

AutoIndexer

version 2.8 - revision 1

Administrator's Guide

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express permission of Autonomy.

Windows is a trademark of Microsoft Corp.

Copyright © 2002 Autonomy

AutoIndexer is a trademark of Autonomy

Table of Contents

1. Introduction: Controlling internal file import.....	1
2. Installing AutoIndexer.....	3
System requirements	3
Installing under Windows.....	4
Directory structure: Windows.....	6
Installing under UNIX	8
Directory structure: UNIX.....	10
3. Configuring AutoIndexer	11
Configuration file sections	12
[License] section	12
[Service] section	13
[Server] section.....	14
[Default] section.....	16
[Configuration] section.....	32
[Job] section	32
4. Importing individual files	33
action=Import: importing individual files.....	33
Index	35

Autonomy

Autonomy employs a fundamentally different and unique combination of technologies to enable computers to form an understanding of a page of text, web pages, e-mails, voice, documents and people.

Autonomy's solution is therefore able to power any application dependent upon unstructured information within every market sector, including: e-commerce, customer relationship management, knowledge management, enterprise information portals and online publishing applications.

This is evidenced by the significant penetration of the technology in a diversity of vertical markets and has been achieved principally because every market sector needs to manage and leverage the benefits of unstructured information.

Autonomy was founded in 1996 and has offices in Boston, Chicago, Dallas, San Francisco, New York, and Washington, D.C. in the United States, as well as offices throughout EMEA, including Amsterdam, Brussels, Cambridge, Frankfurt, Milan, Paris, Oslo, and Sydney. In July 1998, the company went public on the EASDAQ exchange (EASDAQ:AUTN). Autonomy floated on The NASDAQ National Market (NASDAQ: AUTN) in May 2000, and on the London Stock Exchange (LSE: AU.) in November 2000.

To contact Autonomy, please get in touch with your nearest location listed below.

Europe and South Pacific

Autonomy Systems Ltd.
Cambridge Business Park
Cowley Road
Cambridge
CB4 0WZ

Help Desk: +44 (0) 800 0 282 858

Switchboard: +44 (0) 1223 448 000

Fax: +44 (0) 1223 448 001

E-mail for information: autonomy@autonomy.com

for support: uksupport@autonomy.com

The Help Desk operates from 9.30 am to 6.00 pm (GMT) Monday to Friday.

Website: www.autonomy.com

USA

Autonomy Inc.
301 Howard Street
22nd Floor
San Francisco
CA 94105

Help Desk: +1 877 333 7744

Switchboard: +1 415 243 9955

Fax: +1 415 243 9984

E-mail for information: info@us.autonomy.com

for support: support@us.autonomy.com

The Help Desk operates from 9.30 am to 6.00 pm (CST) Monday to Friday, toll-free.

Website: www.autonomy.com

Welcome

Thank you for choosing Autonomy and welcome to your AutoIndexer 2.8.x™ Administrator's Guide.

Autonomy Solutions

Autonomy solutions provide the software infrastructure that automates operations on unstructured information. This software infrastructure is based on IDOL server, the Intelligent Data Operating Layer. IDOL makes it possible for organizations to process digital content automatically and to enable applications to operate with each other. It consists of data operations that integrate information by understanding any type of content, and is therefore data agnostic. The IDOL server software infrastructure is fully scalable and customizable according to customers' present and future needs.

Autonomy solutions include:

- **Autonomy Connectors™** enable automatic content aggregation from any type of local or remote repository (for example, a database, a web site, a real-time telephone conversation etc.), facilitating a unified solution across all information assets within the organization.
- **ACI Servers™** automatically perform a variety of operations on structured, unstructured and semi-structured information (documents, audio, video etc.). These operations include automatic hyperlinking, tag reconciliation, XML tagging, profiling, alerting, categorization, cluster mapping, real-time transcription, targeting etc.
- **Autonomy Application Builder™** is a toolkit that enables companies and partners to customize Autonomy's products according to their individual requirements. It facilitates easy communication between custom-built applications that retrieve data using HTTP commands and the Autonomy ACI servers, as well as simple manipulation of the returned result sets. Communication with the servers is implemented over HTTP using XML and can adhere to SOAP. The API is distributed with a set of sample code.
- **Portal-in-a-Box™**, our comprehensive and fully automated Information Portal for content-rich Internet and Intranet sites.
- **Portlets** are windows that can be set up in Autonomy's Portal-in-a-Box or third party Portals. Each portlet contains an application that allows the Portals' end users to benefit from a variety of IDOL server functionality.
- **Autonomy Desktop Suite™** brings the power of Autonomy to every desktop. Conducting a real-time analysis of the ideas involved in the content of any opened desktop application, Desktop Suite's ActiveKnowledge or Active Windows Extensions module provides real-time links to relevant internal and external information without the user being needlessly diverted from their work in progress to perform an exasperating search or retrieval operation.
- **Autonomy Product Orientated Drop-in Solutions™** allow Autonomy solutions to be easily integrated with third party applications and solution providers. PODS enable organizations to make their existing applications compatible with IDOL with minimal configuration and administration requirements. Making IDOL server a part of any solution delivers the direct benefits of content automation and the ability to perform a vast range of IDOL server operations, irrelevant of file format or location.

1. Introduction: Controlling internal file import

To control how files are imported from an internal location (for example, from a computer on your network), you need to configure AutoIndexer, which you can install as one of the Portal-in-a-Box components.

Depending on how you configure AutoIndexer, it will:

- read a specified text file that lists a set of documents with their file path. It then finds, imports and indexes these documents. (This is called File Polling).
- import and index any file that is contained in a specified directory, provided the file meets the criteria that you have specified. (This is called Directory Polling)

AutoIndexer communicates automatically with a Content or Data DRE, and once you have set both up and configured them, you can leave AutoIndexer running continually. Every time new files are added to the list file or the directory from which AutoIndexer is reading, it processes them automatically.

Note:

- If you have set AutoIndexer to File Polling, AutoIndexer creates a **.pos** file for each of the text files from which it reads which files it should process. The **.pos** file stores the current position in the queue of files that AutoIndexer is processing, and is located alongside the text file from which AutoIndexer reads which files it should process.

If you want to stop the AutoIndexer and re-start its process from scratch, you should delete all **.pos** files and **.pos.bak** files from the AutoIndexer directory. If you don't AutoIndexer will refer to them when it is restarted and carry on its process from where it stopped.

If you want to reprocess the last file that AutoIndexer dealt with, you can replace the contents of the **.pos** file with the contents of the **.pos.bak** file, which is a copy of the **.pos** file before the last file was processed.

- If you have set AutoIndexer to Directory Polling, AutoIndexer creates a **<InstallationName>_AutoIndexer.dirstatn** file for each of the jobs that it carries out. The **<InstallationName>_AutoIndexer.dirstatn** file contains a list of all the files that have been processed and is stored in the AutoIndexer folder in your installation directory.

If you want to stop the AutoIndexer and re-start its process from scratch, you should delete the **<InstallationName>_AutoIndexer.dirstatn** and **<InstallationName>_AutoIndexer.dirstatn.bak** files. If you don't AutoIndexer will refer to them when it is restarted and carry on its process from where it stopped.

If you want to reprocess the last file that AutoIndexer dealt with, you can replace the contents of the **<InstallationName>_AutoIndexer.dirstatn** with the contents of the **<InstallationName>_AutoIndexer.dirstatn.bak** file, which is a copy of the **<InstallationName>_AutoIndexer.dirstatn** file before the last file was processed.

Introduction: Controlling internal file import

- If you have set AutoIndexer to Directory Polling, AutoIndexer automatically processes any new files that appear in the specified directory. You should therefore ensure that no application will create temporary files in this directory.
- The AutoIndexer folder in your installation directory contains a **<InstallationName>_AutoIndexer.log** file. You can use this file to keep track of all actions that AutoIndexer performs.

2. Installing AutoIndexer

AutoIndexer should be installed by the system administrator as part of a larger Autonomy system (that is a system that includes an Autonomy DRE and an interface for the information stored in this DRE).

System requirements

Platforms supported
Microsoft Windows NT 4 and 2000
Linux
Solaris

Note: AutoIndexer also supports other POSIX UNIX versions on request.

Minimum server specification	
For Windows:	For Linux:
200 MHz Pentium processor	128 MB of RAM
64 MB RAM	200 MB hard disk
200 MB hard disk	

Note: this specification is dependent on the amount of data to be indexed.

Installing under Windows

To install under Windows insert the AutoIndexer CD-ROM into your CD-ROM Drive.

If your Windows installation is configured to support it, inserting the CD-ROM will automatically start the AutoIndexer installation program. Otherwise you can start the installation by double-clicking on the AutoIndexer-2.8.x.exe program in the root directory of the CD-ROM through Windows explorer.

Read and follow all installation instructions on the screen carefully. Before the installation program can start to copy files onto your PC, you need to provide it with some information:

1. The installation opens with the **Welcome** dialog. Read the text and click on **Next**.
2. The **License Agreement** dialog is displayed.
Read the agreement and click on **Next** to accept it.
3. The **Installation name** dialog is displayed.
Enter a unique name for the AutoIndexer installation. **Note:** The unique name must not contain any spaces.
4. The **Choose Destination Location** dialog is displayed.
Select the directory in which you want to install AutoIndexer, and click on **Next**. By default the components are installed on **C:\Autonomy\Autoindexer**, but you can use the **Browse** button to navigate to another location.
5. The **Select Program Manager Group** dialog is displayed. Select the Program Manager group to which you want to add icons for AutoIndexer.
6. The **DRE Details** dialog is displayed.
Enter the following details for the DRE you want Autoindexer to index into, and click on **Next**.
 - IP Address**
The IP address (or name) of the machine on which the DRE is running.
 - Index Port**
The port that is used to index documents into the DRE.
 - Database**
The name of the DRE database you want to index into.
7. The **Autoindexer Details** dialog is displayed.
Enter the following details for AutoIndexer and click on **Next**.
 - Query Port**
The port AutoIndexer listens on.
 - Service Port**
The port AutoIndexer uses for DiSH communication.

8. The **Autoindexer Services** dialog is displayed.
Check the box if you want to start the AutoIndexer service immediately after installation, and click on **Next**.
9. The **Start Installation** dialog is displayed.
Click on **Next** to confirm the settings you have made and start the installation. Alternatively click on **Back** to return to previous dialogs if you want to make any changes.
10. The **Installing** dialog is displayed.
The progress of the installation process is indicated. If you want to abort the installation process, click on **Cancel**.
11. The **Installation Complete** dialog is displayed.
AutoIndexer has been installed successfully. Click on **Finish** to exit the installation.

Directory structure: Windows

The following files and folders are created in the installation directory when you install AutoIndexer (note that folders are show in **bold**):

data	Default location from which documents are indexed.
filters	Folder that contains executables used during the importing process.
binslave.cfg	Configuration file that contains settings for Binslave.
binslave.exe	Binslave executable used during the importing process to extract text from binary files.
importslave.exe	Executable that generates IDX files for the DRE.
omnislave.cfg	Configuration file that contains settings for Omnislave.
omnislave.exe	Omnislave executable that parses files not in HTML or PDF format to IDX files.
pdfslave.cfg	Configuration file that contains settings for PDFslave.
pdfslave.exe	PDFslave executable that parses PDF files to IDX files.
various DAT files	Files that Binslave uses.
Various DLL files	Files that Omnislave uses.
importTemp	Folder that contains temporary import data.
<installation_name>.cfg	Configuration file that contains the AutoIndexer settings.
<installation_name>.exe	AutoIndexer executable.
Install.log	Installation log file.
Uninstall.exe	Executable to uninstall AutoIndexer from your computer.

In addition, the following folder and files are created when you start the AutoIndexer service:

queue	Folder that stores queued action commands and the results of queued actions (if you have set the results to be stored).
<installation_name>.dirstat0	Store of which files from the filesystem have been indexed by AutoIndexer. A DIRSTAT file and backup are created for each AutoIndexer job.
<installation_name>.dirstat0.bak	
<installation_name>.lck	Internally used lock file for AutoIndexer.
<installation_name>.log	AutoIndexer log file.
<installation_name>.str	AutoIndexer structured configuration file.
<installation_name>.cfg.log	AutoIndexer configuration log file.

Installing under UNIX

1. Copy the AutoIndexer installer from the CD to your local disk.

2. Uncompress the installer using the command:

```
uncompress <Installer>.tar.Z
```

3. Un-tar the resulting file using the command:

```
tar -xvf <Installer>.tar
```

This creates a subdirectory called **Autoindexer-2.8.x**, which contains the following files:

```
LICENSE.TXT
```

```
Setup.sh
```

And the subdirectory:

```
Autoindexer
```

4. Enter the command **cd Autoindexer-2.8.x** to move to this directory.

5. Run the installer script, **./Setup.sh**.

The **Welcome** text is displayed. Press **v** to read the license agreement. When you have finished, press **y** to accept the agreement and continue with the installation.

6. The **Installation Actions** dialog is displayed.

Enter **1** to continue the AutoIndexer installation.

Enter **2** to cancel the installation.

7. Enter a name for your AutoIndexer installation and press **Enter**. By default this is **Autoindexer**.

8. Enter the full path for the location in which you want to install the AutoIndexer files, and press **Enter**. By default this is **Autonomy/<installation_name>**.

9. Enter the following value for your AutoIndexer installation:

ACI port

The port that AutoIndexer listens on for ACI commands.

Directory

The directory from which you want AutoIndexer to import content. The default is the **data** directory that the installer creates in your AutoIndexer installation directory.

10. Enter the following details for the **DRE** that you want AutoIndexer to index content into:

IP address

The IP address (or name) of the machine on which the DRE is running.

Query port

The port number used to send queries to the DRE.

Index port

The port number used to index content in to the DRE.

11. The **Autonomy Autoindexer Installation** text is displayed. Check that your settings are correct, and press **Enter** to confirm your settings and to install AutoIndexer. If you want to change a setting, enter the corresponding number, press **Enter** and then enter a new value for the setting. Alternatively, type **X** or press **Ctrl+C** to cancel the installation.
12. The **Installation complete** dialog is displayed. You have successfully installed AutoIndexer. Press **Enter** to finish.

Directory structure: UNIX

The following folders and files are created in your installation directory when you install AutoIndexer (note that folders are shown in **bold**):

data	Default location from which documents are indexed.
filters	Folder that contains executables used during the importing process.
binslave.cfg	Configuration file that contains settings for Binslave.
binslave.exe	Binslave executable used during the importing process to extract text from binary files.
importslave.exe	Executable that generates IDX files for the DRE.
omnislave.cfg	Configuration file that contains settings for Omnislave.
omnislave.exe	Omnislave executable that parses files not in HTML or PDF format to IDX files.
pdfslave.cfg	Configuration file that contains settings for PDFslave.
pdfslave.exe	PDFslave executable that parses PDF files to IDX files.
various DAT files	Files that Binslave uses.
various SO files	Files that Omnislave uses.
importTemp	Folder that contains temporary import data.
<installation_name>Autoindexer.cfg	Configuration file that contains the AutoIndexer settings.
<installation_name>Autoindexer.exe	AutoIndexer executable.
Start.sh	Start script for AutoIndexer.
Stop.sh	Stop script for AutoIndexer.
Uninstall.sh	Script to uninstall AutoIndexer from your computer.

In addition, the following folder and files are created when you start the AutoIndexer service:

queue	Folder that stores queued action commands and the results of queued actions (if you have set the results to be stored).
<installation_name>Autoindexer.lock	Internally used lock file for AutoIndexer.
<installation_name>Autoindexer.log	AutoIndexer log file.
<installation_name>Autoindexer.str	AutoIndexer structured configuration file.
<installation_name>Autoindexercfg.log	AutoIndexer configuration log file.

3. Configuring AutoIndexer

The settings that determine how AutoIndexer operates are contained in the AutoIndexer configuration file, which is located in your installation directory. You can modify these settings in order to customize AutoIndexer according to your requirements.

Entering Boolean values

For parameters that require Boolean settings the following settings are interchangeable

TRUE = true = ON = on = Y = y = 1

FALSE = false = OFF = off = N = n = 0

Entering string values

If the value that you want to enter for a parameter that requires a string contains quotation marks, you must put the value into quotation marks and escape each quotation mark that the string contains by putting a slash in front of it.

For example:

```
FIELDSTART0="<font face=\"arial\"size=\"+1\"><b>"
```

Here the beginning and end of the string is indicated by quotation marks while all quotation marks that are contained in the string are escaped.

If you want to enter a comma separated list of strings for a parameter, and one of the strings contains a comma, you must indicate the start and the end of this string with quotation marks.

For example:

```
ParameterName=cat,dog,bird,"wing,beak",turtle
```

If any string within a comma separated list contains quotation marks, you must put this string into quotation marks and escaped the quotation marks in the string by putting a slash in front of them.

For example:

```
ParameterName="<font face=\"arial\"size=\"+1\"><b>",dog,bird,"wing,beak",turtle
```

Applying modifications to AutoIndexer's operation

New configuration settings only take effect once the AutoIndexer service is stopped and restarted.

Configuration file sections

The AutoIndexer configuration file comprises the following sections:

- [License]
- [Server]
- [Service]
- [Configuration]
- [Default]
- [Jobn]

Note: for import parameters that you can specify in the configuration file's [Default] and [MyJob] section, please refer to the **Import module** manual.

[License] section

This section contains the licensing details. You should not edit this section, as this could stop AutoIndexer working.

Key

The license key.

Holder

The name of the license holder.

[Service] section

The [Service] section contains the details that AutoIndexer requires, when it is run as a service under Autonomy's Distributed Service Handler (DiSH).

ServicePort

The number of the port used for DiSH communication.

ServiceControlClients

The IP addresses of machines which can control this service.

ServiceStatusClients

The IP addresses of machines which can obtain the status of the service.

[Server] section

This section contains general settings for indexing and querying.

Port

Specify the port that client machines use to communicate with the AutoIndexer.

QueryClients

Enter the IP address of machines that are permitted to query AutoIndexer.

If you want to permit a number of machines to query AutoIndexer, you must separate the individual addresses with a comma. For example, **196.172.86.220,196.172.87.11** (note that there is no space before or after the comma).

Enter * to permit all machines to query AutoIndexer. This is the default.

AdminClients

Enter the IP address of machines that are permitted to administer AutoIndexer.

If you want to permit a number of machines to administer AutoIndexer, you must separate the individual addresses with a comma. For example, **196.172.86.220,196.172.87.11** (note that there is no space before or after the comma).

Enter * to permit all machines to administer AutoIndexer. This is the default.

Threads

The number of threads that AutoIndexer uses to communicate with clients. By default this is **10** (that is AutoIndexer can have **10** simultaneous communications).

MaxInputString

Specify the maximum number of characters that can be sent in an HTTP request string to AutoIndexer. By default this is **64000**.

For example:

MaxInputString=100

In this example, an HTTP request that is sent to the AutoIndexer can have a maximum of **100** characters.

ExplicitHost

If you are using more than one network interface (that is if you have more than one network card and hence multiple IP addresses), **ExplicitHost** allows you to specify which of the IP addresses AutoIndexer should use. If you have only one network interface, you do not need to set **ExplicitHost**. By default this is not enabled.

QueueCleanSeconds

Enter the number of seconds after which you want a completed action to be deleted from the cache and from disk. (Once a queued action is completed its status and its results are stored in the cache.)

XMLEncoding

Enter the encoding that you want to use for XML that AutoIndexer returns by default. By default this is **UTF-8**.

[Default] section

The [Default] section contains the default settings that the AutoIndexer jobs use. If you want to create a job that uses different settings, you should configure these in the [Job*n*] section of the appropriate job (settings that are specified in the [Job*n*] section override default settings for this job).

If a setting that is specified in the [Default] section is also specified in the [Configuration] section, the setting in the [Configuration] section overrides the setting in the [Default] section.

Note: for import parameters that you can specify in the configuration file's [Default] section, please refer to the **Import module** manual.

PollingMethod

Allows you to specify how the AutoIndexer obtains the files that it will process. You can enter one of the following options:

1

Sets AutoIndexer to File Polling. AutoIndexer reads the **filePollFilename** text file (located in the specified **fileBaseDirectory**) that lists a set of documents with their file path. It then finds, imports and indexes these documents.

AutoIndexer keeps track of which files have been indexed, and when it runs the next time it only looks at new files that have been appended to the bottom of the list (it ignores new files if they are not at the bottom of the list).

File Polling is more scalable than Directory Polling, as it doesn't require directory tree traversing and storage in memory of the directory structure.

2

Sets AutoIndexer to Directory Polling. AutoIndexer imports and indexes any file that is contained in the **directoryPathCSVs** directory, provided it meets the criteria that you have specified.

Directory Polling is slightly more system intensive than File Polling and uses more memory.

PollingPeriod

Enter the amount of time that you want to elapse between each AutoIndexer poll. You can enter one of the following:

0

If you set **PollingPeriod** to 0, AutoIndexer polls just once. This is the default setting.

Note: you must not set **PollingPeriod** to 0 if you are using the **Import** action (please refer to the **Importing individual files** section for further details).

n

The number of milliseconds that you want to elapse between each AutoIndexer poll. If you set **PollingPeriod**, for example, to **86400000**, 1 day will elapse between each AutoIndexer poll (86400000 milliseconds=1 day).

n seconds

The number of seconds that you want to elapse between each AutoIndexer poll followed by seconds. For example, if you enter **3600 seconds**, AutoIndexer will poll every hour (1 hour=3600 seconds).

n minutes

Enter the number of minutes that you want to elapse between each AutoIndexer poll followed by minutes. For example, if you enter **10 minutes**, AutoIndexer will poll every 10 minutes.

n hours

Enter the number of hours that you want to elapse between each AutoIndexer poll followed by hours. For example, if you enter **1 hour**, AutoIndexer will poll every hour.

n days

Enter the number of days that you want to elapse between each AutoIndexer poll followed by days. For example, if you enter **1 day**, AutoIndexer will poll every day (this is the default setting).

n weeks

Enter the number of weeks that you want to elapse between each AutoIndexer poll followed by weeks. For example, if you enter **1 week**, AutoIndexer will poll every week.

n months

Enter the number of months that you want to elapse between each AutoIndexer poll followed by months. For example, if you enter **1 month**, AutoIndexer will poll every month.

RemoveLogFileOnStart

Enter **on** if you want to remove the existing log file every time AutoIndexer starts running.

Enter **off** if you want to append to the log file every time AutoIndexer runs. This is the default setting.

Number

The total number of jobs that you want AutoIndexer to carry out. Note that you must list each job with its number in sequential order, starting from **0**.

For example:

Number=3

0=FirstJob

1=SecondJob

2=ThirdJob

MaxLogKBytes

Enter the maximum size (in Kilobytes) that the log file can reach before it is renamed to **.log.previous** and a new log file is created.

For example:

MaxLogKBytes=4000

In this example the log file is renamed when it reaches a size of 4 MegaBytes (=4000 KiloBytes).

PollingAction

Allows you to specify which action AutoIndexer should execute when it polls. Enter one of the following:

1

All IDX files that are located in the **DirectoryPathCSVs** directory are indexed into the **DREHost** DRE.

2

All files that are located in the **DirectoryPathCSVs** directory are imported into IDX format and then indexed into the **DREHost** DRE.

7

All files that are located in the **DirectoryPathCSVs** directory are imported into IDX format and then indexed into the **DREHost** DRE. AutoIndexer also checks if any files that have previously been indexed into the DRE have been deleted from the **DirectoryPathCSVs** directory. It deletes any files from the DRE that the **DirectoryPathCSVs** directory doesn't contain anymore.

8

AutoIndexer deletes all files from the DRE that it has indexed.

16

Note that this option only applies if you are using DRE 4.

All IDX files that are located in the **DirectoryPathCSVs** directory are indexed into the **DREHost** DRE. If any of the documents already exists in the DRE, the existing document's fields are updated. This does not include the content of the document.

ACIPort

If you have set **PollingAction** to **16**, you need to specify the ACI port that by which ACI action commands are sent to the DRE 4 (this must be the port that you have specified using the **Port** parameter in the DRE's configuration file).

UpdateIndexBufferSize

If you have set **PollingAction** to **16**, **UpdateIndexBufferSize** allows you to specify (in kilobytes) how much memory AutoIndexer can use for buffered indexing. By default this is 1 megabyte. The indexing jobs are buffered until the maximum is reached and then indexed in one go.

UpdateReplaceBatchSize

If you have set **PollingAction** to **16**, **UpdateReplaceBatchSize** allows you to specify how many documents AutoIndexer should accumulate before field replacement is carried out. By default this is **100**. Once AutoIndexer has accumulated the specified number it updates the fields of any documents that already exist in the DRE with the fields of the new documents (except the document's content).

PollingPostAction

Allows you to specify which action AutoIndexer should execute after the file or files that are contained in the **DirectoryPathCSVs** directory have been processed. Enter one of the following:

0

AutoIndexer does not execute any action. This is the default setting.

1

AutoIndexer deletes the files after it has processed it, provided that you have set **AllowOriginalFileDeletion** to **true**.

2

AutoIndexer moves the files to the **MoveToDirectory** directory. It keeps the subdirectory structure.

3

AutoIndexer copies the files to the **CopyToDirectory** directory. It keeps the subdirectory structure.

AllowOriginalFileDeletion

If you have set **PollingPostAction** to **1**, you must also set **AllowOriginalFileDeletion** to **true** to indicate that you are sure that you want to delete all files that you have stored in the **DirectoryPathCSVs** directory. This prevents accidental file deletion.

MoveToDirectory

If you have set **PollingPostAction** to **2**, you can use **MoveToDirectory** to specify to which directory the file or files that are contained in the **DirectoryPathCSVs** directory are moved after they have been processed.

For example:

```
MoveToDirectory=D:\projects\autoindexer\processed\
```

In this example the file (or files) contained in **DirectoryPathCSVs** directory will be moved to **D:\projects\autoindexer\processed** when they have been processed.

CopyToDirectory

If you have set **PollingPostAction** to **3**, you can use **CopyToDirectory** to specify which directory the file or files that are contained in the **DirectoryPathCSVs** directory are copied to after they have been processed.

For example:

```
CopyToDirectory=D:\projects\autoindexer\processed\
```

In this example the file (or files) contained in **DirectoryPathCSVs** directory will be copied to **D:\projects\autoindexer\processed** when they have been processed.

PollingMaxNumber

Enter the maximum number of files that you want AutoIndexer to process for each polling cycle. The default setting is **2000**.

FilenameOutputMode

If you set **FilenameOutputMode** to **true**, AutoIndexer creates a text file that lists the names of the documents, which are contained in the **DirectoryPathCSVs** directory.

DreHost

The name or IP address of one or more machines on which the DREs are running into which AutoIndexer indexes data.

To specify multiple machines enter a comma-separated list (there must be no space before or after a comma).

For example:

```
DreHost=localhost,10.1.1.19
```

In this example AutoIndexer indexes files into the DREs on **localhost** and on machine **10.1.1.19**.

QueryPort

Enter one or more port numbers by which AutoIndexer sends queries to one or more DREs.

To specify multiple port numbers enter a comma-separated list (there must be no space before or after a comma).

For example:

QueryPort= 6000,8000

In this example AutoIndexer sends queries to DREs using the ports **6000** and **8000**.

Note: you must set this parameter if you are setting up AutoIndexer to work with the ACLCheck security plug-in.

IndexPort

Enter one or more port numbers by which AutoIndexer indexes documents into one or more DREs.

To specify multiple port numbers enter a comma-separated list (there must be no space before or after a comma).

For example:

IndexPort=6000,8000

In this example AutoIndexer indexes documents into DREs using the ports **6000** and **8000**.

PollingPostActionProcessIdxFiles

If you set **PollingPostActionProcessIdxFiles** to **true**, **PollingPostAction** will include the **MoveToDirectory** directory IDX files in its action (that is it will delete, move or copy them alongside the other files that the **MoveToDirectory** directory contains).

If you set **PollingPostActionProcessIdxFiles** to **false**, **PollingPostAction** will not include the IDX files. This is the default setting.

ImportIDXFilesAction

Allows you to specify which action AutoIndexer should execute after the IDX files that are contained in the **DirectoryPathCSVs** directory have been indexed. Enter one of the following:

0

AutoIndexer deletes the IDX files after it has processed it.

1

AutoIndexer moves the IDX files to the **ImportIDXFilesMoveTo** directory.

2

AutoIndexer does not execute any action. This is the default setting.

ImportIDXFilesMoveTo

Allows you to specify which directory the IDX files that are contained in the **DirectoryPathCSVs** directory are moved to after indexing, if you have set **ImportIDXFilesAction** to **1**.

For example:

```
ImportIDXFilesMoveTo=D:\projects\autoindexer\processed\idx\
```

In this example the IDX files that are contained in the **DirectoryPathCSVs** directory are moved to **D:\projects\autoindexer\processed\idx** after they have been indexed.

FilePollFilename

Enter the name of the file (with the file path if this file is not contained in the installation directory) from which you want AutoIndexer to read the list of files that should be processed. (This only applies to File Polling, which you have specified by setting **PollingMethod** to **1**).

For example:

```
FilePollFilename=queue
```

In this example AutoIndexer reads the list of files that it will import and index from the **queue** file.

FileBaseDirectory

Enter the path to the directory in which the files are stored that are listed in the **FilePollFilename** file.

For example:

```
FileBaseDirectory=c:\Files\
```

FilePollFileMaxSizeKB

Enter the maximum size (in Kilobytes) that the polled **FilePollFilename** file can reach before it is renamed (given the extension **.previous**) and a new poll file is created. (This only applies to File Polling, which you have specified by setting **PollingMethod** to **1**).

For example:

```
FilePollFileMaxSizeKB=4000
```

In this example the poll file is renamed when it reaches a size of 4 MegaBytes (=4000 KiloBytes).

DirectoryPathCSVs

Enter the full path to one or more directories, which contain the files that you want AutoIndexer to process. (This only applies to Directory Polling, which you have specified by setting **PollingMethod** to **2**).

If you want to enter multiple directory paths, you must separate them with commas (there must be no space before or after a comma).

For example:

```
DirectoryPathCSVs=D:\projects\autoindexer\files,F:\archive
```

Entering a UNC path:

By default AutoIndexer is run as a systems account and cannot access UNC paths. If you want to enter a UNC path for **DirectoryPathCSVs** and you are running AutoIndexer as a service, you must run the service as a user who has permission to access the specified UNC path:

1. Display the Windows **Services** dialog and double-click on the AutoIndexer service.
2. In the **Log On As** field select **This Account** and enter your account and password details.
3. Click on **OK** and close the Services dialog.

DirectoryPathRecurseMatchCSVs

If you have specified **DirectoryFileMatch** but don't want to poll all the subdirectories of **DirectoryPathCSVs**, you can use **DirectoryPathRecurseMatchCSVs** to restrict which subdirectories AutoIndexer polls. Only directories that match one of the **DirectoryPathRecurseMatchCSVs** wildcard strings are polled.

If you want to enter multiple strings, you must separate them with commas (there must be no space before or after a comma).

For example:

```
DirectoryFileMatch=*.doc
```

```
DirectoryPathRecurseMatchCSVs=*archive,current
```

In this example AutoIndexer will only poll files in the **DirectoryPathCSVs** directory that have the extension **.doc** and are contained in subdirectories whose names match ***archive** or **current**.

DirectoryPathRecurseCantHaveCSVs

If you have specified **DirectoryFileMatch** but don't want to poll all the subdirectories of **DirectoryPathCSVs**, you can use **DirectoryPathRecurseCantHaveCSVs** to restrict which subdirectories AutoIndexer polls. Directories that match one of the **DirectoryPathRecurseMatchCSVs** wildcard strings are not polled.

If you want to enter multiple strings, you must separate them with commas (there must be no space before or after a comma).

For example:

```
DirectoryPathRecurseCantHaveCSVs=*archive,current
```

In this example AutoIndexer will not poll files in the **DirectoryPathCSVs** directory that are contained in subdirectories whose names match ***archive** or **current**.

DirectoryFileMatch

Enter a wild card to specify which files in the **DirectoryPathCSVs** directory AutoIndexer should poll.

For example:

```
DirectoryFileMatch=*.doc
```

In this example AutoIndexer will only poll files in the **DirectoryPathCSVs** directory that have the extension **.doc**.

DirectoryRecurse

Enter **true** if you want to poll the subdirectories of the **DirectoryPathCSVs** directory.

Enter **false** if you only want to poll the **DirectoryPathCSVs** directory. This is the default setting.

DirectoryMustHaveCSVs

Allows you to specify a string that must appear in the directory path of a document for the document to be polled.

For example:

```
DirectoryMustHaveCSVs=*/temp/*
```

In this example AutoIndexer only polls documents that are contained in a directory, which contains the string **/temp/**.

DirectoryCantHaveCSVs

Allows you to specify one or more strings that must not appear in the directory path of a document for the document to be polled.

If you want to enter multiple strings, you must separate them with commas (there must be no space before or after a comma).

For example:

```
DirectoryCantHaveCSVs=*.sys,*.bat,*.exe
```

In this example AutoIndexer will not poll documents that are contained in a directory which contains the string **.sys**, **.bat** or **.exe**.

DirectoryBeforeDate and DirectoryAfterDate

You can use **DirectoryBeforeDate** and **DirectoryAfterDate** to define the time span within which documents must have been modified for AutoIndexer to process them.

For example:

```
DirectoryBeforeDate=-3
```

```
DirectoryAfterDate=3
```

In this example AutoIndexer only processes documents whose modification date lies between the current date **minus 3** days and the current day **plus 3** days (which, for example, can be the case if a document stems from another time zone).

DirectoryDeleteCheckFile

If you have set **PollingAction** to **7**, you can use **DirectoryDeleteCheckFile** to force AutoIndexer to check if a specific network file exists before it deletes files from the DRE. If the specified file doesn't exist, AutoIndexer assumes that the network connection has failed, and does not delete any files.

DirectoryDeleteMaxPercentage

If AutoIndexer is about to delete a percentage of documents greater than this value it will cancel the deletion command. This allows you to avoid deleting too many files if the network connection has failed. For example:

```
DirectoryDeleteMaxPercentage=80
```

In this example AutoIndexer will not delete any files from the DRE if they make up more than **80** percent of the documents that the DRE contains.

DirectoryPollHashSlots

Allows you to specify the number of hash slots that are available (a hash table is created for each AutoIndexer job). The default is **16384**.

The total number of filenames that can be stored in a hash table is determined by **DirectoryPollHashSlots** multiplied by **DirectoryPollEntriesPerHashSlot**. If there's no available hash entry for a new filename to be added to the hash table, AutoIndexer reallocates memory to double the size of the appropriate hash slot (by doubling the number of hash entries).

Generally, you should try to ensure that the initial hash table capacity is just large enough for the maximum number of files that AutoIndexer expects to retrieve per job, because reallocating memory to increase capacity can slow down the indexing process. However, if AutoIndexer has a large number of jobs scheduled, you should set a lower value for **DirectoryPollHashSlots**, so that the memory needed to create hash tables for all jobs doesn't exceed the memory available.

If you have a lot of jobs it may also be sensible to set up several machines with an AutoIndexer installation, so the load can be spread (to reduce the amount of memory that the importing process takes up on each machine).

DirectoryPollEntriesPerHashSlot

Allows you to specify the number of entries that are available per hash slot (a hash table is created for each AutoIndexer job). The default is **32**.

DirectoryCheckAllExist

Enter true if you want to check that all **DirectoryPathCSVs** directories exist, which you have instructed AutoIndexer to poll. If any of them does not exist AutoIndexer stops polling.

If you enter false AutoIndexer attempts to poll all the directories that you have specified for **DirectoryPathCSVs**.

DeleteEscapeReferences

If you have set **PollingAction** to **7**, you can set **DeleteEscapeReferences** to **true** if you want AutoIndexer to escape the DRE reference of the files that it is processing.

If you set **DeleteEscapeReferences** to **false** AutoIndexer does not escape the reference of the files that it is processing. This is the default setting.

DeleteReferenceFromContent

Enter **true** if you want the DRE to delete all documents after indexing that contain the DRREFERENCE defined by **DeleteReferenceStart** and **DeleteReferenceEnd**.

DeleteReferenceStart

Enter a string to specify where in the file the content starts that AutoIndexer should use as DRREFERENCE.

DeleteReferenceEnd

Enter a string to specify where in the file the content ends that AutoIndexer should use as DREREFERENCE.

ReplaceByField

If you are using Mapped NT Security and have set AutoIndexer to change a document's DREREFERENCE during the importing process, you need to set **ReplaceByField** to **true**. AutoIndexer indexes the file with an additional reference field which contains the file's filename. This ensures that when the ACL is updated, the DRE can recognize the file and make the appropriate changes.

Note: you also need to set **ReplaceByField** to **true** in the configuration file of the ACL Check executable and add a separate, independent field process to the DRE 4 configuration file, which identifies DREFILENAME as a reference field.

StableCheckMinWaitTime

Allows you to check if a file is in use. By entering the number of seconds in which the file must not have been used. When AutoIndexer cycles it checks if a file has been changed within the specified number of seconds. If it has AutoIndexer assumes that the file is still in use and does not import it.

The default setting is **0** seconds (checking is disabled).

ImportReadChecking

Enter **true** if you want AutoIndexer to attempt to read from a file before it imports this file. If the file is not readable AutoIndexer assumes that the file is in use (for example, being copied) and does not import it.

Enter **false** if you do not want AutoIndexer to check if a file is readable before importing it.

IndexLocalFile

Enter **true** if you want AutoIndexer to index the files from a local location.

Enter **false** if you want AutoIndexer to index the files over the network.

Note: this parameter is equivalent to **IndexOverSocket**.

IndexLocalFilePathReplace

If you are not indexing over socket and your AutoIndexer and DRE are located on different machines on the network, you need to specify **IndexLocalFilePathReplace** and **IndexLocalFilePathString**. These settings allow you to specify the relative path to IDX files, so that the DRE can index them from the machine on which they are located.

Use **IndexLocalFilePathReplace** to enter a string to specify which part of the original references to IDX files should be replaced with the string specified for **IndexLocalFilePathString**.

IndexLocalFilePathString

If you are not indexing over socket and your AutoIndexer and DRE are located on different machines on the network, you need to specify **IndexLocalFilePathReplace** and **IndexLocalFilePathString**. These settings allow you to specify the relative path to IDX files, so that the DRE can index them from the machine on which they are located.

Use **IndexLocalFilePathString** to enter the string with which you want to replace the string that you have specified for **IndexLocalFilePathReplace**.

IndexForceDatabase

You can enter the name of a DRE database to force AutoIndexer to index files into this database.

IndexMode

You can enter one of the following index modes:

NONE

AutoIndexer indexes files.

REFERENCE

If a document is indexed that has the same DREREFERENCE as a document that the DRE already contains, the DRE deletes the document that it already contains and replaces it with the new one. This is the default.

REFERENCEMATCH nn

If a document is indexed whose content more than nn percent similar to the content of a document that the DRE already contains, the DRE deletes the document that it already contains and replaces it with the new one.

DELETE

When AutoIndexer has finished indexing the files, the DRE deletes them.

IndexOverSocket

Enter **true** if you want AutoIndexer to index the files over the network.

Enter **false** if you want AutoIndexer to index the files from a local location.

Note: **IndexOverSocket** is equivalent to **IndexLocalFile**. By default it is set to **true**, unless you have set **IndexLocalFile** to **true**, in which case **IndexOverSocket** defaults to **false**.

IndexToFile

Enter **true** if you want to send any data that AutoIndexer outputs to the specified **IndexDirectory**.

Enter **false** if you want to send any data that AutoIndexer outputs to the DRE. This is the default setting.

IndexDirectory

Enter the path to the directory in which AutoIndexer stores output files if you have set **IndexToFile** to **true**.

MappedNTSecurity

Enter **true** if you want to index NT security information into the **DREFieldName** field.

Enter **false** if you don't want to index NT security information into the **DREFieldName** field. This is the default setting.

MappedNetwareSecurity

Enter **true** if you want to index Netware security information into the **DREFieldName** field.

Enter **false** if you don't want to index Netware security information into the **DREFieldName** field. This is the default setting.

MappedUnixSecurity

Enter **true** if you want to index UNIX security information into the **DREFieldName** field.

Enter **false** if you don't want to index UNIX security information into the **DREFieldName** field. This is the default setting.

EncryptACLEntries

Enter **true** if you don't want to debug the ACL extraction process. This is the default setting.

Enter **false** if you want to be able to debug the ACL extraction process.

Note: this is for debugging only.

DREFieldName

Enter the name of the DRE field in which you want to store the ACL security information. By default this is **AUTONOMYMETADATA**.

NetwareServer

Enter the IP address (or the hostname) of the machine on which the netware server is running.

NetwareUsername

Enter the administrator's netware username.

NetwarePassword

Enter the encrypted netware password.

NetwareCacheSize

Enter the maximum number of items that the cache can hold. When the cache is full the oldest item is replaced with a new item. The default setting is **1000**.

NetwareCacheExpiry

Enter the number of minutes that an item can stay in the cache hold before it is deleted. The default setting is **60**.

SpawnSecurityProcesses

Enter **true** if you want to spawn the security update process. This is the default setting.

Enter **false** if you don't want to spawn the security update process.

NTSecurityPath

Enter the path to the mapped security folder.

For example:

./mapped

NetwareCleanUsernames

Enter **true** if you want to convert distinguished user names to simple usernames.

Enter **false** if you want to convert distinguished user names to simple usernames. This is the default setting.

For example:

NetwareCleanUsernames=true

In this example, the following distinguished user name is converted to **testuser**:

CN=testuser/OU=testdepartment/O=testcompany

[Configuration] section

The [Configuration] section contains the configuration settings that determine how AutoIndexer operates. If a settings that is specified in the [Configuration] is also specified in the [Default] section, the setting in the [Configuration] section overrides the setting in the [Default] section.

Number

The total number of jobs that you want AutoIndexer to carry out. Note that you must list each job with its number in sequential order, starting from **0**.

For example:

Number=3

0=FirstJob

1=SecondJob

2=ThirdJob

[Job] section

Each [Job] section contains settings that apply only to this job.

If you specify settings for a job that are different from settings which you have specified in the [Default] section, the job will use the settings that you have specified for it in its [Job] section and ignore the settings in the [Default] section.

Note: for import parameters that you can specify in the configuration file's [MyJob] section, please refer to the **Import module** manual.

4. Importing individual files

action=Import: importing individual files

http://<host>:<port>/action=Import<mandatory_parameters>&<optional_parameters>

The **Import** action allows you to import an individual file into IDX file format. It also indexes the IDX files that it produces into the DRE if you have configured appropriate settings in the AutoIndexer configuration file.

You can call the **Import** action from your browser or via the ACI API.

Note: if you are using the **Import** action, the **PollingPeriod** parameter in the AutoIndexer configuration file must not be set to **0**.

Command parameters:

Mandatory: **FileName**=<file_name>

or

FileData=<file_content>

Optional: **FileExtension**=<file_extension>

JobName=<job_name>

StubIdxFileData=<stub_idx_content>

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the ACI port by which commands are sent to the DRE.

<file_name>

The path to the directory in which the file you want to import to IDX file format is stored. If the file is stored in the main AutoIndexer directory, you can specify only the name of the file.

<file_content>

Specify the content of the file that you want to import to IDX file format, if you want to send it over the socket rather than have AutoIndexer reading it from disk.

<optional_parameters>

You can enter one or more of the following parameters (note that you must separate individual parameters with an ampersand).

Note: you can also add any configuration parameter that can be set for individual AutoIndexer jobs to the command string. If you add a configuration parameter to the command string that clashes with a parameter that the **JobName** setting picks up, the setting of the added configuration parameter overrides its corresponding **JobName** setting.

FileExtension=<file_extension>

If you are sending the content of the file over the socket instead (that is if you are executing the Import action command using the **FileData** parameter rather than the **FileName** parameter), you need specify the extension of the file that you are importing for **FileExtension**.

JobName=<job_name>

Allows you to specify the name of the AutoIndexer configuration file job from which you want the default parameters to be read that are applied during the importing process of the file. By default the parameters are read from the **Default** section.

StubIdxFileData=<stub_idx_content>

Allows you to add one or more additional pieces of metadata in IDX file format to the file that you are importing (multiple pieces of metadata must be separated by **%OA**).

For example:

```
&StubIdxFileData=%23DREFIELD%20DREREFERENCE=%221234%22%OA%23DREFIELD%20DRETITLE=%22horticulture%22
```

In this example, a DREREFERENCE field (with the value **221234**) and a DRETITLE field (with the value **horticulture**) are added to the file that you are importing.

Example:

```
http://12.3.4.56:4000/action=Import&FileName=MyFile
```

This command uses port **4000** to send the **MyFile** file to AutoIndexer, which is located on a machine with the IP address **12.3.4.56**. AutoIndexer imports the file into IDX file format, and indexes it into the DRE (if it has been configured to index the file). AutoIndexer then returns an index ID for the file.

Index

A

ACI actions
 Import, 33
 ACIPort (AutoIndexer.cfg), 18
 AdminClients (Configuration setting), 14
 AllowOriginalFileDeletion (AutoIndexer.cfg), 19
 Applying modifications, 11
 AutoIndexer, 1, 2, 11, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 32
 Installing, 3
 System requirements, 3
 AutoIndexer.cfg, 13
 ACIPort, 18
 AllowOriginalFileDeletion, 19
 CopyToDirectory, 20
 DeleteEscapeReferences, 26
 DeleteReferenceEnd, 27
 DeleteReferenceFromContent, 26
 DeleteReferenceStart, 26
 DirectoryBeforeDate, 25
 DirectoryCantHaveCSVs, 25
 DirectoryCheckAllExist, 26
 DirectoryDeleteCheckFile, 25
 DirectoryDeleteMaxPercentage, 25
 DirectoryFileMatch, 24
 DirectoryMustHaveCSVs, 24
 DirectoryPathCSVs, 23
 DirectoryPathRecurseCantHaveCSVs, 24
 DirectoryPathRecurseMatchCSVs, 23
 DirectoryPollEntriesPerHashSlot, 26
 DirectoryPollHashSlots, 26
 DirectoryRecurse, 24
 DREFieldName, 30
 DreHost, 20
 EncryptACLEntries, 29
 FileBaseDirectory, 22
 FilenameOutputMode, 20
 FilePollFileMaxSizeKB, 22
 FilePollFilename, 22

ImportIDXFilesAction, 21
 ImportIDXFilesMoveTo, 22
 ImportReadChecking, 27
 indexDirectory, 29
 IndexForceDatabase, 28
 IndexLocalFile, 27
 IndexLocalFilePathReplace, 28
 IndexLocalFilePathString, 28
 IndexMode, 28
 IndexOverSocket, 29
 IndexPort, 21
 IndexToFile, 29
 MappedNetwareSecurity, 29
 MappedNTSecurity, 29
 MappedUnixSecurity, 29
 MaxLogKBytes, 18
 MoveToDirectory, 20
 NetwareCacheExpiry, 30
 NetwareCacheSize, 30
 NetwareCleanUsernames, 31
 NetwarePassword, 30
 NetwareServer, 30
 NetwareUsername, 30
 NTSecurityPath, 30
 Number, 17, 32
 PollingAction, 18
 PollingMaxNumber, 20
 PollingMethod, 16
 PollingPeriod, 16
 PollingPostAction, 19
 PollingPostActionProcessIdxFiles, 21
 QueryPort, 21
 RemoveLogFileOnStart, 17
 ReplaceByField, 27
 ServiceControlClients, 13
 ServicePort, 13
 SpawnSecurityProcesses, 30
 StableCheckMinWaitTime, 27
 UpdateIndexBufferSize, 19
 UpdateReplaceBatchSize, 19

B

Boolean values, 11

C

- Configuration file, 12
 - [License] section, 12
 - [Service] section, 13
- Configuration settings
 - AdminClients, 14
 - ExplicitHost, 15
 - Holder, 12
 - Key, 12
 - MaxInputString, 14
 - Port, 14
 - QueryClients, 14
 - QueueCleanSeconds, 15
 - Threads, 14
 - XMLEncoding, 15
- Configuring AutoIndexer
 - Applying modifications, 11
 - Entering Boolean values, 11
 - Entering string values, 11
- Content DRE, 1
- CopyToDirectory (AutoIndexer.cfg), 20

D

- DeleteEscapeReferences (AutoIndexer.cfg), 26
- DeleteReferenceEnd (AutoIndexer.cfg), 27
- DeleteReferenceFromContent (AutoIndexer.cfg), 26
- DeleteReferenceStart (AutoIndexer.cfg), 26
- Directory Polling, 1, 2, 16, 23
- Directory structure
 - UNIX, 10
 - Windows, 6
- DirectoryAfterDate (AutoIndexer.cfg), 25
- DirectoryBeforeDate (AutoIndexer.cfg), 25
- DirectoryCantHaveCSVs (AutoIndexer.cfg), 25
- DirectoryCheckAllExist (AutoIndexer.cfg), 26
- DirectoryDeleteCheckFile (AutoIndexer.cfg), 25
- DirectoryDeleteMaxPercentage (AutoIndexer.cfg), 25
- DirectoryFileMatch (AutoIndexer.cfg), 24
- DirectoryMustHaveCSVs (AutoIndexer.cfg), 24
- DirectoryPathCSVs (AutoIndexer.cfg), 23
- DirectoryPathRecurseCantHaveCSVs (AutoIndexer.cfg), 24

- DirectoryPathRecurseMatchCSVs (AutoIndexer.cfg), 23
- DirectoryPollEntriesPerHashSlot (AutoIndexer.cfg), 26
- DirectoryPollHashSlots (AutoIndexer.cfg), 26
- DirectoryRecurse (AutoIndexer.cfg), 24
- DREFieldName (AutoIndexer.cfg), 30
- DreHost (AutoIndexer.cfg), 20

E

- EncryptACLEntries (AutoIndexer.cfg), 29
- ExplicitHost (Configuration setting), 15

F

- File Polling, 1, 16, 22
- FileBaseDirectory (AutoIndexer.cfg), 22
- FilenameOutputMode (AutoIndexer.cfg), 20
- FilePollFileMaxSizeKB (AutoIndexer.cfg), 22
- FilePollFilename (AutoIndexer.cfg), 22

H

- Holder (Configuration setting), 12

I

- Import action, 33
- ImportIDXFilesAction (AutoIndexer.cfg), 21
- ImportIDXFilesMoveTo (AutoIndexer.cfg), 22
- Importing individual files, 33
- ImportReadChecking (AutoIndexer.cfg), 27
- indexDirectory (AutoIndexer.cfg), 29
- IndexForceDatabase (AutoIndexer.cfg), 28
- IndexLocalFile (AutoIndexer.cfg), 27
- IndexLocalFilePathReplace (AutoIndexer.cfg), 28
- IndexLocalFilePathString (AutoIndexer.cfg), 28
- IndexMode (AutoIndexer.cfg), 28
- IndexOverSocket (AutoIndexer.cfg), 29
- IndexPort (AutoIndexer.cfg), 21
- IndexToFile (AutoIndexer.cfg), 29
- Installing AutoIndexer, 3
 - Under UNIX, 8
 - Under Windows, 4

K

Key (Configuration setting), 12

L

[License] section (Configuration file), 12
Linux, 3

M

MappedNetwareSecurity (AutoIndexer.cfg), 29
MappedNTSecurity (AutoIndexer.cfg), 29
MappedUnixSecurity (AutoIndexer.cfg), 29
MaxInputString (Configuration setting), 14
MaxLogKBytes (AutoIndexer.cfg), 18
MoveToDirectory (AutoIndexer.cfg), 20

N

NetwareCacheExpiry (AutoIndexer.cfg), 30
NetwareCacheSize (AutoIndexer.cfg), 30
NetwareCleanUsernames (AutoIndexer.cfg), 31
NetwarePassword (AutoIndexer.cfg), 30
NetwareServer (AutoIndexer.cfg), 30
NetwareUsername (AutoIndexer.cfg), 30
NTSecurityPath (AutoIndexer.cfg), 30
Number (AutoIndexer.cfg), 17, 32

P

PollingAction (AutoIndexer.cfg), 18
PollingMaxNumber (AutoIndexer.cfg), 20
PollingMethod (AutoIndexer.cfg), 16
PollingPeriod (AutoIndexer.cfg), 16
PollingPostAction (AutoIndexer.cfg), 19
PollingPostActionProcessIdxFiles (AutoIndexer.cfg), 21
Port (Configuration setting), 14
.pos files, 1
.pos.bak files, 1

Q

QueryClients (Configuration setting), 14
QueryPort (AutoIndexer.cfg), 21
QueueCleanSeconds (Configuration setting), 15

R

RemoveLogFileOnStart (AutoIndexer.cfg), 17
ReplaceByField (AutoIndexer.cfg), 27

S

[Service] section (Configuration file), 13
ServiceControlClients (AutoIndexer.cfg), 13
ServicePort (AutoIndexer.cfg), 13
ServiceStatusClients (AutoIndexer.cfg), 13
SpawnSecurityProcesses (AutoIndexer.cfg), 30
StableCheckMinWaitTime (AutoIndexer.cfg), 27
String values, 11
System requirements, 3

T

Threads (Configuration setting), 14

U

UpdateIndexBufferSize (AutoIndexer.cfg), 19
UpdateReplaceBatchSize (AutoIndexer.cfg), 19

W

Windows, 3

X

XMLEncoding (Configuration setting), 15