

Power Classifier for Files Guide

UM637719

July 2020



© Boldon James Ltd. All rights reserved.
www.boldonjames.com

This document is for informational purposes only, and Boldon James cannot guarantee the precision of any information supplied.

Boldon James makes no warranties, express or implied, in this document.

Contents

1	Introduction	3
2	Installation and Deployment.....	3
2.1	PowerShell 64-bit Environment Considerations	4
2.2	Power Classifier for Files Shell - Client versus Service Mode	4
3	Power Classifier for Files – PowerShell commands	4
3.1	Cmdlet Specific Help.....	4
3.2	Common Parameters	5
3.3	Pipelining.....	5
3.4	Get-Classification	5
	3.4.1 <i>Alternative Output format (Format-List)</i>	7
3.5	Get-SelectorValues	7
3.6	Set-Classification.....	8
	3.6.1 <i>Set a Classification on a set of files using a Text Label</i>	8
	3.6.2 <i>Unable to set the label</i>	9
	3.6.3 <i>Set a Classification on a set of files using a SISL Label</i>	10
	3.6.3.1 <i>Constructing a SISL label</i>	10
	3.6.4 <i>Set a classification on all the files in a folder and its sub-solders</i>	12
3.7	Remove-Classification	12
3.8	Clear Classification.....	13
3.9	Preserve Timestamps option	14
3.10	Set Update Markings option	14
3.11	Error Message Handling	14
	3.11.1 <i>Access Control failures</i>	14
	3.11.2 <i>Verbose option</i>	15
4	Power Classifier for Files (windows) application	16
4.1	Additional Options.....	18
	4.1.1 <i>Copy SISL</i>	18
	4.1.2 <i>PowerShell</i>	18
	4.1.3 <i>Save As Log</i>	18
5	Power Classifier for Files console application.....	19
6	Interworking with Classifier Applications	20
6.1	ADS Marking options	21
6.2	Alternate Data Stream Limitations	21
6.3	Interworking with File Classifier.....	22

1 INTRODUCTION

This document summarises the features supported by *Power Classifier for Files*, which allows you to label multiple files at once. *Power Classifier for Files* provides access to this functionality through three interfaces:

1. **Power Classifier for Files (Shell)** which extends PowerShell by adding a *Classification* noun with *Get*, *Set*, *Remove* and *Clear* verbs.
2. **Power Classifier for Files (Windows)** application allowing you to set label values interactively.
3. **Power Classifier for Files (Console)** application (*PowerClassifier.exe*) allowing you to set label values. For example, from within a .BAT file.

Power Classifier for Files can be used to get, set and clear a label on:

- Document types supported by Office Classifier (i.e. Microsoft Office 2003 / 2007 / 2010 Word, Excel, PowerPoint and Project, but **NOT** Visio),
- PDF and ZIP files allowing Email Classifier to perform attachment checks on these file types.
- JPG files, HTML files and Open Office documents¹ or other types supported subject to the additional configuration of Classifier plug-ins under guidance from Classifier support.
- Optionally storing label information in, and retrieving label information from, an Alternate Data Stream for file types not labelled by Office Classifier (controlled by [ADS Marking Options](#)).

Note: *Power Classifier for Files* is a separate product within the Classifier Suite. It requires a valid Classifier Configuration published using Classifier Administration and including a Power Classifier licence.

Note: Throughout this document, Administration Guide means Administration Guide or Administration Server Guide depending upon whether you are using the Administration Console or Administration Server environment.

2 INSTALLATION AND DEPLOYMENT

Install the product from the Power Classifier bundle by executing **Setup.exe** from the 32-bit (x86) or 64-bit (x64) folder beneath the **Power Classifier for Files** folder.

You may have to Logoff and Logon for all features to be fully available.

The product can be deployed using the usual Group Policy mechanisms.

Installation adds a new command to the Start menu below **Start/ All Programs/ Boldon James/ Power Classifier for Files/**:

- **Power Classifier for Files (Windows)** which will launch a windows application as described in section 4.

It also installs:

- **Power Classifier for Files (Console)** application as described in section 5 that can be invoked from the command prompt, or used in .BAT files.

Note: Boldon James recommends that, in light of its powerful capabilities, access to this software be carefully controlled.

¹ JPG, JPEG and JPE / HTML and HTM / ODF, ODT and ODP
boldonjames.com

2.1 PowerShell 64-bit Environment Considerations

Power Classifier for Files can be installed and used in both 32 and 64-bit environments.

If you use the 32-bit version in a 64-bit environment:

- 1 Run the 32-bit PowerShell (from the `Windows\SysWOW64\WindowsPowerShell\v1.0` folder).
- 2 Load the PowerClassifier module manually, using the following command (if required):
`Import-Module .\PowerClassifierForFiles`
- 3 Use the Power Classifier for Files cmdlets as normal.

2.2 Power Classifier for Files Shell - Client versus Service Mode

By default, *Power Classifier for Files (Shell)* will run in Client Mode – that is, it handles its configuration like the other Classifier applications that store a local copy of the Classifier Configuration.

However, *Power Classifier for Files* may be invoked from a Service or invoked in an environment where there is no user environment, and *Power Classifier for Files* fails to find its Configuration (it may not give any direct indication of the problem, it merely fails to operate correctly).

The solution to this problem is to make *Power Classifier for Files* operate in Service Mode that will inhibit the attempt to store the Classifier Configuration locally by setting the following REG_DWORD Registry Key:

```
HKLM\SOFTWARE\Boldon James\Power Classifier Config\ServiceMode = 1
```

If the registry key is missing or contains any other value, *Power Classifier for Files* will operate in Client Mode.

The only consequence of setting Service Mode is that *Power Classifier for Files* will always retrieve its configuration from the designated master location rather than using any local version.

3 POWER CLASSIFIER FOR FILES – POWERSHELL COMMANDS

This section describes the usage of the *Power Classifier for Files* PowerShell extension.

3.1 Cmdlet Specific Help

The *Power Classifier for Files* module and the associated cmdlets support the `Get-Help` cmdlet. The `Get-Help` cmdlet provides comprehensive help for the cmdlets.

For instance, to see the examples for `Get-Classification`, issue the command

```
Get-Help Get-Classification -examples
```

For more information, issue the command

```
Get-Help Get-Classification -detailed
```

For technical information, issue the command

```
Get-Help Get-Classification -full
```

3.2 Common Parameters

All cmdlets support the following general common parameters

Parameter	Description
<code>-Debug</code>	Specifies whether programmer-level debugging messages are displayed
<code>-ErrorAction</code>	Specifies what action should take place when an error occurs
<code>-ErrorVariable</code>	Specifies the variable in which to place objects when an error occurs
<code>-OutVariable</code>	Specifies the variable in which to place all output objects generated by the cmdlet.
<code>-OutBuffer</code>	Defines the number of objects to store in the output buffer before any objects are passed down the pipeline.
<code>-Verbose</code>	Specifies whether the cmdlet writes explanatory messages that can be displayed at the command line
<code>-WarningAction</code>	Specifies what action should take place when the cmdlet writes a warning message
<code>-WarningVariable</code>	Specifies the variable in which warning messages can be saved.

3.3 Pipelining

All cmdlets accept parameters from the pipeline by name. As a general heuristic, cmdlets accept the file from the pipeline by value.

For example, the following cmdlets show bindings for the parameter `-FileList`

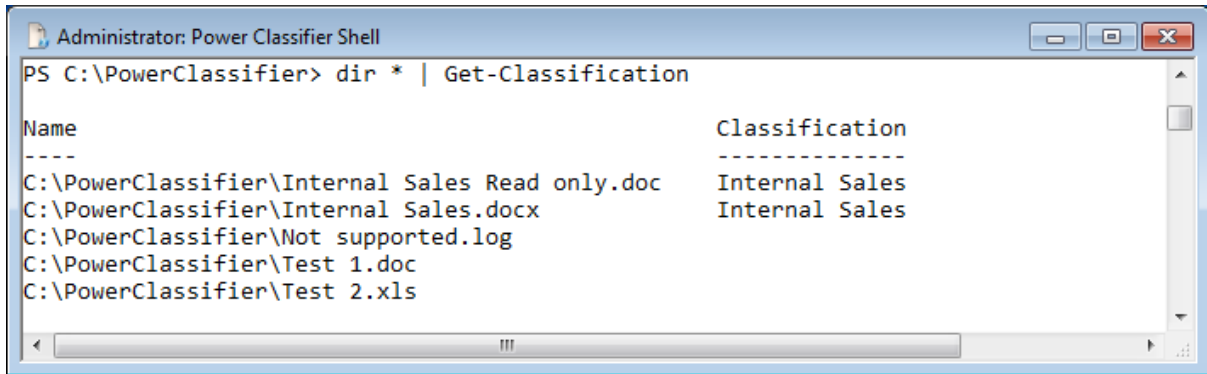
```
dir *.docx | Get-Classification
Get-Classification -FileList "folder list.docx"
```

3.4 Get-Classification

The cmdlet syntax is:

- `Get-Classification [-FileList] <Object[]> [-UseADSMarking] [<CommonParameters>]`
- `Get-Classification [-FileList] <Object[]> -MarkingFormatName <String> [-UseADSMarking] [<CommonParameters>]`
- `Get-Classification [-FileList] <Object[]> -MarkingFormatId <String> [-UseADSMarking] [<CommonParameters>]`

`Get-Classification` presents the label value associated with each of the files supplied:



```
Administrator: Power Classifier Shell
PS C:\PowerClassifier> dir * | Get-Classification

Name                               Classification
----                               -
C:\PowerClassifier\Internal Sales Read only.doc Internal Sales
C:\PowerClassifier\Internal Sales.docx      Internal Sales
C:\PowerClassifier\Not supported.log
C:\PowerClassifier\Test 1.doc
C:\PowerClassifier\Test 2.xls
```

Figure 1: Get Classification

The label presented under the Classification heading above is derived as follows:

- If a Classifier Label value is available (sometimes referred to as a SISL label), display the text version (Marking) of that label. If a marking name or marking identifier is supplied as a parameter then that marking will be used to display the text. If no marking is supplied then the label will be shown using the summary marking.
- If the document has a Classifier Label associated, but no values selected, the configured **NO LABEL** text will be displayed.
- If there is no Classifier Label value, then try to derive a label value from the Label Locations defined within the current Classifier Configuration or from the custom property for PDF, or from the Comment for a ZIP file.
- If the [-UseADSMarking](#) option is specified, check the ADS for label information.
- If none of the above is available then there is no Label. In this instance, the Classification column entry remains **blank** meaning no Label value could be obtained. When no label is found, additional information may be available either using the [-Verbose](#) option or in the Status field available using [Format-List](#).

Note: Classification will be Blank if the file is an unsupported file type, and the [-UseADSMarking](#) option has not been specified, or if the file was labelled using a different Configuration (e.g. by an external organisation).

Any label value found must be valid within the current user's Classifier Configuration; otherwise, the file will be treated as unlabelled.

Nested folder paths can result in a significant part of the file name being truncated. In such circumstances, the [Alternative Output format](#) produced by the use of [Format-List](#) will be helpful.

Some files selected through the `dir` command may not be passed to [Get-Classification](#) and additional 'error' messages may appear as described in [Error Message Handling](#).

3.4.1 Alternative Output format (Format-List)

You can pipe the output from **Get-Classification** or **Set-Classification** through **Format-List** to get a complete list of properties including the SISL information and an indication of the status.

For example,

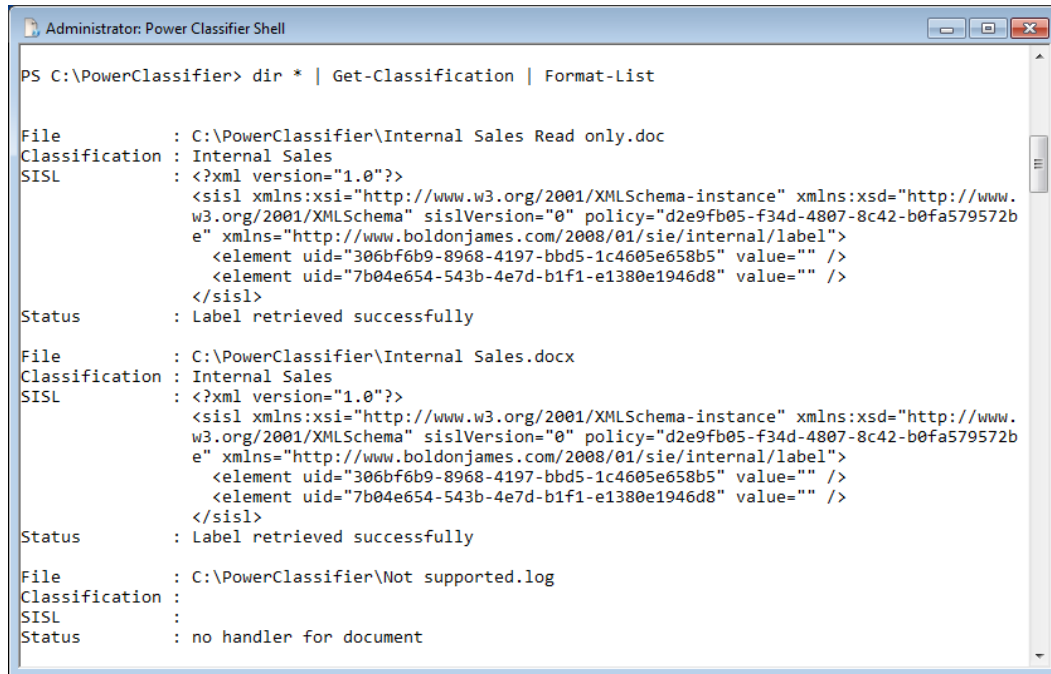


Figure 2: Use of Format-List

3.5 Get-SelectorValues

The cmdlet syntax is:

- 1 **Get-SelectorValues** [-SISL] <String> [<CommonParameters>]
- 2 **Get-SelectorValues -TextLabel** <String> [<CommonParameters>]
- 3 **Get-SelectorValues -TextLabel** <String> -MarkingFormatName <String> [<CommonParameters>]
- 4 **Get-SelectorValues -TextLabel** <String> -MarkingFormatId <String> [<CommonParameters>]

Get-SelectorValues converts a SISL or Text representation of a Label into a hash table of selector names and values. For example,

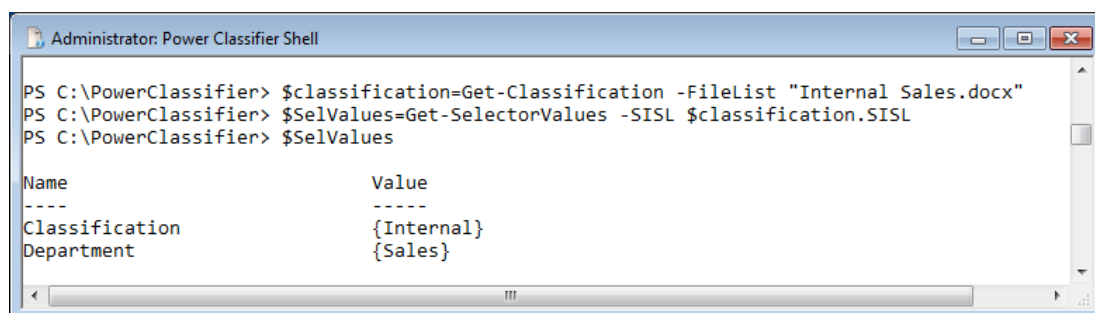


Figure 3: Get-SelectorValues SISL

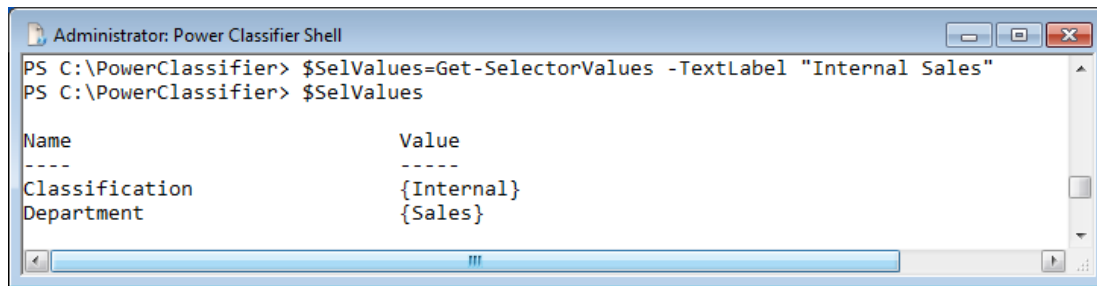


Figure 4: Get-SelectorValues TextLabel

You can specify a marking format to use when decoding the text label through the `-MarkingFormatId` or `-MarkingFormatName` parameters. If a marking format is not specified, then the cmdlet will iterate through the available marking formats until one succeeds.

3.6 Set-Classification

The cmdlet syntax is:

- 1 `Set-Classification [-FileList] <Object[]> -SISL <String> [-LabelOrigin <LabelOrigin>] [-UseADSMarking] [-PreserveTimestamps] [-SetUpdateMarkings] [-UseFileLabelCache] [<CommonParameters>]`
- 2 `Set-Classification [-FileList] <Object[]> -TextLabel <String> [-LabelOrigin <LabelOrigin>] [-UseADSMarking] [-PreserveTimestamps] [-SetUpdateMarkings] [-UseFileLabelCache] [<CommonParameters>]`
- 3 `Set-Classification [-FileList] <Object[]> -TextLabel <String> -MarkingFormatName <String> [-LabelOrigin <LabelOrigin>] [-UseADSMarking] [-PreserveTimestamps] [-SetUpdateMarkings] [-UseFileLabelCache] [<CommonParameters>]`
- 4 `Set-Classification [-FileList] <Object[]> -TextLabel <String> -MarkingFormatId <String> [-LabelOrigin <LabelOrigin>] [-UseADSMarking] [-PreserveTimestamps] [-SetUpdateMarkings] [-UseFileLabelCache] [<CommonParameters>]`

`Set-Classification` allows you to set the label value associated with each of the files supplied. The label value to be applied is supplied either as a [Text String](#) or as a (Classifier) [SISL label](#) value as described in the sub-sections below.

Note: Setting a label value using `Set-Classification` updates the label value on the document, but does NOT change the document in any other way. In particular, any Markings applied within the document (e.g. Header / Footer) will not be updated to reflect the amended label value until the next time the document is opened using Office Classifier. See also Set Update Markings option.

The `-UseADSMarking` option is supported to extend the file types that can be labelled beyond those processed by Office and Email Classifier.

3.6.1 Set a Classification on a set of files using a Text Label

You can pipe files to the `Set-Classification` cmdlet using the command `Dir` as in the example below.

The classification value to be applied to those files is specified using the `-TextLabel` parameter as follows:

```
dir *.* | Set-Classification -TextLabel "Text label value"
```


Note: The Text label value should be enclosed in double-quotes and must exactly match a valid Marking Format structure supported by the current user’s Classifier Configuration. In particular, the component values supplied are case sensitive and must be supplied in the correct order and include relevant delimiters. The examples in this document assume a minimal Marking format that just includes space-separated Selector values.

You can specify a marking format to use when decoding the text label through the `-MarkingFormatId` or `-MarkingFormatName` parameters. If a marking format is not specified, then the cmdlet will iterate through the available marking formats until one succeeds.

The following example applies a `-TextLabel` value of `Internal Sales` to a set of files some of which were initially unlabelled:

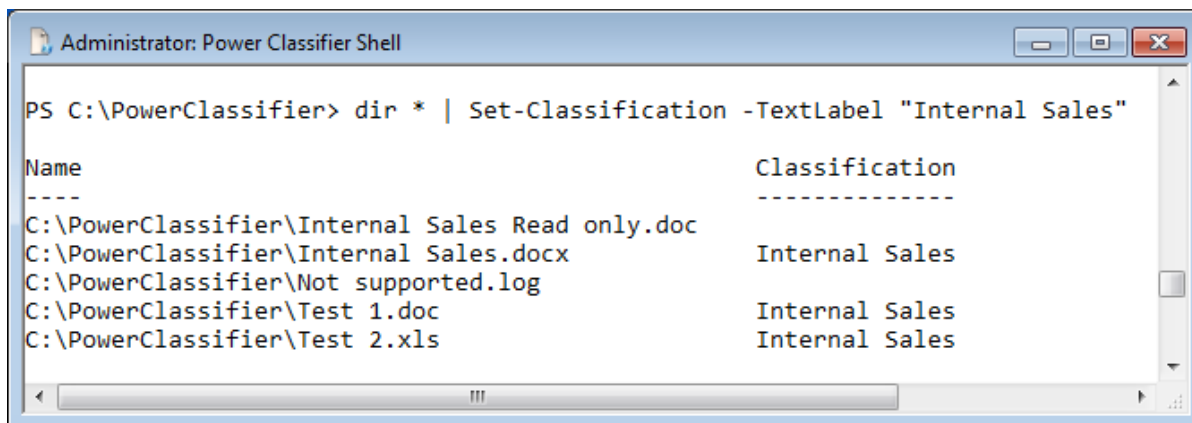


Figure 5: Set-Classification using TextLabel

The Classification is left **blank** for any files that could not be labelled. Additional information may be available using the `-Verbose` option, or in the Status field available if the output is piped through `Format-List`.

An error message is displayed if the supplied Text Label value is invalid (e.g. incorrect spelling or wrong case):

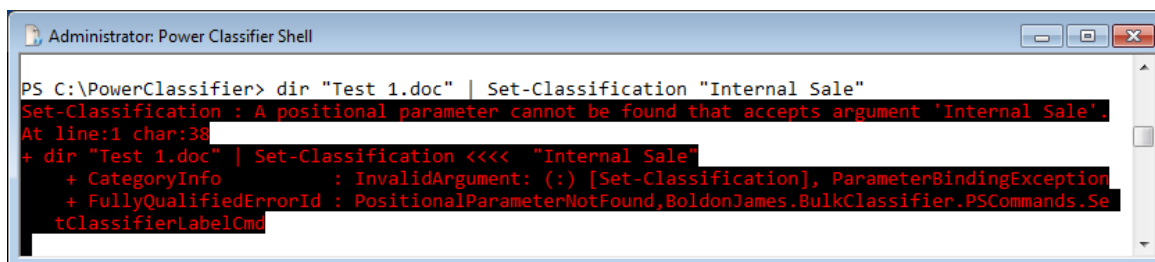


Figure 6: Invalid Text Label

3.6.2 Unable to set the label

Power Classifier for Files will be unable to set a label in the following circumstances:

- The file is in use by another application.
- The file is Read-Only.
- The file is masquerading as a file supported by *Power Classifier for Files* (e.g. File extension changed).
- The file type is not supported by *Power Classifier for Files*. I.e. Outlook message (MSG), EML message format files and Visio files.

- The file is not one of the core types supported by Classifier (e.g. .TXT) and the [-UseADSMarking](#) option has not been specified.

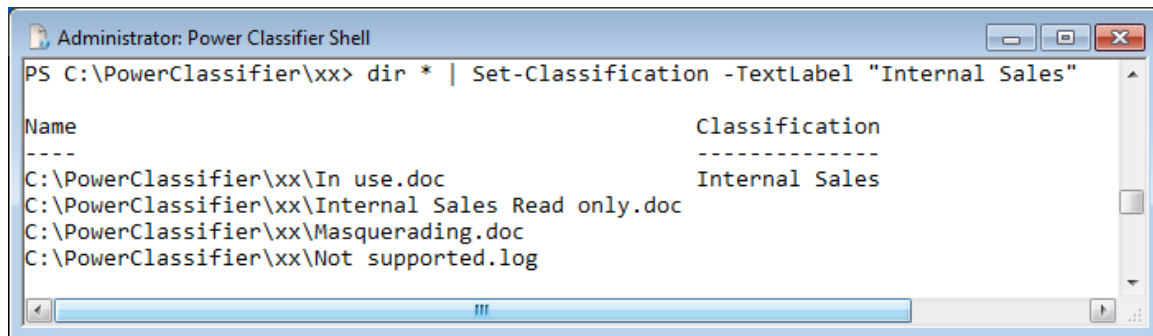


Figure 7: Unable to set a label

A blank Classification entry does **not** mean that there is no classification, only that the Classification could not be set (in the above example the Read-Only document may already have a Classifier label).

Further information on the reason why the label could not be set can be obtained by using the [-Verbose](#) option or in the Status field presented if you pipe the output through [Format-List](#).

3.6.3 Set a Classification on a set of files using a SISL Label

The selected files are passed to *Power Classifier for Files* (using `dir` in the example below).

The label value to be applied to those files is specified using the `-SISL` parameter as follows:

```
dir *.* | Set-Classification -SISL 'SISL label value'
```

A SISL label value can be produced as described in [Constructing a SISL label below](#).

The SISL label value should be enclosed in single quotes (it contains double quotes within). The SISL label value supplied must be valid within the current configuration.

Using this SISL label mechanism may be mandatory if the Label configuration is such that the label to be applied gives ambiguous results when supplied as a Text Label. This might arise for instance if the current Label Configuration contains the same text value under different selectors (not recommended).

3.6.3.1 Constructing a SISL label

Note: The following shows you how to obtain a SISL label using PowerShell mechanisms. An alternative method would be to use the Power Classifier for Files (Windows) application and an Edit Label / Copy SISL sequence to establish the SISL in the clipboard.

The easiest way to produce a SISL label is by using the label from an existing document as in the following sequence.

1. Choose a suitable existing labelled document.
2. Check that it has the required SISL label (e.g. [Get-Classification](#) | [Format-List](#))

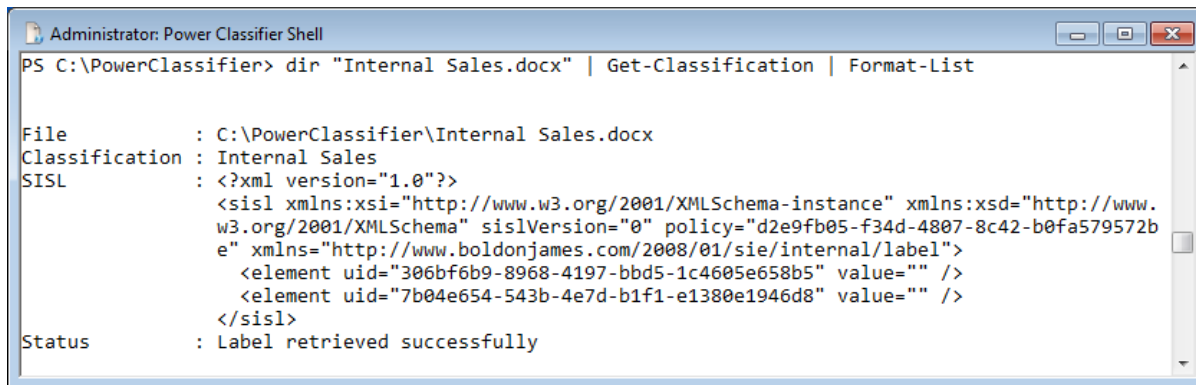


Figure 8: Check Label is correct

3. Get the current label value and store it in a variable (**\$Classification** in the example below).
4. Check that the variable has been populated and contains a SISL value (e.g. **write-host**).

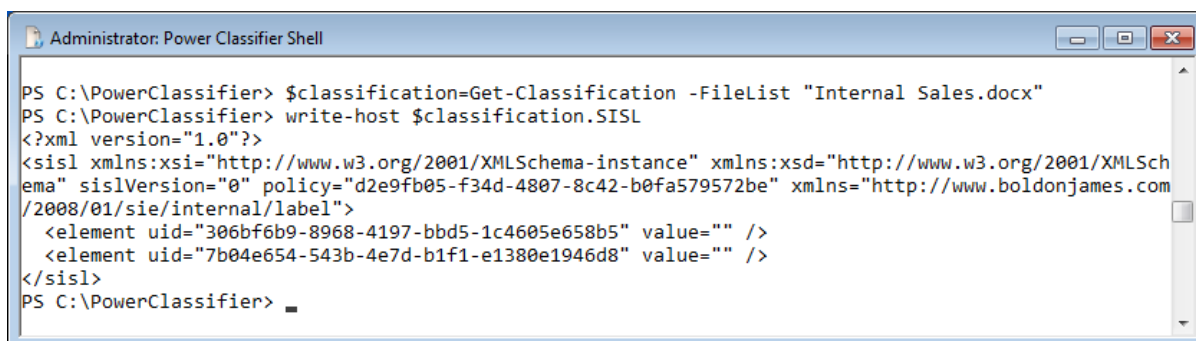


Figure 9: Store in a variable and check

5. Use that SISL value as a parameter to a **Set-Classification** cmdlet.

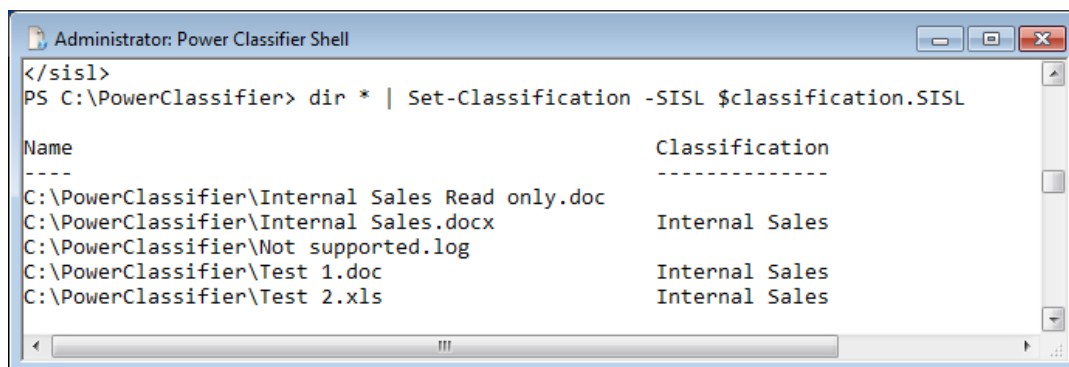


Figure 10: Set Classification using -SISL

If a label cannot be set then the Classification remains blank. See [Unable to set the label](#) for further information on the possible reasons.

An error message is displayed if the supplied SISL Label value is invalid within the current user’s Classifier Configuration / Policy, although there is little indication as to the cause (in the example below a seven has been changed to a one).

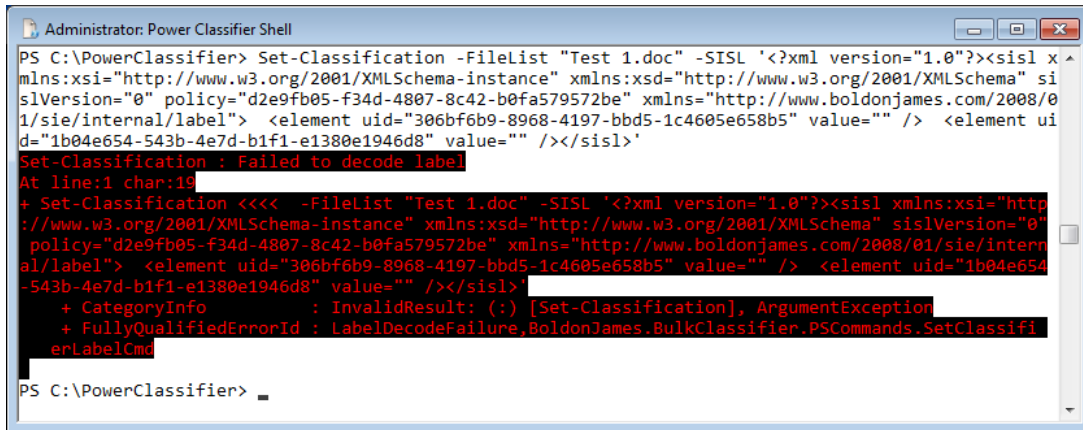


Figure 2 Invalid SISL Label

3.6.4 Set a classification on all the files in a folder and its sub-solders

You can use the **-Recurse** parameter of the PowerShell **dir** command to operate on all files in a folder and all its sub-folders. The following example would find all files in folders starting with **fol** and pass them to *Power Classifier for Files*

```
dir -Recurse fol* | Set-Classification -TextLabel "Internal Sales"
```

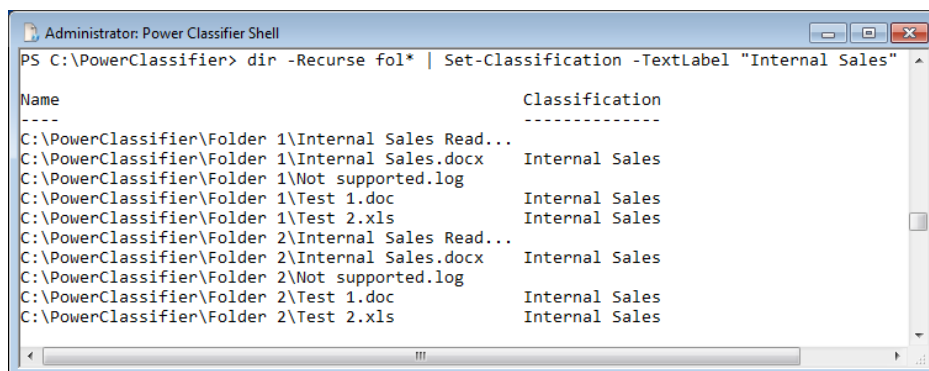


Figure 3 Use of Recurse

3.7 Remove-Classification

The cmdlet syntax is:

- 1 **Remove-Classification** [-FileList] <Object[]> [-UseADSMarking] [-PreserveTimestamps] [-SetUpdatemarkings] [-UseFileLabelCache] [<CommonParameters>]

Remove-Classification removes the Classifier label associated with each of the files supplied. That is, the file will be 'Unlabelled' rather than having a label with no values selected. If you want a label with no values selected then use the **Clear-Classification** cmdlet.

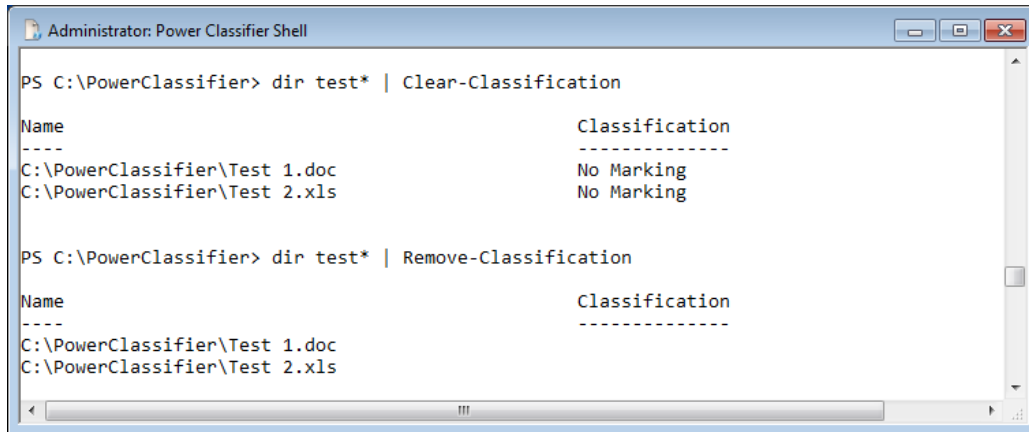


Figure 4 Remove Classification

It may be sensible to use this cmdlet with the `-Verbose` option, as a failure to remove the label (e.g. a Read-only file) will also result in a blank Classification entry.

Including the `-UseADSMarking` parameter will ensure that any label value set via the alternate data stream is removed.

Note: Removing the label deletes the (SISL) label property but does NOT change the document in any other way. Any Markings applied within the document (e.g. Header / Footer) will not be updated to reflect the removed label value until the next time the document is opened using Office Classifier. See also Set Update Markings option

3.8 Clear Classification

The cmdlet syntax is:

- 1 `Clear-Classification [-FileList] <Object[]> [-UseADSMarking] [-PreserveTimestamps] [-SetUpdateMarkings] [-UseFileLabelCache] [<CommonParameters>]`

Clear-Classification allows you to clear the label value associated with each of the files supplied.

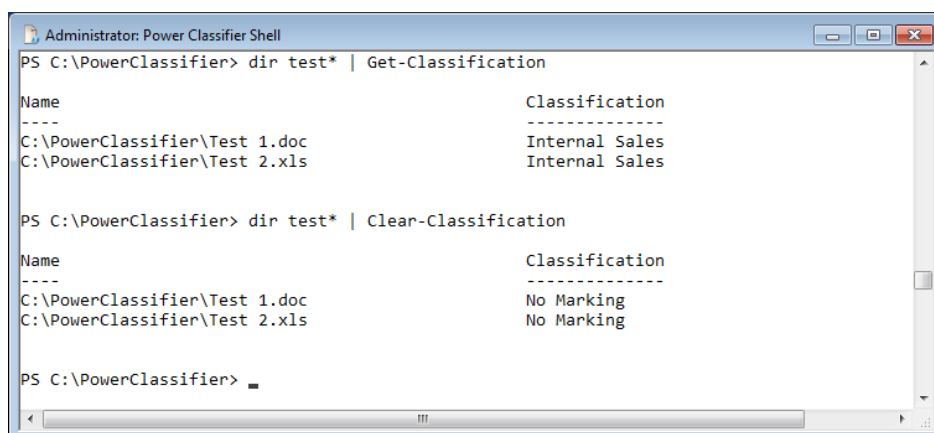


Figure 5 Clear Classification

A blank Classification entry does not mean that there is no Classification, only that the Classification could not be cleared (e.g. a labelled Read-only document would have a blank Classification entry).

Note: Clearing the label clears the (SISL) label value (leaving an empty label) on a document, but does NOT change the document in any other way. In particular, any Markings applied within the document (e.g. Header / Footer) will not be updated to reflect the cleared label value until the next time the document is opened using Office Classifier. See also Set Update Markings option

3.9 Preserve Timestamps option

`Set-Classification`, `Remove-Classification` and `Clear-Classification` cmdlets support the `-PreserveTimestamps` option. By default, performing one of these actions will update the file timestamp. If this option is set, the original timestamp is preserved when changes are made.

3.10 Set Update Markings option

`Set-Classification`, `Remove-Classification` and `Clear-Classification` cmdlets support the `-SetUpdateMarkings` option. By default, this is False meaning that although the label value may have been modified by the action performed, Office Classifier will be unaware of the change and the 'old' marking values will be presented in headers/footers until the document is specifically saved or changed by the user.

Supplying the switch `-SetUpdateMarkings` will set a Classifier specific attribute that is detected by Office Classifier the next time the document is opened, and header/footers etc. are refreshed immediately and the document automatically saved.

Note: If `-SetUpdateMarkings` is supplied with `-PreserveTimestamps`, the next time a user views the document all the markings are refreshed and the document automatically saved – thereby changing the carefully preserved timestamp unless the “Update markings, when indicated” setting is disabled in the Classifier Admin.

3.11 Error Message Handling

3.11.1 Access Control failures

Power Classifier for Files will not process a file passed to it in the following circumstances:

1. You do not have suitable permissions.

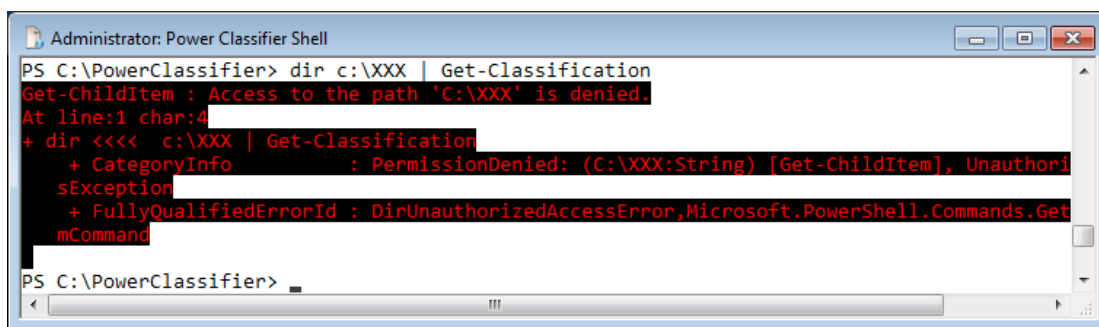


Figure 6 Access permissions failure

Some conditions that occur during processing of files by *Power Classifier for Files* may be reported on the standard error output stream in the shell, namely **StdErr**.

The output from **StdErr** can be collected and redirected to an **error log** file using the '2>' syntax as follows:

```
command 2> ..\error.log
```

By doing this, normal success reporting of the labelling of files by *Power Classifier for Files* will not be captured in the error.log file, only the errors.

If you wish to capture all success output as well as error reporting in a single file, you need to use the '**2>&1**' syntax, which redirects **StdErr** through **StdOut**, as follows:

```
command 2>&1 > output.log
```

For example:

```
dir *.docx | Set-Classification -TextLabel "Confidential" 2>&1 > output.log
```

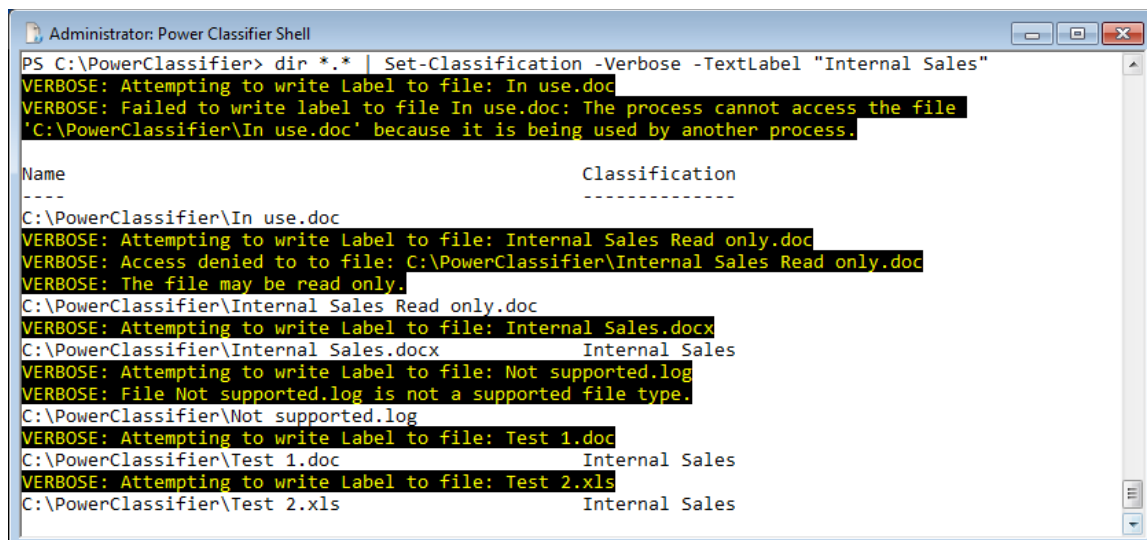
This command will capture all output from *Power Classifier for Files* to the file output.log.

Note: Some conditions such as lack of suitable permissions above are generated by, for example, the `dir` command, and are unaffected by the above example redirections.

3.11.2 Verbose option

The `-Verbose` option can be used to obtain further information on 'failures'. The example below shows the output when `Set-Classification` is unable to set a label value.

```
dir *.* | Set-Classification -Verbose -TextLabel "Text label value"
```



```
Administrator: Power Classifier Shell
PS C:\PowerClassifier> dir *.* | Set-Classification -Verbose -TextLabel "Internal Sales"
VERBOSE: Attempting to write Label to file: In use.doc
VERBOSE: Failed to write label to file In use.doc: The process cannot access the file
'C:\PowerClassifier\In use.doc' because it is being used by another process.
Name                                Classification
----                                -
C:\PowerClassifier\In use.doc
VERBOSE: Attempting to write Label to file: Internal Sales Read only.doc
VERBOSE: Access denied to to file: C:\PowerClassifier\Internal Sales Read only.doc
VERBOSE: The file may be read only.
C:\PowerClassifier\Internal Sales Read only.doc
VERBOSE: Attempting to write Label to file: Internal Sales.docx
C:\PowerClassifier\Internal Sales.docx      Internal Sales
VERBOSE: Attempting to write Label to file: Not supported.log
VERBOSE: File Not supported.log is not a supported file type.
C:\PowerClassifier\Not supported.log
VERBOSE: Attempting to write Label to file: Test 1.doc
C:\PowerClassifier\Test 1.doc              Internal Sales
VERBOSE: Attempting to write Label to file: Test 2.xls
C:\PowerClassifier\Test 2.xls              Internal Sales
```

Figure 7 Verbose option

4 POWER CLASSIFIER FOR FILES (WINDOWS) APPLICATION

The *Power Classifier for Files (Windows)* application can be invoked from the Start menu:

Start/All Programs/Boldon James/Power Classifier for Files/Power Classifier for Files

It will display the following form that allows you to set a label on a file, or all files in a folder, or all files in a sub-tree.

The current Configuration and Policy names appear at the top right of the dialog along with the product version information.

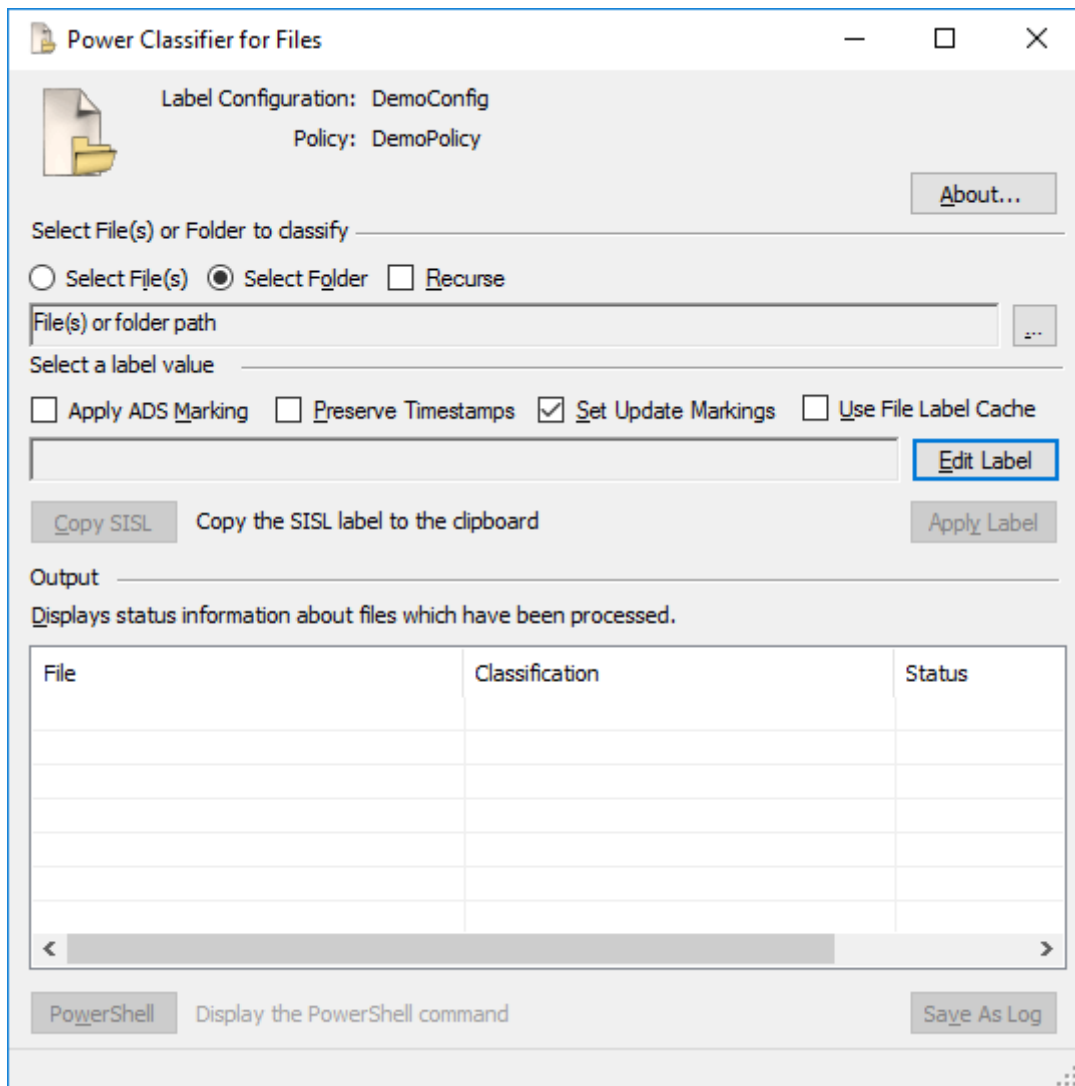



Figure 8 Power Classifier for Files (Windows) Application

1. Choose whether you wish to label a single file or files, or all files in a folder, by checking the **Select File** or **Select Folder** options.
2. If you choose **Select File**, multiple files can be selected from within a single folder in the browse dialog (SHIFT/CTRL + Select).
3. If you choose **Select Folder**, you can also choose whether to recurse the operation through sub-folders by checking the **Recurse** option.
4. Choose the File or Folder that you wish to label via the file selection dialog .

5. By default, your chosen label will only be applied to:
 - File types supported by Office Classifier (excluding Visio files).
 - PDF files
 - ZIP files
6. Checking **Apply ADS Marking** extends the operation of Power Classifier for Files to store label information as NTFS Alternate Data Stream information for additional file types as described [ADS Marking Options](#).
7. Checking the **Preserve Timestamps** option (default **False**) affects the operation as described in the [Preserve Timestamps option](#).
8. Checking the **Set Update Markings** option (default **True**) controls whether or not a flag is set on each file to indicate that next time the document is opened by Office Classifier Headers/Footers etc. should be refreshed. See [Set Update Marking option](#).
9. Checking the **Use File Label Cache** option (default **false**) controls whether or not File Classifier's Label Cache is updated. See [Interworking with File Classifier](#)
10. Press **Edit Label** to access the Label selection dialog.

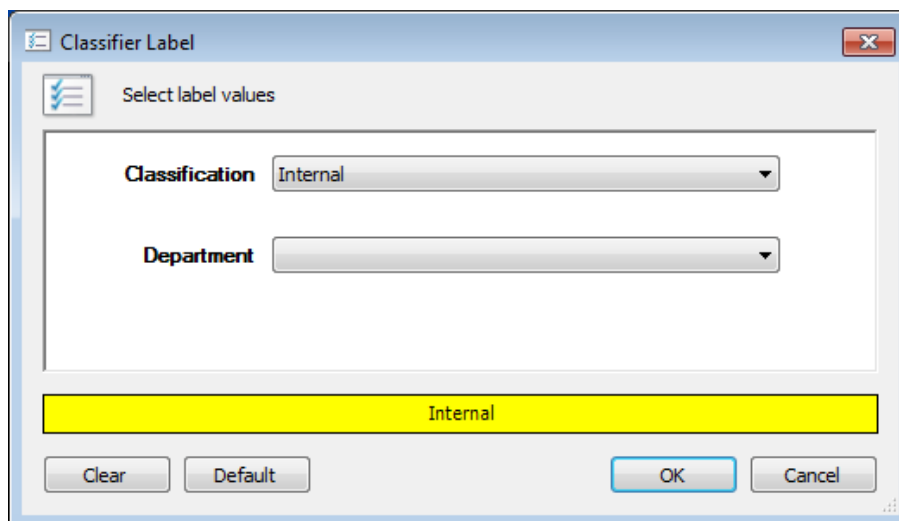


Figure 9 Label Selection

This dialog presents the available label values and, if Favourites is enabled, any Favourites / Recently used label values available.

11. Choose the relevant label values and press **OK**. Note that clearing all label values allows you to clear existing label values when the **Apply label** button is pressed. This will result in the configured **NO LABEL** text appearing.

Note: There is a subtle distinction between Unlabelled (There is no Classifier label information), and NO LABEL which means Classifier label information exists, but no label values were selected.

12. Press the **Apply Label** button to apply the chosen label to the file(s).
13. The results of the operation can be viewed in the **Output** display area below.
14. Whilst labels are being applied, the **Apply Label** button is replaced by a **Stop** button that can be used to terminate an Apply Label sequence at the current point in the sequence

4.1 Additional Options

4.1.1 Copy SISL

The **Copy SISL** button becomes available once you have selected a Label Value. It allows you to copy the Classifier internal (SISL) format label to the clipboard. This may be of use as the SISL Label value when using the **-SISL** option with the **Set-Classification** PowerShell cmdlet.

4.1.2 PowerShell

Pressing the **PowerShell** button presents a separate window containing the Power Classifier for Files PowerShell cmdlet that will be executed if you press **Apply Label**.

This feature may be of use when developing PowerShell scripts to invoke **Set-Classification -SISL** operations.

4.1.3 Save As Log

Pressing the **Save As Log** button allows you to save the contents of the display area in CSV (Comma Separated Values) format to a file.

5 POWER CLASSIFIER FOR FILES CONSOLE APPLICATION

The Power Classifier for Files console application can be invoked from within the console application (or within a .BAT file):

C:\Program Files\Boldon James\Power Classifier for Files\Power Classifier for Files \PowerClassifier.exe

It will display the following window that allows you to set a label on the files in a folder and its sub-folders.

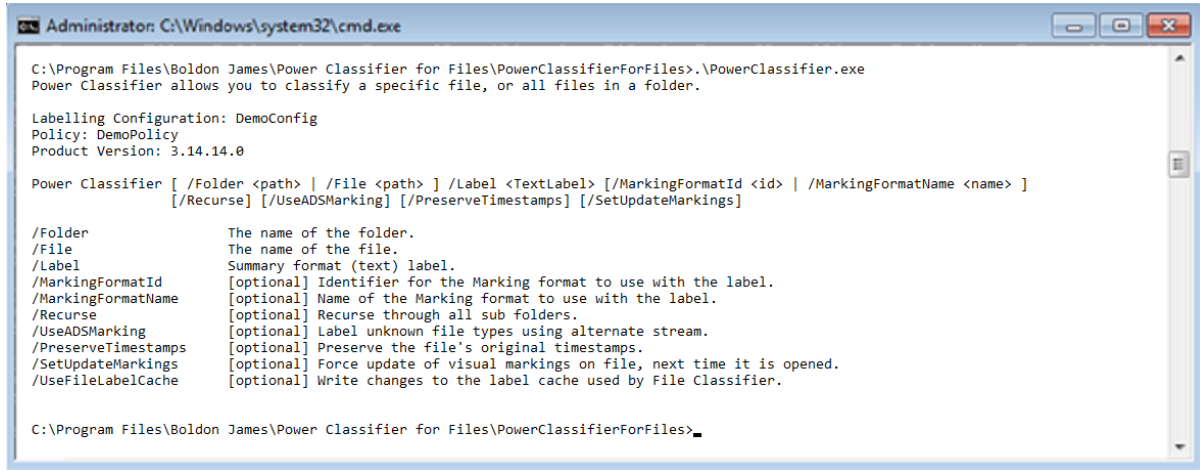


Figure 19: Command options

The various parameters operate in a similar manner to the equivalent ones under **Power Classifier for Files (Shell)** as in the following example:

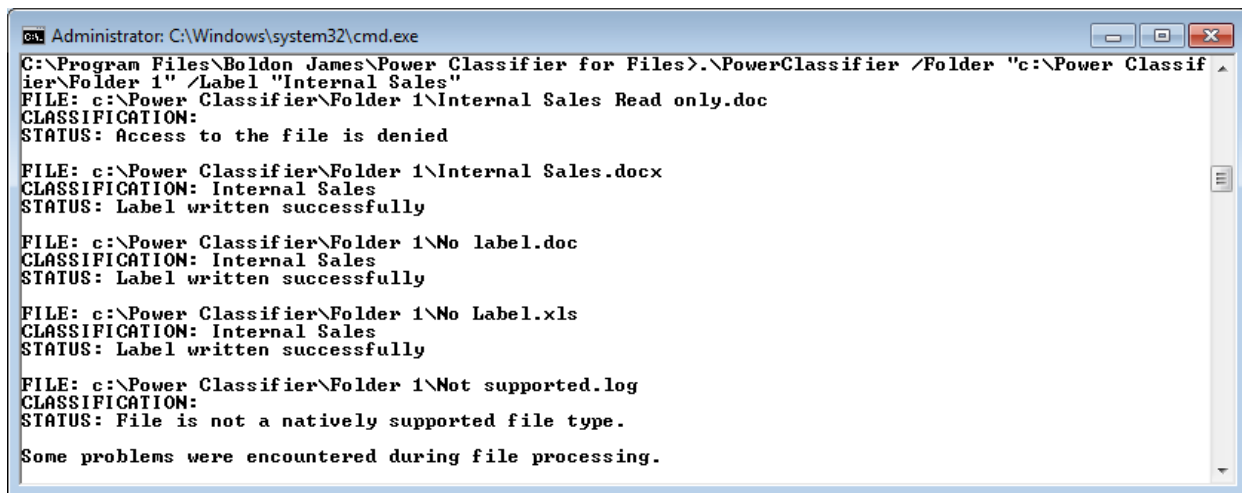


Figure 10 Command Application example

The command returns an exit code:

- 0 - No error
- 1 - Error

6 INTERWORKING WITH CLASSIFIER APPLICATIONS

Note: More detailed information on the use of labels by, and propagation of, labels between, Classifier applications can be found in the Administration Guide / Appendix – Label Propagation Mechanisms.

Key information is summarised below.

1. *Power Classifier for Files* uses the same Label information as Office Classifier and CAD Classifier (Classifier 'private' label property and Label Locations) when reading labels from file types supported (e.g. Microsoft Word, Excel, PowerPoint and Project - note VISIO files must be labelled using the VISIO application).

Using *Power Classifier for Files* to apply a label value to (for example) a Word document update the Classifier 'private' label property, but consequential effects on Markings will only be applied the next time the document is opened using Office Classifier depending upon the [SetUpdateMarkings option](#).

Note: Power Classifier for Files cannot label zero-length Word documents established using New Microsoft Office Word document in Windows Explorer.

2. For PDF files and ZIP files, *Power Classifier for Files* applies a BJ Label value to the file (PDF as a custom property / ZIP as the comment). These label values can be accessed by Email Classifier to perform High Water Mark checks for instance.
3. For JPG² files, HTML³ files and Open Office documents⁴ (or other types supported subject to the additional configuration of Classifier plug-ins under guidance from Classifier support): *Power Classifier for Files* applies a BJ Label value as a custom property to the file. These label values can subsequently be accessed by Email Classifier to perform High Water Mark checks for instance.

Note: Support for these file types requires additional configuration of Classifier plug-ins under guidance from Classifier support. Be aware that the plugin for DXF/DWG files only supports reading of BJ Label values.

4. Outlook message (.MSG), EML message format files and Visio files cannot be labelled using *Power Classifier for Files*.
5. For other file types (e.g. .TXT), *Power Classifier for Files* can (under control of the [UseADSMarking options](#)) store label information in an NTFS Alternate Stream which is not used by Office Classifier, Email Classifier, OWA Classifier or SharePoint Classifier.

Note: When Power Classifier for Files stores a label on a PDF file, it modifies the file. The effects may vary depending on when and how the PDF was originally written. As a minimum, a Classifier defined custom property is added. For older files, this might indirectly cause insertion of a whole PDF Metadata block and effectively update the PDF structure to a newer standard. Note that Power Classifier does not modify the PDF version and Producer properties for Files.

² JPG, JPEG and JPE

³ HTML and HTM

⁴ ODF, ODT and ODP

6.1 ADS Marking options


By default *Power Classifier for Files* can be used to manage labels for file types supported by Office Classifier, PDF and ZIP files (items 1 and 2 above). Email Classifier can access labels for these file types when executing attachment-related rules (e.g. High Water Mark checks).

File types covered by item 5 above (e.g. TXT, LOG, PNG etc.) can optionally have label information stored in an NTFS Alternate Data Stream.

The `-UseADSMarking` parameter and **Apply ADS Marking** option can be set to extend Power Classifier for Files operation to read or store label information in an NTFS Alternate Data Stream (ADS) for file types covered by item 5 above which would not normally be processed. Note that this ADS information is not always accessible, and in particular is not available in Outlook or OWA environments.

The `-UseADSMarking` / **Apply ADS Marking** options are available in the following situations:

- Optional parameter to PowerShell cmdlets
 - `Get-Classification -UseADSMarking`,
 - `Set-Classification -UseADSMarking`,
 - `Remove-Classification -UseADSMarking`,
 - `Clear-Classification -UseADSMarking`.
- **Apply ADS Marking** checkbox in the Power Classifier for Files window
- Optional parameter to console application `/UseADSMarking`.

If the File Classifier product is installed, setting a label on a file that is not supported by Office Classifier, nor PDF, nor ZIP, JPG will apply an annotation to the icon  visible through windows explorer.

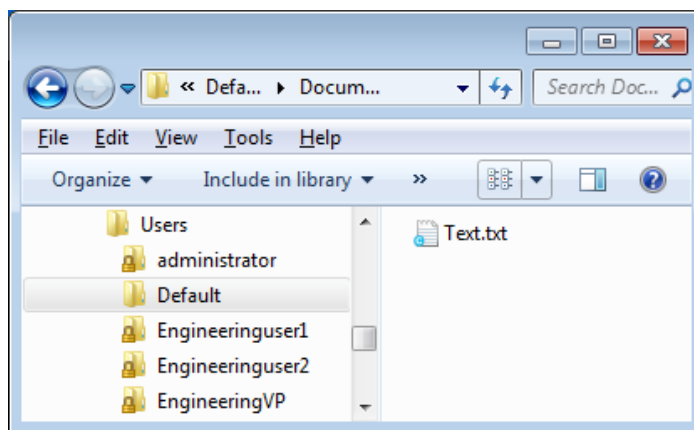


Figure 11 ADS labelled Icon

6.2 Alternate Data Stream Limitations

Moving a file labelled by Power or File Classifier (but not supported by Office Classifier) to a non-NTFS location (e.g. a memory stick) will lose additional Alternate Data Stream (ADS) information (a warning is presented). The file will then be unlabelled. This also relates to emailing such files, cloud storage, etc.

1. Email Classifier will not perform High Water Mark checking on Power and File Classifier labelled files that use the Alternate Data Stream mechanism, nor include such files in Attachment Lists as the Alternate Stream information is unavailable in that environment.

2. Office Classifier will only detect label values stored in standard Office Classifier locations. So for example, opening a .TXT file using Word will not detect any Alternate Data Stream Label and the file will be treated as unlabelled.
3. SharePoint Classifier will not take account of ADS stored label values when uploading files.

6.3 Interworking with File Classifier

File Classifier optionally reads and writes a cached label to improve icon overlay performance.

To help with this feature, *Power Classifier for Files* can optionally use a switch parameter *-UseFileLabelCache* that indicates whether labels are written to the cache.

If you use File Classifier in this mode with files that are being classified by Power Classifier for Files, then supply the *-UseFileLabelCache* switch when using the cmdlets *Set-Classification*, *Remove-Classification* and *Clear-Classification*.