 to 31.

nn specifies the month numerically. Valid values are 01 to 12, where the numbers correspond to the sequential months of the year in the Gregorian calendar.

@hh:mm[-hh:mm] specifies the scan start and stop time. Twenty-four-hour time is required with valid values of 00:00 to 23:45 (that is, midnight to 11:45 p.m.). The first two units specify hours from 00 to 23. The second two units specify minutes in 15-minute increments of 00, 15, 30, or 45.

**Restrictions:** This option cannot be combined with the following frequency options:

```
-manual  
-hourly  
-daily  
-weekly  
-monthly  
-quarterly  
-custom
```

## Example

The scan runs on day 15, month 6, starting at 1:00 p.m. and running until complete:

```
-yearly="15:06@13:00"
```

## -custom

Specifies the custom dates that you want to run the policy.

## Syntax

```
-custom=<"dd:nn:<yy|yyyy>[ ,dd:nn:<yy|yyyy> , . . . ]">
```

Where:

dd = calendar day of the month (01 to 31)

nn = month (01 to 12)

yy or yyyy = year (e.g., 11 or 2011 is the year 2011)

**Restrictions:** This option cannot be combined with the following frequency options:

```
-manual  
-hourly  
-daily  
-weekly  
-monthly  
-quarterly  
-yearly
```

## Example

Schedule a policy to run on March 30, 2011, April 15, 2011, and June 30, 2011:

```
-custom="30:03:2011,15:04:11,30:06:11"
```

## Add Policy Schedule Examples

Each of the following `-schedule` commands creates a policy schedule with the frequency specified. A policy schedule is separately associated with one or more pairs.

### Example: Weekly Policy Schedule

The following example specifies the start and stop times as part of the frequency parameter:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-schedule
-add
-scheduleType="policy"
-name="Weekly on Sunday"
-weekly="01@01:00-05:00"
```

The following example specifies the start and stop times using the `-time` option:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-schedule
-add
-scheduleType="policy"
-name="Weekly on Sunday"
-time="01:00-05:00"
-weekly="01"
```

### Example: Monthly Policy Schedule

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-schedule
-add
-scheduleType="policy"
-name="Monthly Last Day of the Month"
-time="01:30"
-monthly="last"
```

## 3.10.4 Delete a Schedule

### -delete

The `-delete` action removes the specified schedule from the database. All links to any retention pairs are removed.

### Syntax

```
DswCli.exe [authentication_parameters] -schedule
-delete
-scheduleID=<"schemulename"|"GUID">
```

### Delete Schedule Example

Delete a schedule named `custom_financial`:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-schedule
-delete
-scheduleID="custom_financial"
```

## 3.10.5 Modify a Review Schedule

### -modify

The `-modify` action changes the settings for an existing schedule. You cannot modify the schedule type. The configuration changes are written to the schedule's XML file in the `C:\ProgramData\Dynamic File Services\Schedules` folder.

## Syntax

```
DswCli.exe [authentication_parameters] -schedule
    -modify
    -scheduleID=<"schemulename" | "GUID">
    [-name=<schemulename>]
    [-description=<"text">]
    [-custom=<SEE FORMAT>]
    [-monthly=<SEE FORMAT>]
    [-quarterly=<SEE FORMAT>]
    [-yearly=<SEE FORMAT>]
```

For modifying a schedule, the required parameters are schedule and scheduleID. Name, description, frequency, and time options can be changed. Schedule GUIDs and types cannot be changed.

The changes apply to all retention pairs that are associated with the review schedule.

### Modify Schedule Dates Options

For information, see the “Add Schedule Dates Option” in [“Add a Review Schedule” on page 56](#).

### Modify Schedule Examples

Each of the following `-schedule` commands modifies an existing schedule with the frequency specified. The changes apply to all retention pairs associated with the schedule.

#### Example: Modify a Schedule to Review Retained Data on Custom Dates

A custom schedule is currently set for the dates March 30, 2011, April 15, 2011, June 30, 2011, and January 2, 2012. In December 2011, you decide to use the same dates for 2012. You must create a new schedule, or modify the existing custom dates. Make sure to account for the pending date in January. The revised schedule has five custom dates: January 2, 2012, March 30, 2012, April 15, 2012, June 30, 2012, and January 2, 2013.

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
    -schedule
    -modify
    -scheduleID="custom_financial"
    -description="Custom schedule for the Finance department"
    -custom=
        "02:01:12,30:03:2012,15:04:12,30:06:12,02:01:13"
```

#### Example: Modify a Schedule to Review Retained Data Quarterly

A current schedule is set for the first day of each quarter, beginning on July 1 of the current year. You want to modify the schedule to the 15th. The schedule name of `quarterly_1st` can be modified, such as `quarterly_15th`. You decide to modify the dates anyway. The revised schedule is planned for July 15, October 15, January 15, and April 15.

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
    -schedule
    -modify
    -scheduleID="quarterly_1st"
    -name="quarterly_15th"
    -quarterly="15:07"
```

## 3.10.6 Modify a Policy Schedule

### -modify

The -modify action changes the settings for an existing schedule. You cannot modify the schedule type. The configuration changes are written to the schedule's XML file in the C:\ProgramData\Dynamic File Services\Schedules folder.

#### Syntax

```
DswCli.exe [authentication_parameters] -schedule
    -modify
    -scheduleID=<"schemulename" | "GUID">
    [-name=<schemulename>]
    [-description=<"text">]
    [-time=<SEE FORMAT>]
    [-hourly=<SEE FORMAT>]
    [-daily=<SEE FORMAT>]
    [-weekly=<SEE FORMAT>]
    [-monthly=<SEE FORMAT>]
    [-quarterly=<SEE FORMAT>]
    [-yearly=<SEE FORMAT>]
    [-custom=<SEE FORMAT>]
```

For modifying a schedule, the required parameters are schedule and scheduleID. Name, description, frequency and time options can be changed. Schedule GUIDs and types cannot be changed.

The changes apply to all policies that are associated with the policy schedule.

#### Modify Schedule Dates Options

For information, see the "Add Schedule Dates Option" in ["Add a Policy Schedule" on page 59](#).

#### Modify Schedule Examples

Each of the following -schedule commands modifies an existing policy schedule with the frequency specified. The changes apply to all policies associated with the schedule.

##### Example: Modify a Schedule to Run on Custom Dates

A custom schedule is currently set to run on the dates March 30, 2011, April 15, 2011, June 30, 2011, and January 2, 2012. In December 2011, you decide to use the same dates for 2012. You must create a new schedule, or modify the existing custom dates. Make sure to account for the pending date in January. The revised schedule has five custom dates: January 2, 2012, March 30, 2012, April 15, 2012, June 30, 2012, and January 2, 2013.

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
    -schedule
    -modify
    -scheduleID="custom_financial_policy"
    -description="Custom policy schedule for the Finance
department"
    -custom=
        "02:01:12,30:03:2012,15:04:12,30:06:12,02:01:13"
```

##### Example: Modify a Policy Schedule to Run Quarterly on a Different Day of the Month

A current policy schedule is set to start a run on the first day of each quarter, beginning on July 1 of the current year. You want to modify the schedule to the 15th. The schedule name of quarterly\_1st can be modified, such as quarterly\_15th. You decide to

modify the dates rather than creating a new schedule. The revised policy schedule is planned for July 15, October 15, January 15, and April 15. Any policies associated with the schedule runs on those dates.

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-schedule
-scheduleID="quarterly_1st"
-modify
-name="quarterly_15th"
-quarterly="15:07"
```

### 3.10.7 List All Schedules

#### (no action options)

When `-schedule` action option is used without any other parameters or options, all review schedules on the server are listed.

#### Syntax

```
DswCli.exe [authentication_parameters] -schedule
```

#### Example

The following `-schedule` command displays a list of all schedules on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-schedule
```

### 3.10.8 List Details for a Schedule

#### -detail

The `-detail` action provides a details for a schedule, including its schedule type and frequency. For review schedules, the associated pairs are identified. For policy schedules, the associated pairs are identified. You must specify the `scheduleId` parameter for the policy.

#### Syntax

```
DswCli.exe [authentication_parameters] -schedule
-detail
-scheduleId=<"schedulename" | "GUID">
```

#### Example

The following `-schedule` command lists details for the schedule named `custom_financial` on the specified server:

```
DswCli.exe -servername=localhost -u=Administrator -p=novell
-schedule
-detail
-scheduleId="custom_financial"
```

### 3.10.9 Associate a Review Schedule to a Retention Pair

See [Section 3.8.8, "Associate a Review Schedule to a Retention Pair,"](#) on page 41.

### 3.10.10 Disassociate a Review Schedule from a Retention Pair

See [Section 3.8.9, "Disassociate a Review Schedule from a Retention Pair,"](#) on page 42.

## 3.10.11 Associate a Policy Schedule to a Policy

See [Section 3.9.7, “Associate a Policy Schedule to a Policy,”](#) on page 54.

## 3.10.12 Disassociate a Policy Schedule from a Policy

See [Section 3.9.8, “Disassociate a Policy Schedule from a Policy,”](#) on page 54.

# 3.11 Using CLI Commands in Scripts

When you use Dynamic File Services commands in scripts, you can use the `-silent` option to prevent the results from displaying to a console. The script should check for an exit code to report whether a command succeeded or failed. The `-silent` option can be added to any pair or policy command.

### **-silent**

Use this option when you do not want to display errors or help on execution, such as in scripts for batch files where the batch file checks for exit codes.

If a command fails, you can enter the command at the Command Prompt console without the `-silent` option to view the error messages.

### **Syntax**

```
DswCli.exe [authentication_parameters] -pair  
-silent  
[pair_actions]  
[pair_parameters]  
[pair_options]
```

```
DswCli.exe [authentication_parameters] -policy  
-silent  
[policy_actions]  
[policy_parameters]  
[policy_options]
```

For information about the Dynamic File Services command options, see the following resources:

Option	Reference
authentication_parameters	<a href="#">Section 3.3, “Authentication Parameters,”</a> on page 20
-pair pair_actions pair_parameters pair_options	<a href="#">Section 3.8, “Pair Actions,”</a> on page 33
-policy policy_actions policy_parameters policy_options	<a href="#">Section 3.9, “Policy Actions,”</a> on page 45

### **Exit Codes**

The `DswCli.exe` command returns an exit code of 0 for success or 1 for failure.

## 3.12 Example Scripts Using CLI Commands

This section provides example scripts for Dynamic File Services.

- ♦ [Section 3.12.1, “Creating a Pair and Policy,” on page 70](#)
- ♦ [Section 3.12.2, “Running the Policies Manually,” on page 71](#)
- ♦ [Section 3.12.3, “Deleting the Policies,” on page 71](#)
- ♦ [Section 3.12.4, “Deleting the Pair Relationship,” on page 71](#)

### 3.12.1 Creating a Pair and Policy

This example uses the Dynamic File Services CLI to set up one pair named `myPair` with a primary path of `e:\Primary` and a secondary path of `f:\Secondary`. It creates three policies and associates them to the pair:

- ♦ The `GRAPHICS` policy moves graphic files to the secondary path. It is scheduled to run daily at midnight.
- ♦ The `OLD FILES` policy moves files that have not been modified within two weeks to the secondary path. It is scheduled to run weekly on Saturday at midnight.
- ♦ The `MODIFIED FILES` policy moves files modified during the past day from the secondary to the primary. It is scheduled to run hourly.

```
ECHO Create Dynamic File Services Setup of one pair and three policies
```

```
ECHO Create the pair named myPair
```

```
DswCli.exe -pair -add -name=myPair -primaryPath=e:\Primary  
-secondaryPath=f:\Secondary -description="Pair linking user home directories and  
old storage" -serverName=192.168.1.3 -port=8999 -userName=Administrator  
-password=myPassword
```

```
ECHO Create the policy named GRAPHICS to secondary
```

```
DswCli.exe -policy -add -name="GRAPHICS to secondary" -  
fileExtension=JPG,JPEG,BMP,GIF,PNG,RAW -daily="00:00" -primaryToSecondary  
-description="Moves all graphic files to secondary storage. Runs daily at  
midnight." -serverName=192.168.1.3 -port=8999 -userName=Administrator  
-password=myPassword
```

```
ECHO Associate the policy to myPair
```

```
DswCli.exe -associate -policyId="GRAPHICS to secondary" -pairId=myPair  
-serverName=192.168.1.3 -port=8999 -userName=Administrator  
-password=myPassword
```

```
ECHO Create the policy named OLD FILES to secondary
```

```
DswCli.exe -policy -add -name="OLD FILES to secondary" -lastModified="gt2w"  
-primaryToSecondary -description="Moves files that have not been modified for over  
two weeks to secondary. Runs weekly on Saturday morning at midnight" -  
weekly="05@00:00" -serverName=192.168.1.3  
-port=8999 -userName=Administrator -password=myPassword
```

```
ECHO Associate the policy to myPair
```

```
DswCli.exe -associate -policyId="OLD FILES to secondary" -pairId=myPair  
-serverName=192.168.1.3 -port=8999 -userName=Administrator  
-password=myPassword
```

ECHO Create the policy named MODIFIED FILES to primary

```
DswCli.exe -policy -add -name="MODIFIED FILES to primary" -lastModified="ltld" -primaryToSecondary -description="Moves files that were modified within the last day to the primary. Runs hourly every day." -hourly -serverName=192.168.1.3 -port=8999 -userName=Administrator -password=myPassword
```

ECHO Associate the policy to myPair

```
DswCli.exe -associate -policyId="MODIFIED FILES to primary" -pairId=myPair -serverName=192.168.1.3 -port=8999 -userName=Administrator -password=myPassword
```

## 3.12.2 Running the Policies Manually

The following command uses a Dynamic File Services CLI command to run the three policies manually for the pair named myPair:

ECHO Manually run all policies

```
DswCli.exe -execute -policyIdList="GRAPHICS FILES to secondary","MODIFIED FILES to primary","OLD FILES to secondary" -pairId=myPair -serverName=192.168.1.3 -port=8999 -userName=Administrator -password=myPassword
```

## 3.12.3 Deleting the Policies

The following example uses the Dynamic File Services CLI to delete the three policies. The policies' associations to the pair are also deleted.

ECHO Delete the policies

```
DswCli.exe -policy -delete -policyId="GRAPHICS to secondary" -serverName=192.168.1.3 -port=8999 -userName=Administrator -password=myPassword
```

```
DswCli.exe -policy -delete -policyId="OLD FILES to secondary" -serverName=192.168.1.3 -port=8999 -userName=Administrator -password=myPassword
```

```
DswCli.exe -policy -delete -policyId="MODIFIED FILES to primary" -serverName=192.168.1.3 -port=8999 -userName=Administrator -password=myPassword
```

## 3.12.4 Deleting the Pair Relationship

The following example uses the Dynamic File Services CLI to delete the link between the primary path and secondary path. No files are deleted.

```
DswCli.exe -pair -delete -pairId="myPair" -serverName=192.168.1.3 -port=8999 -userName=Administrator -password=myPassword
```



---

# 4 Dynamic File Services File System Inventory Utility

The Novell Dynamic File Services (DynamicFS) File System Inventory utility (`DswInventory.exe`) creates an inventory of the files in a pair, and writes the report to a file. On computers where the DynamicFS Service is installed, the File System Inventory utility is found in the directory where you installed DynamicFS. By default, DynamicFS is installed in the `C:\Program Files\Dynamic File Services` directory. The output inventory files are written in XML format. You can use a text editor or an XML editor to display the output of the inventories that you run manually.

---

**IMPORTANT:** A file system inventory of each pair is automatically generated daily with the pair history run. For information, see [“Viewing the Pair History”](#) in the *Dynamic File Services 2.0 Administration Guide*.

---

- ♦ [Section 4.1, “DswInventory,”](#) on page 73
- ♦ [Section 4.2, “Preparing a Command File for a File System Inventory,”](#) on page 74
- ♦ [Section 4.3, “Example of a General Inventory,”](#) on page 77
- ♦ [Section 4.4, “Example of a Filtered Inventory,”](#) on page 81
- ♦ [Section 4.5, “Additional Information,”](#) on page 84

## 4.1 DswInventory

- ♦ [Section 4.1.1, “Description,”](#) on page 73
- ♦ [Section 4.1.2, “Syntax,”](#) on page 74

### 4.1.1 Description

The Dynamic File Services File System Inventory utility is used to create an inventory of the file system and write the desired output to a file. You must create a command file for each report that you want to create that specifies the options to use and the output file where you want write the results.

For information about creating the command file, see [Section 4.2, “Preparing a Command File for a File System Inventory,”](#) on page 74.

## 4.1.2 Syntax

Log in to the server as the Administrator user or as a user with Administrator privileges. It does not matter if the user is also a member of the Dynamic File Services group. Navigate to the C:\Program Files\Dynamic File Services folder (or the folder where you installed Dynamic File Services), then issue the command.

```
DswInventory.exe commandFile [commandFile2...]
```

---

**IMPORTANT:** If remote shares are used in pairs, you must log in as a domain user with Administrator privileges on the DynamicFS server that also has Active Directory rights on the remote shares and NTFS file system access rights on the secondary storage locations. Otherwise, a secondary location is reported as missing. One way to do this is to add the administrator user as a member of the Dynamic File Services Storage Rights group.

---

## 4.2 Preparing a Command File for a File System Inventory

Before you can use the Dynamic File Services File System Inventory tool, you must create a text file (referred to as the command file) where you specify the path to be inventoried, the type of reports to create, the filenames to use for the output reports, and the inventory options to use. You can create multiple command files to generate different inventory reports that meet the criteria that are specified in the different command files.

In a command file, specify the following information about the inventory report that you want to create. Put each command on a separate line in the following order:

```
inventory_path report_type  
[inventory_option]
```

---

**IMPORTANT:** If there are errors in any parameter that you specify, the generated report files are empty.

---

The syntax for each of the required and optional elements is described in more detail in the following sections:

- [Section 4.2.1, “Inventory Path,” on page 74](#)
- [Section 4.2.2, “Report Types and Filenames,” on page 75](#)
- [Section 4.2.3, “Action List Filename,” on page 75](#)
- [Section 4.2.4, “Inventory Options,” on page 75](#)

### 4.2.1 Inventory Path

Specify the path that you want to inventory for this report. The inventory path can be the primary path for the pair or for any path in the pair.

```
inventory_path
```

Specify the full path, including the drive letter. For example:

```
C:\DATA\dir1\dir2
```

A command file can contain multiple paths. Place each path on a separate line. The combined output is written to the same report file. To write the inventory for each path to separate output files, create a separate command file for each path.

## 4.2.2 Report Types and Filenames

You can generate a summary report, a detailed report, or both reports. The summary report contains the statistics about the files, but does not contain the filenames. The detailed report includes both statistics and the matching filenames.

Specify a different filename for each report. The output files are in the `.xml` format.

Report Option	Description
<code>/summary=summary_report_filename.xml</code>	After an inventory is performed, creates a summary report file with the specified filename.
<code>/detailed=detailed_report_filename.xml</code>	After an inventory is performed, creates a detailed report file with the specified filename.

For example:

```
/summary=DATA_summary.xml  
/detailed=DATA_detailed.xml
```

## 4.2.3 Action List Filename

You can request that a list of the filtered files be written to a separate file by using the `/actionlist` parameter. Specify a filename for a file where a list of files will be written.

Action List Option	Description
<code>/actionlist=filename</code>	Requests a list of files to be generated in the specified file.
<code>/actionlistappend</code>	Append the list of files to the specified action list file instead of overwriting its existing content.

## 4.2.4 Inventory Options

Specify one inventory option per line. If no inventory options are specified, then all options are enabled for the report. The inventory results are written to the report files that you requested.

You can specify none, one, or multiple general inventory options in any combination:

<b>Inventory Option</b>	<b>Description</b>
<code>/create</code>	Inventories files by creation time.
<code>/modify</code>	Inventories files by modification time.
<code>/owner</code>	Inventories files by owner name.
<code>/size</code>	Inventories files by file length.
<code>/extension</code>	Inventories files by file extension.
<code>/empty</code>	Tracks empty files and folders and adds the list to the report.

If you want to specify constraints for any one of the inventory options, you must specify constraints for each of the inventory options that you use in the same command file. For each parameter type, you can specify only one of its constraints.

In the following commands, the time duration notation `ddd.hh:mm:ss` indicates days (0 to 999), hours (0 to 24), minutes (0 to 60), and seconds (0 to 60). You can also specify days (`ddd`) alone.

<b>Inventory Option</b>	<b>Description</b>
<code>/create&gt;ddd.hh:mm:ss</code>	Inventories files by creation times that are greater than the specified time duration.
<code>/create&lt;ddd.hh:mm:ss</code>	Inventories files by creation times that are less than the specified time duration.
<code>/modify&gt;ddd.hh:mm:ss</code>	Inventories files by modification times that are greater than the specified time duration.
<code>/modify&lt;ddd.hh:mm:ss</code>	Inventories files by modification times that are less than the specified time duration.
<code>/owner=name[,nextName...]</code>	Inventories files by the specified owner name or names. Delimit multiple names with a comma and no spaces.
<code>/size&gt;amount</code>	Inventories files by file length that is greater than the amount specified. Specify the amount in bytes.  For example, to inventory files greater than 1 MB, specify:  <code>/size&gt;1024</code>
<code>/size&lt;amount</code>	Inventories files by file length that is less than the amount specified. Specify the amount in bytes.  For example, to inventory files less than 1 MB, specify:  <code>/size&lt;1024</code>
<code>/extension=*.extension[,*.extension2...]</code>	Inventories files with the specified file extension. Separate entries with a comma and no spaces.

## 4.3 Example of a General Inventory

In this example, assume that you want an inventory for the Dynamic File Services pair that has a primary path of e:\. You perform the following tasks to manually generate a report:

- ♦ [Section 4.3.1, “Create a Command File,” on page 77](#)
- ♦ [Section 4.3.2, “Run the File System Inventory Utility,” on page 77](#)
- ♦ [Section 4.3.3, “View the Summary Report,” on page 77](#)
- ♦ [Section 4.3.4, “View the Detailed Report,” on page 78](#)

### 4.3.1 Create a Command File

Use a text editor to create a command file called `FScommands.txt` in the `Dynamic File Services` folder. The first line is the primary path for the pair. The second line requests a summary report to be created and specifies the output file for the summary report. The third line requests a detailed report to be created and specifies the output file for the detailed report. No inventory options are specified, so all inventory options are enabled.

```
e:\
/summary=e:\mySummary.xml
/detailed=e:\myDetails.xml
```

### 4.3.2 Run the File System Inventory Utility

- 1 Log in to the server as the Administrator user, or as a user in the Administrators group.
- 2 Open a Command Prompt console, then go to the `Dynamic File Services` folder.
- 3 At the prompt, enter

```
DswInventory.exe FSCommands.txt
```

### 4.3.3 View the Summary Report

The summary results are written to the `e:\mySummary.xml` file. The following is sample output of a summary report:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
- <DswInventory GenerationDateTime="8/27/2009 3:10:37 PM" generation="summary"
platform="Win32NT" OS="Microsoft Windows NT 5.2.3790 Service Pack 2">
- <folderlist>
- <folder>
- <![CDATA[ e:\
  ]]>
  </folder>
</folderlist>
- <list type="Modified">
  <criteria filter="Within Last Day" filecount="31" totalsize="74811"
sizeunit="1024" />
</list>
- <list type="Creation">
  <criteria filter="Within Last Day" filecount="31" totalsize="74811"
sizeunit="1024" />
</list>
- <list type="EmptyFiles">
  <criteria filter="Empty Files" filecount="11" totalsize="0" sizeunit="0" />
</list>
- <list type="Extension">
```

```

<criteria filter=".10file" filecount="10" totalsize="0" sizeunit="1024" />
<criteria filter=".doc" filecount="1" totalsize="57" sizeunit="1024" />
<criteria filter=".docx" filecount="1" totalsize="57" sizeunit="1024" />
<criteria filter=".exe" filecount="1" totalsize="16" sizeunit="1024" />
<criteria filter=".gif" filecount="1" totalsize="888" sizeunit="1024" />
<criteria filter=".jpeg" filecount="1" totalsize="468" sizeunit="1024" />
<criteria filter=".jpg" filecount="5" totalsize="2569" sizeunit="1024" />
<criteria filter=".ods" filecount="1" totalsize="3" sizeunit="1024" />
<criteria filter=".odt" filecount="4" totalsize="828" sizeunit="1024" />
<criteria filter=".pdf" filecount="1" totalsize="641" sizeunit="1024" />
<criteria filter=".rtf" filecount="1" totalsize="1" sizeunit="1024" />
<criteria filter=".txt" filecount="1" totalsize="0" sizeunit="1024" />
<criteria filter=".xml" filecount="2" totalsize="6" sizeunit="1024" />
<criteria filter=".zip" filecount="1" totalsize="69277" sizeunit="1024" />
</list>
- <list type="Modified">
  <criteria filter="Within Last Day" filecount="3" totalsize="69283"
sizeunit="1024" />
  <criteria filter="1 Month - 2 Months" filecount="1" totalsize="16"
sizeunit="1024" />
  <criteria filter="6 Months - 1 Year" filecount="27" totalsize="5512"
sizeunit="1024" />
</list>
- <list type="Owner">
  <criteria filter="Administrators" filecount="31" totalsize="74811"
sizeunit="1024" />
</list>
- <list type="Length">
  <criteria filter="Less than 1KB" filecount="2" totalsize="2" sizeunit="1024" />
  <criteria filter="1KB - 4KB" filecount="1" totalsize="3" sizeunit="1024" />
  <criteria filter="4KB - 16KB" filecount="1" totalsize="5" sizeunit="1024" />
  <criteria filter="16KB - 64KB" filecount="6" totalsize="193" sizeunit="1024" />
  <criteria filter="64KB - 256KB" filecount="1" totalsize="161" sizeunit="1024" />
  <criteria filter="256KB - 1MB" filecount="8" totalsize="5170" sizeunit="1024" />
  <criteria filter="64MB - 256MB" filecount="1" totalsize="69277" sizeunit="1024" />
>
</list>
</DswInventory>

```

### 4.3.4 View the Detailed Report

The detailed results are written to the e:\myDetails.xml file. The following is sample output from a detailed report. Omitted entries are replaced by an ellipsis (...).

```

<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
- <DswInventory GenerationDateTime="8/27/2009 3:10:37 PM" generation="detailed"
platform="Win32NT" OS="Microsoft Windows NT 5.2.3790 Service Pack 2">
- <folderlist>
- <folder>
- <![CDATA[ e:\
]]>
  </folder>
</folderlist>
- <list type="Modified">
- <criteria filter="Within Last Day" filecount="31" totalsize="74811"
sizeunit="1024">
- <information name="Spreadsheet.ods" matchvalue="8/26/2009 7:50:17 PM">
- <![CDATA[ e:\Primary\subfolder\Text Files\Spreadsheet.ods
]]>
  </information>
...
  </criteria>
</list>
- <list type="Creation">
- <criteria filter="Within Last Day" filecount="31" totalsize="74811"
sizeunit="1024">
- <information name="Spreadsheet.ods" matchvalue="8/26/2009 7:50:17 PM">

```

```

- <![CDATA[ e:\Primary\subfolder\Text Files\Spreadsheet.ods
  ]]>
</information>
...
</criteria>
</list>
- <list type="EmptyFiles">
- <criteria filter="Empty Files" filecount="11" totalsize="0" sizeunit="0">
- <information name="10file1.10file" matchvalue="0">
- <![CDATA[ e:\Primary\subfolder\10Files.10file\10file1.10file
  ]]>
</information>
...
</criteria>
</list>
- <list type="Extension">
- <criteria filter=".10file" filecount="10" totalsize="0" sizeunit="1024">
- <information name="10file1.10file" matchvalue=".10file">
- <![CDATA[ e:\Primary\subfolder\10Files.10file\10file1.10file
  ]]>
</information>
...
</criteria>
- <criteria filter=".doc" filecount="1" totalsize="57" sizeunit="1024">
- <information name="Proposal.doc" matchvalue=".doc">
- <![CDATA[ e:\Primary\subfolder\Text Files\Proposal.doc
  ]]>
</information>
</criteria>
- <criteria filter=".docx" filecount="1" totalsize="57" sizeunit="1024">
- <information name="Advanced&.docx" matchvalue=".docx">
- <![CDATA[ e:\Primary\subfolder\Text Files\Advanced&.docx
  ]]>
</information>
</criteria>
- <criteria filter=".exe" filecount="1" totalsize="16" sizeunit="1024">
- <information name="GimmeFiles.exe" matchvalue=".exe">
- <![CDATA[ e:\Primary\subfolder\GimmeFiles.exe
  ]]>
...
</information>
</criteria>
- <criteria filter=".gif" filecount="1" totalsize="888" sizeunit="1024">
- <information name="poppies1920.gif" matchvalue=".gif">
- <![CDATA[ e:\Primary\subfolder\Pictures\poppies1920.gif
  ]]>
</information>
</criteria>
- <criteria filter=".jpeg" filecount="1" totalsize="468" sizeunit="1024">
- <information name="backtrack.jpeg" matchvalue=".jpeg">
- <![CDATA[ e:\Primary\subfolder\Pictures\backtrack.jpeg
  ]]>
</information>
</criteria>
- <criteria filter=".jpg" filecount="5" totalsize="2569" sizeunit="1024">
- <information name="arctica1920.jpg" matchvalue=".jpg">
- <![CDATA[ e:\Primary\subfolder\Pictures\arctica1920.jpg
  ]]>
</information>
</criteria>
- <criteria filter=".ods" filecount="1" totalsize="3" sizeunit="1024">
- <information name="Spreadsheet.ods" matchvalue=".ods">
- <![CDATA[ e:\Primary\subfolder\Text Files\Spreadsheet.ods
  ]]>
</information>
</criteria>
- <criteria filter=".odt" filecount="4" totalsize="828" sizeunit="1024">
- <information name="Investment Final.odt" matchvalue=".odt">
- <![CDATA[ e:\Primary\subfolder\Text Files\Investment Final.odt
  ]]>

```

```

</information>
...
</criteria>
- <criteria filter=".pdf" filecount="1" totalsize="641" sizeunit="1024">
- <information name="dep_stats.pdf" matchvalue=".pdf">
- <![CDATA[ e:\Primary\subfolder\Text Files\dep_stats.pdf
]]>
</information>
</criteria>
- <criteria filter=".rtf" filecount="1" totalsize="1" sizeunit="1024">
- <information name="Another Document.rtf" matchvalue=".rtf">
- <![CDATA[ e:\Primary\subfolder\Text Files\Another Document.rtf
]]>
</information>
</criteria>
- <criteria filter=".txt" filecount="1" totalsize="0" sizeunit="1024">
- <information name="Some Document.txt" matchvalue=".txt">
- <![CDATA[ e:\Primary\subfolder\Text Files\Some Document.txt
]]>
</information>
</criteria>
- <criteria filter=".xml" filecount="2" totalsize="6" sizeunit="1024">
- <information name="myDetails.xml" matchvalue=".xml">
- <![CDATA[ e:\myDetails.xml
]]>
</information>
...
</criteria>
- <criteria filter=".zip" filecount="1" totalsize="69277" sizeunit="1024">
- <information name="Text Files.zip" matchvalue=".zip">
- <![CDATA[ e:\Primary\subfolder\Text Files.zip
]]>
</information>
</criteria>
</list>
- <list type="Modified">
- <criteria filter="Within Last Day" filecount="3" totalsize="69283"
sizeunit="1024">
- <information name="Text Files.zip" matchvalue="8/26/2009 4:01:18 PM">
- <![CDATA[ e:\Primary\subfolder\Text Files.zip
]]>
</information>
...
</criteria>
- <criteria filter="1 Month - 2 Months" filecount="1" totalsize="16"
sizeunit="1024">
- <information name="GimmeFiles.exe" matchvalue="7/17/2009 2:00:44 PM">
- <![CDATA[ e:\Primary\subfolder\GimmeFiles.exe
]]>
</information>
</criteria>
- <criteria filter="6 Months - 1 Year" filecount="27" totalsize="5512"
sizeunit="1024">
- <information name="Spreadsheet.ods" matchvalue="12/16/2008 8:29:00 AM">
- <![CDATA[ e:\Primary\subfolder\Text Files\Spreadsheet.ods
]]>
</information>
...
</criteria>
</list>
- <list type="Owner">
- <criteria filter="Administrators" filecount="31" totalsize="74811"
sizeunit="1024">
- <information name="myDetails.xml" matchvalue="Administrators">
- <![CDATA[ e:\myDetails.xml
]]>
</information>
...
</criteria>
</list>

```

```

- <list type="Length">
- <criteria filter="Less than 1KB" filecount="2" totalsize="2" sizeunit="1024">
- <information name="Another Document.rtf" matchvalue="7">
- <![CDATA[ e:\Primary\subfolder\Text Files\Another Document.rtf
]]>
</information>
...
</criteria>
- <criteria filter="1KB - 4KB" filecount="1" totalsize="3" sizeunit="1024">
- <information name="Spreadsheet.ods" matchvalue="2274">
- <![CDATA[ e:\Primary\subfolder\Text Files\Spreadsheet.ods
]]>
</information>
</criteria>
- <criteria filter="4KB - 16KB" filecount="1" totalsize="5" sizeunit="1024">
- <information name="myDetails.xml" matchvalue="4597">
- <![CDATA[ e:\myDetails.xml
]]>
</information>
</criteria>
- <criteria filter="16KB - 64KB" filecount="6" totalsize="193" sizeunit="1024">
- <information name="Proposal.doc" matchvalue="57856">
- <![CDATA[ e:\Primary\subfolder\Text Files\Proposal.doc
]]>
</information>
...
</criteria>
- <criteria filter="64KB - 256KB" filecount="1" totalsize="161" sizeunit="1024">
- <information name="Eastern Fire.jpg" matchvalue="164026">
- <![CDATA[ e:\Primary\subfolder\Pictures\Eastern Fire.jpg
]]>
</information>
</criteria>
- <criteria filter="256KB - 1MB" filecount="8" totalsize="5170" sizeunit="1024">
- <information name="Rough (Formatted).odt" matchvalue="782959">
- <![CDATA[ e:\Primary\subfolder\Text Files\Rough (Formatted).odt
]]>
</information>
- <information name="dep_stats.pdf" matchvalue="655518">
...
</criteria>
- <criteria filter="64MB - 256MB" filecount="1" totalsize="69277" sizeunit="1024">
- <information name="Text Files.zip" matchvalue="70938669">
- <![CDATA[ e:\Primary\subfolder\Text Files.zip
]]>
</information>
</criteria>
</list>
</DswInventory>

```

## 4.4 Example of a Filtered Inventory

In this example, assume that you want a filtered inventory for the Dynamic File Services pair that has a primary path of e:\. You want to find JPG files with file sizes greater than 1 MB. You perform the following tasks to manually generate a report:

- ◆ [Section 4.4.1, “Create a Command File,” on page 82](#)
- ◆ [Section 4.4.2, “Run the File System Inventory Utility,” on page 82](#)
- ◆ [Section 4.4.3, “View the Summary Report,” on page 82](#)
- ◆ [Section 4.4.4, “View the Detailed Report,” on page 82](#)

## 4.4.1 Create a Command File

Use a text editor to create a command file called `FSCommands.txt` in the Dynamic File Services folder. The first line is the primary path for the pair. The second line requests a summary report to be created and specifies the output file for the summary report. The third line requests a detailed report to be created and specifies the output file for the detailed report. No inventory options are specified, so all inventory options are enabled.

```
e:\
/summary=e:\mySummary.xml
/detailed=e:\myDetails.xml
/size>1024
/extension=*.jpg
```

## 4.4.2 Run the File System Inventory Utility

- 1 Log in to the server as the Administrator user, or as a user in the Administrators group.
- 2 Open a Command Prompt console, then go to the Dynamic File Services folder.
- 3 At the prompt, enter

```
DswInventory.exe FSCommands.txt
```

## 4.4.3 View the Summary Report

The summary results are written to the `e:\mySummary.xml` file. The following is sample output of a summary report:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
- <DswInventory GenerationDateTime="8/27/2009 3:13:10 PM" generation="summary"
platform="Win32NT" OS="Microsoft Windows NT 5.2.3790 Service Pack 2">
- <folderlist>
- <folder>
- <![CDATA[ e:\
  ]>
  </folder>
</folderlist>
- <list type="Extension">
<criteria filter=".jpg" filecount="5" totalsize="2569" sizeunit="1024" />
</list>
- <list type="Length">
<criteria filter="64KB - 256KB" filecount="1" totalsize="161" sizeunit="1024" />
<criteria filter="256KB - 1MB" filecount="4" totalsize="2408" sizeunit="1024" />
</list>
</DswInventory>
```

## 4.4.4 View the Detailed Report

The detailed results are written to the `e:\myDetails.xml` file. The following is sample output from a detailed report:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
- <DswInventory GenerationDateTime="8/27/2009 3:13:10 PM" generation="detailed"
platform="Win32NT" OS="Microsoft Windows NT 5.2.3790 Service Pack 2">
- <folderlist>
- <folder>
- <![CDATA[ e:\
  ]>
  </folder>
</folderlist>
```

```

- <list type="Extension">
- <criteria filter=".jpg" filecount="5" totalsize="2569" sizeunit="1024">
- <information name="arctical1920.jpg" matchvalue=".jpg">
- <![CDATA[ e:\Primary\subfolder\Pictures\arctical1920.jpg
]]>
</information>
- <information name="Eastern Fire.jpg" matchvalue=".jpg">
- <![CDATA[ e:\Primary\subfolder\Pictures\Eastern Fire.jpg
]]>
</information>
- <information name="eventide1920.jpg" matchvalue=".jpg">
- <![CDATA[ e:\Primary\subfolder\Pictures\eventide1920.jpg
]]>
</information>
- <information name="lastlight1920.jpg" matchvalue=".jpg">
- <![CDATA[ e:\Primary\subfolder\Pictures\lastlight1920.jpg
]]>
</information>
- <information name="vigil1920.jpg" matchvalue=".jpg">
- <![CDATA[ e:\Primary\subfolder\Pictures\vigil1920.jpg
]]>
</information>
</criteria>
</list>

- <list type="Length">
- <criteria filter="64KB - 256KB" filecount="1" totalsize="161" sizeunit="1024">
- <information name="Eastern Fire.jpg" matchvalue="164026">
- <![CDATA[ e:\Primary\subfolder\Pictures\Eastern Fire.jpg
]]>
</information>
</criteria>
- <criteria filter="256KB - 1MB" filecount="4" totalsize="2408" sizeunit="1024">
- <information name="vigil1920.jpg" matchvalue="642827">
- <![CDATA[ e:\Primary\subfolder\Pictures\vigil1920.jpg
]]>
</information>
- <information name="lastlight1920.jpg" matchvalue="705456">
- <![CDATA[ e:\Primary\subfolder\Pictures\lastlight1920.jpg
]]>
</information>
- <information name="eventide1920.jpg" matchvalue="544622">
- <![CDATA[ e:\Primary\subfolder\Pictures\eventide1920.jpg
]]>
</information>
- <information name="arctical1920.jpg" matchvalue="572311">
- <![CDATA[ e:\Primary\subfolder\Pictures\arctical1920.jpg
]]>
</information>
</criteria>
</list>
</DswInventory>

```

## 4.5 Additional Information

A file system inventory of each Dynamic File Services pair is automatically generated daily with the pair history run. You can use the Management Console to view graphical displays of this report, and to drill down into the report to see the individual files in various categories. For information, see [“Viewing the Pair History”](#) in the *Dynamic File Services 2.0 Administration Guide*.

---

# 5 Dynamic File Services Synchronize Pair Utility

The Novell Dynamic File Services (DynamicFS) Synchronize Pair utility (`DswSyncPair.exe`) provides a way to detect and report conflicts for duplicate files and mismatched ACL permissions and attributes on folders. This report is helpful for identifying information about duplicate files or mismatched folder metadata that might exist after restoring the primary and secondary data from separate backup media.

The Dynamic File Service can be running or not running when you run the Synchronize Pair utility, but policies must not be running against the pair.

- ◆ [Section 5.1, “DswSyncPair,” on page 85](#)
- ◆ [Section 5.2, “Additional Information,” on page 89](#)

## 5.1 DswSyncPair

- ◆ [Section 5.1.1, “Description,” on page 85](#)
- ◆ [Section 5.1.2, “Syntax,” on page 85](#)
- ◆ [Section 5.1.3, “Using the Utility,” on page 86](#)
- ◆ [Section 5.1.4, “Check Action Options,” on page 86](#)
- ◆ [Section 5.1.5, “Control Options,” on page 87](#)
- ◆ [Section 5.1.6, “Report Options,” on page 87](#)
- ◆ [Section 5.1.7, “Examples,” on page 88](#)

### 5.1.1 Description

The Dynamic File Services Synchronize Pair utility is used to detect duplicate files in the pair structure or to detect folders with attribute or ACL permission differences. It can generate reports in CSV and XML format.

The Dynamic File Service can be running or not running when you run the Synchronize Pair utility, but policies must not be running against the pair.

### 5.1.2 Syntax

```
DswSyncPair.exe -pair="<pairname | guid>" [options]
```

```
DswSyncPair.exe -source="path" -target="path" [options]
```

Make sure there are no policies running against the pair (that is, the pair status is *Idle*), then run the `DswSyncPair.exe` command on the server where Dynamic File Services is installed.

The Synchronize Pair utility can be run by any user with Administrator privileges on the primary server and that has file system rights on the primary path and secondary path.

---

**IMPORTANT:** Typically, the username provided is a user identity that has Administrator privileges. If a remote secondary path is used in a pair, you must provide the username of a user that also has Administrator privileges on the DynamicFS server, rights on the remote share, and NTFS file system access rights on the secondary path. Otherwise, the secondary location is reported as missing. One way to do this is to add the username as a member of the Dynamic File Services Storage Rights group. It does not matter if the user is also a member of the `Dynamic File Services` group.

---

If you are using this command in a script, you must escape the quotation mark characters by preceding them with a backslash (`\`). You can also use the `-silent` option to turn off the screen output. For example:

```
DswSyncPair.exe -pair="\<pairname | guid>" [options] -silent
```

```
DswSyncPair.exe -source="\path\" -target="\path\" [options] -silent
```

### 5.1.3 Using the Utility

- 1 Log in to the Dynamic File Services server.
- 2 Open the Management Console, select the pair, then verify that the pair status is in the *Idle* state. Policies must not be running against the pair when you start the Synchronize Pair utility. If policies are running, wait until they are done, or stop them manually. Wait until the pair status is idle before continuing.
- 3 Open a Command Prompt console, then change directory to go to the `C:\Program Files\Dynamic File Services` folder (or the folder where you installed Dynamic File Services). Select *Start > All Programs > Accessories*, then click *Command Prompt*.
- 4 At the command prompt, enter one of the following commands:

```
DswSyncPair.exe -pair="\<pairname | guid>" [options]
```

```
DswSyncPair.exe -source="\path\" -target="\path\" [options]
```

- 5 When you are prompted, enter the username of a user with rights to all of the paths in the pair.

### 5.1.4 Check Action Options

#### **-files**

If this option is specified, a check is performed for duplicate files on the specified source and target paths.

#### **Example**

```
-files
```

#### **-folders**

If this option is specified, a check is performed for folder attribute and ACL permission differences on the specified source and target paths.

#### **Example**

```
-folders
```

## 5.1.5 Control Options

### **-h, --help**

Displays help for `DswSyncPair.exe`, then exits.

### **-pair="*<pairname | guid>*"**

Use this option to look for the specified pair in the pair database to determine the paths to use for the source and target paths. Do not use this option with the `-source` and `-target` options.

#### **Example**

```
-pair="MyPair"
```

### **-silent**

If this option is specified, screen output is not generated.

#### **Example**

```
-silent
```

### **-source="*path*"**

Use this option to specify the path to use for the primary path. This option must be used in combination with the `-target` option.

Do not use this option with the `-pair` option.

#### **Example**

```
-source="c:\primary"
```

### **-target="*path*"**

Use this option to specify the path to use for the secondary path. This option must be used in combination with the `-source` option.

Do not use this option with the `-pair` option.

#### **Example**

```
-target="g:\shadow"
```

## 5.1.6 Report Options

### **-csv="*reportname*"**

Use this option to generate an output report in CSV format.

If this option is used with the `-files` option, the output file is named `reportname.files.csv`.

If this option is used with the `-folders` option, the output file is named `reportname.folders.csv`.

#### **Example**

```
-csv="cvsReport"
```

The possible reports generated are `cvsReport.files.csv` and `cvsReport.folders.csv`.

### **-xml="*reportname*"**

Use this option to generate an output report in XML format.

If this option is used with the `-files` option, the output file is named `reportname.files.xml`.

If this option is used with the `-folders` option, the output file is named `reportname.folders.xml`.

### Example

```
-xml="xmlReport"
```

The possible reports generated are `xmlReport.files.xml` and `xmlReport.folders.xml`.

## 5.1.7 Examples

- ♦ [“Check for Duplicate Files and Produce a CSV Report” on page 88](#)
- ♦ [“Check for Folders with Mismatched Attributes and ACLs and Produce an XML Report” on page 88](#)
- ♦ [“Check Files and Folders and Produce CSV and XML Reports” on page 88](#)

### Check for Duplicate Files and Produce a CSV Report

```
DswSyncPair.exe -pair="My DSW Pair" -files -csv="myCsvReport"
```

Looks in the pair database for the source and target paths of the pair named My DSW Pair. Checks for duplicate files on the source and target paths. Produces a report in CSV format. The report file generated is `myCsvReport.files.csv`.

### Check for Folders with Mismatched Attributes and ACLs and Produce an XML Report

```
DswSyncPair.exe -pair="My DSW Pair" -folders -xml="myXmlReport"
```

Looks in the pair database for the source and target paths of the pair named My DSW Pair. Checks for folders that have mismatched attributes and ACLs on the source and target paths. Produces a report in XML format. The report file generated is `myXmlReport.folders.xml`.

### Check Files and Folders and Produce CSV and XML Reports

```
DswSyncPair.exe -pair="My DSW Pair" -folders -files -xml="myXmlReport" -csv="myCsvReport"
```

Looks in the pair database for the source and target paths of the pair named My DSW Pair. Checks for duplicate files and for folders that have mismatched attributes and ACLs on the source and target paths. Produces a report in XML format and in CSV format.

The report files generated are:

```
myCsvReport.files.csv  
myCsvReport.folders.csv  
myXmlReport.files.xml  
myXmlReport.folders.xml
```

## 5.2 Additional Information

For information about using the Dynamic File Services Synchronize Pair utility for reporting on files and folders in a pair, see the following sections in the *Novell Dynamic File Services Administration Guide* ([http://www.novell.com/documentation/dynamic\\_file\\_services/](http://www.novell.com/documentation/dynamic_file_services/)):

- ◆ “Reporting Conflicts for Attributes and ACL Permissions on Folders”
- ◆ “Reporting Conflicts for Duplicate Files”



---

# 6 Dynamic File Services Configuration Dump Utility

The Novell Dynamic File Services (DynamicFS) Configuration Dump utility (`DswDump.exe`) reports information about the configuration settings, pairs, policies, files, error events, and logs to a file called `Config.txt` in the folder where you installed Dynamic File Services. You can also output the file in `.html` format. This report is helpful for record-keeping and troubleshooting. The tool can be run at any time, with the Service running or not running, including in Windows Safe Mode.

- ♦ [Section 6.1, “DswDump,” on page 91](#)
- ♦ [Section 6.2, “Config.txt Output,” on page 93](#)

## 6.1 DswDump

- ♦ [Section 6.1.1, “Description,” on page 91](#)
- ♦ [Section 6.1.2, “Syntax,” on page 92](#)
- ♦ [Section 6.1.3, “Options,” on page 92](#)
- ♦ [Section 6.1.4, “Examples,” on page 93](#)

### 6.1.1 Description

The Dynamic File Services Configuration Dump utility consolidates current information about Dynamic File Services running on the server, and writes it all to a single file. It includes the following information:

- ♦ Product Information (registration)
- ♦ Windows Configuration (such as the operating system version, domain, machine name, processor type, and total memory)
- ♦ Service configuration (such as the installation and data folders, the port numbers, and the certificate)
- ♦ Workgroup or Domain information
- ♦ Folders and files in the `Dynamic File Services` folder (or the folder where you installed Dynamic File Services)
- ♦ Folders and files in the program data folder
- ♦ Pairs information
- ♦ Policies information
- ♦ Schedules information
- ♦ Audit event messages

- ◆ Audit configuration information
- ◆ Microsoft Event Logger event messages
- ◆ Log files

The information is output by default to the `Config.txt` file in the `C:\Program Files\Dynamic File Services` folder (or the folder where you installed Dynamic File Services). You can also output the file in `.html` format and specify a target output file. For an overview of the output, see [Section 6.2, "Config.txt Output,"](#) on page 93.

## 6.1.2 Syntax

Log in to the server as the Administrator user or as a user with Administrator privileges. It does not matter if the user is also a member of the `Dynamic File Services` group. Navigate to the `C:\Program Files\Dynamic File Services` folder (or the folder where you installed Dynamic File Services), then issue the command.

```
DswDump.exe [-html] ["output_filename"]
```

If no options are specified, it generates the information in `.txt` format and writes it to the default `C:\Program Files\Dynamic File Services\Config.txt` file. The tool can be run at any time, with the Service running or not running, including in Windows Safe Mode.

## 6.1.3 Options

The following options are available for the `DswDump.exe` command:

### **-h**

Displays help.

### **(no options)**

Generates the file output in `.txt` format.

If you do not specify a filename, the information is written to the `Config.txt` file in the `Dynamic File Services` folder.

If you specify a filename, the information is written to the specified file and path.

### **-html**

Generates the file output in `.html` format.

If you do not specify a filename, the information is written to the `Config.htm` file in the `Dynamic File Services` folder.

If you specify a filename, the information is written to the specified file and path.

### **"output\_filename"**

Generates the file output and writes it to the specified file and path. Provide the full path and filename, including the extension. Enclose the filename in double quotes if the name contains spaces.

If no format is specified, the file is generated in `.txt` format.

If the `-html` format is specified, the file is generated in `.html` format.

## Examples

```
"F:\path\ndfs_config_20110704.txt"  
"G:\path\ndfs_config.htm"  
"\\servername\sharename\server1_ndfs_config_20110704.htm"
```

### 6.1.4 Examples

The following command generates the information in .txt format and writes it to the default C:\Program Files\Dynamic File Services\Config.txt file:

```
DswDump.exe
```

The following command generates the information in .txt format and writes it to the F:\path\ndfs\_config\_20110704.txt file:

```
DswDump.exe "F:\path\ndfs_config_20110704.txt"
```

The following command generates the information in .html format and writes it to the default C:\Program Files\Dynamic File Services\Config.htm file:

```
DswDump.exe -html
```

The following command generates the information in .html format and writes it to the \\servername\sharename\ndfs\_config\_20110704.htm file:

```
DswDump.exe -html "\\servername\sharename\ndfs_config_20110704.htm"
```

## 6.2 Config.txt Output

The results of the Dynamic File Services Configuration Dump utility are written to the Config.txt file in the C:\Program Files\Dynamic File Services folder (or the folder where you installed Dynamic File Services). The report includes the following major categories:

```
[ DswDump Runtime Information ]  
  Startup path  
  
[Product Information]  
  Registered  
  
[ Windows Configuration ]  
  OS Version  
  OS Service Pack  
  Domain  
  Machine Name  
  Processor  
  Total Memory  
  
[ Dynamic File Services Configuration ]  
  [ Registry Settings = SOFTWARE\Novell\Dynamic File Services\Setup ]  
  [ Registry Settings = Hardware\Description\System\CentralProcessor\0 ]  
  [ Snapshot Information ]  
  [ Filter Information ]
```

```

[ Active Directory Information ]
  Domain name
  Domain context
  [ Published Shares ]
    [ Share ]

[ Dsw Folder Hierarchy ]
  [ Folder ]
    [ Child Folders ]
    [ Child Files ]
    [ File ]

[ Pairs ]
  [ Pair Database Contents ]
  [ Pair List ]
  [ Pair Details ]
    [ Pair ]
      [ Associated Pair Policies ]
      [ Pair Policy ]
  Folder flags = < None | IncludeFolders | ExcludeFolders >
  [ Include Folders ] or [ Exclude Folders ]
  [ Pair Health ]
  [ Pair Summary History ]

[ Policies ]
  [ Policy Database Contents ]
  [ Policies List ]
  [ Policy Details ]
  [ Policy ]

[ Schedules ]
  [ Schedule Entry Information ]
    [ Schedule Database Contents ]
    [ Schedule Details ]

[ Audit Entry Information ]
  [ Audit Log Contents ]

[ Audit Config Information ]
  [ Audit Config Contents ]

[ MS Event Logger ]
  [ Dynamic File Services ]
  [ Dynamic File Services Snap Shot System ]
  [ DswStandard Policy ]

[ Log files ]
  [ Log File Contents for ...DswDump.log ]
  [ Log File Contents for ...DswMcpCore.log ]
  [ Log File Contents for ...DswStandardPolicy.log ]
  [ Log File Contents for ...install.log ]
  [ Log File Contents for ...upgrade.log ]

```