

# Content Manager

Software Version 24.2

IDOL and Document Content Indexing  
Installation and Configuration

**opentext**<sup>™</sup>

Document Release Date: October 2023  
Software Release Date: April 2024

## Legal notices

Copyright 2008-2024 Open Text

The only warranties for products and services of Open Text and its affiliates and licensors (“Open Text”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

## Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for updated documentation, visit <https://www.microfocus.com/support-and-services/documentation/>.

## Support

Visit the [MySupport portal](#) to access contact information and details about the products, services, and support that OpenText offers.

This portal also provides customer self-solve capabilities. It gives you a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the MySupport portal to:

- View information about all services that Support offers
- Submit and track service requests
- Contact customer support
- Search for knowledge documents of interest
- View software vulnerability alerts
- Enter into discussions with other software customers
- Download software patches
- Manage software licenses, downloads, and support contracts

Many areas of the portal require you to sign in. If you need an account, you can create one when prompted to sign in.

# Contents

Introduction .....	9
Audience .....	9
Chapter summary .....	9
Related documents .....	9
Introduction to IDOL .....	11
About IDOL .....	11
OEM IDOL .....	11
Enterprise IDOL .....	11
IDOL components .....	11
IDOL in Content Manager .....	12
Document content searching .....	12
Searching metadata .....	13
Document Content, Record Titles and Record notes .....	13
Use IDOL Indexes Where Possible .....	13
IDOL search method .....	13
Optical Character Recognition (OCR) rendering .....	13
Auto-Classification .....	13
Connectors .....	13
IDOL OEM Installation .....	15
Requirements .....	15
Supported operating systems .....	15
System requirements .....	15
Other considerations .....	15
Separate disks .....	15
Separate computers .....	15
Separate computers for content engines .....	16
Additional content engines .....	16
Installing Additional IDOL Components .....	17
Installing CM IDOL service and Content services .....	17
Prerequisites .....	17
Installation steps .....	17
Command line installation .....	22
Content Manager Media Server .....	22
Repairing or removing Content Manager IDOL components .....	23
Upgrading IDOL components .....	23
Upgrading OEM IDOL .....	24

Upgrading OEM IDOL using Setup_CM_x64.exe .....	24
Upgrading OEM IDOL manually .....	24
Upgrading your document content indexes .....	26
Document Content Searching .....	28
Requirements .....	28
Document volume .....	28
Site size .....	28
Small sites .....	28
Medium sites .....	28
Large sites .....	29
Performance considerations .....	29
Failover and mirroring .....	29
Configuring document content indexing .....	29
Document content index menu in Content Manager Enterprise Studio .....	33
Configuring content index event processing .....	33
The re-indexing process .....	35
The Content Manager process .....	35
The IDOL process .....	35
Content Manager URN .....	36
Performing content indexing .....	37
Re-indexing records .....	37
Content indexing with multiple Content Manager clients .....	39
Indexing steps .....	40
Maintaining an IDOL index .....	45
Checking the IDOL index for errors .....	45
Optimizing the IDOL index .....	45
Optimization steps .....	46
Monitoring an IDOL Server .....	46
Steps to monitor IDOL Servers .....	46
Removing a document content index .....	47
Removal steps .....	47
Deleting an IDOL database .....	48
Suppressing document content indexing .....	49
For all files of a particular file type .....	49
For all electronic documents with a particular Record Type .....	49
Using IDOL to search record metadata .....	50
Configuring Content Manager to use IDOL to search metadata fields .....	50
Document Content, Record Titles and Record Notes .....	50
Use IDOL Indexes Where Possible .....	50
Configuration steps .....	50

Examples of metadata searching .....	51
Using the IDOL search method .....	51
Field specifiers for common restrictions .....	51
Field specifiers for advanced restrictions .....	52
Field specifiers for biasing result scores .....	52
OCR Rendering .....	53
Requirements .....	53
OCR rendering installation and configuration .....	53
Troubleshooting OCR rendering .....	57
File types .....	57
Content Manager Media Server log files .....	57
Auto-Classification .....	58
Requirements .....	58
Setting up Content Manager for Auto-Classification .....	58
Setting up IDOL for Auto-Classification .....	58
Training Classification terms .....	58
Fine-tuning training .....	63
Enabling Auto-Classification event processing .....	64
Running Auto-Classification .....	64
Reviewing Auto-Classification .....	65
Auto-Classification Confidence Level .....	65
Storing Auto-classification details .....	66
Connectors .....	67
Requirements .....	67
Configuration .....	67
Re-Indexing User Locations .....	67
IDOL Configuration .....	73
Securing the IDOL Server .....	73
Content Manager IDOL Service .....	73
Content Manager Content Service .....	74
Content Manager Media Server Service .....	74
Restricting Content Manager IDOL Service Control Commands .....	74
Mirroring .....	75
Simple .....	75
Tiered .....	75
Deactivating or removing a content engine .....	76
Service Changes .....	76
Configuration changes .....	76
Adding an additional content engine .....	77
Manual installation .....	77

Configuration changes .....	79
Moving a content engine .....	80
Backing up the content index .....	80
Manual backup .....	81
Scheduled backup .....	81
Example configuration .....	82
Exporting data .....	82
Restoring a backup .....	82
Importing data .....	83
Threads .....	83
Maximum queued transactions .....	83
DistributeByReference .....	84
DCI round-robin .....	84
MaxResults .....	84
MaxSyncDelay .....	85
Timeout .....	85
EngineTimeout .....	85
StateTokens .....	86
IndexCacheMaxSize .....	86
StopwordIndex .....	86
Paths .....	87
Database Read-only .....	87
Default Operator .....	87
IDOL component logs .....	87
Changing the log file locations .....	88
Searching options .....	88
AugmentSeparators .....	88
DiminishSeparators .....	89
IDOL stop words .....	89
Setting up IDOL to index Additional Fields .....	89
Monitoring IDOL .....	90
Services .....	90
Content index monitor .....	90
IDOL logs .....	90
IDOL logs .....	90
Application logs .....	90
Index logs .....	91
Action logs .....	91
Restoring or changing IDOL license .....	91
Changing IDOL license .....	91

Converting to an Enterprise IDOL license .....	91
Sample [License] Section .....	92
Troubleshooting .....	93
General troubleshooting .....	93
IDOL content service does not start .....	94
Investigate the IDOL Content Service.log .....	94
Investigate the license.log .....	94
Investigate the application.log .....	94
Troubleshooting license issues .....	95
Document content search returns records without documents .....	95
Content search does not find header or footer content .....	95
IDOL searches are timing out .....	95
Adjusting timeouts .....	95
Timeout-related configuration settings .....	96
Performance .....	96
Searching for special characters .....	96
Search structure .....	97
Example searches .....	97
Common log messages .....	97
IDOL and ISYS .....	98
Text conversion .....	98
Chains vs. content engines .....	98
Transition from ISYS .....	98
Appendix A IDOL Configuration Settings .....	99
Main IDOL Server configuration parameters .....	99
Service section .....	99
Server section .....	100
Distribution Settings section .....	109
Virtual Database Settings section .....	116
DAHEngines section .....	118
DIHEngines section .....	119
Security section .....	119
Logging section .....	120
Language Types section .....	126
IDOL content service configuration parameters .....	129
Service section .....	129
Server section .....	130
Term Cache section .....	143
Index Cache section .....	144
Section Breaking section .....	144

Paths section .....	145
Logging section .....	146
Field Processing section .....	152
Properties section .....	153
Language Types section .....	157
Databases section .....	160
XML document handling .....	161
Appendix B Default Ports Used By IDOL Components .....	163
Main IDOL Service .....	163
AgentStore .....	163
Category .....	163
DAH .....	163
DIH .....	163
Content Service 1 .....	163
Content Service 2 .....	164
Appendix C Supported File Types for IDOL Content Indexing .....	165
Appendix D Content Manager Fields Available in IDOL .....	166
Appendix E FieldText Methods .....	168



## Introduction

Content Manager can index the content of electronic documents for searching by using IDOL indexing software. While Content Manager comes with an OEM version of IDOL, it can also be used with Enterprise IDOL. For information about purchasing and implementing Enterprise IDOL in your organization, please contact your Sales Executive.

See [Installing Additional IDOL Components](#) in this document for installation and configuration of the Content Manager IDOL Components, including the IDOL Services and Content Services, and [Configuring document content indexing](#) for information on configuring document content indexing.

This Content Manager IDOL Installation and Configuration Guide describes the process of installing, upgrading, removing, configuring and backing up Content Manager document content indexing.

Even if you are familiar with Content Manager from earlier versions, it is recommended you read this guide in its entirety before you set out to deploy document content indexing.

## Audience

This guide assumes you are qualified IT support personnel with extensive knowledge of Microsoft Windows and the client server architecture, and system administrator with a sound understanding of Windows Services and Group Policy.

This guide also assumes you are familiar with Content Manager and its terms and functions.

## Chapter summary

The introduction to IDOL chapter outlines what IDOL is, what it is made up of, and how Content Manager uses IDOL.

The next two chapters cover the installation of the main IDOL package and the additional components available with Content Manager.

The chapters on searching, OCR rendering, auto-classification and connectors contain more detail about how Content Manager uses IDOL, and how to configure and operate those Content Manager modules.

The IDOL configuration chapter contains details about some of the configuration options available to a site, and some of the more common configuration changes sites make.

The first appendix is a glossary of all of the parameters in the default configuration files shipped with Content Manager.

The final chapter is on troubleshooting IDOL. It details common problems, as well as giving advice on how to investigate issues and the information that your Support representative may need.

## Related documents

IDOL documentation

You can find all IDOL and associated documentation at <https://softwaresupport.softwaregrp.com>. Customers need to create an account before accessing the content.

- IDOL Server Administration Guide
- Content Manager Connector Administration Guide
- Connector Framework Server Administration Guide
- IDOL DIH Administration Guide

#### Content Manager documentation

You can find all Content Manager documentation on the installation media, and in the installation directory folder **Documentation**.

- Content Manager client Help - **TRIM.chm**
- Specifications and Behaviors - **CM24.2\_Spec.pdf**

## Introduction to IDOL

### About IDOL

The Intelligent Data Operating Layer (IDOL) server integrates unstructured, semi structured, and structured information from multiple repositories through an understanding of the content. It delivers a real time environment to automate operations across applications and content, removing all the manual processes involved in getting information to the right people at the right time.

### OEM IDOL

An Original Equipment Manufacturer (OEM) version of IDOL can be installed using the **CM\_IDOLComponents\_x64.msi**. Unless otherwise stated, this document assumes you are working with the OEM IDOL package. If you're upgrading Content Manager, the OEM version of IDOL will be upgraded as a part of the upgrade process.

The OEM package is the same as the standard Enterprise IDOL, but is made up of only the IDOL components that Content Manager uses. The OEM license also means that only Content Manager is able to communicate with IDOL.

There is no requirement to install the OEM IDOL package with the Content Manager Workgroup Server, or any other Content Manager component. The only requirement is that the Content Manager Workgroup Server can communicate via HTTP with the main IDOL server.

### Enterprise IDOL

Enterprise IDOL is referred to in this document to differentiate between the OEM IDOL package shipped with Content Manager, and the fully licensed IDOL. Unlike the OEM IDOL, Enterprise IDOL (or "IDOL") can be used by any other compatible application or even by users directly.

Content Manager can be configured to use an Enterprise IDOL server in place of the Content Manager OEM IDOL package. While some of the back-end configuration may be different, the end-user experience will be identical.

For additional information about purchasing and implementing Enterprise IDOL in your organization please contact your Sales Executive.

### IDOL components

IDOL is made up of multiple components. Depending on a site's needs and resources, components can be installed on separate computers or turned off.

Each component has its own set of log files and in most cases its own configuration file. The configuration files (\*.cfg) for each component are located with the component's application (\*.exe) file.

By default, the OEM IDOL components are installed to the Content Manager application folder. The default working directory for each component is the Content Manager IDOL directory, for example, **C:\Micro Focus Content Manager\IDOL\**.

The OEM IDOL installation establishes:

- Content Manager IDOL Service
- Content Manager IDOL Content Service

**NOTE:** Records Manager 8.1 and earlier included two content services.

The Content Manager IDOL Content Services is an IDOL content engine. An IDOL installation may contain multiple content engines, each of which controls a portion of the IDOL index.

The main Content Manager IDOL Service controls several IDOL components:

- Distributed Index Handler (DIH)
- Distributed Action Handler (DAH)
- AgentStore
- Category

DIH and DAH manage the IDOL configuration and handle indexing and searching.

The IDOL AgentStore component and the IDOL Category component are used by the Content Manager Auto-Classification feature.

Content Manager Media Server is an additional IDOL component that is used by the Content Manager OCR feature and can convert images to text. For information on installing and setting up Media Server, see **CM24.2\_Install.pdf**.

## IDOL in Content Manager

IDOL functions in Content Manager are described below. Some of these functions may require separately licensed modules.

**NOTE:** Any search that can be completely handled by IDOL, whether it is a content type search (including Title/Notes), or a specialized IDOL syntax query string, will now return results in weighted order, with the highest scoring results presented first.

### Document content searching

Document content searching enables Content Manager users to search for records by the content of their attached electronic documents, rather than by the record metadata like a date, Location or record title. Anyone who has searched the Internet for information is aware how cumbersome it can be when a search returns thousands of documents. Similarly in Content Manager, it is a lot easier to find the record you want when you search for it by a combination of its metadata and the content of the document.

For example, search for all documents that have been checked in in March 2008 and that also have Project X in the text. This avoids finding all documents about Project X, which may be thousands.

In Content Manager, document content indexing (DCI) enables users to carry out word or phrase searches including synonyms, proximity searches and to use the Boolean operators AND, OR, NOT and exclusive OR.

**NOTE:** The content of some documents does not get indexed as they do not contain any text, or because the text is not accessible, for example, when the document is password protected. When a document's content has not been indexed, you may not be able to find the document using a document content search.

### Searching metadata

Content Manager can be configured to use IDOL for searching certain metadata fields. This can improve the performance of certain searches, in particular, searches involving both content and metadata search terms. See [Using IDOL to search record metadata](#).

### Document Content, Record Titles and Record notes

This option uses IDOL for any search that involves document content, record titles or record notes or any combination of those fields.

### Use IDOL Indexes Where Possible

This option uses a search optimizer to analyze any searches. If the optimizer decides the search can be completed exclusively by IDOL, for example, when the search only uses metadata available in IDOL, then the search is sent directly to IDOL.

### IDOL search method

Content Manager includes a specific IDOL search method. When using the IDOL search method, the Content Manager client passes the search query directly to the IDOL Server. This allows for a search across any fields that are indexed by IDOL, similar to an **AnyWord** query using the Content Manager client.

It also enables users to use many of the advanced IDOL search queries.

### Optical Character Recognition (OCR) rendering

Normally, text in an image file such as JPGs or TIFs cannot be extracted by the content indexing process. As a result, users cannot search the content of images that are attached to records in Content Manager.

With the OCR Rendering module, records that have an image file type can be automatically passed through an OCR engine, which produces a text rendition of the target image. This text rendition can then be added to the content index and searched.

### Auto-Classification

Content Manager includes a feature called Auto-Classification. When activated, this feature enables the automatic classification of suitable records.

### Connectors

The Content Manager connector facilitates the indexing by maintaining the security and access control settings from Content Manager on the records indexed into IDOL. This is useful for sites with an Enterprise IDOL instance, and enables other IDOL-compatible applications to interact with the Content Manager data while maintaining Content Manager's security.

Some examples of applications that require the Content Manager connector are ControlPoint and Universal Search.

The Content Manager **IDOL Connector Manage in Place** feature uses other IDOL connectors to provide records management features for documents located in other repositories. See *Content Manager Help > Administration > System Options > System Options Features page* for feature details.

## IDOL OEM Installation

This section refers to the installation of the OEM IDOL package included in the Content Manager installation. For details on installing Enterprise IDOL, see the IDOL Getting Started Guide for IDOL. For details on installing OEM IDOL packages from earlier versions of Content Manager, refer to the install documentation of the appropriate version.

### Requirements

The following are the requirements for the OEM IDOL package included with Content Manager. For requirements for IDOL packages shipped with earlier versions of Records Manager/TRIM, see the relevant version Specifications and Limitations PDF, or requirements for Enterprise IDOL, see the relevant IDOL documentation.

### Supported operating systems

Please see **CM24.2\_Spec.pdf** for OEM IDOL operating system requirements.

### System requirements

The minimum system requirements for IDOL are:

- A dedicated SCSI disk
- 4 GB RAM
- 100 GB disk space
- A minimum of two dedicated CPU - Intel Xeon or AMD Opteron or above

### Other considerations

The OEM IDOL installation consists of a main IDOL service and an IDOL content service (Records Manager 8.1 and earlier included two content services). While this package is generally suitable for small to medium sites or for test environments, there are some adjustments that can be made.

For more information on how to implement each of these options, see [Document content searching](#).

### Separate disks

If possible, each content service should be configured to use its own physical disk for its working directories. For further information, see [Paths](#).

### Separate computers

In an ideal situation, each of the IDOL components would be installed on their own dedicated computer. This enables each component to use the maximum amount of resources.

### **Separate computers for content engines**

Similar to the above configuration, this option involves installing each content engine on a dedicated computer. The main IDOL service can be installed on a Workgroup Server or another non-dedicated computer.

This option maximizes the resources available to the content services, which are the most resource-intensive components.

### **Additional content engines**

For larger sites or sites that require higher performance or greater redundancy, additional content engines can be installed and configured.



## Installing Additional IDOL Components

**NOTE:** From 10.0 onwards, the Content Manager Connector, Connector Framework and OMNIGroup server are not shipped with Content Manager. Use the **CM\_IDOLComponents\_x64.msi** to install the IDOL Service and additional content services.

### Installing CM IDOL service and Content services

The Content Manager IDOL Service and Content Manager IDOL content services can be installed using the **CM\_IDOLComponents\_x64.msi**. It installs a single, additional content service that is identical to the default content service that the standard OEM IDOL installation sets up with Content Manager.

For additional information on installing additional Content Manager IDOL content services please see Tech Note: [Adding additional IDOL content engines to Content Manager](#).

### Prerequisites

These instructions assume you have already successfully installed and configured:

- Content Manager Workgroup Server (64-bit) 10.x
- The OEM installation of IDOL that was installed with Content Manager; or the IDOL and Content Services have been installed using **CM\_IDOLComponents\_x64.msi**

### Installation steps

1. On your installation medium, locate the installation file and run it as administrator:

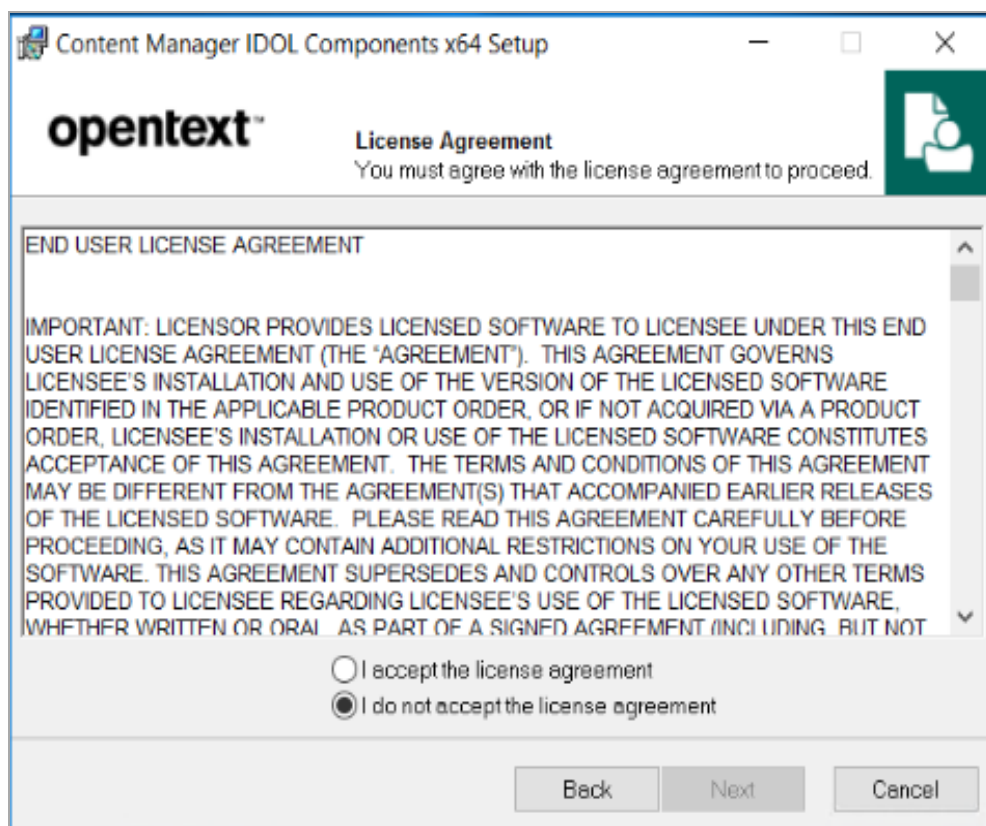
**CM\_IDOLComponents\_x64.msi**

The **Welcome to the Content Manager IDOL Components x64 Installation Wizard** dialog appears:



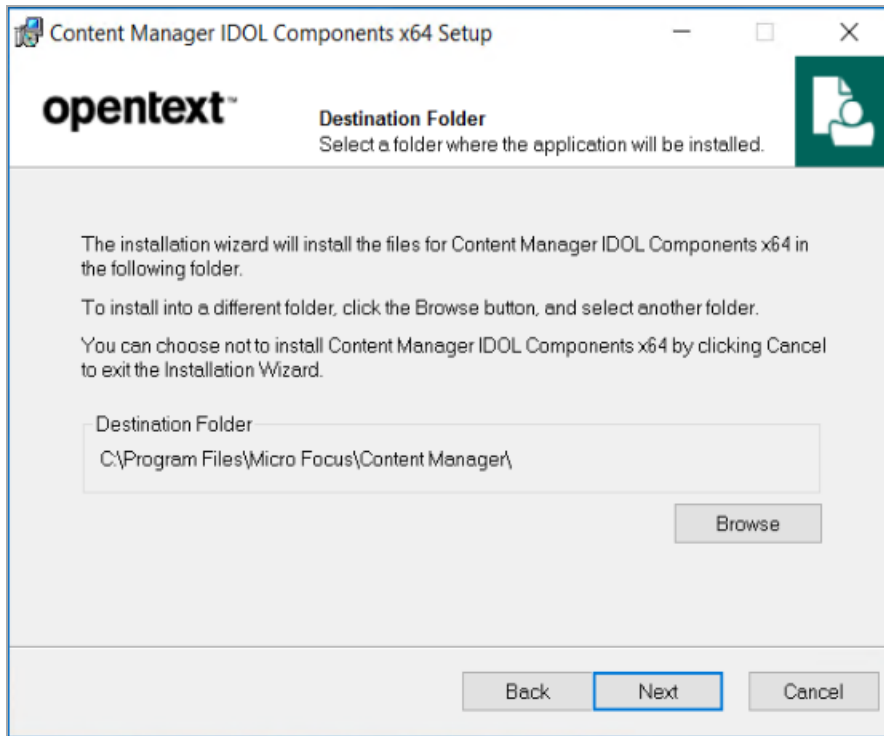
2. Click **Next**.

The **License Agreement** dialog appears:



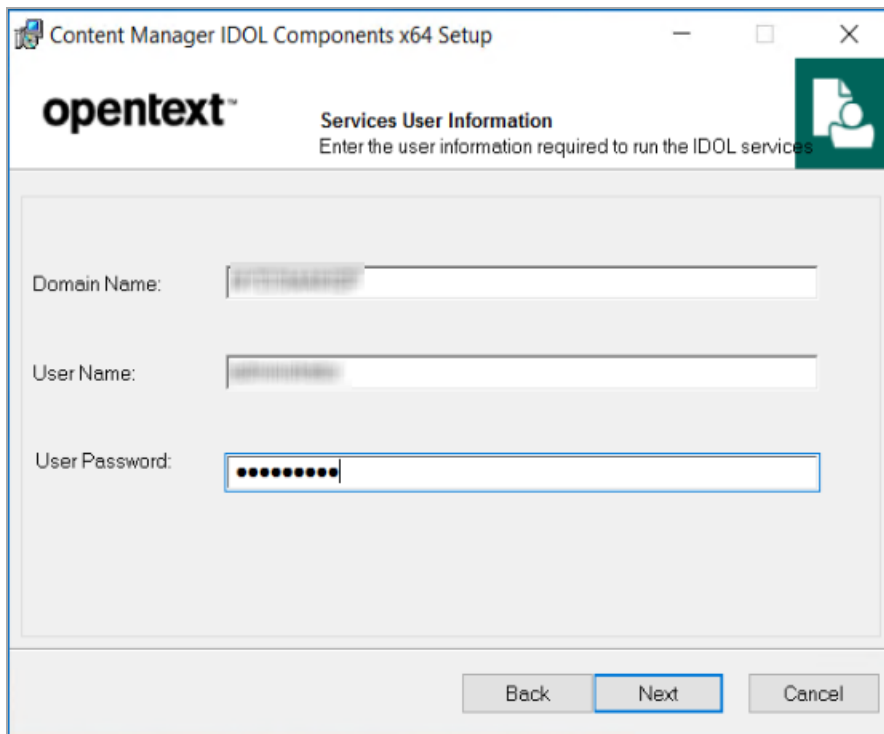
3. Select **I accept the license agreement** and click **Next**.

The **Destination Folder** dialog appears:



4. Click **Next**.

The **Services User Information** dialog appears:

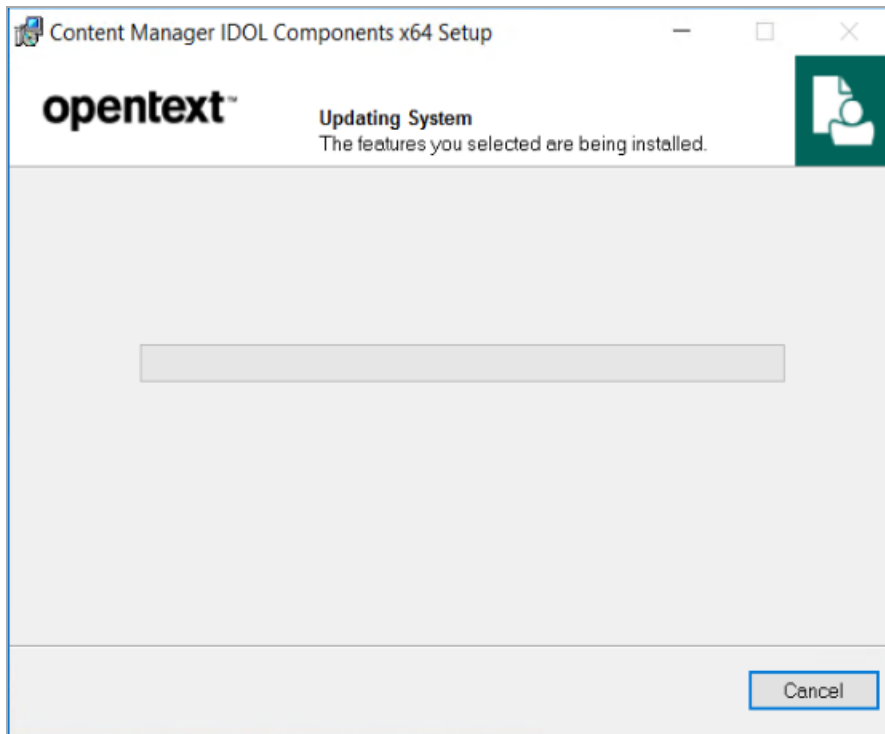


5. Enter the information and click **Next**.

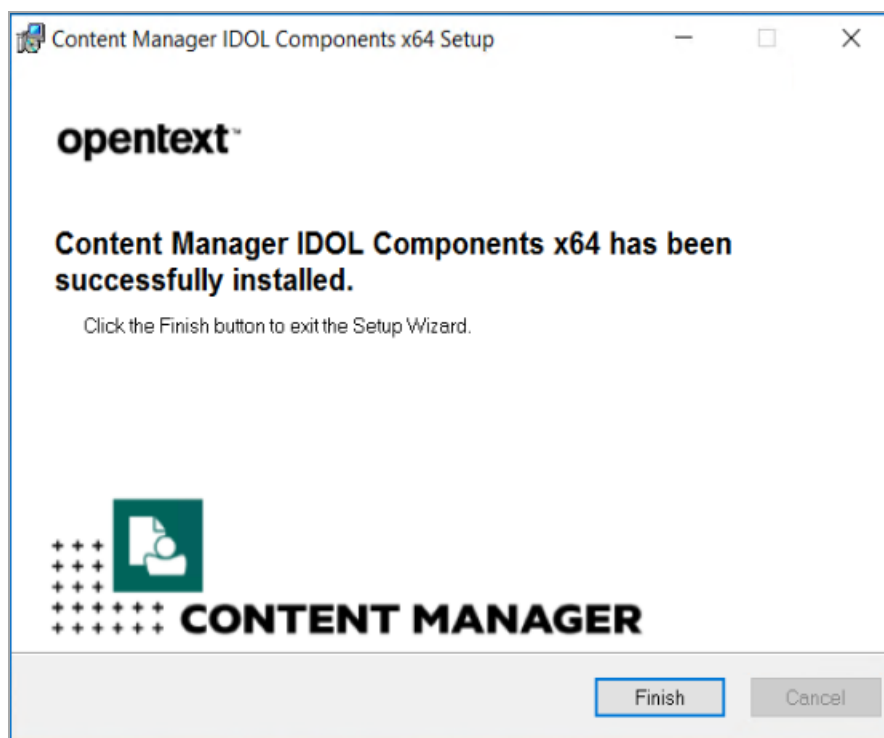
The **Ready to Install the Application** dialog appears:

6. Click **Next**.

The **Updating System** dialog appears:



The **IDOL Components installation confirmation** dialog appears:



7. Click **Finish**.

The installation of the **Content Manager IDOL Service and Content Service** is complete.

In the list of services on the computer, the services appear as **Content Manager IDOL Service** and **Content Manager Content Service**.

### Command line installation

Command line installations can be performed using `msiexec.exe`. To specify the credentials of the installed service, use the **DOMAINNAME**, **SERVICEUSER** and **SERVICEPASS** MSI properties as follows:

```
msiexec /i CM_IDOLComponents_x64.msi DOMAINNAME=<domain> SERVICEUSER=<username>  
SERVICEPASS=<password>
```

### Content Manager Media Server

Content Manager uses Media Server to perform optical character recognition (OCR) on images. This enables users to search the content of image files like any other document.

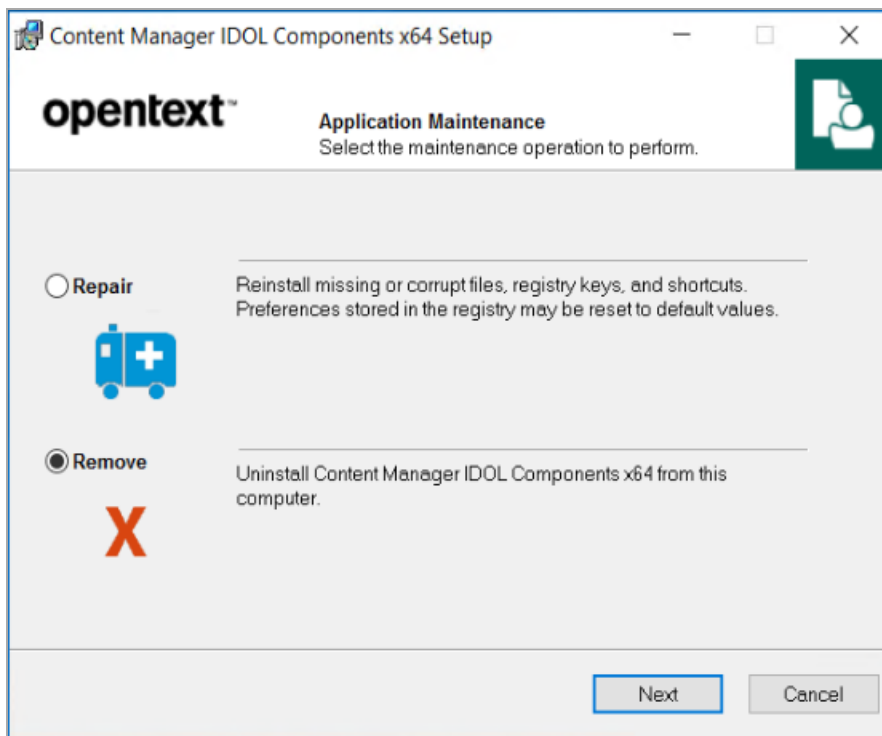
**NOTE:** Now, you cannot install Media Server from IDOL component using the **CM\_IDOLComponents\_x64.msi**. Also, upgrading the IDOL components removes the existing Media Server. You will have to install Media Server separately by using **CM\_MediaServer\_x64.msi**. For information on installing and setting up Media Server, see **CM24.2\_Install.pdf**.

## Repairing or removing Content Manager IDOL components

1. On your installation medium, locate the installation file and run it as administrator:

**CM\_IDOLComponents\_x64.msi**

The **Application Maintenance** dialog appears:



2. Select from the options:
  - Repair** – to repair your IDOL components installation.
  - Remove** – to remove your IDOL components installation.
3. Click **Next**.

The installer carries out the task(s) you selected.

## Upgrading IDOL components

This section describes the sequence of steps you need to follow to upgrade the IDOL components.

**IMPORTANT:** Make sure to upgrade the Content Manager before upgrading the IDOL components.

1. Stop all the services.
2. Upgrade from Content Manager 9.x version to Content Manager 24.2.

3. Start all the services.
4. Check that the IDOL is still working.
5. Stop the services.
6. Upgrade IDOL components to latest.
7. Start the services.
8. Verify that the IDOL indexes are working and return search results.

## Upgrading OEM IDOL

With the evolving Content Manager system, the IDOL default configuration files may change in terms of their parameters from time to time.

As you may well have customized your OEM IDOL configuration files in your previous version of Content Manager, the Content Manager upgrade process will not overwrite your existing configuration files, but rather copy them, insert the installation date and time into their file names, and keep them for your reference.

For example, if one of your own configuration files was called **TRIM Content Service 1.cfg**, after the upgrade, you will find it in your new installation directory folder **IDOL** as **TRIM Content Service 1\_05\_22\_2013\_10\_29.cfg**, where date and time are in the format **\_DD\_MM\_YYYY\_HH\_MM**.

After upgrading, you should therefore update your new configuration files manually with the properties from your old configuration files that you want to keep.

### Upgrading OEM IDOL using Setup\_CM\_x64.exe

If you have not customized any of your OEM IDOL configuration files, Content Manager will upgrade OEM IDOL to the latest version when you use **Setup\_CM\_x64.exe** for the upgrade of the Content Manager installation.

### Upgrading OEM IDOL manually

If your organization's OEM IDOL configuration is changed from the out-of-the-box setup, you will have to upgrade all your Content Manager OEM IDOL components manually, which means replacing the existing files with the ones that come with Content Manager 9.x as follows:

Stop all IDOL services on the computer hosting OEM IDOL

**IMPORTANT:** See <https://softwaresupport.softwaregrp.com/group/softwaresupport/search-result/-/facetsearch/document/KM00209246> for the correct way to shutdown the IDOL services.

For the Content Manager Content Service components, navigate to the installed directory, by default, **C:\Program Files\Micro Focus\Content Manager\IDOL\**



1. In each **TRIM Content Service** and additional **Content Service** folder:
  - a. Add or replace the following files:
    - autnxsIt.dll
    - licensekey.dat
    - TRIM Content Service.exe
  - b. Delete the folders:
    - license
    - uid
  - c. Insert the following in the existing configuration files **TRIM Content Service.cfg**:

Section [Logging]:

    - Below the line 2=IndexLogStream, insert:  
**3=ActionLogStream**  
**[ActionLogStream]**  
**LogFile=action.log**
    - Below the section [IndexLogStream], insert:  
**LogTypeCSVs=action**
2. For the Content Manager IDOL Service component, navigate to the installed directory, by default, **C:\Program Files\Micro Focus\Content Manager\IDOL\**
3. Add or replace the following files manually:
  - autnxsIt.dll
  - licensekey.dat
  - TRIM IDOL Service.exe
  - a. In the folder **dah**, add or replace the following files manually:
    - autnxsIt.dll
    - dah.exe
    - federated.dll
    - licensekey.dat
    - sca.dat
    - yaz.dll
  - b. In the folder **dih**, add or replace the following files manually:
    - autnxsIt.dll
    - dih.exe

- licensekey.dat
- c. In the folders **TRIM IDOL Service** and its subfolders **dah** and **dih**, delete the following folders:
- license
  - uid
- d. Edit the following in the existing configuration file **TRIM IDOL Service.cfg**:
- Section [Server]:
- Insert **Timeout=60**
  - Remove **MaxResults** and **QueryTimeoutInMilliseconds**
- Section [DistributionSettings]:
- Insert:  
**MaxResults=100000**  
**EngineTimeout=60**  
**StateTokensDirectoryUsesHash=true**  
**StoredStateTokenExpiryPeriod=300**  
**StoredStateTokenLifetime=0**
- Section [Logging]:
- Insert below the line 2=IndexLogStream:  
**3=ActionLogStream**  
**[ActionLogStream]**  
**LogFile=action.log**
  - Insert below the section [IndexLogStream]:  
**LogTypeCSVs=action**
4. Restart the IDOL services.
- Each of the content services will spend some time converting the existing indexes to the new version of OEM IDOL, which may make IDOL unavailable for some time. The duration depends on the size of the existing indexes.

## Upgrading your document content indexes

TRIM before version 7.2 used ISYS for document content indexing (DCI). From version 7.2, TRIM, Records Manager and Content Manager use IDOL for DCI, which is not compatible with the legacy ISYS engine.

If you are upgrading Content Manager from TRIM before 7.2, use Enterprise Studio to remove the ISYS index and then create the new IDOL index. Depending on your version of TRIM, you may not have to remove the ISYS indexes. For data security, it is recommended you delete the legacy ISYS index files.

If you are upgrading Content Manager from TRIM 7.2x, then after the Content Manager upgrade, edit the existing IDOL configuration files to include the additional properties from the new configuration files. To do that, stop all IDOL services on the indexing computer, make the changes and re-start the IDOL services. See also [IDOL and ISYS](#).

## Document Content Searching

Document content index searching is an invaluable tool in many circumstances; however, there are costs that must be recognized.

Each content engine is an OEM licensed version of IDOL. This enables Content Manager to index up to 10 million documents per content engine, although the recommended size is only 5 million documents. You can create additional content engines for:

- Handling greater document volumes
- Increased performance
- Failover and mirroring options

## Requirements

### Document volume

Ensure that there is enough room for the index.

The physical space required for the document content index depends on the number of content indexed electronic documents in the dataset and their file formats. To most accurately predict the document content index size, we suggest testing it with a large number of example documents in your own organization, enough to cause the initial index size to grow.

### Site size

When considering how big a site is in terms of the IDOL content index, the “number of records” refers to both the number of Content Manager records in a dataset, and the number of electronic documents in the dataset. Each Content Manager record will translate to 1 or more records in IDOL.

Each record in Content Manager will have a corresponding record in IDOL. If the Content Manager record has an electronic document associated with it, there may be multiple IDOL records created for the Content Manager record. Examples of documents that result in multiple IDOL records are email and archive files (e.g. .zip files).

### Small sites

Sites with fewer than five million IDOL records are considered small sites from a Content Manager perspective. A single content engine could be installed on a separate server or a dedicated Workgroup Server.

### Medium sites

Sites with a document volume between five and 20 million are considered medium from a Content Manager perspective. Medium sites will need at least one additional content service installed. It is worth taking into consideration the index growth and making allowances in the initial configuration of the IDOL content engines.

For medium sites, you should also consider the physical location of the content engines. It is recommended to have these engines on separate disks to optimize performance. Contact Content Manager Product Support for specific hardware requirements for your environment.

One or more additional content engines should be created before the initial indexing process to spread the load and allow for index growth.

If mirroring or failover is required, it may be necessary to add additional content engines.

### **Large sites**

For larger/enterprise class sites, it is recommended that you consult your Content Manager support team, business partner or consulting team for specifics on scaling your IDOL solution.

## **Performance considerations**

Disk I/O plays a significant part in a successful Content Manager IDOL implementation, as well as the free space on each disk. As virtualization technologies such as VMware have a great impact on disk I/O, physical servers are strongly recommended.

Substandard write speeds may impact the initial index creation and synchronization of new content. Substandard read speeds may impact search performance.

## **Failover and mirroring**

The Content Manager dataset, electronic store and document content index require regular backups to retain internal referential integrity.

For full redundancy and failover of the system, all of the above systems may be replicated on identical hardware. For IDOL Server, the recommended number of servers should be mirrored and the DIH should be used for index mirroring. Note that the mirrored systems can also be used to increase performance. DIH mirroring is configured through the Content Manager IDOL Service configuration file.

See [Backing up the content index](#) for information about backing up your content indexes.

## **Configuring document content indexing**

Content Manager uses the Intelligent Data Operating Layer (IDOL) for document content indexing (DCI). You can use the OEM version of IDOL, which comes with Content Manager, or your own implementation of Enterprise IDOL, for which you need to install and configure the Content Manager connector as described in [Installing Content Manager Connector](#).

You have to set up DCI if you have not used it before, and also if you have upgraded from TRIM version 7.1 or earlier, which did not use IDOL content indexing. See also [Transition from ISYS](#).

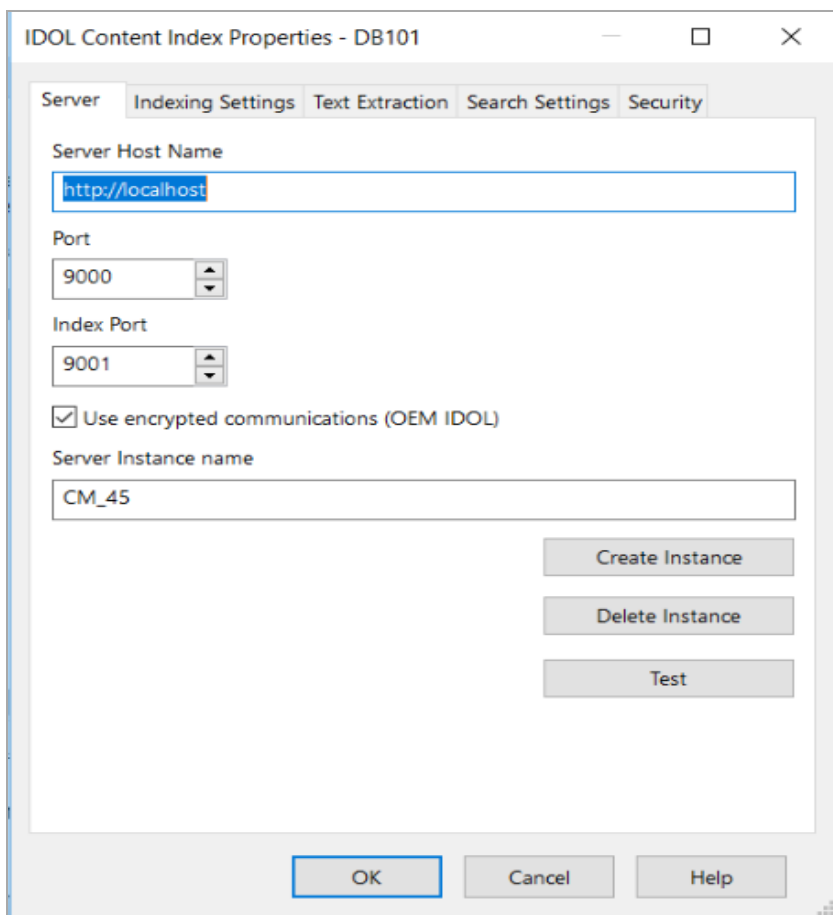
You can also use these steps to configure an existing Workgroup Server to use an existing IDOL document content index by using the properties of the existing index in the Create Content Index dialog.

These instructions assume that you have already installed, set up and configured your IDOL server or servers.

1. Using Content Manager Enterprise Studio, expand **Datasets**.
2. Right-click the dataset, point to **Content Index** and then click **Create IDOL Content Index**.

If IDOL content indexes have already been created, new indexes can be added by clicking **Properties**.

The **Create context index - <Dataset name> - Server** tab appears:



- **Server Host Name** – type the fully qualified name of the computer that runs the Content Manager IDOL Service - the main IDOL Server
- **Port** – type the IDOL Server port number, used for searching. By default it is 9000.
- **Index Port** – type the Index Port number, used for sending documents to the server. By default it is 9001.

The port numbers should remain at their defaults unless you specifically changed the IDOL configuration file in the folder C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOL Service or your equivalent.

- **Use encrypted communications (OEM IDOL)** - select this option for encrypted communication.

- **Server Instance name** – displays the name of the instance on the IDOL server that this dataset communicates with. Defaults to CM\_[database ID]. Should not be changed, unless you re-index the dataset afterwards, as Content Manager Enterprise Studio would create a new instance on the IDOL server when the instance name changes.
  - **Create Instance** – creates the content index database in IDOL.
  - **Delete Instance** – deletes the content index database in IDOL. This will destroy the index, requiring a full content reindex to recreate.
3. Click **Test** to test the setup. If it fails, it means that Content Manager cannot communicate with the IDOL Server. Check that the Content Manager Enterprise Studio IDOL configuration data above is correct, for example the server name. Next, check that the IDOL Server is running correctly, for example as described in [Monitoring an IDOL Server](#) and the log files as described in [IDOL logs](#):

For the IDOL OEM version, check your setup against the installation and configuration instructions; see [Installing Additional IDOL Components](#).

If you are using IDOL Enterprise Server, check your IDOL setup against the IDOL documentation.

4. Click **Indexing Settings** tab.
- **Index Directly into IDOL via CM Event Processor** - this option allows to index directly in IDOL through the Content Manager event processor.
  - **A CFS Connector is indexing this dataset (for an external database, or for searching within Content Manager)** - you are using your own Enterprise version of IDOL, and have also installed and configured the CFS connector. Content Manager will send messages to the CFS connector to ensure that the IDOL index is up to date with changes to the Content Manager database.
  - **Maximum queued transactions** – select the maximum number of records in the queue. By default this is 20. The smaller this number, the shorter the queue, which makes processing more reliable, but takes longer to be processed by IDOL.
  - **Maximum size of any file that can be content indexed (MB)** - set the maximum size of a file that can be content indexed. By default this is set to 200.
  - **Size limit for archive files (zip, gz, tar) that can be content indexed (MB)** - set the maximum size of archive file that can be content indexed. By default this is set to 50.
  - **Provide extensive logging for index management** – select this option to create detailed log files.
5. Click **Text Extraction** tab.
- **Index text that was deleted from a document with revision tracking** - select this option to index text that was deleted from document.
  - **Index hidden text from Microsoft Office files** - select this option to index hidden text.
  - **Index the tag names within XML files** - select this option to index XML Format text files.

- **Set Microsoft Rights Management Services (RMS) credentials** - enter the credentials for the Microsoft Rights Management Services (RMS) to allow Microsoft Information Protection (MIP) encrypted documents to be indexed.
  - **Tenant ID** - enter the RMS Tenant ID.
    - **Set Client Secret** - if required, click the Set Client Secret option to set the password.
  - **Client ID** - enter the RMS Client ID.

6. Click **Search Settings** tab.

- **IDOL Connection Timeout** – select the number of milliseconds the Workgroup Server will wait for IDOL before timing out. By default it is 60000 milliseconds (1 minute).

If you are finding that searches are timing out, you may want to try to change this value. It should have a higher value than the value of Timeout and EngineTimeout in the IDOL configuration file C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOL Service\TRIM IDOLService.cfg.

To diagnose the issue:

When IDOL times out before Content Manager, then it returns a subset of the result, which appears in a Content Manager search list as usual. When Content Manager times out before IDOL, then the Workgroup Server displays an error message.

- **Maximum number of results in any search** - default value of 100,000 - the maximum number of results Content Manager will get back from IDOL. If the IDOL part of a search finds more than this number, a warning displays and the user only gets this number. The maximum value this can be set to is 10,000,000.
- **Search result buffer size** - select the buffer size for search result.
- **Find records from other datasets** – this option will allow IDOL to find records that were indexed from other Content Manager datasets. For example, using a copy of the content index from the production database with the training database.

7. Click **Security** tab.

The options on this tab allow for a connection to an Enterprise IDOL instance via SSL/TLS.

- **Enable SSL/TLS connection to IDOL** - select this option to enable TLS between the client and the IDOL server. The version of TLS will depend on what is configured in IDOL and what is supported by the client operating system. Any errors due to invalid certificates will be ignored by the client.

Communication from the client to an IDOL service is done using both a HTTP/REST interface, as well as using IDOL's ACI API. The REST component requires a certificate from the local certificate store and the ACI API component requires separate certificate/key file in PEM format. For more secure environments where certificate checking is enabled, or required, the following fields are required:

- **Client Certificate** - from the drop-down list select the SSL certificate used to establish a secure connection with the IDOL Proxy component using HTTP/REST. This is a certificate that has been added to the local certificate store and must include the private key.



- **View Certificate** - click **View Certificate** to view the associated certificate.
  - **ACI Client API Certificate Path** - click Browse to navigate to and select the SSL Certificate file used to identify the client to the IDOL Proxy component via the ACI API. The required file format is PEM.
  - **ACI Client API Private Key Path** - click Browse to navigate to and select the Private Security Key file corresponding to the selected SSL Certificate (Client Certificate). The required file format is PEM.
  - **ACI Client API CA Certificate Path** - click Browse to navigate to and select the Certificate Authority (CA) certificate file of a trusted authority. The required file format is PEM. The IDOL Proxy component trusts communication only with a peer that provides a certificate signed by the specified CA.
8. Click **OK**.

The setup is complete.

A status message is displayed. Click **OK** in the dialog.

When a content index is set up, the dataset right-click **Content Index** menu commands **Check**, **Optimize**, **Monitor**, **Remove** and **Properties** are available.

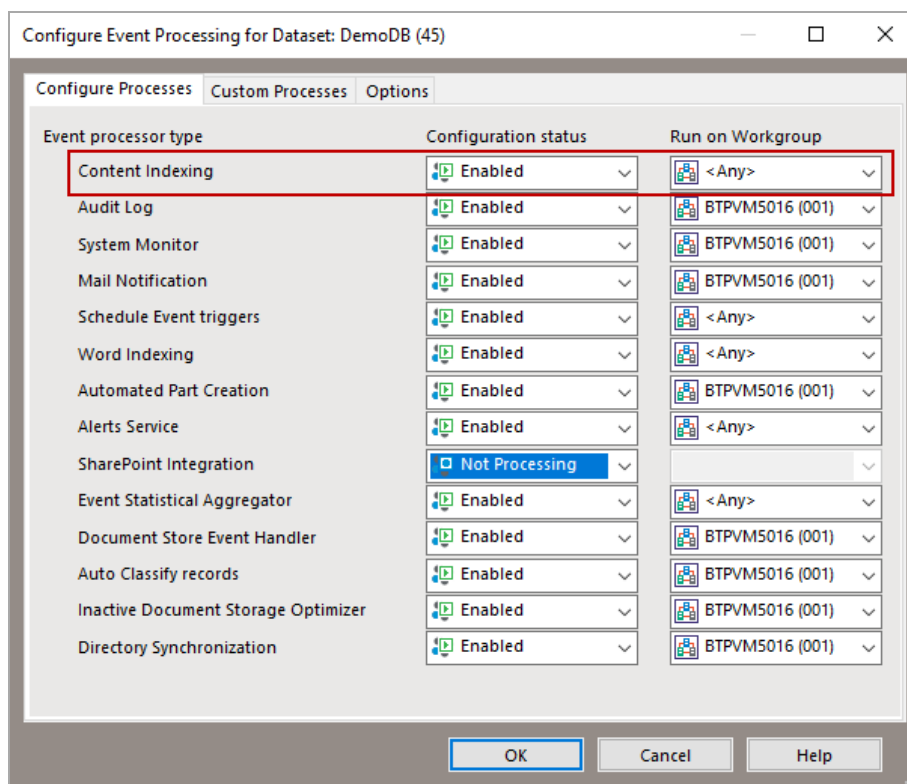
## Document content index menu in Content Manager Enterprise Studio

1. Using Content Manager Enterprise Studio, expand **Datasets**.
2. Right-click the dataset, point to **Content Index** and then choose from:
  - **Create** - see [Configuring document content indexing](#)
  - **Check** - see [Checking the IDOL index for errors](#)
  - **Optimize** - see [Optimizing the IDOL index](#)
  - **Monitor** - displays the Monitor IDOL Server dialog box with the last 25 events that the IDOL Server received
  - **Remove** - see [Removing a document content index](#)
  - **Properties** - to modify the dataset's content index details, which are described in [Configuring document content indexing](#)
3. Click **OK** to save the changes you have made or **Cancel** to close the dialog without saving your changes.

## Configuring content index event processing

Content index event processing is configured using Content Manager Enterprise Studio.

1. Using Content Manager Enterprise Studio, expand **Datasets**.
2. Right-click the dataset, point to **Event Processing** and then click **Configure**.  
The **Configure Event Processing for Dataset: <Dataset Name>** appears:



3. For the event processor type **Content Indexing**, from the Configuration status drop-down list, select **Enabled**.
4. From the **Run on Workgroup** drop-down list, select a particular Workgroup Server name to run the processor on, or select **Any** to run it on any available Workgroup Server.
5. Click **OK**.

Content Manager indexes the content of electronic documents in the dataset.

**NOTE:** To pause the content index processing, set **Configuration status** to **Suspended**. To re-enable the processor type, change **Configuration status** back to **Enabled**.

To set a limit on the size of a document that Content Manager should index the content of, on the **IDOL Content Index Properties - Indexing Settings** tab, set the size limit using the **Maximum size of any file that can be content indexed (MB)** option. See *Content Manager Enterprise Studio Help - Dataset setup and maintenance - Indexing in Content Manager - Document content index - Content Index Properties dialog box* for details.

You may want to exclude particular types of files from your content index.

See [Suppressing document content indexing](#).

If you have created a new IDOL index for an existing dataset, start content indexing the existing dataset by using the Content Manager client command Tools - Content Manager Administration - IDOL Record Reindex. See [Performing content indexing](#). Content Manager indexes new documents after they are checked in.

## The re-indexing process

### The Content Manager process

- The Workgroup Server runs the select query based on the search to identify the records to index, taking into account the documents that are excluded due to their file type, Record Type, and size. When complete, the reindex window appears in the Content Manager client, and you can start the re-indexing process.
- The re-indexing process selects the documents in order of URI and passes them to the KeyView text converter. The number of text converters (KVOOP.exe) depends on the number of cores available on the client computer that is executing the indexing process.
- Each core creates a separate TRIMtrans\_x.txt file, which adds text converted documents until the number of processed records has reached the number specified in the field **Maximum queued transactions** in the **IDOL Record Reindex** dialog box, **Protocol** tab.

These transaction files are stored in the directory

**C:\Users\**

Note that crashes place crash dump files in the same directory. ZIP files for text extraction will be opened and extracted to the same location.

- When reaching the limit, the transaction text file is passed to the IDOL Server as a .dat file along with a .cmd file. The .cmd file is a DREADD command for the IDOL Server.
- The IDOL Server then passes the .dat and .cmd files to every content service engine for processing.

At this point, Content Manager has completed its processing and now, IDOL processes the data into the IDOL databases.

### The IDOL process

Each content engine receives the text extraction and command file. Taking into account the options set in IDOL, they then process the command file and update the content engine database.

The IDOL databases are distributed so that each of them processes separate parts of the text file as determined by the IDOL Server. They hold the information in memory until the synchronization limit is reached. At that point, they commit the changes to the IDOL database.

The defaults for IDOL are:

- DelayedSync=true
- MaxSyncDelay=600
- IndexCacheMaxSize=102400

This means that the IDOL database waits 600 seconds before committing all changes to the IDOL content database, or until there is 102400 kbs or more data in the index cache. When multiple threads are processing text extractions, this commit submits all changes to the database. Increasing the parameter reduces the number of commits during an indexing process. Memory use increases, but updates happen faster.

- If the IDOL content service is busy, the command and text extraction file are copied to the Status directory and queued for processing. When disk write speeds are slow, the likelihood of queuing is much greater. The content service processes from the queue as soon as it is available again.

Before processing a queued command, the IDOL service checks whether the batch has already been processed by another content service. If so, it discards the batch and picks up the next queued batch file for processing.

- You can see main updates in an IDOL database in the Dynterm and Nodetable directories.
- Content Manager is oblivious to the state of the IDOL content databases and continues to pass text extractions and .cmd files to the IDOL Server.

### **Content Manager URN**

The URN is the unique identifier of a Content Manager record in IDOL. It has a structure of:

trim:[DBID]/[OBJECT TYPE]/[OBJECT URI]

It may also contain a sub-part identifier. In which case, the structure looks like:

trim:[DBID]/[OBJECT TYPE]/[OBJECT URI]:[SUBPART]

The [DBID] is the two-character dataset ID of the Content Manager dataset the record can be found in.

The [OBJECT TYPE] is a three-letter abbreviation of the Content Manager object type. This is usually “rec” for a Content Manager Record, but could be “loc” for a Location or “pln” for a Classification.

The [OBJECT URI] is the unique identifier of the object in the original Content Manager dataset.

When a record in Content Manager contains a document that is made up of subdocuments, such as an email with attachments or a .zip archive, each document becomes a unique record in IDOL. These subdocuments are identified by a URN that matches their parent, but which also includes a [SUBPART].

### **Example URNs**

- trim:F1/rec/9000003457
- trim:45/rec/3437:3
- trim:ST/loc/234

The IDOL databases are distributed so that each of them processes separate parts of the text file as determined by the IDOL Server. They hold the information in memory until the synchronization limit is reached. At that point, they commit the changes to the IDOL database.

The defaults for IDOL are:

- DelayedSync=true
- MaxSyncDelay=600
- IndexCacheMaxSize=102400

## Performing content indexing

This function enables you to rebuild the document content indexes, IDOL metadata indexes and Location indexes with the user security settings.

It is not for re-indexing the words in the text fields. For this function, see Index Text in Content Manager client Help.

Highly recommended:

- Use a temporary index for re-indexing so that users can continue to work. To do this, create the new index in a new location. When the re-index is complete, copy it to the location of the original index.
- Run the re-indexing process on the Workgroup Server for efficiency and to pick up the defaults
- If users still need to access Content Manager, run the re-indexing process on a secondary Workgroup Server (otherwise the server may not have enough resources to handle user requests)

## Re-indexing records

You can use this function to select existing Content Manager records with electronic documents to index the document content for the first time or to re-index if they have not been indexed correctly, for example because of a network failure.

You should also use it when an entire Content Manager document content index has been lost or damaged, for an upgrade of a TRIM 7.2 IDOL index and when the indexing parameters have been changed, for example which Additional Fields to index.

To improve the very first document content indexing process, see **Content Manager Enterprise Studio Help > Dataset setup and maintenance > Indexing in Content Manager > Reindexing the dataset using IDOL > Options to improve the initial indexing process.**

1. Start the Content Manager client with the dataset to re-index.
2. On the **Administration** tab, in the **IDOL Index** group, click **Records**.

The **IDOL Record Reindex** dialog box appears.

3. **Select** tab - to select the records to reindex:
  - **Enter search string to select items to be reindexed** - type or use the KwikSelect button to form a search string to select the records to index
  - **Maximum size of any file that can be content indexed (MB)** - default: 200. Content Manager does not index the content of documents whose file size is greater than this number in megabytes. This data should be the same as the **IDOL Content Index Properties - Indexing Settings** dialog in Content Manager Enterprise Studio. Content Manager fills in the data automatically when the client runs on the same computer as the Workgroup Server.
  - **Maximum size of any archive file (zip, gz, tar) that can be content indexed (MB)** - default: 50. Content Manager does not index the content of archive files whose file size is

greater than this number in megabytes. Content Manager populates this field automatically, when possible. See previous option for details.

4. **Protocol** tab - to specify the indexing computer:

- **Create notifications for the CFS Connector** - select to use the CFS connector to pass data to Enterprise IDOL.

See [Installing Content Manager Connector](#) for installation and configuration of the Content Manager connector.

- **Call directly to IDOL Service** - select to use the Content Manager event processor for indexing. Content Manager populates the server details, if available.
- **IDOL Server Name** - the computer that runs the Content Manager IDOL Service, the main IDOL Server
- **IDOL Server Port** - default: 9000. IDOL Server port number, used for searching. See next option for more.
- **IDOL Server Port (Index)** - default: 9001. Used for sending documents to the server.

The port numbers should remain at their defaults unless you specifically changed the IDOL configuration file in the folder C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOLService or your equivalent.

- **Instance name for Content Manager data** - name of the instance on the IDOL server that this dataset communicates with. Defaults to CM\_[database ID].

When you create a content index for a dataset, Content Manager creates an instance on the IDOL server, which is what Content Manager uses for all IDOL searches.

**CAUTION:** While you can change the value in this field and re-index to a different instance, it is not recommended. When you re-index into a different instance, then Content Manager will not be able to find anything in that index, and creates unusable data in the other index.

The only scenario when an experienced Content Manager and IDOL administrator may want to re-index to a different instance, is to create your IDOL index offline, for example, for a test dataset or to create a new index while still using an old content index.

- **IDOL Server is the OEM service shipped with Content Manager** - select when you are using Content Manager's OEM IDOL version for indexing
- **Ignore read-only warnings for IDOL instance** – If Content Manager detects that the selected IDOL instance has been marked as read-only, it will prevent the reindex from being run. Check this checkbox if you want the reindex to proceed regardless of the detected read-only state.
- **Reindex As Dataset** – reindexes records from this dataset as if the dataset ID was the value specified. This is useful when needing to create an index for a copy of the current dataset that is not available yet (eg using the test dataset to create the production index before production is made live).

- **Maximum queued transactions** - default: 20, which is the number of records in the queue. The smaller this number, the shorter the queue, which makes processing more reliable, but takes longer to be processed by IDOL.

5. **Options** tab - to specify the indexing parameters.

- **Number of processing threads** - default: 4. You can use a number from 1 to 20. The greater the number of threads, the faster the indexing process will run. It depends on your hardware capacity how many threads you can run.
- **Log output to** - the log file location
- **Produce detailed log messages** - select to see more detail in the log files
- **Continue after an individual item error** - should always be selected. Errors may occur when for example an electronic document cannot be converted for indexing.

When the option is not selected, the indexing process stops when it encounters an error.

- **Perform metadata updates only** - select to only update the IDOL record metadata. Useful when you have changed the metadata to index, for example by adding an Additional Field to index, but when the electronic documents have not changed.
- **Only reindex missing records** - select to re-index only records that have not previously been indexed. Useful to index the records that were not indexed, for example because the indexing run did not complete for some reason.
  - **Only report missing records** - available when the option Only reindex missing records is selected - select to merely find the records that have not been indexed before and then report them in the log files, but not re-index them.

6. Click **OK**.

Content Manager starts re-indexing and the **IDOL Re-indexing Progress** dialog appears with the status of the items to index, and the status of IDOL's processing of the index jobs:

- **View Log** - click to see the log file
- **Pause** - click to pause the indexing process for all threads. Useful for example to add disk space for the index. When paused, click Resume to continue indexing.

If the index jobs are not being cleared by IDOL, the process will automatically pause, allowing an administrator to ensure IDOL is running properly. When indexing has completed, the Pause button becomes unavailable.

7. Click **Close** to close the dialog box.

## Content indexing with multiple Content Manager clients

Content indexing a large volume of records in Content Manager or TRIM can take a considerable length of time.

Due to the distributed nature of both Content Manager and the IDOL engine, it is fairly straightforward to increase or decrease the content indexing performance of Content Manager. This enables a site to ramp up its capabilities to deal with peak periods such as the initial indexing of a dataset or a bulk import of new documents, and then scale back for normal day-to-day loads.

The Content Manager client can perform a bulk index operation, which bypasses the normal method of indexing content by using the event processor. When using this method, the majority of the indexing work – extracting the metadata from a record and the text from any associated document – is performed by the Content Manager client, which uses the client computer's resources.

Using this method may shift a processing bottleneck from the Content Manager client to the IDOL Server. You should pay attention to the performance of the main IDOL Server and the content engines while adding indexing clients, and during the re-indexing process.

When using this approach, the main IDOL Service and each IDOL content engine in the IDOL configuration should each run on their own dedicated computers.

The Content Manager clients performing the index should not be located on the same computer as any of the IDOL components.

The load on the Workgroup Server will also increase dramatically. As such, this operation should be done out of hours, or on a secondary Workgroup Server. Otherwise, users may have issues connecting.

The IDOL index status table will only report the status of index jobs sent from that client.

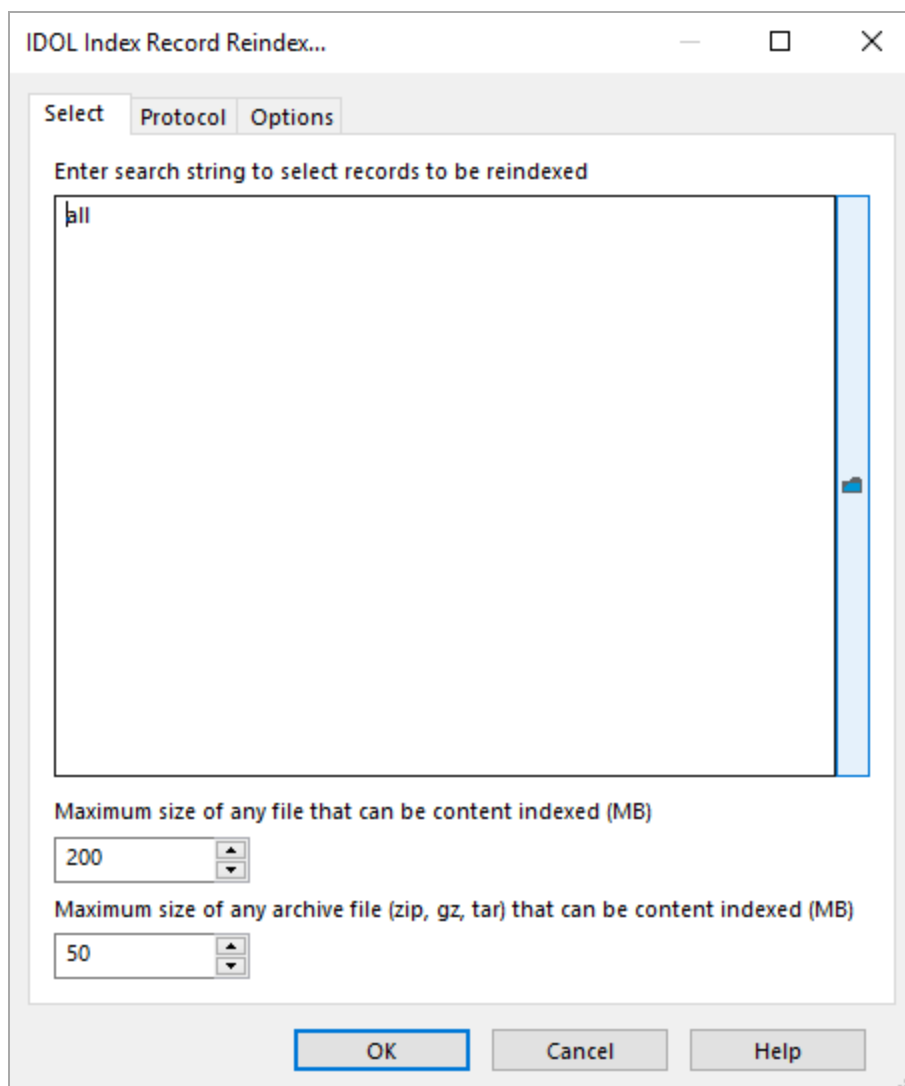
### Indexing steps

To be able to use the **IDOL Record Reindex** dialog, a Content Manager user must have the **IDOL Index Administrator** permission. Multiple Content Manager clients can run **IDOL Record Reindex** in parallel by launching additional Content Manager clients on different computers:

1. Start an Content Manager client on each available computer, on the **Administration** tab, in the **IDOL Index** group, click **Records**.

The **IDOL Index Record Reindex** dialog appears:





2. On the **Select** tab of each client, enter a search string for a portion of the target record set, for example:

- On client 1: registeredOn:1/01/2012 to 31/12/2012
- On client 2: registeredOn:1/01/2013 to 31/12/2013
- On client 3: registeredOn:1/01/2014 to 31/12/2014

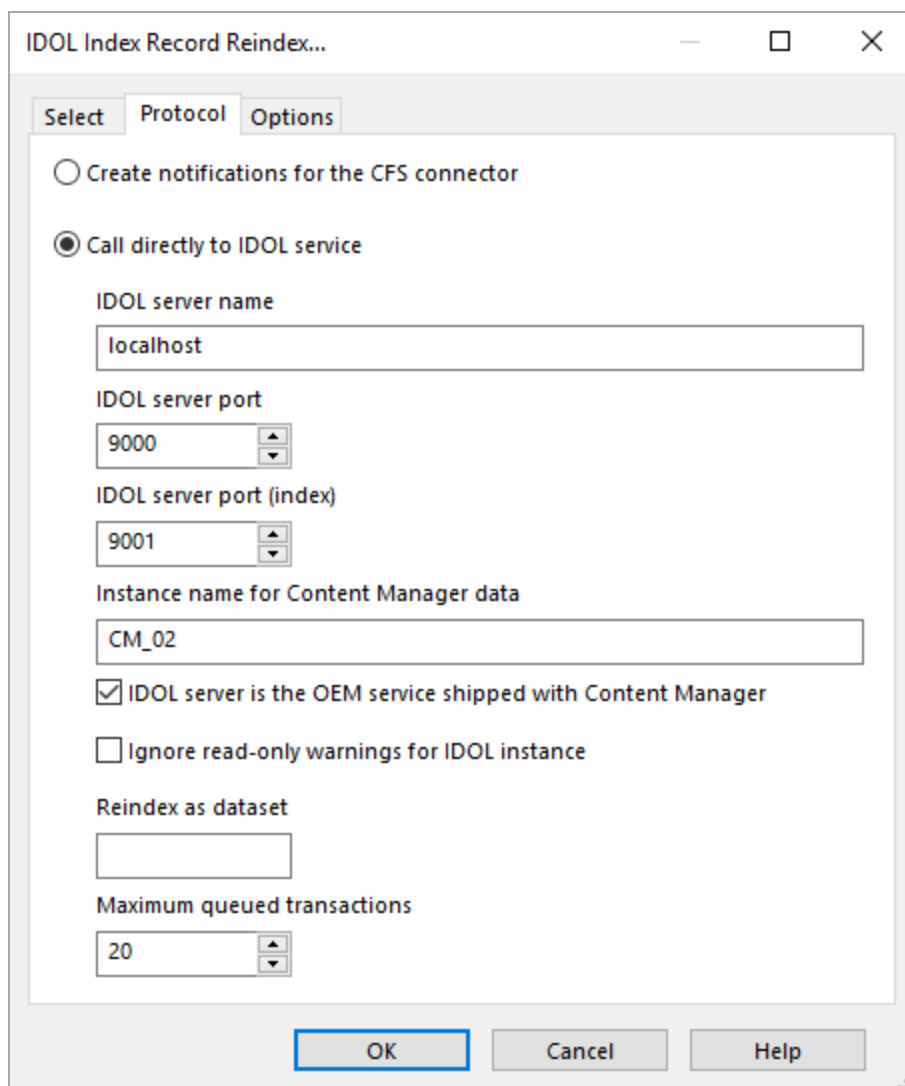
When using multiple clients, avoid having any overlap between the record searches.

3. Set the options to a value of your choosing:

- **Maximum size of any file that can be content indexed (MB)**
- **Maximum size of any archive file (zip, gz, tar) that can be content indexed (MB)**

Content Manager will not index the content of files that exceed these file sizes.

4. On the **Protocol** tab of each client:



- **Create notifications for the CFS Connector** - select to use the CFS connector to pass data to Enterprise IDOL
- **Call directly to IDOL Service** - select to use the Content Manager event processor for indexing. Content Manager populates the server details, if available.
  - **IDOL Server Name** - the computer that runs the Content Manager IDOL Service, the main IDOL Server. Do not set any client to point to localhost.
  - **IDOL Server Port** - default: 9000. IDOL Server port number, used for searching. See next option for more.
  - **IDOL Server Port (Index)** - default: 9001. Used for sending documents to the server.The port numbers should remain at their defaults unless you specifically changed the IDOL configuration file in the folder C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOL Service or your equivalent.

- **Instance name for Content Manager data** - name of the instance on the IDOL server that this dataset communicates with. Defaults to CM\_[database ID].

When you create a content index for a dataset, Content Manager creates an instance on the IDOL server, which is what Content Manager uses for all IDOL searches.

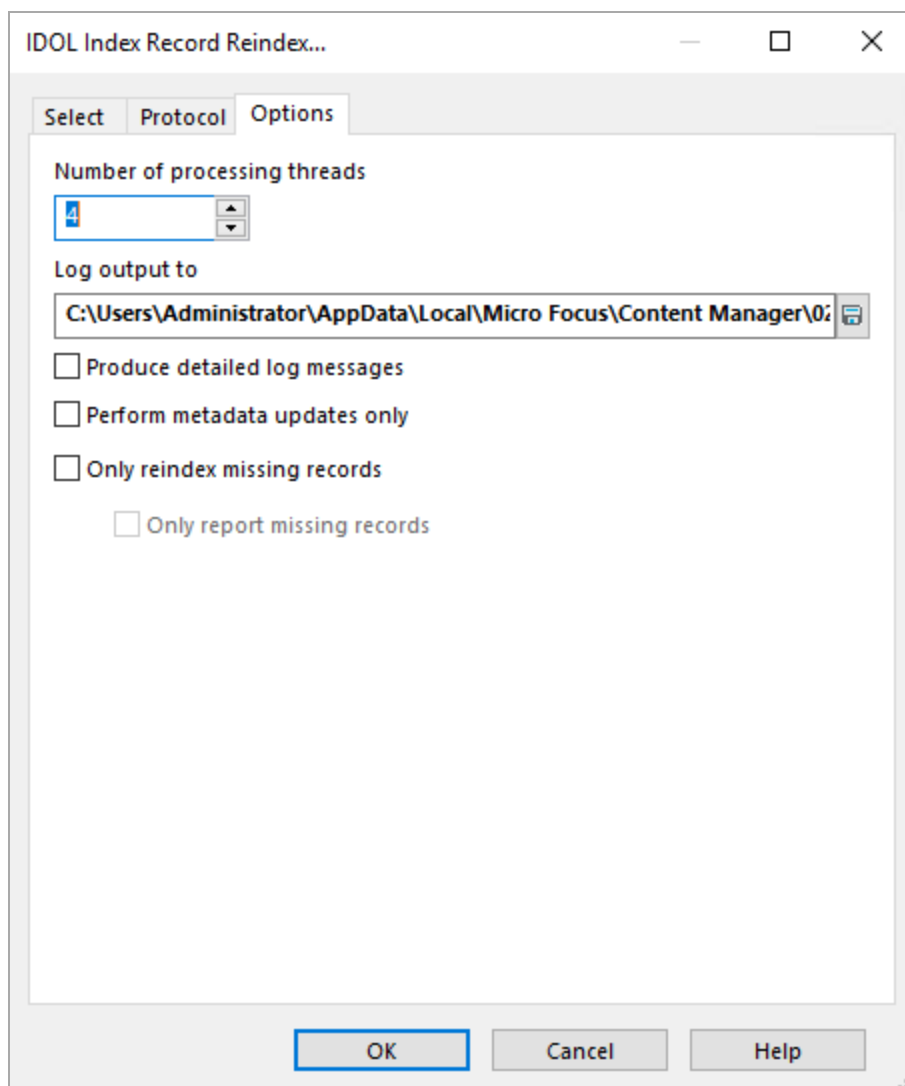
Should not be changed, unless you reindex the dataset afterwards, as Content Manager Enterprise Studio would create a new instance on the IDOL server when the instance name changes.

**CAUTION:** While you can change the value in this field and re-index to a different instance, it is not recommended. When you re-index into a different instance, then Content Manager will not be able to find anything in that index unless the “Find records from other datasets” option has been configured in Content Manager Enterprise Studio.

**CAUTION:** The only scenario when an experienced Content Manager and IDOL administrator may want to re-index to a different instance, is to create your IDOL index offline, for example, for a test dataset or to create a new index while still using an old content index.

- **IDOL Server is the OEM service shipped with Content Manager** - select when you are using Content Manager's OEM IDOL version for indexing.
- **Ignore read-only warning for IDOL instance** – only select when you wish to override the read-only protection. For example, if one content engine has been deliberately marked as read-only.
- **Reindex As Dataset** – only use when you wish to create an IDOL index for a copy of the current dataset. If the target dataset is not an exact copy of the current dataset, the content index will cause incorrect search results.
- **Maximum queued transactions** - default: 20, which is the number of records in the queue. The smaller this number, the shorter the queue, which makes processing more reliable. There should be no need to change this number other than for troubleshooting. In that case, you could even reduce it to 1 to process one record at a time.

5. On the **Options** tab:



- Set the Number of processing threads on each client to the number of available CPUs on each client. This number can be different for each client.
- **Log output to** - the log file location
- **Produce detailed log messages** - select to see more detail in the log files
- **Perform metadata updates only** - only relevant for IDOL metadata that is available to third-party applications using IDOL indexes, for example ControlPoint. Select to only update the IDOL record metadata. Useful when for example you have changed the metadata to index, for example by adding an Additional Field to index, but when the electronic documents have not changed.
- **Only reindex missing records** - select to re-index only records that have not previously been indexed. Useful to index the records that were not indexed, for example because the indexing run did not complete for some reason.

- **Only report missing records** - available when the option Only reindex missing records is selected - select to merely find the records that have not been indexed before and then report them in the log files, but not re-index them.
6. Click **OK** on each client.

Each client displays a progress dialog until processing is complete.

## Maintaining an IDOL index

Content Manager Enterprise Studio enables you to check and optimize IDOL document content indexes.

### Checking the IDOL index for errors

To check your IDOL index for errors:

1. Using Content Manager Enterprise Studio, expand **Datasets**.
2. Right-click the dataset, point to **Content Index**, and then click **Check**.

The Check content index validity dialog appears.

3. Check the log for details.
4. Click **OK**.

To rectify errors that may have been found during the check process:

- restore the IDOL data from a backup and reindex all documents from the date of that backup
- or
- recreate the complete content indexing database using Content Manager Enterprise Studio

5. Enable content index processing, and then save and deploy your configuration changes to all Workgroup Servers.

### Optimizing the IDOL index

The Content Manager command Content Index - Optimize issues a DRECOMPACT command to the IDOL server. This command instructs IDOL to reduce the space left from documents that have been deleted. It is similar to a defragmentation process.

This process should never be run while any indexing jobs are being processed. Doing so can corrupt the IDOL index, requiring a full re-index. If Content Manager Enterprise Studio detects that there are unprocessed index jobs, it will prevent the Optimization. The DRECOMPACT command can also be a time- and resource-intensive process, so should be run during low-usage periods.

The DRECOMPACT command also performs a backup of the index as part of the process. By default, this uses the internalbackup directory in the IDOL server's data directory. This can be changed by adding the PreCompactionBackupPath setting to the [Schedule] section of the main IDOL configuration file. This backup can be turned off by setting PreCompactionBackup=false

**[Schedule]**

**PreCompactionBackupPath=D:/IDOL/InternalBackup**

### PreCompactionBackup=true

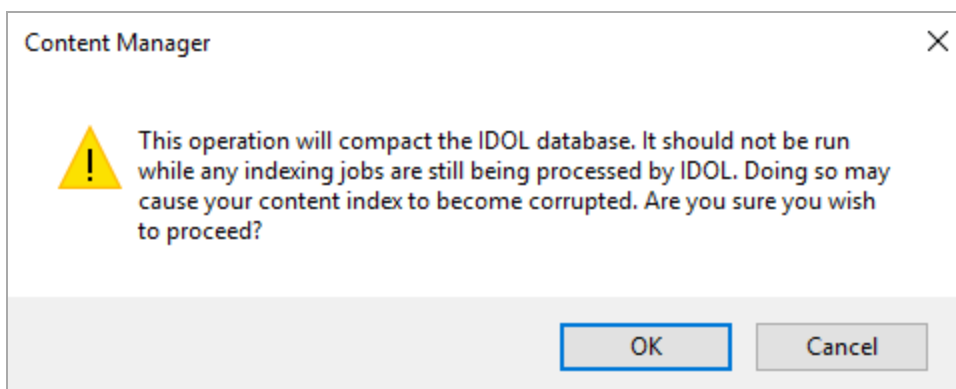
Regardless of the automatic pre-optimisation backup, it is also good practice to back up the child IDOL servers before optimizing them.

**CAUTION:** You must suspend the content indexing processor before using the Content Manager Enterprise Studio Content Index - Optimize option. See [Configuring content index event processing](#).

### Optimization steps

1. Using Content Manager Enterprise Studio, expand **Datasets**.
2. Right-click the dataset, point to **Content Index** and then click **Optimize**.

A warning message advising the operation will compact the IDOL database appears:



3. Click **OK** to proceed.  
The **Optimize content index database** dialog appears.
4. Check the log for details.
5. Click **OK**.
6. Enable the content index processing, and then save and deploy your configuration changes to all Workgroup Servers.

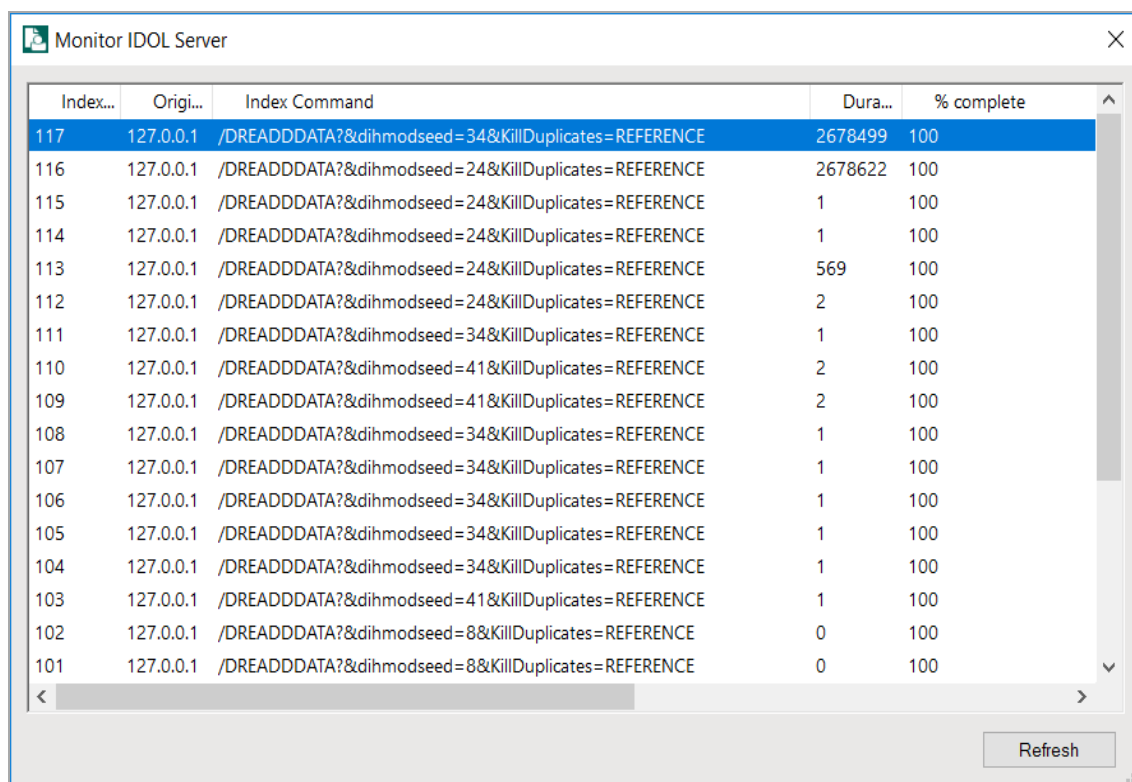
### Monitoring an IDOL Server

The activity of an Content Manager IDOL Server can be monitored via Content Manager Enterprise Studio.

#### Steps to monitor IDOL Servers

1. Using Content Manager Enterprise Studio, expand **Datasets**.
2. Right-click the dataset, point to **Content Index** and then click **Monitor**.

The **Monitor IDOL Server** dialog appears.



3. Click **Refresh** to update the view or click **Close** to close the monitor.

## Removing a document content index

The Content Manager Enterprise Studio **Content Index - Remove** command removes all references to the content index from Content Manager Enterprise Studio. It will also give the option to delete the content index from IDOL.

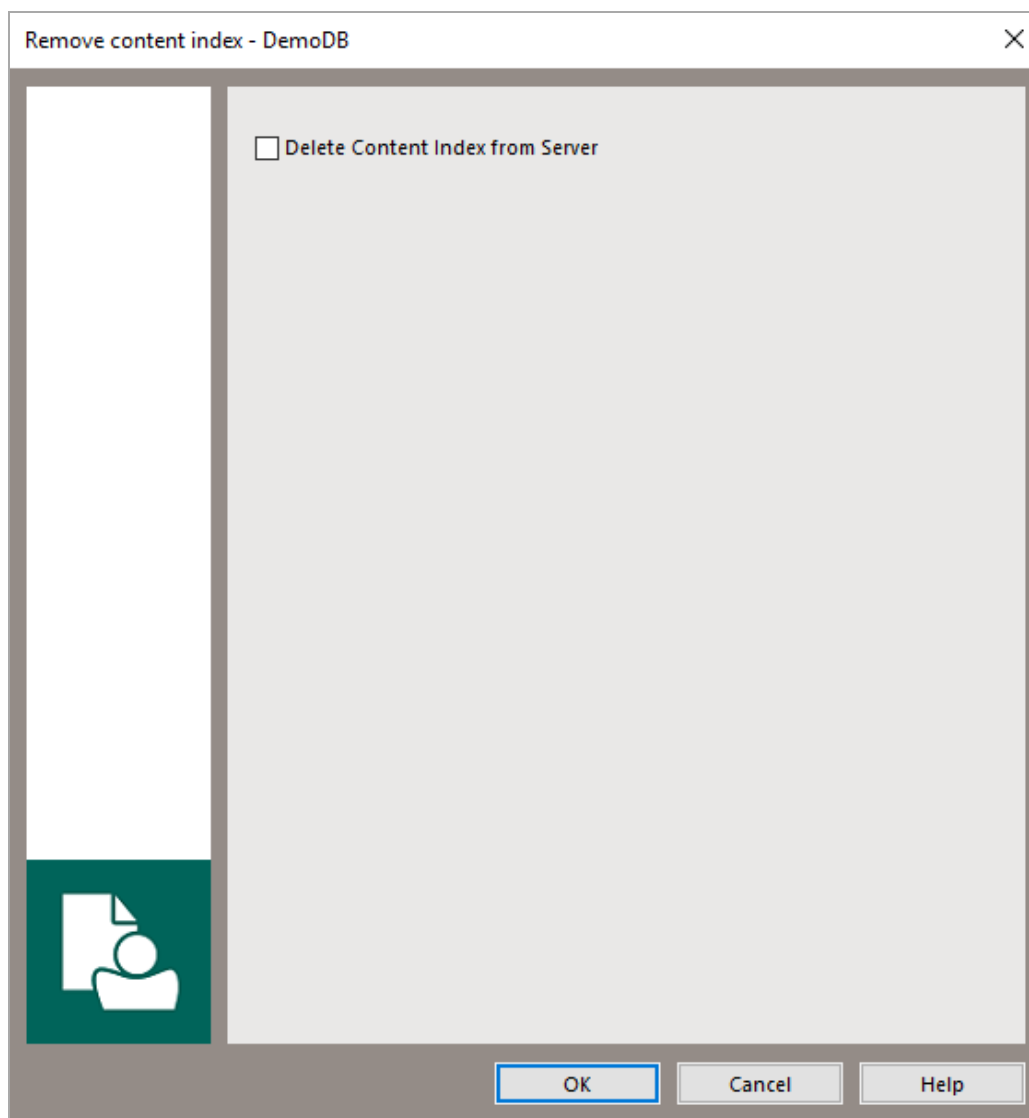
The content index is removed from Content Manager Enterprise Studio, and if selected, the content index is deleted from IDOL.

### Removal steps

**CAUTION:** You must suspend the content indexing processor before using the Content Manager Enterprise Studio > Content Index > Check option. See [Configuring content index event processing](#).

1. Using Content Manager Enterprise Studio, expand **Datasets**.
2. Right-click the dataset, point to **Content Index** and then click **Remove**.

The **Remove content index** dialog appears:



3. Select **Delete Content Index from Server**, if required.
4. Click **OK**.

The content index is removed from Content Manager Enterprise Studio, and if selected, the content index is deleted from IDOL.

### Deleting an IDOL database

It is possible to delete an existing IDOL database.

1. Using Content Manager Enterprise Studio, expand **Datasets**.
2. Right-click the dataset, point to **Content Index** and then click **Properties**.

The **Content index properties** dialog appears:



3. On the **Server** tab of the **Content Index properties** dialog, click **Delete Instance** to remove the IDOL instance with the name specified in the Server Instance Name field.

Alternatively, you can run the index command DREDELDBASE from a Web browser to delete all documents from an IDOL Server database.

DREDbName specifies the database from which to delete the documents. The database name is in the format CM\_[DBID], where DBID is the database ID that is set in Content Manager Enterprise Studio.

The example below uses localhost for the name of the server hosting IDOL.

Example:

**http://localhost:9001/DREDELDBASE?DREDbName=CM\_45**

This command uses port 9001 to delete all documents from the IDOL Server CM\_45 database.

After deleting, compact the database using Optimize through Content Manager Enterprise Studio. See [Maintaining an IDOL index](#).

## Suppressing document content indexing

You can exclude electronic documents from having their content indexed. This may be desirable because of policy, or because a particular file is causing problems for the indexing engine.

### For all files of a particular file type

1. Using Content Manager, from the **Administration** tab, in the **Options** group, click **System**.  
The **Content Manager System Options** dialog appears.
2. On the **File Types** tab, select a file type in the list and click **Properties**.
3. Deselect **Content indexing** and click **OK**.
4. Click **OK** to save the changes to the file types.

This will exclude all electronic documents of this type from having their document content indexed.

### For all electronic documents with a particular Record Type

1. In the Content Manager client, on the **Manage** tab, in the **Records** group, click **Record Types**.
2. Right-click a Record Type and click **Properties**.
3. On the **Electronic** tab, select **Suppress Content Indexing**.
4. Click **OK**.

## Using IDOL to search record metadata

### Configuring Content Manager to use IDOL to search metadata fields

Content Manager has several options for using the IDOL search engine for searching metadata fields. This can improve the performance of certain searches – in particular, searches involving both content and metadata search terms.

#### Document Content, Record Titles and Record Notes

This option will use IDOL for any search that involves document content, record titles or record notes (or any combination of those fields). To enable this option, select **Document Content, Record Titles and Record Notes** on the **Search for Records - Options** tab in Content Manager. See [Configuration steps](#) below for further information.

#### Use IDOL Indexes Where Possible

This option uses a search optimizer to analyze any searches. If the optimizer decides the search can be completed exclusively by IDOL (e.g. all of the search methods being used search against metadata available in IDOL), then the search is sent directly to IDOL. To enable this option, select **Use IDOL Indexes where possible** on the **Search for Records - Options** tab in Content Manager. See [Configuration steps](#) below for further information.

#### Configuration steps

1. To configure these options, on the Content Manager administrator computer, open the Content Manager client, on the **Search** tab, in the **Find** group, click **Records** and then click **Find Records**.

The **Search for Records** dialog appears.

2. On the **Options** tab, in the **Search optimizer** section, from the drop-down list on the **User of content engine for search optimization** option, select the required option (see below) and click **OK**:
  - **Only Document Content** - default. Content Manager uses IDOL to search the content of electronic documents only
  - **Document Content, Record Titles and Record Notes** - where possible, Content Manager uses IDOL for searches that use electronic document content, record titles and notes
  - **Use content indexes where possible** – Content Manager uses IDOL for all search methods that can be supported by IDOL
3. Restart your Workgroup Server and IDOL services.

Content Manager always uses IDOL for electronic document content searches. In addition to that, Content Manager passes a search to IDOL when it is compatible with the option that is selected here.

## Examples of metadata searching

When **Document Content, Record Titles and Record Notes** is selected, an Additional Field search does not go through IDOL, as the option does not include Additional Fields.

Even when **Use IDOL indexes where possible** is selected, a search for **title: reef OR datecreated: 1/1/2001 - 1/1/2002** does not use IDOL, because IDOL does not easily support OR date searches.

### Using the IDOL search method

When using the IDOL search method, the Content Manager client will pass whatever is typed into the query field directly across to the IDOL server. This enables users to run a basic search across any fields that are indexed by IDOL, similar to an **AnyWord** query using the Content Manager client. It also enables users to use many of the advanced IDOL search methods.

The value entered into the search field is sent to IDOL as the “text” value of a query action. To use any additional methods, simply add an ampersand character (“&”) before the parameter. If no specific “text” value is to be specified, use an asterisk (“\*”) before the ampersand.

The main advanced method that can be used is the FieldText parameter. FieldText allows specific fields that the results must contain, and conditions that these fields must meet.

Each FieldText condition must be associated with a particular field or fields. These fields correspond to metadata of Content Manager records. For a full list of the metadata indexed into IDOL and the associated field name, see [Appendix E FieldText Methods](#).

The format of a FieldText search as entered into the Content Manager client looks like:

```
*&FieldText=MATCH{DOC140516-4}:TS_NUMBER
```

This example would find search IDOL for the Content Manager record with a Record Number of DOC140516-4

To specify how records must match fields and field values to return as results, the FieldText parameter uses field specifiers that match the pattern of values in fields. These field specifiers fall into the following groups:

### Field specifiers for common restrictions

To find documents in which a specified field contains	Use the field specifier
a value that exactly matches one or more specified strings	MATCH
a number	EQUAL, GREATER, LESS, NRANGE or NOTEQUAL
a date	GTNOW, LTNOW or RANGE
a value that matches a specified Wildcard string	WILD

### Field specifiers for advanced restrictions

To find documents in which a specified field	use the field specifier
contains a value that falls within a specific alphabetical range	ARANGE
does not exist or does not contain a value	EMPTY
exists, irrespective of its value	EXISTS
contains a value that is similar to a specified string	FUZZY
contains a value whose length is a specific number of characters	LENGTH
contains multiple instances, whose values include at least one match for each of the specified strings or numeric values	MATCHALL or EQUALALL
contains multiple instances, all of whose values match in the specified strings or numeric values	MATCHCOVER or EQUALCOVER
contains multiple instances, at least one of whose values does not match the specified string	NOTMATCH, NOTSTRING or NOTWILD
contains a specified string	STRING, STRINGALL or SUBSTRING
whose value match specific terms or phrases	TERM, TERMALL, TERMEXACT, TERMEXACTALL, TERMEXACTPHRASE or TERMPHRASE

### Field specifiers for biasing result scores

To find documents in which a specified field	use the field specifier
To bias the score of results according to the numerical proximity of the specified field to a given value:	BIAS
To boost the score of any document in which the specified field has a specified date:	BIASDATE
To boost the score of any document in which the specified field has a specified value:	BIASVAL

For details on each mentioned FieldText method, please see [Appendix E FieldText Methods](#).

## OCR Rendering

### Requirements

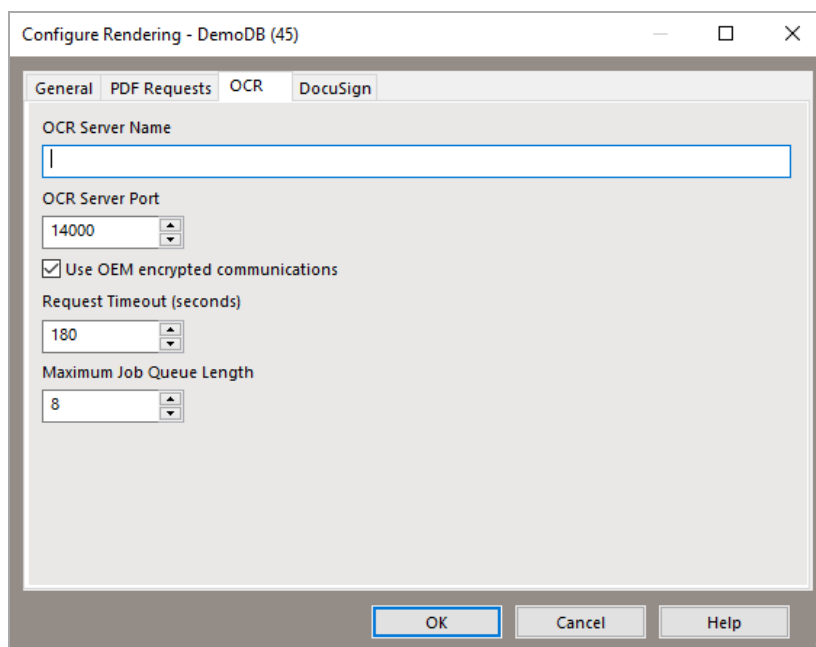
To use the Media Server with Content Manager for OCR, your system must meet the following requirements:

- Content Manager Media Server must be installed on a computer on your network. You can install it by using **CM\_MediaServer\_x64.msi** from your Content Manager installation media. It installs as the service **Content Manager Media Server**. For more details, see **CM24.2\_Install.pdf**.
- You must have a license for the **OCR Image Processing** module and the **Document Rendering** Feature must be enabled in the Content Manager client.
- The **OCR Server Name** and **OCR Server Port** must be specified in the **Configuring Rendering - OCR** tab in Content Manager Enterprise Studio.

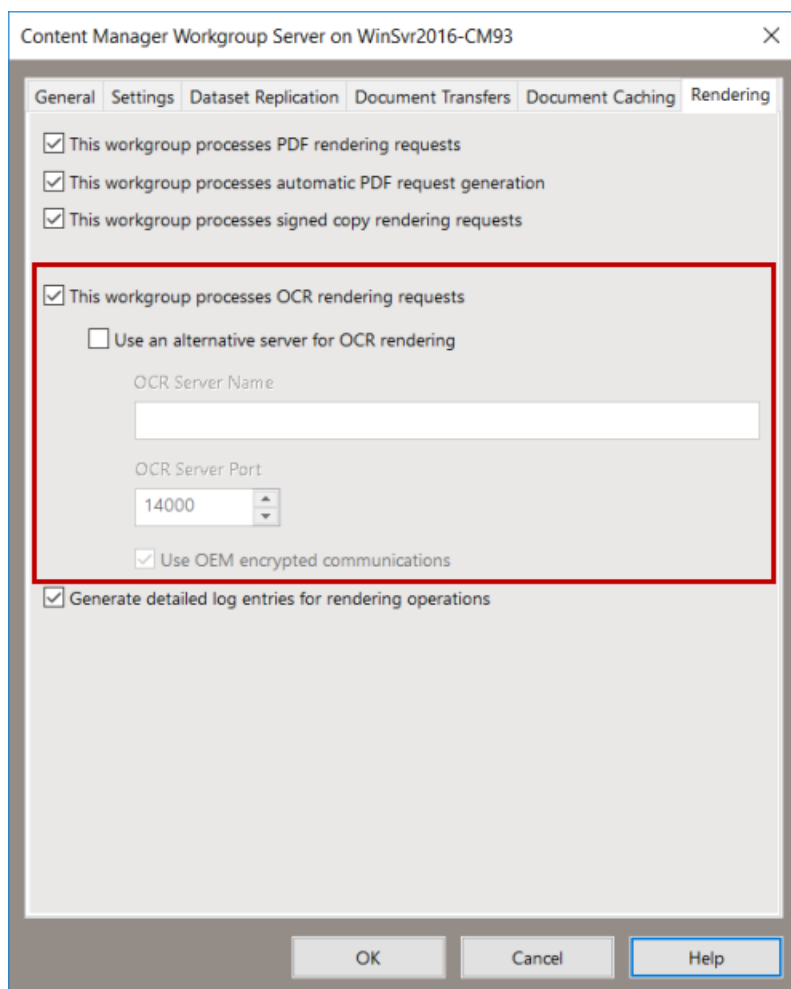
### OCR rendering installation and configuration

The instructions assume that you have a working Content Manager environment with clients, at least one Content Manager Workgroup Server and have the Document Rendering Event processing enabled and configured.

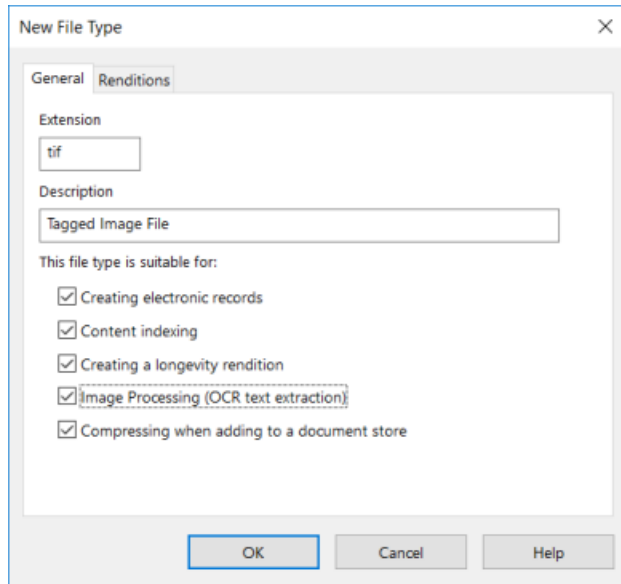
1. On the Media Server computer, use **CM\_MediaServer\_x64.msi** to install Media Server. For more details, see **CM24.2\_Install.pdf**.
2. Ensure all the Content Manager services are started, including **Content Manager Media Server**.
3. In Content Manager Enterprise Studio on the **Home** tab, in the **General** group, click **License**. Apply the license file that you received from Software Support that enables OCR Image Processing functionality in Content Manager.
4. In the Content Manager client, on the **Administration** tab, in the **Options** group, click **System**. On the **Features** tab, enable **Document Rendering**. Click **OK**.
5. In Content Manager Enterprise Studio, expand the **Dataset** node and right-click on the dataset name, point to **Event Processing** and then click **Rendering**. The **Configuring Rendering <datasetname>** dialog appears.



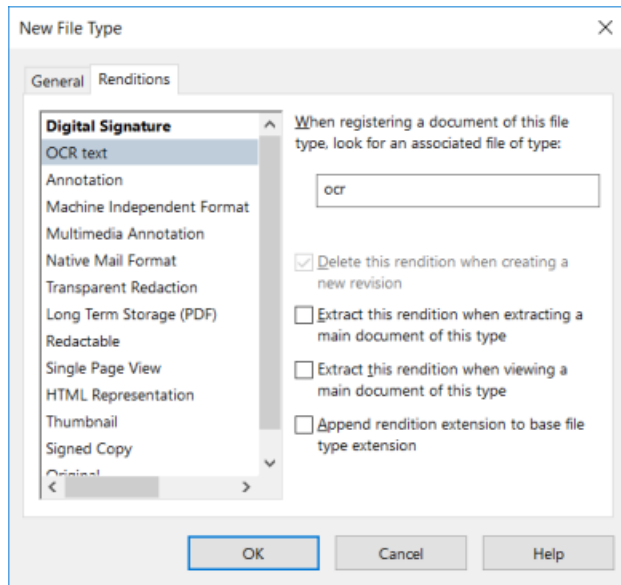
6. On the **OCR** tab, set **OCR Server Name** to point to where the Media Server was installed. The default port is 14000.
  - **Use OEM encrypted communications** - default: enabled with the OEM IDOL license. Disable if you are using Enterprise IDOL and require communications outside of Content Manager.
  - **Request Timeout (seconds)** - The duration in seconds before a request that has been submitted to the Media Server is automatically canceled by Content Manager. Once the timeout period has elapsed, Content Manager will mark the request as failed.
  - **Maximum Job Queue Length** - The maximum number of concurrent jobs that can be submitted to the Media Server.
7. Click **OK**.
8. In Content Manager Enterprise Studio, expand the **Workgroup Servers** node, right-click on the Workgroup Server name that is to process OCR rendering requests and then click **Properties**. The **Content Manager Workgroup Server <WGS name>** dialog appears.



9. On the **Rendering** tab, select **This workgroup processes OCR rendering requests**. If there is an alternative server processing OCR requests, the properties of this server can be defined using the **Use an alternative server for OCR rendering** option.
10. Ensure all Workgroup servers that are processing OCR rendition requests have this option enabled. Click **OK**.
11. In Content Manager Enterprise Studio, **Save** and **Deploy** your configuration changes to the Workgroup Servers.
12. In Content Manager, on the **Administration** tab, in the **Options** group, click **System**.
13. On the **File Types** tab, click **Add** and add an appropriate file type for OCR, e.g. TIF or JPG and select **Content Indexing and Image Processing (OCR text extraction)**.



14. On the **Renditions** tab, select **OCR text**. On the right for the field **When cataloging a document of this file type...**, type **ocr**.
15. If not enabled, select **Delete this rendition when creating a new revision** and click **OK** twice.



16. To automatically create OCR Renditions when records of a suitable type are checked into Content Manager, in the Content Manager client, on the **Manage** tab, in the Records group, click **Record Types**.
17. On the Record Type name, right-click and click **Properties**. On the **Record Type Properties** dialog, on the **Electronic** tab, select **Automatically create OCR rendition (for tif, jpg, etc.)** and then click **OK**.

Make sure that the services are running:



- **Content Manager Media Server**
- **Content Manager Workgroup Service**

The setup is complete.

Now, when you check in an electronic document to Content Manager, for example, a .tif file, a text rendition will automatically be created and will be attached to the record, where it becomes visible in the record **Properties - Renditions** tab.

The rendering processing queue can be monitored within the Content Manager client. From the **Administration** tab, in the **Other** group, click **Monitor Render Queue**. The **Monitor Render Queue** dialog appears displaying all documents that are in the render queue. See the Content Manager Help file for additional information.

After that, Content Manager indexes the document content, which makes it retrievable by using a document content search.

**NOTE:** There is a delay before the document content is available for searches. This is because by default, the delay is set to be 600 seconds in the content service configuration files, using the parameter `MaxSyncDelay`. For testing purposes, you could change this value to 30 temporarily by editing the configuration files.

For advanced OCR configuration details (including the OCRing of rotated text or other languages), please refer to the Media Server Admin guide (which can be downloaded as a ZIP file from <https://www.microfocus.com/documentation/idol/>).

## Troubleshooting OCR rendering

### File types

Ensure that the file types for the target documents have been configured correctly in Content Manager **System Options - File Types** tab as the file type extension, and not with a leading asterisk (\*).

For example, the file type should be entered as **JPG** and not **\*.JPG** - this would appear in the **File Types** dialog in Enterprise Studio as **\*.\*.JPG** instead of **\*.JPG**.

### Content Manager Media Server log files

The Content Manager Media Server log files, **TRIMRender.log**, can be used to check on how files are progressing through the Media Server. These logs can be useful for determining if files are being sent to the Media Server, and if the problem is occurring before, during or after the Media Server steps.

## Auto-Classification

### Requirements

To use the Content Manager Auto-Classification feature, your system must meet the following requirements:

- IDOL AgentStore and Category components must be installed and configured for the IDOL server. For OEM IDOL, these components are included in the OEM package, so no extra configuration should be required.
- You must have a license for the Content Manager Auto-Classification module. If your license key does not include the Auto-Classification module, the Auto-Classification configuration options are unavailable or hidden.

### Setting up Content Manager for Auto-Classification

To set up the Content Manager client for auto-classification, see the client Help file TRIM.chm in your Content Manager installation folder, topic **Auto-Classification**.

### Setting up IDOL for Auto-Classification

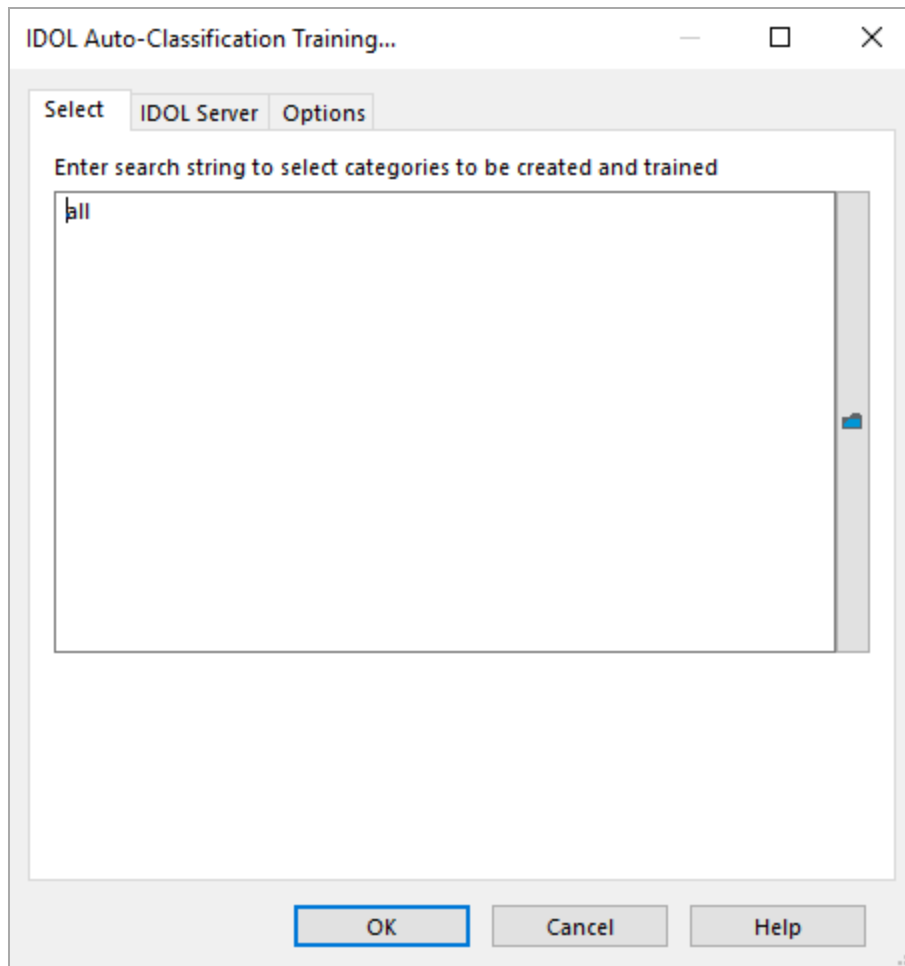
#### Training Classification terms

1. On the **Administration** tab, in the **IDOL Index** group, click **Auto-Classification**.

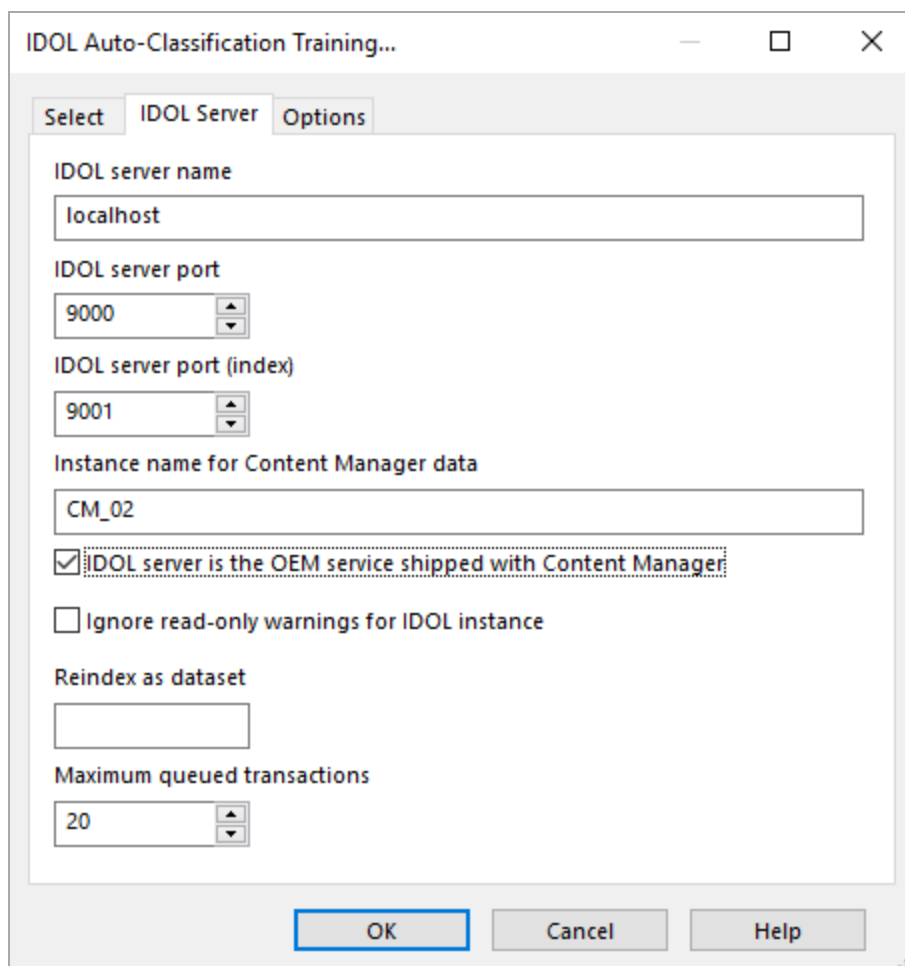
The **IDOL Auto-Classification Training** dialog appears for you to create and train categories in IDOL.

The **Select** tab features a Classification search dialog.

The Classifications that are selected here will be created in IDOL as Categories.



2. The **IDOL Server** tab allows you to specify the details of the indexing computer.



- **IDOL Server Name** - the computer that runs the **Content Manager IDOL Service**, the main IDOL Server.
- **IDOL Server Port** - IDOL Server port number, used for searching. See next option for more.
- **IDOL Server Port (Index)** - Used for sending documents to the server.

The port numbers should remain at their defaults unless you specifically changed the IDOL configuration file in the folder **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOL Service** or your equivalent.

- **Instance name for Content Manager data** - name of the instance on the IDOL server that this dataset communicates with. Defaults to **CM\_[database ID]**.

When you create a content index for a dataset, Content Manager creates an instance on the IDOL server, which is what Content Manager uses for all IDOL searches.

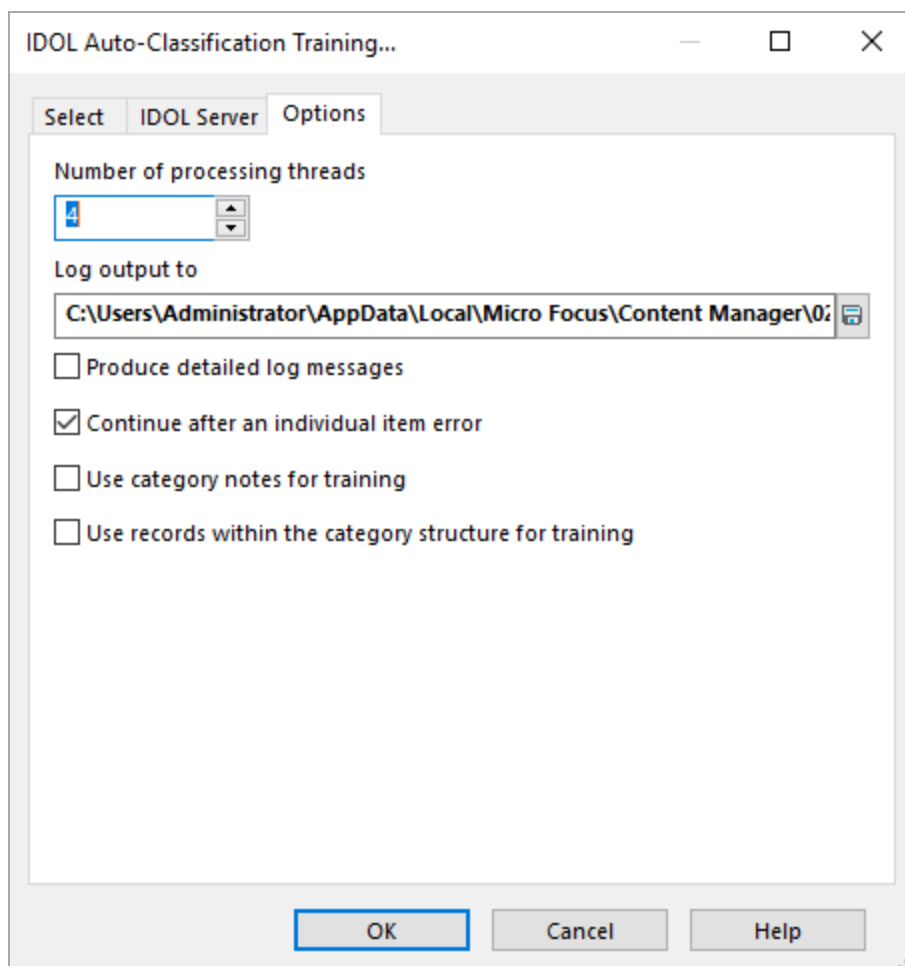
Should not be changed, unless you reindex the dataset afterwards, as Content Manager Enterprise Studio would create a new instance on the IDOL server when the instance name changes.

**CAUTION:**

While you can change the value in this field and re-index to a different instance, it is not recommended. When you re-index into a different instance, then Content Manager will not be able to find anything in that index, and creates unusable data in the other index.

The only scenario when an experienced Content Manager and IDOL administrator may want to re-index to a different instance, is to create your IDOL index offline, for example, for a test dataset or to create a new index while still using an old content index.

- **IDOL Server is the OEM service shipped with Content Manager** - select when you are using Content Manager's OEM IDOL version for indexing.
  - **Ignore read-only warnings for IDOL instance** - Select this option to ignore read-only warnings for content engines that have been set to read-only. Only suppress the warning if some of the content engines have been specifically marked as read-only.
  - **Reindex as Dataset** - To reindex content to a different dataset, specify in this field the target dataset value (e.g. CM\_45). This makes it possible to create a valid content engine for a production database while running the reindex on a copied dataset.
  - **Maximum queued transactions** - default: 20, which is the number of records in the queue. The smaller this number, the shorter the queue, which makes processing more reliable. There should be no need to change this number other than for troubleshooting. In that case, you could even reduce it to 1 to process one record at a time.
3. The **Options** tab controls how the Classifications will be processed, and how the Categories will be trained.



- **Number of processing threads** - controls how many Classifications are processed simultaneously.
- **Log output to** - location of the log file,
- **Produce detailed log messages** - toggles how detailed the logs are,
- **Continue after an individual item error** - select for IDOL to keep training, even when an individual item causes an error,

#### Training options

Any or all of the training options can be selected. More weight is given to the classification notes, but more representative terms can be extracted from existing documents.

- **Use category notes for training** - the notes on the selected Classification are used to train the category. This is useful the Classification terms have scope notes or other descriptions about the types of documents that should be located
- **Use records within the category structure for training** - the IDOL category can be trained based on a selection of records already classified under the target term. IDOL automatically identifies the 50 oldest (based on Date Registered) records under the target Classification. It is assumed that the oldest records are the most likely to have

been correctly classified, as they would have had the longest time to have any mistakes corrected.

4. Click **OK**.

Category creation and training begins.

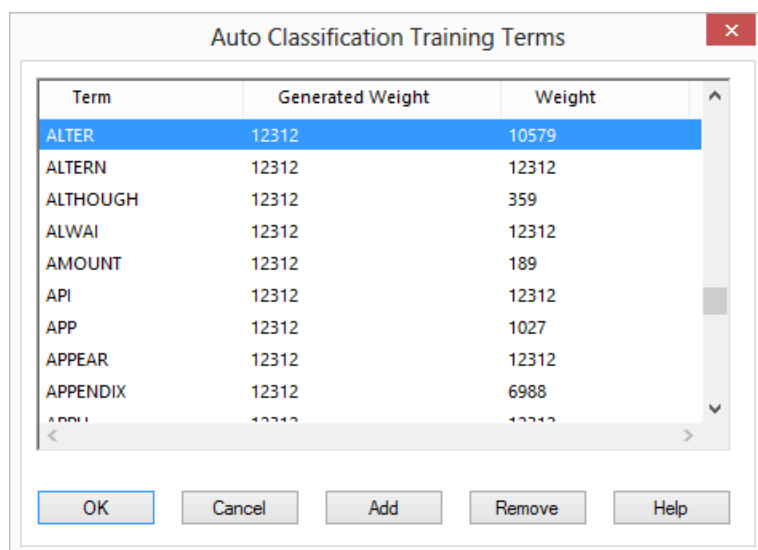
Content Manager works through the selected Classification terms, creates them in IDOL as categories and sets the training information.

Content Manager displays a message when processing is complete.

### Fine-tuning training

When IDOL trains a category, it generates a list of terms it feels best represents the “uniqueness” of the training material. These are the terms IDOL then uses to match a document to a particular category.

On the **Properties** dialog of each Classification term in the Content Manager client, on the **Auto-Classification** tab, there is a button labeled **Category Training**. This button launches a dialog that enables you to adjust the training terms:



The terms are listed in their “stemmed” form, so may appear to be truncated.

Each term is listed with its IDOL-generated weight in the **Generated Weight** column, and its current weight in the **Weight** column. Generated weight is the initial weighting IDOL determined that term should have. If this value is a dash (-), it means that the term was not generated by IDOL.

You can edit each term by double-clicking it. This launches another dialog where you can modify the values. If the term is a generated term, the term itself cannot be changed, but the weight can be.

Setting the weight of a generated term to 0 will set that term to be ignored by the auto-classification process.

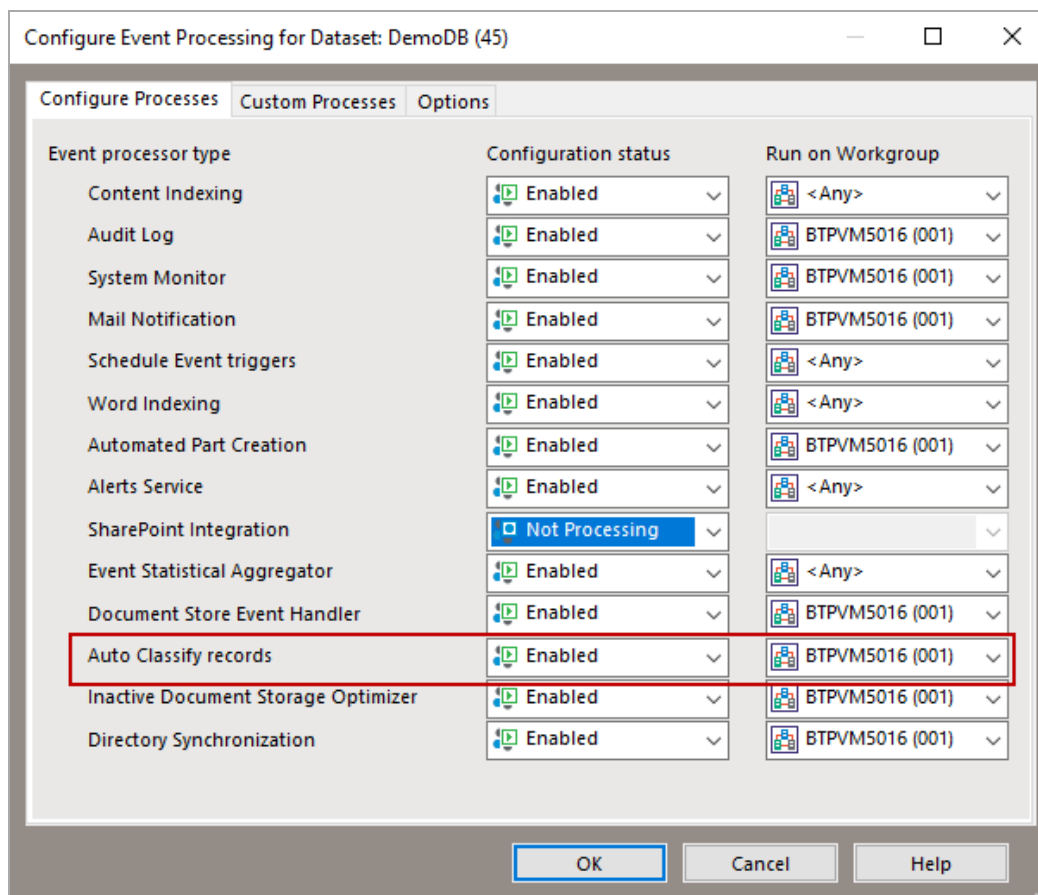
Non-generated terms can be added or removed by using the **Add** and **Remove** buttons.

When you are finished updating the terms, pressing **OK** will synchronize the modified term data with IDOL. If an IDOL connection cannot be found, no changes can be made.

## Enabling Auto-Classification event processing

1. Using Content Manager Enterprise Studio, expand **Datasets**.
2. Right-click the dataset, point to **Event Processing** and then click **Configure**.

The **Configure Event Processing for Dataset: <Dataset Name>** appears:



3. For the event processor type **Auto Classify Records**, from the **Configuration status** drop-down list, select **Enabled**.
4. From the **Run on Workgroup** drop-down list, select a particular Workgroup Server name to run the processor on
5. Click **OK**, and then save and deploy the configuration

## Running Auto-Classification

Once auto-classification has been setup and configured, and the Auto-Classify events have been enabled, auto-classification will start processing.

Content Manager will periodically check all of the holding bay Classification terms. For each record it finds, it will ask IDOL to suggest a Classification. If IDOL's confidence in the suggested Classification is equal to or higher than the configured minimum confidence level, the record is moved to that





Classification. Security, retention and any other inherited properties are automatically updated when the record is moved.

IDOL needs to have already content indexed the record, so if the content index event has not yet been processed or if IDOL has not yet updated its index, the record will be skipped until it is available in the IDOL index.

## Reviewing Auto-Classification

The auto-classification process requires monitoring and fine-tuning, particularly in the early stages.

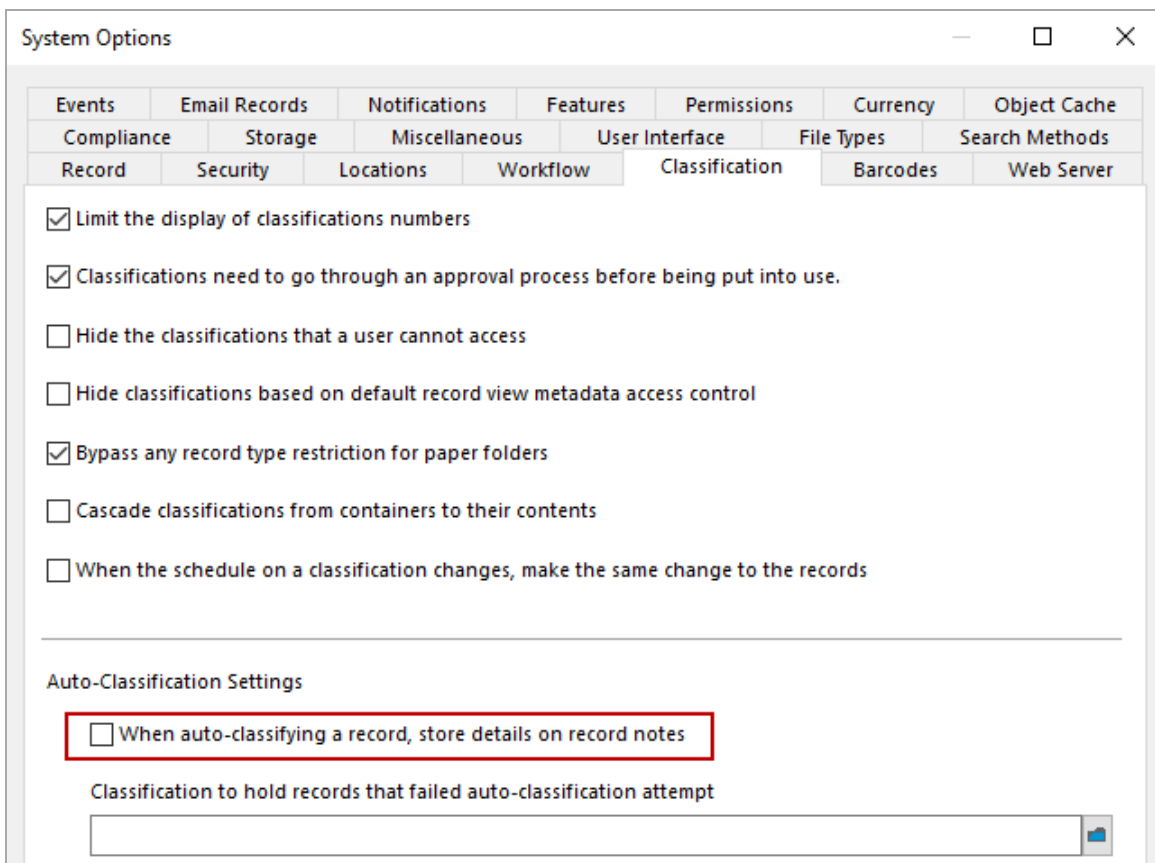
### Auto-Classification Confidence Level

Record Number	 REC_3453
Title	 Document1
Auto-Classification Confidence Level	82
Date Created	Wednesday, 28 May 2014 at 12:05 PM
Date Registered	Wednesday, 28 May 2014 at 12:05 PM

When a record is auto-classified, IDOL's confidence level is stored in the record field **Auto-Classification Confidence Level**.

You should review this field to help adjust the minimum confidence levels of the classifications.

## Storing Auto-classification details



The system option, **When auto-classifying a record, store details on record notes**, **Classification** tab, allows Content Manager to insert the details of an auto-classification operation to the record notes.

If enabled, then when an attempt is made to auto-classify a record, the details from IDOL are appended to the record notes. This information includes the confidence level and the terms that IDOL matched against.

This information should clearly explain why IDOL decided to classify a particular record the way it did.

This information is recorded even if the auto-classification failed – if IDOL was not confident enough for a record to be moved, the notes can be reviewed to see where IDOL felt the record should go, and how confident it was.

## Connectors

### Requirements

Content Manager Connector should be installed by a system administrator as part of a larger IDOL system that includes an IDOL Server, the CFS Connector Framework and Content Manager . The Content Manager Connector should be installed on a computer with access to the Content Manager SDK.

**NOTE:** For Content Manager Connector to work, you need to install Content Manager COM components separately. Run the **CM\_COMComponents\_x64.msi** to register the COM components.

### Configuration

You can configure the Content Manager Connector and the Connector Framework Service by editing the configuration file. The configuration files **HPRMConnector.cfg** and **ConnectorFramework.cf** are located in the connector installation folder. These files can be modified using a text editor.

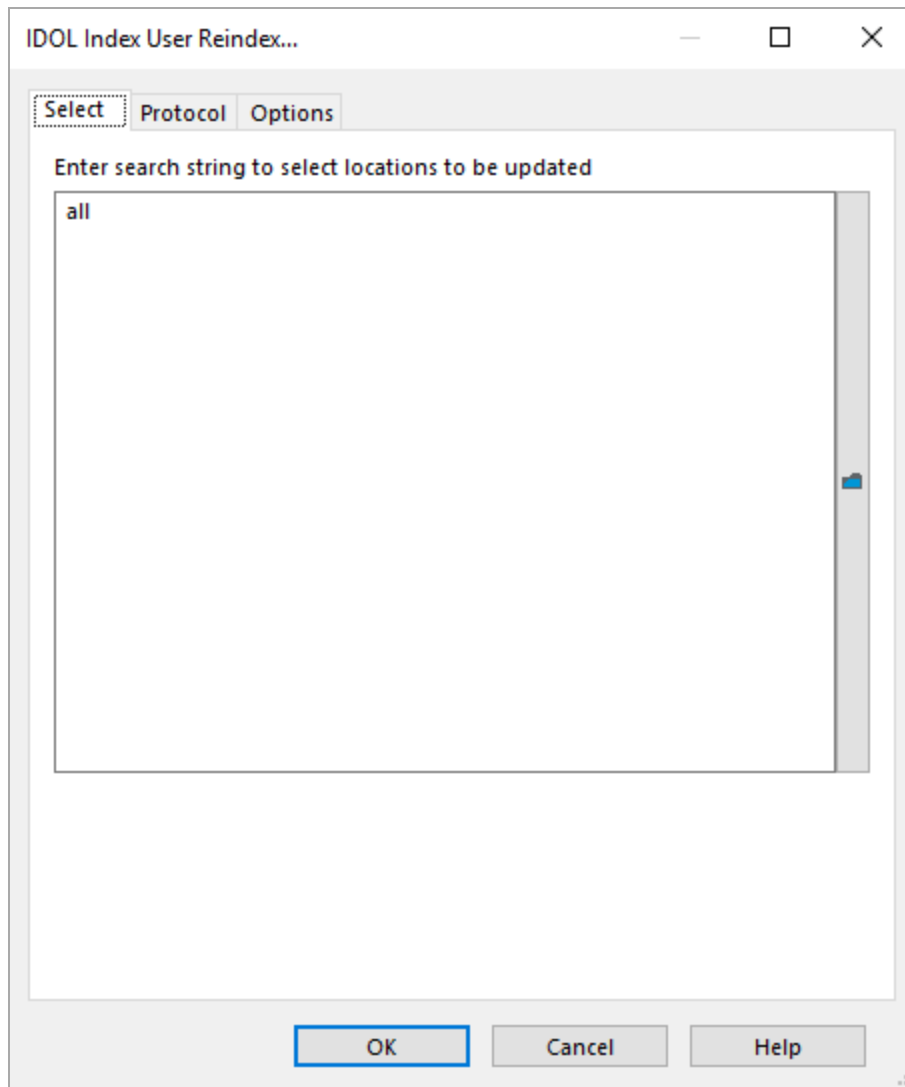
### Re-Indexing User Locations

You can use this function to update the user security data in IDOL by using the CFS connector. This is required when you first set up the Content Manager connector, and it is useful when an administrator has made changes to user security data like Content Manager Access Controls and user permissions.

1. Start the Content Manager client that has the Location's in it that require re-indexing.
2. On the **Administration** tab, from the **IDOL Index** group, click **Users**.

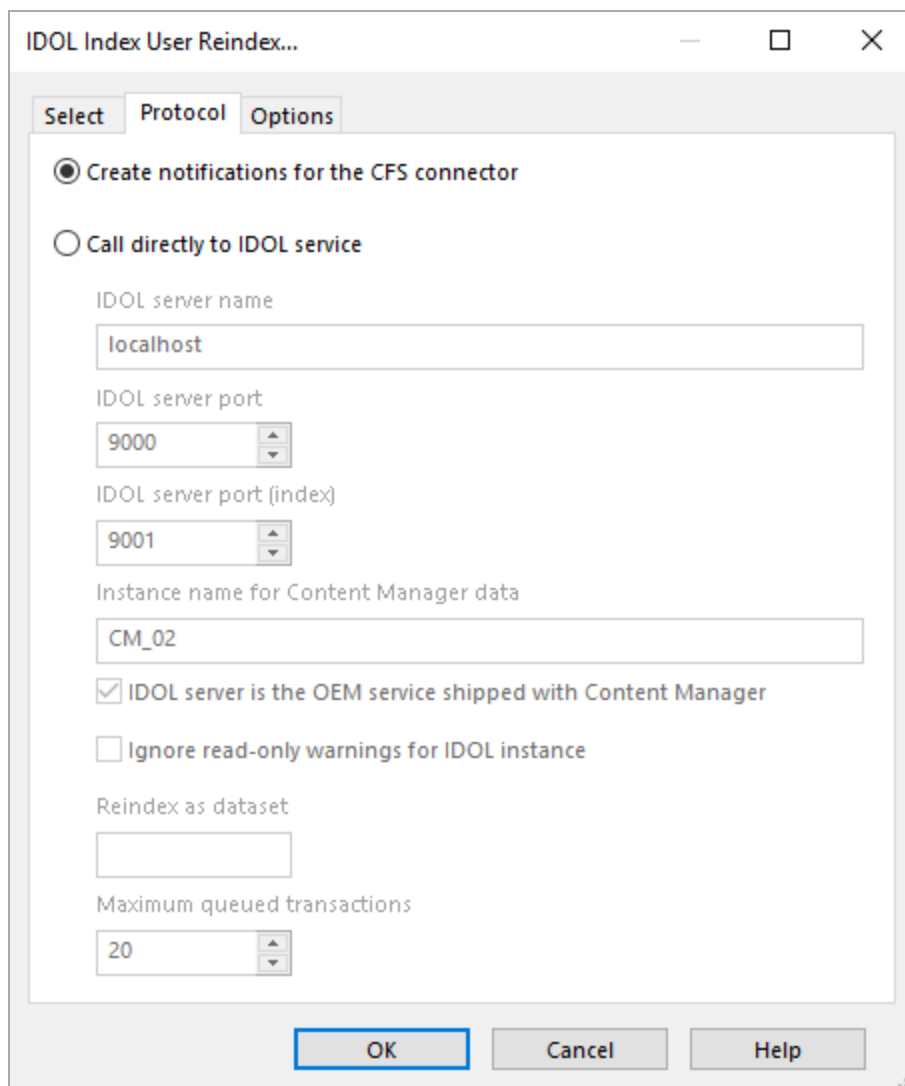
The **IDOL Index User Reindex** dialog appears.

3. On the **Select** tab - to select the Content Manager Locations to re-index:



- **Enter search string to select items to be reindexed** - type or use the KwikSelect button to form a search string to select the Locations to index.

4. On the **Protocol** tab - to specify the indexing computer:



- **Create notifications for the CFS Connector** - select to use the CFS connector to pass data to Enterprise IDOL.
- **Call directly to IDOL service** -
  - **IDOL Server Name** - the computer that runs the **Content Manager IDOL Service**, the main IDOL Server.
  - **IDOL Server Port** - IDOL Server port number, used for searching. See next option for more.
  - **IDOL Server Port (Index)** - Used for sending documents to the server.

The port numbers should remain at their defaults unless you specifically changed the IDOL configuration file in the folder **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOL Service** or your equivalent.

- **Instance name for Content Manager data** - name of the instance on the IDOL server that this dataset communicates with. Defaults to **CM\_[database ID]**.

When you create a content index for a dataset, Content Manager creates an instance on the IDOL server, which is what Content Manager uses for all IDOL searches.

Should not be changed, unless you reindex the dataset afterwards, as Content Manager Enterprise Studio would create a new instance on the IDOL server when the instance name changes.

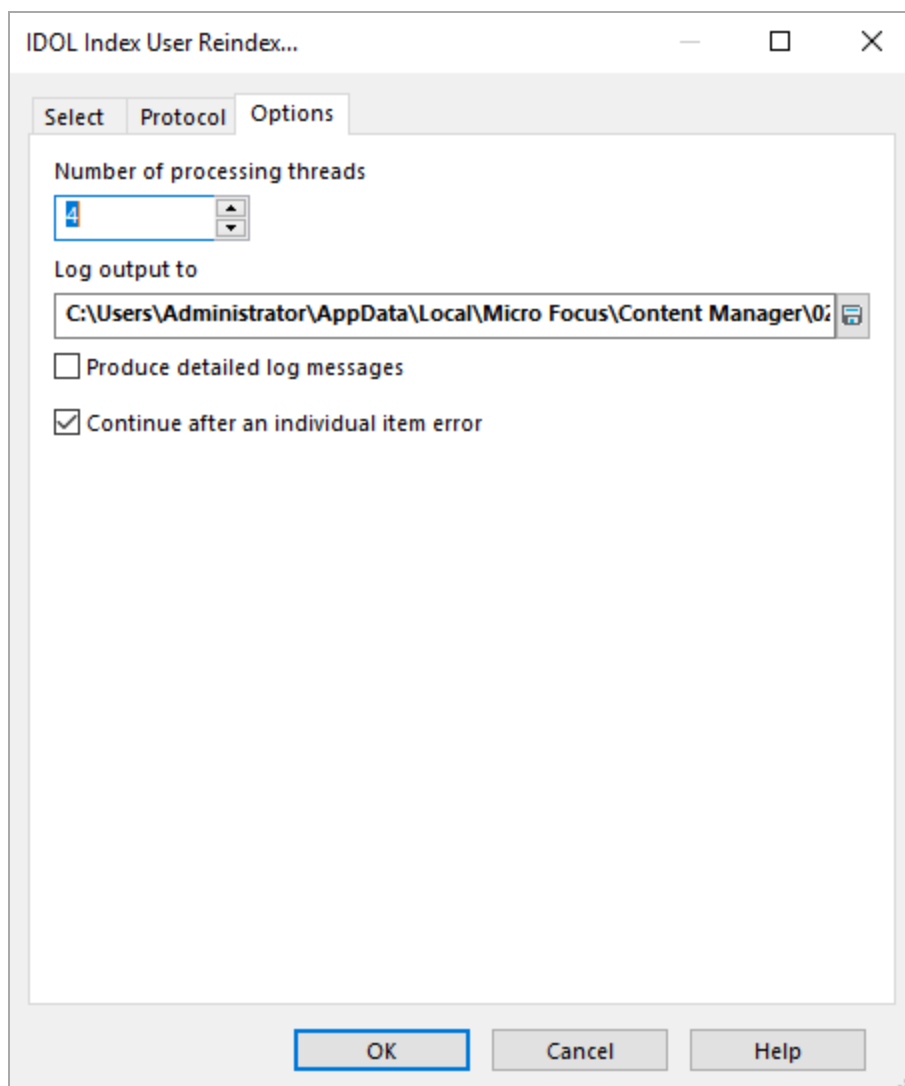
**CAUTION:**

While you can change the value in this field and re-index to a different instance, it is not recommended. When you re-index into a different instance, then Content Manager will not be able to find anything in that index, and creates unusable data in the other index.

The only scenario when an experienced Content Manager and IDOL administrator may want to re-index to a different instance, is to create your IDOL index offline, for example, for a test dataset or to create a new index while still using an old content index.

- **IDOL Server is the OEM service shipped with Content Manager** - select when you are using Content Manager's OEM IDOL version for indexing.
- **Ignore read-only warnings for IDOL instance** - Select this option to ignore read-only warnings for content engines that have been set to read-only. Only suppress the warning if some of the content engines have been specifically marked as read-only.
- **Reindex as Dataset** - To reindex content to a different dataset, specify in this field the target dataset value (e.g. CM\_45). This makes it possible to create a valid content engine for a production database while running the reindex on a copied dataset.
- **Maximum queued transactions** - default: 20, which is the number of records in the queue. The smaller this number, the shorter the queue, which makes processing more reliable. There should be no need to change this number other than for troubleshooting. In that case, you could even reduce it to 1 to process one record at a time.

5. On the **Options** tab - to specify the indexing parameters:

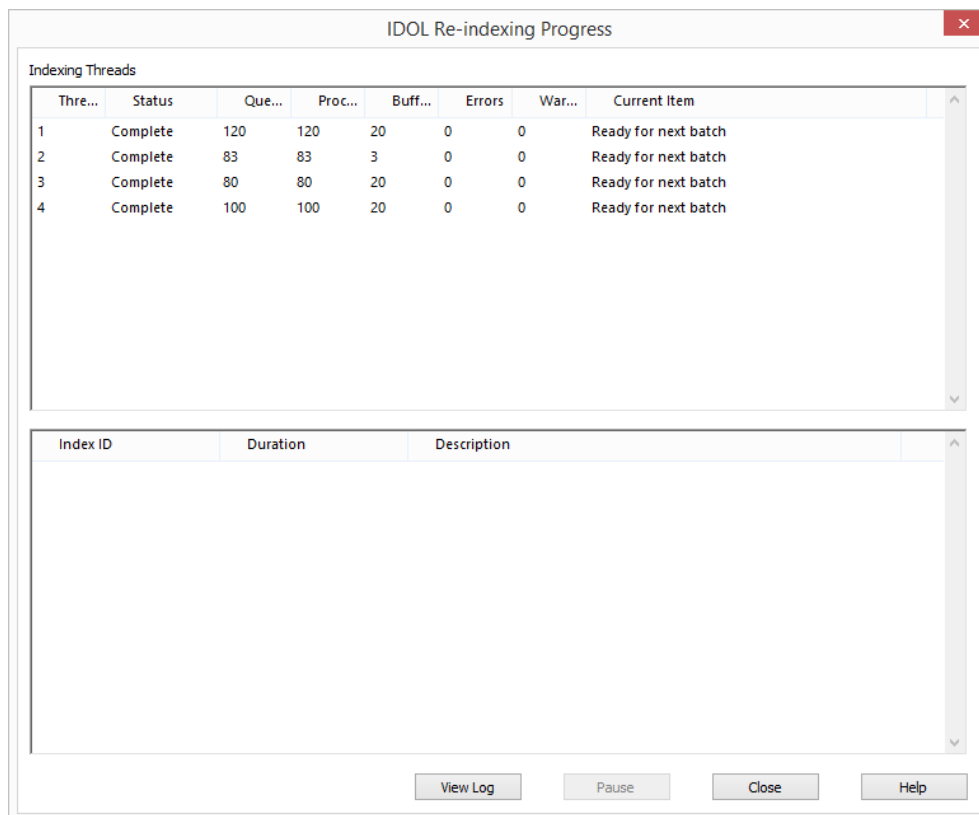


- **Number of processing threads** - default: 4. You can use a number from 1 to 20. The greater the number of threads, the faster the indexing process. It depends on your hardware capacity how many threads you can run.
- **Log output to** - the log file location
- **Produce detailed log messages** - select to see more detail in the log files
- **Continue after an individual item error** - should always be selected. When the option is not selected, the indexing process stops when it encounters an error.

6. Click **OK**.

Content Manager starts re-indexing and the **IDOL Re-indexing Progress** dialog appears with the status of the items to index:

- **View Log** - click to see the log file.
- **Pause** - click to pause the indexing process for all threads. Useful for example to add disk space for the index. When paused, click **Resume** to continue indexing.



When indexing has completed, the **Pause** button becomes unavailable.

7. Click **Close** to close the dialog box.

When an indexing process does not complete, for example because it was cancelled or a hardware failure, then when you start another indexing process with the same search parameters, Content Manager displays a message box to confirm whether it should resume the interrupted indexing process from where it stopped.

If you click **Yes**, Content Manager inserts **ADD URI>XXX** to the record search, which excludes the records that were already indexed.

If you click **No**, Content Manager starts the indexing process without resuming the previous process.



## IDOL Configuration

This section outlines some of the more common configuration options available to a site. Unless otherwise stated, these configuration options are either general IDOL advice, or options related to content searching.

### Securing the IDOL Server

In theory any user can issue commands to an IDOL server from an Internet browser including (but not limited to) index delete, backup and initialization commands, as well as IDOL service control commands. While OEM IDOL will only allow searches of the content database to be made from Content Manager, by default any client computer can be used to send other commands such as index maintenance or service control commands. Therefore, it is important you restrict access to the Content Manager IDOL servers to the appropriate computers.

By default, in the [Server] section of the TRIM IDOL Service and TRIM Content Service configuration files, you will find the following parameters and values:

```
QueryClients=*. *.*.*.*
```

```
AdminClients=*. *.*.*.*
```

```
IndexClients=*. *.*.*.*
```

This means by default there are no restrictions on which computers/IP addresses are allowed to send commands to IDOL. Instead of leaving these IP ranges open, it is recommended to restrict them to only the names/IP addresses of the computers that need to be able to issue commands to IDOL.

**NOTE:** You can specify IP addresses, IP address ranges (e.g. 15.146.146.\*), specific computers, or a combination of these (separated by commas with no spaces). In some environments DNS hostname resolution may not reliably translate server names to IP addresses, hence specifying IP addresses rather than hostnames is preferred.

### Content Manager IDOL Service

An example configuration of these parameters for the **TRIM IDOL Service.cfg** file would be:

```
QueryClients=localhost,CMWGS1,CMWGS2,CMAAdminPC1,CMAAdminPC2
```

```
AdminClients=localhost,CMWGS1,CMWGS2
```

```
IndexClients=localhost,CMWGS1,CMWGS2,CMAAdminPC1,CMAAdminPC2
```

**NOTE:** The above example shows a sample configuration of these parameters in the TRIM IDOL Service.cfg file, based on the following rules:

- All Workgroup Servers should be included in QueryClients, AdminClients and IndexClients. This will ensure that any IDOL commands used within a Content Manager IDOL OEM installation will be accepted by IDOL.

- Any Content Manager client machines that will be performing IDOL Record Reindexing, IDOL User Reindexing or IDOL Category Training need to be included in QueryClients and IndexClients.

### Content Manager Content Service

An example configuration of these parameters for the **TRIM Content Service 1.cfg** file would be:

```
QueryClients=myIDOLserver,CMWGS1,CMWGS2,CMAAdminPC1,CMAAdminPC2
```

```
AdminClients=localhost,myIDOLserver
```

```
IndexClients=myIDOLserver
```

**NOTE:** The above example shows a sample configuration of these parameters in the TRIM Content Service 1.cfg file, based on the following rules:

- The Content Manager IDOL Service needs to send administrative actions, indexing jobs and queries to the Content Manager IDOL Content engines, and hence needs to be included in QueryClients, AdminClients and IndexClients.
- Content Manager Enterprise Studio needs to confirm the online status of all content engines when creating and configuring DCI for a dataset, and hence needs to be included in QueryClients.
- Any Content Manager client machines that will be performing IDOL Record Reindexing, IDOL User Reindexing or IDOL Category Training need to confirm the online status of all content engines, and hence need to be included in QueryClients.

### Content Manager Media Server Service

An example configuration of these parameters for the **TRIM Media Server.cfg** file would be:

```
AdminClients=127.0.0.1,CMRenderServer
```

```
QueryClients=CMRenderServer
```

**NOTE:** The above example shows a sample configuration of these parameters in the **TRIM Media Server.cfg** file, based on the following rules:

- If Content Manager Render is installed and configured to perform OCR, its hostname or IP address needs to be included in the QueryClients and AdminClients parameters in the Media Server configuration.
- The IndexClients parameter is not applicable to the Content Manager Media Server and hence is not included in its configuration file.

### Restricting Content Manager IDOL Service Control Commands

By default, in the [Service] section of Content Manager IDOL configuration files, you will find the following parameters and values:

```
ServiceStatusClients=*. *.*.*
```

```
ServiceControlClients=*. *.*.*
```

This means by default there are no restrictions on which computers/IP addresses are allowed to send service control commands to IDOL. Instead of leaving these IP ranges open, recommends restricting them to only the names/IP addresses of the computers that need to be able to issue service control commands (e.g. action=stop) to IDOL services.

An example configuration of these parameters for any of the Content Manager IDOL configuration files would be:

```
ServiceStatusClients=*. *.*.*
```

```
ServiceControlClients=localhost,CMWGS1,CMWGS2,CMAAdminPC1,CMAAdminPC2
```

**NOTE:** The above example shows a sample configuration of these parameters in any of the Content Manager IDOL configuration files, based on the following rules:

- Any machine that needs to be able to send a command such as `http://MyIDOLServer:9002/action=stop` to any Content Manager IDOL service must be included in `ServiceControlClients`.

## Mirroring

### Simple

The default installation of the OEM IDOL content services are configured with mirroring disabled. Enabling mirroring will require extra disk space, memory and CPU resources. Normally, the content index is distributed across each content service instance. However, in mirroring mode, the entire index is available on each content service. There must be sufficient disk space and resources to accommodate this setting. The disk space required is the size of the entire index multiplied by the number of content services.

The advantage of enabling mirror mode is high availability and fault tolerance. Before configuring mirroring, the recommendations in [KM02602193 - Mirroring a CM IDOL \(OEM\) environment for High Availability](#) should be considered.

To activate `MirrorMode`, in the main IDOL configuration file, set the `MirrorMode` property to **TRUE**.

When the `MirrorMode` property is changed, the `\dih\Main\` subdirectory default location **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOL Service\dih\Main\** needs to be deleted.

### Tiered

The simple configuration above allows for the index to be replicated across all the configured content engines. It is also possible to tier multiple IDOL servers together, mirroring data between them.

1. Install and configure the OEM IDOL package on two separate servers. Each OEM package can be configured individually as described in the rest of this document. For this example, we will refer to these as `IDOLServer1` and `IDOLServer2`
2. Install a third instance of the OEM IDOL package. For this example, we will refer to this computer as `IDOLMainServer`.
3. On `IDOLMainServer`, follow the instructions for removing a content service to deactivate both content services (but ignore the configuration changes).
4. On `IDOLMainServer`, update the **TRIM IDOL Service.cfg** file, replacing each **DIHEngine** and **DAHEngine** with the details for the OEM IDOL packages installed on `IDOLServer1` and `IDOLServer2`.

Note that the port values should be set to the port of the main Content Manager IDOL Service on each of the `IDOLServers` - default **9000**.

5. On IDOLMainServer, update the **TRIM IDOL Service.cfg** file to set **MirrorMode=TRUE**

Example configuration for IDOLMainServer:

```
[DAEngines]
```

```
Number=2
```

```
[DAEngine0]
```

```
Host=IDOLServer1
```

```
Port=9000
```

```
[DAEngine1]
```

```
Host=IDOLServer2
```

```
Port=9000
```

```
[DIEngines]
```

```
Number=2
```

```
[DIEngine0]
```

```
Host=localhost
```

```
Port=9000
```

```
[DIEngine1]
```

```
Host=localhost
```

```
Port=9000
```

## Deactivating or removing a content engine

If there is a requirement to remove or deactivate a content engine, such as when upgrading from an older version of Records Manager. The extra content engine can be disabled so it is not impacting on the available resources.

### Service Changes

In the Windows Services function, locate the Content Service that is to be uninstalled. If running, stop the service, and set its startup type to **Manual**.

If needed, the content service can be uninstalled.

1. Launch the Windows command line as an administrator
2. Change the path to the installation directory of the Content Service and type "**Name of content service.exe**" -**uninstall** and press **Enter**.

The Content Service should disappear from the Windows **Services** control panel.

### Configuration changes

The redundant content engine can be removed from the configuration file of the main IDOL service. This will stop IDOL from attempting to communicate with that content engine.

1. Locate and edit the **TRIM IDOL Service.cfg**, default location **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOL Service\TRIM IDOL Service.cfg**
2. Change both the **[DAEngines]** and **[DIEngines]** number to 1
3. Remove or comment out the **[DAEngine1]** and **[DIEngine1]** sections
4. Save the configuration file and restart the main IDOL service.

Example configuration:

```
[DAEngines]
Number=1
```

```
[DAEngine0]
Host=localhost
Port=9100
```

```
[DIEngines]
Number=1
```

```
[DIEngine0]
Host=localhost
Port=9100
```

The above instructions assume Content Service 2 is being deactivated. If Content Service 1 is being deactivated, the ports in the example configuration would be different.

## Adding an additional content engine

There are two steps to adding additional content engines. First, the new content engine must be installed on a computer. Secondly, the main IDOL server needs to be made aware of the new content engine by editing the **TRIM IDOL Service.cfg** file.

For additional information on installing additional Content Manager IDOL content services please see Tech Note: [Adding additional IDOL content engines to Content Manager](#).

### Manual installation

This example creates a second Content Manager IDOL content service. For each additional content service, replace the number 2 with the appropriate number and pick suitable unused port numbers.

**CAUTION:** Before creating additional content engine services, stop all Content Manager IDOL services.

See <https://softwaresupport.softwaregrp.com/group/softwaresupport/search-result/-/facetsearch/document/KM00209246> for the correct way to shutdown the IDOL services.

1. Navigate to **C:\Program Files\Micro Focus\Content Manager\IDOL** and copy the folder **TRIM Content Service 1** to a temporary folder.
2. Rename the new folder to **TRIM Content Service 2**.

3. In the folder **TRIM Content Service 2**, rename the files **TRIM Content Service 1.cfg** and **TRIM Content Service 1.exe** to **TRIM Content Service 2.cfg** and **TRIM Content Service 2.exe**, respectively.
4. Edit the file **TRIM Content Service 2.cfg** and change the sections Service Settings and Server Settings to the following:

```
//----- Service Settings -----//  
[Service]  
ServicePort=9202  
ServiceStatusClients=*. *.*.*  
ServiceControlClients=*. *.*.*  
//----- Server Settings -----//  
[Server]  
Port=9200  
IndexPort=9201
```

You can use any value for **Port**, **IndexPort**, and **ServicePort** number that is available in your environment. However, what you use must be reflected correctly in the Content Manager IDOL configuration file **TRIM IDOL Service.cfg**.

5. Change the sections Path Settings and Logging Settings to the following:

```
//----- Path Settings -----//  
[Paths]  
BitFieldPath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\bitfield  
DyntermPath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\dynterm  
IndexQueuePath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\indexqueue  
IndexTempPath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\indextmp  
MainPath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\main  
NodetablePath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\nodetable  
NumericPath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\numeric  
RefIndexPath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\refindex  
SecIndexPath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\secindex
```

```
SortfieldPath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\sortfield
StatusPath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\status
StatePath=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\storedstate
TagPath=D:\Micro FocusE Content Manager\IDOL\TRIM Content Service 2>tagindex
LanguageDirectory=..\shared\langfiles
//----- Logging Settings -----//
[Logging]
LogDirectory=D:\Micro Focus Content Manager\IDOL\TRIM Content Service 2\logs
```

In this example, the path for the IDOL indexes is **D:\Micro Focus Content Manager\IDOL**. You can change it to any location on any server by changing it in the configuration file **TRIM IDOL Service.cfg**. You can also use UNC paths.

6. Save the configuration file.
7. Copy the entire directory **TRIM Content Service 2** to **C:\Program Files\Micro Focus\Content Manager\IDOL**.
8. Open a command prompt as administrator and navigate to the directory **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM Content Service 2**.
9. Run the command **Trim Content Service 2.exe –install**.

This will install **Content Manager Content Service 2** as a service. For the new service, you may want to change the value for **Log on as**.

### Configuration changes

The configuration file for the main IDOL service needs to be updated to include a reference to the new content engine. This example adds a third content engine, which is already installed and configured a different computer to the main IDOL service (named “server 2”), configured to listen on port 9300.

1. Navigate to the directory **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOL Service** and as administrator, edit the file **TRIM IDOL Service.cfg**.
2. Set:

```
[DAHEngines]
```

```
Number=3
```

1. Add:

```
[DAHEngine2]
```

```
Host=server2
```

```
Port=9300
```

2. Set:

```
[DIHEngines]
```

```
Number=3
```

3. Add:

```
[DIHEngine2]
```

```
Host=server2
```

```
Port=9300
```

Ensure the port number matches the one that you used in the file TRIM Content Service 3.cfg.

3. Save **TRIM IDOL Service.cfg** .

4. Open Windows Services and start all Content Manager content engine services including the newly installed content service, and then start the **Content Manager IDOL Service**.

## Moving a content engine

“Moving” a content engine is a combination of deactivating or removing an engine, and adding a content engine: Remove the content service from the original computer, and then install another content service on the new computer.

Example configuration:

```
[DAHEngines]
```

```
Number=2
```

```
[DAHEngine0]
```

```
Host=originalServer
```

```
Port=9100
```

```
[DAHEngine1]
```

```
Host=newServer
```

```
Port=9200
```

```
[DIHEngines]
```

```
Number=2
```

```
[DIHEngine0]
```

```
Host= originalServer
```

```
Port=9100
```

```
[DIHEngine1]
```

```
Host=newServer
```

```
Port=9200
```

## Backing up the content index

Due to its distributed and volatile nature, backing up an IDOL content index can be a complex task.



## Manual backup

A manual backup of an IDOL server's index can be triggered using the DREBACKUP command. This command copies all the IDOL server data index's \*.DB files to a specified location. These files can then be restored at a later point. See [Restoring a backup](#) for details.

To initiate the manual backup, use a web browser and navigate to:

**http://[IDOLSERVER]:[INDEXPORT]/DREBACKUP?[BACKUP PATH]**

where IDOLSERVER is the hostname or IP address of the main IDOL server, INDEXPORT is the value configured for the IndexPort parameter in the main IDOL service's CFG file (default 9001), and [BACKUP PATH] is the path to use for the backed up files.

For example: **http://localhost:9001/DREBACKUP?C:\Backup**

**NOTE:** The above method assumes the IDOL server has only one content engine. For IDOL implementations with multiple content engines the 'HostDetails' parameter should be added to ensure each engine writes its backup to its own unique subfolder.

For example: **http://localhost:9001/DREBACKUP?C:\Backup&HostDetails=true**

For a more detailed discussion about backing up CM IDOL indexes, please refer to knowledgebase article *KM03471260 - Backing up and Restoring CM IDOL Content Indexes*.

## Scheduled backup

Each content engine, also known as a Content Manager content service or TRIM content service, can be configured to back up its portion of the IDOL index to a specific location at a regular interval.

The configuration changes outlined below need to be made for each content engine in the site's IDOL configuration, otherwise the index may only be partially backed up.

1. Locate the Content Service configuration file (TRIM Content Service X.cfg or Content Manager Content Service X.cfg), which is located in the directory with the Content Service.exe. By default, this will be installed at C:\Program Files\Micro Focus\Content Manager\IDOL\
2. Add a [Schedule] section to the configuration file, and add the following properties as appropriate:
  - a. **Backup** – set to TRUE (required for backup scheduling)
  - b. **BackupCompression** – set to TRUE to compress the backup files, and use BackupCompressionLevel to control the level of compression
  - c. **BackupCompressionLevel** – set between 0-9. 0 is no compression, but fastest; 9 is maximum compression, but slowest. Default is 6.
  - d. **NumberofBackups** – number of backups that are created before the earliest backup files are overwritten (required)
  - e. **BackupDirN** – path to directory where this backup will be written. You must specify one directory for each backup as specified in **NumberofBackups** (required)
  - f. **BackupInterval** – number of hours between backups. When the IDOL component starts, the specified time must elapse before the first backup occurs. Set to 0 for daily backups.

- g. **BackupMaintainDirStructure** – set to TRUE to maintain the directory structure when IDOL component is backed up
  - h. **BackupTime** – time (hh:mm) of first backup in 24 hour time.
  - i. **BackupRetryAttempts** – number of times to retry if the backup operation is unsuccessful
  - j. **BackupRetryPause** – number of seconds to wait between backup attempts specified by BackupRetryAttempts
3. Save the changes to the configuration file, and then restart the content service.

### Example configuration

```
[Schedule]
Backup=TRUE
BackupCompression=TRUE
BackupCompressionLevel=6
NumberOfBackups=3
BackupDir0=\\bgSvr\Contentbackup1\day1\
BackupDir1=\\bgSvr\Contentbackup1\day2\
BackupDir2=\\bgSvr\Contentbackup1\day3\
BackupInterval=0
BackupMaintainDirStructure=true
BackupTime=00:20
BackupRetryAttempts=3
BackupRetryPause=60
```

### Exporting data

Data in the IDOL index can be manually exported by using the DREEXPORTIDX command. This command exports all of the IDOL IDX documents to the specified path (or IDOL's working directory if no path is specified). The files created in this directory are prefixed with AUTN-IDX-EXPORTTo initiate the export, use a web browser and navigate to:

**http://[IDOLSERVER]:[INDEXPORT]/DREEXPORTIDX?FileName=[BACKUP PATH]**

where IDOLSERVER is the hostname or IP address of the main IDOL server, INDEXPORT is the value configured for the IndexPort parameter in the main IDOL service's CFG file (default 9001), and [BACKUP PATH] is the path to use for the backed up files.

For example: **http://localhost:9001/DREEXPORTIDX?C:\Backup**

### Restoring a backup

An IDOL backup – created either from a [scheduled backup](#) or from the DREBACKUP command - see [Manual backup](#) – can be restored using the DREINITIAL command. To execute, use a web browser and navigate to

**http://[IDOLSERVER]:[INDEXPORT]/DREINITIAL?[BACKUP PATH]**

where IDOLSERVER is the hostname or IP address of the main IDOL server, INDEXPORT is the value configured for the IndexPort parameter in the main IDOL service's CFG file (default 9001), and [BACKUP PATH] is the path to use for the backed up files.

For example, **http://localhost:9001/DREINITIAL?C:\Backup**

**NOTE:** DREINITIAL will delete all data from an IDOL server's data index if no backup path is specified. Please use caution when using it.

**NOTE:** The above method assumes the IDOL server has only one content engine. For IDOL implementations with multiple content engines the 'HostDetails' parameter should be added to both the backup and restore commands to ensure each engine writes its backup to its own unique subfolder, and each engine restores its own backup from its own unique subfolder.

For example: **http://localhost:9001/DREINITIAL?C:\Backup&HostDetails=true**

For a more detailed discussion about restoring CM IDOL indexes, please refer to knowledgebase article *KM03471260 - Backing up and Restoring CM IDOL Content Indexes*.

## Importing data

The DREADD command allows you to manually add an IDX file to the IDOL index.

**http://[IDOLSERVER]:[INDEXPORT]/DREADD?[IDX PATH]**

where IDOLSERVER is the hostname or IP address of the main IDOL server, INDEXPORT is the value configured for the IndexPort parameter in the main IDOL service's CFG file (default 9001), and [IDX PATH] is the path to use for the IDX file to be imported.

For example, **http://localhost:9001/DREADD?C:\Backup\AUTN-IDX-EXPORT-12.04.2005-02.15.41-1.idx**

**NOTE:** DREADD can import \*.IDX files or \*.IDX.gz files

## Threads

Threads applies to the Content Manager IDOL service and sets the maximum number of parallel actions that can be handled by an IDOL Server and the number of action commands that the server can process in parallel at any one time. You can calculate the optimum number of threads as follows:

$(1 \times \text{number of cores}) + 1 \text{ spare thread}$ .

You can set the option in the configuration file TRIM IDOL Service.cfg, under the [server] section. The default is 4.

Example:

Threads=9

This option is not identical with the Content Manager property **Number of processing threads** that you can set when re-indexing in the **IDOL Record Reindex** dialog box, **Options** tab.

## Maximum queued transactions

Content Manager sends the transaction and command files to all content engines when it has processed a certain number of documents. That number of documents is determined by the value of **Maximum queued transactions** in the **Content Index Properties** dialog box. See [Configuring document content indexing](#).

## DistributeByReference

Plain non-mirror mode sends all commands to all engines and lets the child engines determine which documents to index. With reference-based indexing, DIH performs these calculations, reducing network traffic and load on the child servers.

In conjunction with the previous option, you can configure IDOL to send each content service only part of the transaction file. To do so, in the IDOL service configuration file **TRIM IDOL Service.cfg**, [server] section, add:

```
DistributeByReference=true
```

## DCI round-robin

An alternative to `DistributeByReference` is to use a round-robin approach to the processing of transaction files in IDOL. Rather than breaking up the transaction file and distributing it to the content engines, round-robin sends the transaction file to one engine at a time and cycles through the available content engines.

The advantage here is that the Distributed Index Handler (DIH) does not need to break up the transaction file or send it to all content engines.

See the Distributed Index Handler help file at <https://www.microfocus.com/documentation/idol/> for more information.

## MaxResults

The `MaxResults` setting in the Content Manager IDOL configuration files limits how many results IDOL will return to Content Manager in a single search. This is included as a performance consideration.

Change the value, for example to:

```
MaxResults=100000
```

By default, the value is 10000.

The upper limit for `MaxResults` is 1000000. Any IDOL search or combined IDOL / metadata search that includes a Content query that exceeds the limit set by `MaxResults` may return inconsistent results. Setting the `MaxResults` value to the maximum may impact on the overall search performance of your IDOL and hybrid IDOL/Metadata queries.

To change the figure, edit the **TRIM Content Service [number].cfg** configuration files, by default in **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM Content Service [number]** and the **TRIM IDOL Service.cfg** configuration file, by default in **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOL Service**.

In the **TRIM Content Service [number].cfg** files, you will find `MaxResults` under [Server].

In the file **TRIM IDOL Service.cfg**, you will find it under [DistributionSettings].

**NOTE:** If the `MaxResults` value is modified in the configuration files, you must also modify the **Maximum number of results in any search** value on the database's Content index properties -

Search Settings dialog in Content Manager Enterprise Studio. The defined MaxResults value in the configuration files and Enterprise Studio must be the same.

The value to choose for MaxResults depends on your environment and performance expectations:

- A lower value means that IDOL will return fewer results, but faster
- A higher value means that IDOL will return more results, but it will take longer, which in turn may cause the Content Manager Workgroup Server to time out before IDOL has responded

Therefore, if you are using a higher value for MaxResults, you should probably also increase the time before a query times out by using the Content Manager Enterprise Studio option IDOL Connection Timeout. See [Configuring document content indexing](#).

## MaxSyncDelay

MaxSyncDelay in the **TRIM IDOL Service.cfg** configuration file, [Server] section, and **TRIM Content Service [number].cfg**, enables you to specify the maximum time in seconds that can elapse before the IDOL Server is forced to store index data on disk.

This option is particularly useful when running a re-index, or for the initial indexing of the document store. As the content services databases grow, index data processing to disk takes more time.

This setting increases the time it takes for the information to be available for searching. However, during re-indexing, write performance is more important than read performance.

By default, the value is 120.

For re-indexing, it is recommended to set it to 360 to 720 seconds:

```
MaxSyncDelay=360
```

Set the value in all relevant configuration files.

MaxSyncDelay requires the DelayedSync option in the same configuration file to be set to true. For Content Manager OEM IDOL, the default is true.

When set to false, the IDOL server stores index data every time it finishes indexing an IDX or XML file, or when the specified value has been reached.

Enter true for the IDOL Server to only store index data on disk when the index cache is full, or when the specified value has been reached:

```
DelayedSync=true
```

## Timeout

The value for Timeout controls how long the main IDOL Server will wait for an action to be completed.

```
Timeout=60
```

## EngineTimeout

The value for EngineTimeout controls how long the main service will wait for a content engine to return its results.

```
EngineTimeout=60
```

## StateTokens

There are two settings in the TRIM IDOL Service.cfg file, section Distributed Settings that manage StateTokens in IDOL:

- StoredStateTokenExpiryPeriod
- StoredStateTokenLifetime

If you are running IDOL 10.4 or IDOL 10.5, the setting for StoredStateTokenLifetime is 0 by default and you should not change this setting. If you are running IDOL 10.8 or later, this restriction does not apply. Defaults are:

```
StoredStateTokenExpiryPeriod=300
```

```
StoredStateTokenLifetime=3600
```

## IndexCacheMaxSize

This setting in the TRIM Content Service [number].cfg configuration files determines the amount of index data to store in the cache before writing it to disk. Increasing the size of the cache can improve indexing performance. However, you should ensure that the host computer has sufficient memory available. The default value is 102400, which refers to kbytes.

To improve performance during reindexing, increase this value to 612000 to 1024000, which is 512MB to 1GB. This setting works with the option MaxSyncDelay, and the content service will write index data to disk only when either the delay timer is reached or the cache is full. This setting is in the [IndexCache] section:

```
IndexCacheMaxSize=102400
```

**NOTE:** You can dynamically grow or shrink the indexing cache after IDOL Server startup by using the DRERESIZEINDEXCACHE index command.

For example, to dynamically resize the index cache and increase the sync delay, use the following command in a browser on the IDOL Server:

```
http://localhost:9001/DRERESIZEINDEXCACHE?IndexCacheMaxSize=1024000&maxSyncDelay=360
```

## StopwordIndex

By default, IDOL does not index or search on a number of operator words, such as AND, NOT, TO, DO, NEAR, MORE, etc. You can enable indexing of these operator words by editing the IDOL configuration files. If you are changing this option after the index has already been created, you should perform a full Content Manager document content reindex.

In the Content Manager IDOL Service configuration file and each of the Content Manager Content Service configuration files, in the section [LanguageTypes], add:

```
StopwordIndex=1
```

## Paths

The default location for the content engines and log files is determined by the settings in the **TRIM IDOL Service.cfg** file and the respective Content Manager Content Service configuration file. It is **C:\Micro Focus Content Manager** by default.

To specify a different location for the Content Manager IDOL log files, insert the following [paths] statement into the file **TRIM IDOL Service.cfg**:

[Paths]

Incoming=D:\Micro Focus Content Manager\IDOL\TRIM IDOL Service\incoming

Archive=D:\Micro Focus Content Manager\IDOL\TRIM IDOL Service\archive

Failed=D:\Micro Focus Content Manager\IDOL\TRIM IDOL Service\failedTasks

This is particularly relevant for systems that have limited space on the C: drive. In the event of a problem, the failed directory receives all unprocessed transaction files and this may fill the drive.

## Database Read-only

You can set an Content Manager content engine to read-only by adding the following to the Content Manager Content Service configuration file in the respective content service in C:\Program Files\Micro Focus\Content Manager\IDOL:

[CM\_PM]

DatabaseReadOnly=TRUE

PM represents the database ID of the Content Manager dataset.

## Default Operator

The default IDOL operator is not strictly an OR, but similar to an OR with calculations that affect relevancy sorting.

In Content Manager 9.3, the default IDOL operator was update to be AND.

To change this, in the file TRIM Content Service [number].cfg, [Server] section, update DefaultQueryOperator=<Operator> to be OR, e.g. DefaultQueryOperator=OR

## IDOL component logs

Logs relevant to the content engines are by default in **C:\Micro Focus Content Manager\IDOL\TRIM Content Service [number]\logs**. There are **application.log**, **index.log**, and **query.log**.

Logs relevant to the Content Manager IDOL services including DIH and DAH logs are by default in **C:\Micro Focus Content Manager\IDOL\IDOL Service\logs**.

Location of the transaction files and working directory for opening compressed files during text extraction: **C:\Users\<services\_user>\AppData\Local\Micro Focus\Content Manager**

The Workgroup Server directory **C:\Micro Focus Content Manager\ServerLocalData\TRIM\Log** on the Workgroup Servers holds all relevant Workgroup Server, Event Server and document content indexing logs.

### Changing the log file locations

In the default configuration, the log files are located in **C:\Micro Focus Content Manager**. You can change this by changing the **TRIM Content Service 1.cfg** configuration files in **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM Content Service 1** folder. The path to the log files is in the section [Logging]:

```
//----- Logging Settings -----//  
[Logging]  
LogDirectory=C:\Micro Focus Content Manager\IDOL\TRIM Content Service 1\logs  
LogTime=TRUE  
LogOutputLogLevel=TRUE  
LogLevel=Normal
```

Change the directory to your desired location.

In the Content Manager IDOL Service configuration file at **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM IDOL Service.cfg**, there is a similar path:

```
//-----Logging-----//  
[Logging]  
LogDirectory=C:\Micro Focus Content Manager\IDOL\TRIM IDOL Service\logs  
LogTime=TRUE
```

Change this path, as well.

### Searching options

#### AugmentSeparators

By default, IDOL uses certain special characters when users run a search for document content, which effect the search result for terms that contain these characters. IDOL therefore does not index these characters when they appear in the document content. Therefore, it is good to be aware of them or, if required, modify the list that IDOL uses. The list is in the Content Manager IDOL Service Configuration file under Languages types [English]:

```
AugmentSeparators=---@“”‘’_....
```

The most significant here are the at symbol (@), comma (,), dash (-) and full stop (.). A search for a word or string that contains these characters, for example a search for TPS-1234 is treated as a search for TPS or 1234. Therefore, although Content Manager will return the documents with TPS-1234, it will also return other records containing only TPS or 1234. To search for TPS-1234 exactly, you would put the text within quotation marks, like "TPS-1234".

The same applies to an e-mail address: Searching for John.Smith@company.com would become a search for John or Smith or company or com. To find the e-mail address exactly, you would search for "John.Smith@company.com".



The same applies to the use of wildcard characters. A search for \*ps-1234 would return all records with a word ending in ps, for example soups, troops and groups. If you would rather find the records with ps-1234 exactly, you can put the text within quotation marks as in “\*ps-1234”.

To have IDOL index these special characters, remove them from the list of AugmentSeparators. If you are changing the setup of an existing index, this will require a full reindex.

To enable searching of entire email addresses - eg 'john.smith@company.com', both the '@' and '.' characters need to be included in the Diminish Separators section.

### **DiminishSeparators**

A list of characters to disallow as word separators, entered as UTF-8.

IDOL Server uses separators to determine where one word ends and the next one starts. For each language it supports, it has a list of appropriate characters that it automatically reads as separator, such as spaces, tabs, line-feeds, and carriage returns.

To ensure that IDOL Server does not read a specific character as separator, list it under DiminishSeparators (using un-encoded UTF-8).

Type: String

Default:

Required: No

Configuration Section: LanguageTypes or MyLanguage

Example: DiminishSeparators=.

**NOTE:** DiminishSeparators is not included in the default Content Manager IDOL configuration files. It must be added prior to re-indexing the content.

If you change this setting after you have indexed content into IDOL Server, you must re-index the content.

### **IDOL stop words**

Content Manager IDOL uses a default set of stop or noise words when building the document content index, which IDOL does not index when they appear in document content. These noise words do not relate to the set of noise words in Content Manager that are applied to metadata fields such as record titles and notes.

The IDOL noise words are stored in .dat files in the directory C:\Program Files\Micro Focus\Content Manager\IDOL\shared\langfiles. By default, they contain common English words.

After modifying the list of noise words, you should re-index the dataset for the new settings to take effect.

### **Setting up IDOL to index Additional Fields**

Content Manager IDOL can index the content of Additional Fields as metadata. To configure it to do so, you need to insert the Additional Field in the configuration files as a two-letter prefix that indicates the field type, followed by an underscore and the Additional Field name without any spaces.

For example, for an Additional Field of type integer called Batch number, it would be **TN\_BATCHNUMBER**.

The prefixes:

- TN\_ - represents the types integer, currency, decimal and large number
- TD\_ - represents the types date and date and time
- TS\_ - represents the types string, text and Location

Insert your TN\_ and TS\_ Additional Field code in the **TRIM Content Service.cfg** configuration file in the [SetIndexFields] block, parameter PropertyFieldCSVs, with \*/ in front of it.

Insert your TD\_ Additional Field code in the block [SetDateFields] instead.

After modifying the list of indexed Additional Fields here, you should reindex the dataset metadata for the fields to be added to the index.

## Monitoring IDOL

### Services

Using the Windows Services function, you can monitor whether the IDOL services are all running.

### Content index monitor

The Monitor option in Enterprise Studio will display the last 25 instructions sent to the IDOL server, and the current status of each of those instructions. This can be very helpful in determining if the IDOL server is currently operating correctly, particularly when running a large content re-index. It will also report when IDOL has finished processing all the indexing jobs.

See [Monitoring an IDOL Server](#) for more information.

### IDOL logs

If you suspect one of the IDOL components is not operating correctly, you can investigate the IDOL log files. See [IDOL component logs](#) for more information.

### IDOL logs

Each IDOL component generates its own set of log files. To differentiate the logs of different components, the component may prefix its name to the start of the log file name. For example, there will be a “dih\_application.log”, a “dah\_application.log” as well as an “application.log”.

### Application logs

Each IDOL component creates an application.log file. This log records information about the component and its general functioning.

When the component starts, its version number will be logged into the application.log, along with any working paths it will use. If there are any issues starting or running the component, or if the component shuts down for some reason, details will be recorded in this log file.

### **Index logs**

The index logs for each component will detail any requests to add or modify data that the component received. Some components may not record much information in the index.log, or even generate an index.log.

The most significant index log files are the dih\_index.log, and the index.log files from the content services.

### **Action logs**

The action logs record all of the instructions a component receives. Every query or action request should be logged in the main action.log. These requests can be traced through the IDOL system by examining the component action.log files.

### **Query logs**

The query logs record any searches the component is asked to do. These logs can be used to trace a search from the Content Manager client, and see the results that are found.

## **Restoring or changing IDOL license**

### **Changing IDOL license**

If the IDOL license for a component ever changes – e.g., if the OEM license has changed to include a new IDOL component or the site is migrating to an enterprise IDOL license – some manual cleanup needs to be done to force the component to pick up the new license.

1. Stop the target component's service.
2. Navigate to the component's installed location, e.g. **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM Content Service 1**.
3. Delete the **license** and **uid** directories

The component's service can now be restarted.

### **Converting to an Enterprise IDOL license**

It is possible to convert existing components from the Content Manager OEM IDOL installation to use an enterprise IDOL license without having to reinstall or reindex any data.

At the start of each component's cfg file, a [License] section needs to be added. This section contains the location of the DISH. When the component starts up, if this section is present, it will attempt to contact the DISH and check license.

In addition to the steps outlined in the [Changing IDOL License](#) section, the licensekey.dat file will also need to be deleted.

**Sample [License] Section**

```
[License]  
LicenseServerHost=192.168.1.1  
LicenseServerACIPort=20000  
LicenseServerTimeout=600000  
LicenseServerRetries=1
```

Upgrading to Enterprise IDOL should be done in consultation with Software Support, Professional Services or an Business Partner.

## Troubleshooting

### General troubleshooting

When you cannot use document content searching in your Content Manager client or cannot find any records by using document content searching, check that your system meets these requirements:

- You have created an IDOL document content index - to check, in Content Manager Enterprise Studio, expand Datasets. Right-click the dataset and on the Content Index menu, if the commands under Create IDOL are available, there is an IDOL content index. If there is no IDOL content index, use Create IDOL to create one. This is also what you need to do after you have upgraded Content Manager from version 7.1 (TRIM) or earlier. See [Configuring document content indexing and searching](#).
- You have content indexed your dataset – see [Performing content indexing](#).
- The IDOL Service is running - to check, on the computer running the Workgroup Server, make sure that the services Content Manager IDOL Service and at least one Content Manager Content Service have the status Started
- There is sufficient disk space - the Content Manager IDOL content engine requires a minimum of 1024MB of free space for the operation of the content service. You can change that figure in the IDOL configuration files. If there is insufficient space when starting the Content Manager IDOL Content Service, the message Error 1067: The process terminated unexpectedly may appear.

During a re-indexing process, the Content Manager error **DCI server is unreachable** message may appear.

The log file dih\_index.log may also report I/O errors

In normal day-to-day operations, document content indexing is handled through the event processor. If the document content service has insufficient space, the following appears in the event processor log file:

```
04:54:06 0 21 Error : 21 User: <account> - DCI server TRIMIDOL1 is unreachable
```

The IDOL application log in the content service\log directory may report that there is insufficient space.

To resolve the issue, increase the available space and restart the process.

- Content Manager IDOL database is not set to read-only - there may be circumstances that make the IDOL services start up without details of the current Content Manager IDOL database configuration. In that case, the IDOL service detects that a database existed that was not in the configuration file and sets the database to read-only by appending it to the end of the Content Service configuration file.

To resolve this issue, edit the TRIM Content Service.cfg configuration file and change the DatabaseReadOnly value to False:

```
[Database0]
```

Name=CM\_PM

DatabaseReadOnly=FALSE

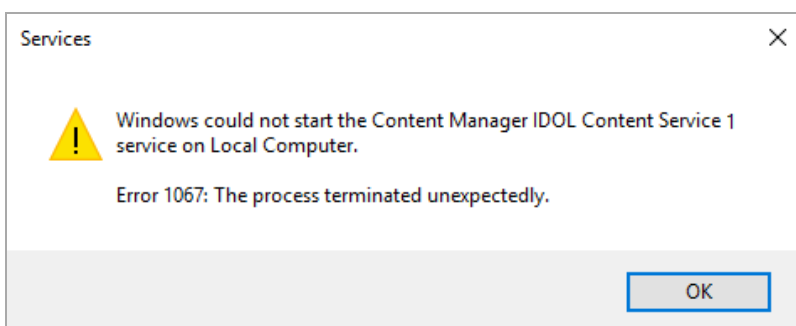
Internal=FALSE

InvertedAgent=FALSE

PM represents the Content Manager database ID of the dataset.

## IDOL content service does not start

For this example, we assume that Content Manager IDOL Content Service 1 is not starting when the computer starts or when manually started using the Windows Services function. The only error message is that the process terminated unexpectedly or similar:



### Investigate the IDOL Content Service.log

It is located with the Content Service 1.exe in the default location **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM Content Service 1\TRIM Content Service 1.log**.

Any errors related to the creation or initialization of log streams can be ignored – they will stop the service from starting. Any other errors mentioned should be investigated.

### Investigate the license.log

Located with the Content Service 1.exe, default location **C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM Content Service 1\license.log**.

This log should report that the license for the IDOL server is valid. An invalid or missing license will stop the component from starting. See [Restoring or changing IDOL license](#) for more information on fixing this issue.

### Investigate the application.log

Located in the Content Service 1 log directory, as set by the LogDirectory configuration parameter in the TRIM Content Service 1.cfg file, default location **C:\Micro Focus Content Manager\IDOL\TRIM Content Service 1\logs\application.log**.

This log will report the general status of the TRIM Content Service 1 service. It contains information about the component's version, any license restrictions and changes to working paths. It also contains information about the service starting, initialization and shutdown stages. If the component can start but is then forced to exit during its initialization, the reason should be recorded here.

## Troubleshooting license issues

Each component will create a license.log file when it starts. The log file will detail how the component is license (“Using license details from licensekey.dat” for an OEM license, “Successfully retrieved license from license server” for an enterprise license), as well as if the license was successfully loaded.

If the type of license the component is trying to use appears to be wrong, try:

- Stop the target component’s service
- Navigate to the component’s installed location (e.g. C:\Program Files\Micro Focus\Content Manager\IDOL\TRIM Content Service 1)
- Delete the “license” and “uid” directories

The component’s service can now be restarted.

## Document content search returns records without documents

Content Manager will index all records – both with and without documents – into IDOL. This allows us to be able to use IDOL for other types of searches. This was implemented in TRIM 7.3. When running a “document content” search using the Content Manager client, the search results should be restricted to only those Content Manager records with electronic documents.

If the IDOL index has been created using TRIM 7.3 or later (or any version of Records Manager/Content Manager), and the search is being run from an older client (e.g. TRIM 7.2), this restriction isn’t applied to the search results. So the user will see both document-records and metadata-only records in their search results.

## Content search does not find header or footer content

Text from the headers and footers of documents (such as \*.docx) was not being extracted by the text conversion process. These means that records cannot be found by searching for content in their headers or footers (CR 49652)

## IDOL searches are timing out

In some situations, IDOL-based searches may result in timeout errors for users. These timeouts can be caused by slow server hardware, network issues or query complexity.

## Adjusting timeouts

In some situations the best solution may be to increase the timeout values and allow all of the components longer to complete their work. As a general rule, the IDOL Connection Timeout (located in Enterprise Studio) should be equal to or slightly higher than the highest IDOL timeout value.

When adjusting these timeout values, ensure that the IDOL settings are located in the correct CFG files and in the correct configuration section (as marked by [section]). In some versions, these configuration values were located in the wrong place (and so were ignored).

### Timeout-related configuration settings

Location	Setting	Section	Comment
TRIM IDOL Service.cfg	Timeout	[Server]	The number of seconds before the requested action will timeout
TRIM IDOL Service.cfg	EngineTimeout	[DistributionSettings]	The number of seconds before the main IDOL service will timeout if a content engine has not responded
TRIM Content Service 1.cfg	QueryTimeoutInMilliseconds	[Server]	The number of milliseconds before the query will timeout if it has not finish processing
Enterprise Studio	IDOL Connection Timeout	Content Index Properties Dialog	The number of milliseconds before the workgroup server will timeout if IDOL has not responded

### Performance

Ideally, each IDOL component should be located on its own dedicated computer.

See [IDOL OEM Installation](#) for more information.

### Searching for special characters

Characters list “&” and “%” are special characters as far as IDOL is concerned. Since IDOL uses HTTP for its communication, these characters need to be handled differently.

IDOL also does not index “&” or “-“ characters, but rather treats them a separators. So a word like “ABC&DEF” will be indexed by IDOL as “ABC” and “DEF”.

Currently the Content Manager client cannot correctly deal with IDOL-based searches that contain ampersands or percent signs. As a work around, the “idol:” search method using the FieldText parameter can be used. See [Using the IDOL search method](#) for more info.

The FieldText methods that need to be used are the STRING{} and WILD{} methods. The STRING method will match whatever phrase is put in the {} brackets. The WILD{} method is used if you need to include a wildcard, but you will need to include a wildcard character at the beginning and end of your search phrase if you're not after an exact match.

To search for an ampersand using either method, it needs to be URL encoded - so instead of typing &, you need to type %26.



### Search structure

In the following examples, “searchquery” should be replaced with what the user is looking for, and “field” should be replaced with the field the user is trying to search against. Typically “field” will be DRECONTENT or TS\_TITLE. See [Content Manager Fields Available in IDOL](#) for more information.

To search for a special character as part of a word:

```
*&FieldText=STRING{searchquery}:field
```

To search for a special character as part of a word, including a wildcard character:

```
*&FieldText=WILD{*searchquery*}:field
```

### Example searches

To find the value "AB&DCE" in the document content:

```
*&FieldText=STRING{AB%26DCE}:DRECONTENT
```

To find the value "&DCE" in the document content:

```
*&FieldText=STRING{%26DCE}:DRECONTENT
```

To find the value "\*"&DCE" (matching anything that ended in "&DCE" such as "ABC&DCE") in the document content:

```
*&FieldText=WILD{*%26DCE*}:DRECONTENT
```

To find the value "\*"&DCE" in the record title:

```
*&FieldText=WILD{*%26DCE*}:TS_TITLE
```

### Common log messages

Log Message	Comments
Action failed - unencrypted communication is not allowed	Something attempted to communicate with OEM IDOL without the OEM key.  This message often appears when the OEM components send ACTION=GETSTATUS. In this case, this message can be ignored
Insufficient disk space. Exiting	Check the MinFreeSpaceMB setting in the component's cfg file. There needs to be at least this much available space on disk for this component to start
The drive on which [path] is mounted has only [space] MB disk space remaining. The folder requires at least [space] MB free space.	Check the MinFreeSpaceMB setting in the component's cfg file. There needs to be at least this much available space on disk for this component to start
Some of the configured components did not	There was a problem starting one or more of the

start	IDOL components. Investigate the logs to determine which component(s) were not starting
ACI Server did not load online help. File not found.	The online help files are not shipped with the OEM IDOL package. This message can be ignored
Failed to initialize log stream	The log of the indicated type could not be created.

## IDOL and ISYS

IDOL is a distributed and modular system. This means it can be easily scaled up to cope with large enterprise needs, or unneeded modules can be excluded for better results in environments with less available resources.

ISYS was the software used to provide document content searching in TRIM 7.1 and earlier. In TRIM 7.2, ISYS was replaced with IDOL and is no longer supported.

### Text conversion

ISYS used to manage the text-extraction of documents as well as managing the content index. Content Manager uses KeyView for the text-extraction step. This allows Content Manager to isolate the text-extraction process from the content indexing, which means if there is a problem with a document (such as the document is corrupt), it can be skipped without damaging the content index.

This also allows multiple documents to be processed simultaneously, which allows more resources to be utilized in the content indexing process, which translates to faster indexing speeds.

### Chains vs. content engines

ISYS spread its index across multiple “index chains”. IDOL uses one or more “content engines” to manage its index. This gives sites more options in terms of performance, availability and recovery from problems.

### Transition from ISYS

If you have been using document content indexing (DCI) in Content Manager before TRIM 7.2, your indexes would have been created by the ISYS indexing engine. Since TRIM 7.2, ISYS technology is not in use in Content Manager any longer and document indexing is powered by IDOL.

A Content Manager upgrade to the latest version does not delete the legacy ISYS files. However, after the upgrade, Content Manager Enterprise Studio does not recognize the ISYS indexes any more, which makes document content searching in Content Manager unavailable. A mixed ISYS/IDOL environment is not supported.

It is recommended for upgrading customers to set up IDOL document content indexing and then index the entire dataset to create the indexes. Follow the instructions in [Configuring document content indexing and searching](#) to start the process.

## Appendix A

# IDOL Configuration Settings

### Main IDOL Server configuration parameters

#### Service section

The Service section contains parameters that determine which machines are permitted to use and control a service.

For example:

```
[Service]
ServicePort=9102
ServiceStatusClients=*. *.*.*
ServiceControlClients=*. *.*.*
```

#### ServicePort

Type: Long

Default: 40010

Allowed range

- Minimum: 1
- Maximum: 65535

Recommended range

- Minimum: 1024
- Maximum: 65535

Required: yes

Description: The port on the host server on which the service listens for service status and control requests. To display the available service commands, click the online help's Service commands tab.

Example

```
ServicePort=40010
```

#### ServiceStatusClients

Type: String

Required: yes

Description

Addresses of clients that can query a service's status. The IP addresses or names of clients that can request status information from a service. These clients cannot control the service.

To enter multiple addresses, separate the individual addresses with commas (there must be no space before or after the comma).

Alternatively, you can use wildcards in the IP address. Enter for example 187.\*.\* to permit any machine whose IP address begins with 187 to access the service's status.

Example

```
ServiceStatusClients=localhost,127.0.0.1
```

### **ServiceControlClients**

Type: String

Required: yes

Description

IP addresses or names of clients that can issue service control commands to the service. To enter multiple addresses, separate the individual addresses with commas (there must be no space before or after the comma).

Alternatively, you can use wildcards in the IP address. Enter for example 187.\*.\* to permit any machine whose IP address begins with 187 to control the connector.

Example

```
ServiceControlClients=localhost,127.0.0.1
```

### **Server section**

The [Server] section contains general parameters.

For example:

```
Port=9100  
IndexPort=9101  
Threads=4  
AdminClients=*. *.*.*  
IndexClients=*. *.*.*  
MaxInputString=-1  
MaxQueryTerms=-1  
MaxResults=100000  
DelayedSync=true  
MaxSyncDelay=600  
KillDuplicates=*/DRREFERENCE  
MinFreeSpaceMB=1024
```

### **Port**

Type: Integer

Default: 9000

Allowed range

Minimum: 0

Maximum: 65536

Recommended range

Minimum: 1025

Maximum: 65536

Required: yes

Description

Action (ACI) port number. The port by which action commands are sent to IDOL server. You can display the available action commands by clicking on the online help's Action commands tab.

Example

Port=9000

### **IndexPort**

Type: Long

Default: 9001

Allowed range

Minimum: 0

Maximum: 65536

Recommended range

Minimum: 1025

Maximum: 65536

Description

Port number by which indexing commands are sent to IDOL server.

Example

IndexPort=59001

### **Threads**

Type: Long

Default: 10

Allowed range

Minimum: 1

Maximum: 32

Recommended range

Minimum: 4

Maximum: 32

Description

Maximum number of parallel actions. The number of action commands that the server can process in parallel at any one time. You can calculate the optimum number of threads as follows:

$(1 \times \text{Num CPU's}) + 1$  spare thread.

Example

Threads=2

### **QueryClients**

Type: String

Default: \*

Required: yes

Description

Addresses of hosts that can send action commands. Enter the IP address (or name) of one or more machines that are permitted to send action commands to the Port. If you want to enter multiple addresses, you must separate them with commas (there must be no space before or after a comma).

Example

QueryClients=10.1.1.\*,127.0.0.1

### **AdminClients**

Type: String

Default: localhost

Required: yes

Description

Addresses of administrative hosts. Enter the IP address (or name) of one or more machines that are permitted to send administrative commands to the Port. If you want to enter multiple addresses, you must separate them with commas (there must be no space before or after a comma).

Example

AdminClients=localhost,127.0.0.1

### **IndexClients**

Type: String

Description

Addresses of allowed indexing clients. One or more IP addresses or hostnames for machines that are permitted to index data into IDOL server (using the specified IndexPort).

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

#### Example

`IndexClients=10.1.1.10,127.0.0.1`

### **MaxInputString**

Type: Long

Default: 64000

Recommended range

Minimum: 100

Maximum: 100000

Description

Maximum characters in a request string. The maximum number of characters that can be sent in an HTTP request string to the server. Enter -1 if you do not want to restrict this number.

#### Example

`MaxInputString=100`

In this example, an HTTP request that is sent to the server can comprise a maximum of 100 characters.

### **MaxQueryTerms**

Type: Long

Default: 300

Recommended range

Minimum: 40

Maximum: 5000

Description :Maximum number of terms in a query. Allows you to specify the maximum number of words in the query text that can be used to query IDOL server (excluding stoplisted words, which are not used to query IDOL server). Any query text that exceeds the specified number is cut off (in case of Boolean or phrase strings this can result in query failure).

If an action's MaxQueryTerms parameter conflicts with the configuration file's MaxQueryTerms setting, the action's setting overrides the configuration file setting.

#### Example

`MaxQueryTerms=200`

### **MaxFileUploadSize**

Type: Long

Default: 10000000

Description

The maximum number of bytes to upload. Specify the maximum size (in bytes) of files that can upload to the server. To specify an unlimited upload size, set MaxFileUploadSize to -1.

**NOTE:** This setting applies to all actions sent using HTTP POST to the ACI port, even if they are not file uploads.

Example

```
MaxFileUploadSize=5000000
```

### Timeout

Type: Long

Default: 60

Description

The number of seconds to timeout for actions sent to IDOL Server.

Example

```
MaxFileUploadSize=90
```

### DateFormatCSVs

Type: String

Default: YYYY/MM/DD

Description

List of supported date formats. One or more date formats that you want IDOL server to be able to read from the files that it contains. If you want to enter multiple date formats, you must separate them with commas (there must be no space before or after a comma). You can enter one of the following:

AUTNDATE

a string that contains one or more of the following:

String	Description
YY	Year (2 digits). For example, 99, 00, 01 and so on.
YYYY	Year (4 digits). For example, 1999, 2000, 2001 and so on.
#YY+	Year (2 or 4 digits). If 2 digits are provided, then the YY format is used. If 4 digits are provided, then the YYYY format is used. For example, 07 is interpreted as 2007 AD and 1007 is interpreted as 1007 AD.
#Y	Year (1 to a maximum of 16 digits) and may be followed by "AD" or "BC". An apostrophe (') immediately before the year denotes a truncated year. For example, 2008, '97 (interpreted as 1997), 97 (interpreted as 97 AD), '08 (interpreted as 2008), 2008 AD and 200 BC. A truncated year with a BC identifier is invalid ('08 BC).



String	Description
<b>#FULLYEAR</b>	Year (1 to a maximum of 16 digits). For example 8, 98, 108, 2008, each of which is taken literally. The year is taken relative to the common EPOCH (0AD).
<b>#ADBC</b>	<p>Time Period. For example, AD, CD, BC, BCE or any predefined list of EPOCH indicators. Typically, the year specified using the above Year formats is interpreted as untruncated and relative to the EPOCH. For example, 84 AD is interpreted as 1984 AD and 84 BC is interpreted as 84 BC.</p> <p>The only exception to this is when #YY+ and #ADBC are both used. In this case, the format is interpreted as untruncated even if the year was set to truncated by #YY+. For example, 99 AD is interpreted as the year 99 AD.</p> <p>It is recommended you only use YY, YYYY or #FULLYEAR with #ADBC.</p>
<b>LONGMONTH</b>	A long month, for example, January, February and so on.
<b>SHORTMONTH</b>	A short month, for example, Jan, Feb and so on.
<b>MM</b>	Month (2 digits). For example, 01, 10, 12 and so on.
<b>M+</b>	Month (1 or 2 digits). For example, 1,2,3,10 and so on.
<b>DD</b>	Day (2 digits). For example, 01, 02, 03, 12, 23 and so on.
<b>D+</b>	Day (1 or 2 digits). For example, 1, 2, 12, 13, 31 and so on.
<b>LONGDAY</b>	2 digits with a postfix. For example, 1st, 2nd and so on.
<b>HH</b>	Hour (2 digits). For example, 01, 12, 13 and so on.
<b>H+</b>	Hour (1 or 2 digits).
<b>NN</b>	Minute (2 digits).
<b>N+</b>	Minute (1 or 2 digits).
<b>SS</b>	Second (2 digits).
<b>S+</b>	Second (1 or 2 digits).
<b>ZZZ</b>	Time Zone, for example, GMT, EST, PST, and so on.
<b>ZZZZZ</b>	Time Difference (1 to 9 digits). For example, +04 denotes 4 hours ahead of UTC. Other examples include +4, +04, +0400, +0400 MSD (the string MSD is ignored). A further example is +030, in this case the time differences is interpreted as 30 minutes.
<b>#PM</b>	AM or PM indicator (2 characters). For example, 2001/09/09 02:46:40 pm
<b>#S</b>	A space.

**NOTE: Notes:** Format strings are matched in the order in which they are listed. You should therefore put the format in order of length (starting with the longest). This prevents the matching of, for example, 19/10/2002 with DD/MM/YY (if this format has been listed before DD/MM/YYYY) and the document from being indexed with the date 19/10/20.

**NOTE:** To enter one or more formats, separate them with commas (there must be no space before or after a comma). To specify a format that contains a space, put the format in quotation marks.

**NOTE:** When documents containing fields with partial dates are indexed into IDOL, the document is assigned a metadata field specifying a full date. This will be the first day of the month if the day is not specified, and the first of January if neither the day or the month are specified. The partial date in the document must still be specified using DateFormatCSVs.

#### Example

DateFormatCSVs=D+ /SHORTMONTH/YYYY,DDMMYY

In this example, only document dates with the format D+ /SHORTMONTH/YYYY (for example, 2/Jan/2001) or DDMMYY (for example, 020101) can be indexed with a date.

DateFormatCSVs="D+SHORTMONTH YYYY", "Date: D+ LONGMONTH, YYYY"

In this example, only document dates with the format D+ SHORTMONTH YYYY (for example, 2 Jan 2001) or Date: D+ LONGMONTH, YYYY (for example, Date: 2 January, 2001) can be indexed with a date.

DateFormatCSVs=LONGMONTH/D+/#Y H+:N+:S+

In this example, only document dates with the format LONGMONTH/D+/#Y H+:N+:S+ (for example, September/9/2001 AD 2:46:40 or September/9/2001 2:46:40 or September/9/'01 2:46:40 or September/9/'01 AD 2:46:40) can be indexed with a date.

DateFormatCSVs=LONGMONTH/D+/#FULLYEAR #ADBC H+:N+:S+

In this example, only document dates with the format LONGMONTH/D+/#FULLYEAR #ADBC H+:N+:S+ (for example, September/9/8 AD 2:46:40 or September/9/2008 AD 2:46:40 or September/9/'01 2:46:40 or September/9/'01 AD 2:46:40) can be indexed with a date.

DateFormatCSVs=D+ LONGMONTH YYYY HH:NN:SS ZZZZ

In this example, only document dates with the format D+ LONGMONTH YYYY HH:NN:SS ZZZZ (for example, 17 May 2003 10:41:07 -0400) can be indexed with a date.

#### KillDuplicates

Type: String

Default: Reference

Description

Determines how IDOL server handles duplicate documents. This parameter allows you to prevent the same document or document content from being stored in IDOL server more than once. Enter one of the following options to determine how IDOL server handles duplicate text. Note that if you

postfix any of these options with =2, the KillDuplicates process is applied to all IDOL server databases (rather than just the database into which the current IDX or XML file is being indexed):

NONE

Duplicate documents are allowed in IDOL server and are not replaced or deleted.

REFERENCE

If the document being indexed has the same DREREFERENCE field value as a document that already exists in the IDOL server, IDOL server deletes the existing document and replaces it with the new document.

REFERENCEMATCHN

If the content of the document being indexed is more than N percent similar to the content of a document that already exists in the IDOL server database, IDOL server deletes the existing document and replaces it with the new document.

FieldName

If the document being indexed contains a FieldName Reference field with the same value as the FieldName Reference field in a document that already exists in IDOL server, IDOL server deletes the existing document and replaces it with the new document.

To specify multiple Reference fields, separate the fields with a plus sign (+) or a space. IDOL server deletes documents containing any of the specified fields with identical content.

**NOTE:** Fields are identified as Reference fields through field processes in the IDOL server configuration file. If you use a FieldName Reference field to eliminate duplicate documents, IDOL server automatically reads any fields listed alongside this field for the PropertyFieldCSVs parameter in the field process, and also uses these fields to eliminate duplicate documents. If you want to define multiple reference fields but do not want them all to be used for document elimination, you must set up multiple field processes (see the IDOL Server Administrator Guide).

If documents are indexed into IDOL server using a DREADD or DREADDATA command, and the command uses a KillDuplicates setting that conflicts with the KillDuplicates parameter in IDOL server's configuration file, the command's setting overrides the IDOL server configuration file's parameter.

Example

```
KillDuplicates=REFERENCE
```

In this example, if the document being indexed has the same DREREFERENCE as a document that already exists in IDOL server, IDOL server deletes the existing document and replaces it with the new document.

### DocumentDelimiterCSVs

Type: String

Default: \*/Document

Description

The fields in a file that mark the start and end of a document. This option ensures that documents are indexed individually. You must have only one document level in each XML schema.

If you index documents with a DREADD or DREADDDATA index action that has a DocumentDelimiters parameter that conflicts with the DocumentDelimiterCSVs configuration parameter, the index action parameter overrides the configuration parameter.

When identifying fields use the formats:

- `FieldName` to match root-level fields
- `*/FieldName` to match all fields except root-level
- `/Path/FieldName` to match fields that the specified path points to.

Example

```
DocumentDelimiterCSVs=*/DOCUMENT,*/SPEECH
```

In this example, the beginning and end of individual documents in a file is marked by opening and closing DOCUMENT and SPEECH tags (or fields if you are running IDOL Server in DRE3 mode).

### CombineIgnoreMissingValue

Type: Boolean

Default: False

Description

Determines how documents are combined in query results if they do not have the specified combine field. If the CombineIgnoreMissingValue parameter is set to true, documents in which the combine field (either FieldCheckType field or ReferenceType field) is absent are treated as distinct. When this parameter is set to false, the combined results include only one document with no such field.

Example

```
CombineIgnoreMissingValue=true
```

### MaxDocumentCount

Type: Long

Required: No

Description Set the maximum number of documents that can be indexed into an engine. When the limit is reached, the engine is marked <FULL>.

To prevent loss of data, engines that are marked as full continue to process index actions. The <FULL> marker is a guide for the component that sends index actions to the engine. For example, DIH does not send index actions to full engines when it is configured with RespectChildFullness set to true.

**NOTE:** For greater control of the stopping and starting of document indexing into an engine, use a combination of MaxDocumentCountUpper and MaxDocumentCountLower.

Example

```
MaxDocumentCount=20000
```

### **MaxDocumentCountUpper**

Type Long

Required No

Description

Set the maximum number of documents that can be indexed into an engine.

Use this setting in combination with `MaxDocumentCountLower` for greater control of the stopping and starting of document indexing into an engine. When the upper limit is reached, the engine is marked `<FULL>`. The `<FULL>` status is removed only when a sufficient number of documents are removed to reach `MaxDocumentCountLower`.

To prevent loss of data, engines that are marked as full continue to process index actions. The `<FULL>` marker is a guide for the component that sends index actions to the engine, for example DIH.

Example

```
MaxDocumentCountUpper=25000
```

### **MaxDocumentCountLower**

Type Long

Required No

Description

Set the lower limit for the number of documents that an engine can contain to allow documents to be indexed into the engine.

Use this setting in combination with `MaxDocumentCountUpper` for greater control of the stopping and starting of document indexing into an engine. When the upper limit is reached, the engine is marked `<FULL>`. The `<FULL>` status is removed only when a sufficient number of documents are removed to reach `MaxDocumentCountLower`.

To prevent loss of data, engines that are marked as full continue to process index actions. The `<FULL>` marker is a guide for the component that sends index actions to the engine, for example DIH.

Example

```
MaxDocumentCountLower=20000
```

### **Distribution Settings section**

Use the `[DistributionSettings]` section when you are using a unified configuration with an IDOL component. There are no specific configuration options that belong to this section. Enter the options that would appear in the `[Server]` section of the `<IDOL Component>.cfg` file in a standalone configuration.

For example, for a unified configuration with the DIH:

```
[DistributionSettings]  
Port=16000
```

```
DIHPort=16001  
IndexClients=*. *.*.*.*  
MirrorMode=true
```

### **MirrorMode**

Type: Boolean

Default: TRUE

Description

Run the DIH in Mirror mode.

In Mirror mode the DIH distributes all the index data it receives to all the IDOL servers that it is connected to. The IDOL servers are exact copies of each other and must all be configured in the same way.

Run the DIH in Mirror mode to ensure uninterrupted service if one of the IDOL servers fails. While one IDOL server is inoperable, data continues to be indexed into its identical copies which at the same time are still available to return data for queries.

The DIH periodically checks if all the IDOL servers that it connects to are operating. If an IDOL server breaks down, the DIH queues the data that this IDOL server normally receives, and after the IDOL server starts operating again, it indexes the queued data into it.

Enter false to run the DIH in Non-Mirror mode.

In Non-Mirror mode the DIH distributes the index data it receives evenly across the IDOL servers that it is connected to. For example, if the DIH is connected to 4 IDOL servers, it indexes approximately one quarter of the data into each one of the IDOL servers (individual documents are not split up).

Run the DIH in Non-mirror mode if the amount of data that you want to index is too large for a single IDOL server. If the IDOL servers that the DIH indexes into are situated on different machines, the index process requires less time.

The DIH periodically checks if all the IDOL servers that it connects to are operating. If an IDOL server breaks down, the DIH queues the data that this IDOL server normally receives, and after the IDOL server starts operating again, it indexes the queued data into it.

Example: MirrorMode=FALSE

### **DistributionMethod**

Type: Long

Default: 1

Description

The method for distributing actions. Set DistributionMethod to 0 for failover, and 1 for load-balancing. The DistributionMethod determines how the DAH distributes actions if:

- it is run in mirror mode
- it is run in non-mirror mode, using Distributor VDBs databases

Type one of the following to determine which distribution method the DAH uses:

1. The DAH uses the Failover distribution method. Depending on the mode the DAH is run in, this method distributes actions as follows:
  - Failover in Mirror mode. The DAH forwards incoming actions to the first IDOL server that is listed in the Distributed Action Handler configuration file [DistributedEngineN] section. If this IDOL server stops responding for any reason (for example, a hardware failure or network outage), the DAH marks it as down and switches to the next IDOL server (the second one listed in the [DistributedEngineN] section) and so on, seamlessly continuing to service client actions.
  - Failover in Non-mirror mode. The Distributor VDBs forward incoming actions to the first IDOL server database that is listed for the virtual database MapsTo parameter. If this IDOL server database stops responding for any reason (for example, a hardware failure or network outage), the DAH marks it as down and switches to the next IDOL server database (the second one listed for the MapsTo parameter) and so on, seamlessly continuing to service client actions.

The DAH periodically checks IDOL servers that are down. If it finds that an IDOL server has come online again (for example, because the hardware has been rebooted or the network connection repaired), it adds it to the active list of IDOL servers. This IDOL server database takes over as the sole destination for actions (as it takes precedence over the IDOL server database that is currently serving results).

2. The DAH uses the Load balancing distribution method. Depending on the mode the DAH is run in, this method distributes actions as follows:
  - Load balancing in Mirror mode. The DAH assigns each incoming action to just one of the IDOL servers that it connects to (using a cumulative predictive algorithm that is designed to spread the action load efficiently). When this IDOL server responds with a result, the DAH forwards it to the client software.

If an IDOL server stops responding for any reason (for example, a hardware failure or network outage), the DAH marks it as down and assigns incoming actions only to the IDOL servers that are running, saving the time it takes to attempt to communicate with the failed IDOL server.

- Load balancing in Non-mirror mode. The Distributor VDBs assign each incoming action to just one of the IDOL server databases that they map to (using a cumulative predictive algorithm that is designed to spread the action load efficiently). When this IDOL server database responds with a result, the DAH forwards it to the client software.

If an IDOL server stops responding for any reason (for example, a hardware failure or network outage), the DAH marks it as down and assigns incoming actions only to the IDOL server databases that are running, saving the time it takes to attempt to communicate with the failed IDOL server database.

The DAH periodically checks IDOL servers that are down. If it finds that an IDOL server has come online again (for example, because the hardware has been rebooted or the network connection repaired), it adds it to the active list of IDOL servers. This IDOL server database becomes a valid choice for actions again.

3. The DAH uses the distribute by reference distribution method. You can use this method only when distributing View, GetLink, and ViewGetDocInfo actions in mirror mode. The DAH distributes View, GetLink and ViewGetDocInfo actions between view servers based on the reference of the requested document. All actions that request the same document are then

always sent to the same View server.

- When View server first receives a request to view a document, it caches any embedded images in the result document. When distributing between View servers, use distribute by reference mode to ensure that each document is cached in only one View server, and it uses the cached images in subsequent requests for the same document.

Example

```
DistributionMethod=300
```

### **MaxResults**

Type: Long

Default: 10000

Recommended range

Minimum: 0

Maximum: 100000

Description

Maximum number of query results. Specify the maximum number of result documents that can be returned by a query.

The MaxResults action parameter specifies the number of results you want returned for a query from the total number of results returned.

**NOTE:** The maximum value for MaxResults is only limited by the amount of allocated RAM.

Example

```
MaxResults=300
```

### **IndexQueueSize**

Type: Integer

Default: 4096

Description

The maximum number of index actions that the index queue can contain.

Example

```
IndexQueueSize=5000
```

### **BlockIfFull**

Type: Boolean

Default: false

Description



Set to true if you want Indextasks to block and not accept commands when the incoming queue becomes full, instead of overwriting the oldest command that has not been processed.

Example

```
BlockIfFull=true
```

### **EngineTimeout**

Type: Long

Default: 60

Allowed range

Minimum: 2

Maximum: 86400

Recommended range

Minimum: 2

Maximum: 600

Description

The number of seconds after which the DAH connection to a child server times out if the DAH does not receive a reply.

Example

```
EngineTimeout=60
```

### **StateTokensDirectoryUsesHash**

Type: Boolean

Default: False

Description

Set this parameter to true to use a hashed directory structure (64 folders, each containing 64 folders) for the directory specified by StateTokensDirectory to store the stored state memory-mapped files. This option prevents a single directory from containing too many files, which can improve performance for file access. By default, all files are stored in the single directory. Example

```
StateTokensDirectory=TRUE
```

```
StoredStateTokenExpiryPeriod
```

Type: Long

Default: 60

Allowed range

Minimum: 1

Maximum: 86400

Recommended range

Minimum: 10

Maximum: 3600

#### Description

The number of seconds between checks for expired stored state token. If any token is found to have exceeded its lifetime, it is expired.

#### Example

```
StoredStateTokenExpiryPeriod=60
```

### **StoredStateTokenLifetime**

Type: Long

Default: 3600

#### Allowed range

Minimum: 0

Maximum: 320000000

#### Recommended range

Minimum: 60

Maximum: 86400

#### Description

The number of seconds until stored state tokens expire.

If you do not want to limit the stored state lifetime, set `StoredStateTokenLifetime` to 0. This disables automatic cleanup.

#### Example

```
StoredStateTokenLifetime=86400
```

### **CollectChildFullness**

Type: Boolean

Default: False

Required: No

#### Description

Instructs the DIH to send `GetStatus` actions to its child servers to collect fullness information. Use this setting when the DIH is running in non-mirror mode to distribute documents dynamically across a bank of child servers based on user-specified document limits. Use the `GetChildStatusMode` parameter to determine when the DIH sends the `GetStatus` actions.

If `CollectChildFullness` is set to true, the DIH aggregates the fullness information from its child servers, and reports the results in the `<FULL>` and `<FULL_RATIO>` tags when it receives a `GetStatus` action.

If the DIH has no information from its child servers (For example, when `CollectChildFullness` is set to `false`) then the DIH reports that it is full.

Use the `MaxDocumentCount` option in the `[Server]` section of the IDOL server configuration file to set the number of documents that the child IDOL servers accept before returning `<FULL>`.

Example

```
CollectChildFullness=true
```

### **RespectChildFullness**

Type: Boolean

Default: False

Required: No

Description

Instructs the DIH to stop indexing documents into a child server that reports that it is full. Use this setting when the DIH is running in non-mirror mode to distribute documents dynamically across a bank of child servers based on user-specified document limits.

Setting `RespectChildFullness` to `true`, implicitly sets `CollectChildFullness` to `true`.

Use the `MaxDocumentCount` option in the `[Server]` section of the IDOL Server configuration file to set the number of documents that the child IDOL Servers accept before returning `<FULL>`.

Example

```
RespectChildFullness=true
```

### **SimpleCombinatorMode**

Type: Boolean

Default: False

Description

Set `SimpleCombinatorMode` to `true` to use IDOL Server databases, rather than virtual databases (VDBs) to distribute and merge results. In `SimpleCombinatorMode`, you cannot configure VDBs, and DAH sends all queries to all child servers. DAH forwards the `DatabaseMatch` to child servers, and merges the responses from all child servers.

If an IDOL server stops responding for any reason (for example, a hardware failure or network outage), the DAH marks it as down and only assigns incoming actions to the databases in IDOL server installations that are running. The DAH periodically checks IDOL servers that are down. If it finds that an IDOL server has come online again (for example, because the hardware has been rebooted or the network connection repaired), it adds it back into the list of active IDOL servers and resumes sending actions to that IDOL server.

**NOTE:** This configuration parameter only applies in non-mirror mode.

Example

```
SimpleCombinatorMode=true
```

### VirtualDatabases

Type: Long

Allowed Range:

Minimum: 1

Maximum: 100000

Recommended Range:

Minimum: 1

Maximum: 10000

Required: No

Configuration Section: Server

Description

The number of virtual databases to set up. For each virtual database, you must set up a corresponding [VDBN] section in the Distributed Action Handler configuration file.

Example

```
VirtualDatabases=2
```

### ShowVDBInfo

Type: Boolean

Default: True

Required: No

Configuration Section: Server

Description

Whether to include virtual database information.

Set ShowVDBInfo to true to return details about the configured Distributed Action Handler virtual databases in the database information that the GetStatus action returns.

Set it to false to return only IDOL server database details.

Example:

```
ShowVDBInfo=false
```

### Virtual Database Settings section

**NOTE:** Make sure that the SimpleCombinatorMode parameter in Distribution Settings configuration parameter is set to False.

Use the [VDBM] section when you are using a virtual database configuration with an IDOL component. There are no specific configuration options that belong to this section. Enter the options

that would appear in the [Server] section of the <IDOL Component>.cfg file in a standalone configuration.

For example:

```
DbName=CM_8P
Type=Distributor
MapsTo=0:CM_8P
```

### **DbName**

Type: String

Required: Yes

Configuration Section: VDBN

Description

The name of the virtual database.

**NOTE:** If you use the name of a virtual database in the DatabaseMatch parameter of an IDOL server action DatabaseMatch parameter, the DAH returns results from all databases that the virtual database maps to.

Example

```
DbName=News
```

### **Type**

Type: String

Default: Combinator

Required: Yes

Configuration Section: VDBN

Description

The type of virtual database. This value is either Distributor or Combinator.

**Distributor.** The virtual database forwards client actions to one of the IDOL server databases that it maps to (these databases must be identical copies of each other and contain the same data). The way it forwards the query is determined by the DistributionMethod.

**Combinator.** The virtual database forwards client actions to all the IDOL server databases that it maps to. It collates and sorts the results before it returns them to the clients.

Example

```
Type=Distributor
```

### **MapsTo**

Type: String

Required: Yes

## Configuration Section: VDBN

### Description

The databases to map the virtual database to. The databases you specify can be IDOL server databases or other virtual databases that you have set up in the Distributed Action Handler.

Separate multiple databases with commas. There must be no space before or after a comma.

### To specify an IDOL server database

Use the format `IDOLServerNumber:DatabaseName`.

Where,

**IDOLServerNumber** - The number of the IDOL server installation listed in the [Engines] section of the configuration file.

**DatabaseName** - The name of the IDOL server database to which you want to map the virtual database.

For example, `MapsTo=0:Today`.

### To specify a Distributed Action Handler virtual database

Type the `DbName` of the virtual database that you want to map to.

For example, `MapsTo=Accounts`.

**NOTE:** You must ensure the [VDBN] section to which you map the virtual database has a lower number than the current [VDBN] section. That is, the virtual database in [VDB2] can map to the virtual database in [VDB1], but it is not possible for the virtual database in [VDB1] to map to the virtual database in [VDB2]. This ensures that virtual database mappings are not circular.

### Example

```
MapsTo=0:Today,2:News,Accounts
```

In this example, the virtual database is mapped to the Today database in IDOL Server 0, the News database in IDOL Server 2, and the Accounts virtual database.

## DAHEngines section

When you install and operate the DAH as integrated with IDOL, you can configure child servers for DAH and DIH separately. This process allows you increased flexibility when querying and indexing content in remote servers. For example, you can configure different child servers to perform actions or index actions exclusively.

Use the [DAHEngines] section to set the total number of IDOL Servers to which the DAH sends actions.

You must create a [DAHEngineN] subsection for each of the child servers that the DAH sends actions to, and specify the configuration details for that server.

For example:

```
[DAHEngines]  
Number=2
```

## DIHEngines section

When you install and operate the DIH as integrated with IDOL, you can configure child servers for DAH and DIH separately. This process allows you increased flexibility when querying and indexing content in remote servers. For example, you can configure different child servers to perform indexing or actions exclusively.

Use the [DIHEngines] section to set the total number of IDOL Servers that the DIH sends index actions to.

You must create a [DIHEngineN] subsection for each of the child servers that the DIH sends index actions to. In these sections you specify the server configuration details.

For example:

```
[DIHEngines]  
Number=2
```

## Security section

The [Security] section lists the security types you are using, and contains a subsection for each of these security types. Within each subsection, you can specify the settings you want to apply to each security type.

For example:

```
[Security]  
SecurityInfoKeys=123,144,564,231
```

## SecurityInfoKeys

Type: String

Required: yes

Description

The security keys The IDOL server uses to encrypt and decrypt the security information that it uses. Set SecurityInfoKeys to four signed 32-bit integers (between 0 and 2147483647), separated with commas (there must be no space before or after the commas).

To use AES encryption, you can set SecurityInfoKeys to the filename and path of the AES keyfile. IDOL Server then uses this keyfile to encrypt and decrypt the SecurityInfo tokens.

You must always change SecurityInfoKeys from the value in the default configuration file when you install IDOL server.

**NOTE:** You must set SecurityInfoKeys to the same value as you use for all other components that use a four-number security identifier or AES keyfile.

**NOTE:** In cases where you use a DAH within an IDOL server (unified configuration), if you use the user security encryption keys, then you must configure SecurityInfoKeys in both the [Security] and [DistributionSettings] sections.

Example

```
SecurityInfoKeys=123456789,144135468,56443234,2000111222
```

## Logging section

The Logging section lists the logging streams used to create separate log files for different log message types (such as, query, index, and application). It also contains a subsection for each of the listed logging streams in which you can configure the parameters that determine how each stream is logged.

For example:

```
[Logging]
LogDirectory=${Folder}\logs
LogTime=TRUE
LogOutputLogLevel=TRUE
LogLevel=Normal
LogExpireAction=Compress
LogOldAction=delete
LogMaxOldFiles=10
MaxLogSizeKbs=20480
```

```
0=ApplicationLogStream
1=QueryLogStream
2=IndexLogStream
3=ActionLogStream
```

```
[ApplicationLogStream]
LogFile=application.log
LogTypeCSVs=application
```

```
[QueryLogStream]
LogFile=query.log
LogTypeCSVs=query
```

```
[IndexLogStream]
LogFile=index.log
LogTypeCSVs=index
```

**NOTE:** All queries are truncated to 4000 characters in query logs.

## LogDirectory

Type: String

Default: ./logs

Description

Path to log directory. Enter the directory in which you want the application to store the log files it creates.

Example



```
LogDirectory=./logs
```

### **LogTime**

Type: Boolean

Default: true

Description

Display time with each log entry. Enable this parameter to display the current time next to each log entry in the log file.

Example

```
LogTime=false
```

### **LogOutputLogLevel**

Type: Boolean

Default: false

Description

Determines whether the log level is included in the log. Enable this parameter to include the log level of a message in the log entry.

Example

```
LogLevel=Always  
LogOutputLogLevel=TRUE
```

In this example, Always is added to the log message:

```
21/12/2006 12:34:56 [10] Always: ACI Server attached to port 1622
```

### **LogExpireAction**

Type: String

Default: Datestamp

Description

Determines how log files are handled when they exceed the maximum size. Enter one of the following to determine how log files are handled when they exceed the MaxLogSizeKBs size:

Compress

The log file's name is appended with a time stamp, compressed and saved in the log directory. By default, this is a Zip file. Use the LogCompressionMode parameter to specify another compression format.

Consecutive

The log file's name is appended with a number and saved in the log directory. When the next log file reaches its LogMaxSizeKBs size, it is appended with the next consecutive number.

Datestamp

The log file's name is appended with a time stamp and saved in the log directory.

#### Previous

The log file's name is appended with .previous and saved in the log directory. Every time a log file reaches its LogMaxSizeKBs size, it is given the same postfix so it overwrites the old log file.

#### Day

Only one log file is created per day and is appended with the current time stamp. Log files are archived once they reach the LogMaxSizeKBs size.

**NOTE:** The LogMaxSizeKBs parameter takes precedence over the LogExpireAction parameter. Therefore, if you set LogExpireAction to Day, and the value for LogMaxSizeKBs results in more than one log file, multiple log files will be generated per day.

#### Example

```
LogExpireAction=Compress
```

### LogOldAction

Type: String

Default: Delete

#### Description

Determines how log files are handled when the maximum number of log files is exceeded. Enter one of the following to determine how log files are handled when the LogDirectory has reached the maximum number of log files, as determined by the LogMaxOldFiles parameter:

#### Delete

The log files are deleted.

#### Move

The log files are moved to the specified LogArchiveDirectory.

#### Example

```
LogOldAction=Move
```

### LogMaxOldFiles

Type: Long

Default: -1 (unlimited)

#### Description

Maximum number of log files in the log directory. The maximum number of log files the specified LogDirectory can store before the application executes the specified LogOldAction. If you do not want to restrict how many log files the LogDirectory can store, enter -1.

#### Example

```
LogMaxOldFiles=1000
```

### **LogMaxSizeKBs**

Type: Long

Default: 1024

Description

Maximum log file size (in KB). If you do not want to restrict the log file size, enter -1. The LogExpireAction parameter determines how a log file is handled once it has reached its maximum size.

Example

LogMaxSizeKBs=1000

### **LogFile**

Type: String

Required: yes

Description

Name of the log file. The name of the log file the application creates in the specified LogDirectory.

Example

LogFile=query.log

### **LogTypeCSVs**

Type: String

Required: yes

Description

List of message types to log. Enter one or more of the following message types to specify the type of messages written to the associated log file. If you want to enter multiple message types, separate them with commas (there must be no space before or after a comma):

All Components.

Action

Logs actions and related messages.

Application

Logs application-related occurrences.

IDOL Server.

Agent

Logs agent actions and related messages.

Category

Logs category actions and related messages.

#### Cluster

Logs cluster actions and related messages.

#### Community

Logs community actions and related messages.

#### ExtendedIndex

Logs index actions as well as index commands that are issued after IDOL server has routed incoming data through other processes.

#### Index

Logs index actions and related messages.

#### Mailer

Logs mailer actions and related messages.

#### Profile

Logs profile actions and related messages.

#### Query

Logs query actions and related messages.

#### QueryTerms

Logs each query term, after stemming, conversion to UTF8, capitalization and punctuation removal. This is mainly used by the DiSH server for statistical reports.

#### Role

Logs role actions and related messages.

#### Schedule

Logs schedule actions and related messages.

#### Taxonomy

Logs taxonomy actions and related messages.

#### User

Logs user actions and related messages.

#### User\_Audit

Logs USERADD and USERDELETE actions and related messages.

#### UserTerm

Logs terms that IDOL server uses to form a user's agents and profiles.

#### DIH.

#### Index

Logs index actions and related messages.

#### Query

Logs query actions and related messages.

DAH.

Security

Logs security action results.

Connectors.

FailureList

Logs details of files that were not imported successfully.

Import

Logs import actions and related messages.

Index

Logs index actions and related messages.

Spider

Logs spider actions and related messages. (HTTP Connector only)

CFS.

Indexer

Logs the status of indexing into IDOL.

Import

Logs import actions and related messages.

CFS Connectors.

Collect

Logs document collection for Legal Hold application.

Delete

Logs the deletion of documents from the repository.

Hold

Logs details of documents that are put on hold in Legal Hold applications.

Identifiers

Logs details of requests for document lists from repositories.

Synchronize

Logs data synchronization when indexing into IDOL.

Update

Logs details of documents whose metadata is updated in the repository.

View

Logs details of documents that are viewed from the repository.

## Example

```
LogTypeCSVs=Application,Index
```

## Language Types section

The [LanguageTypes] section lists the languages you want to use and defines some generic parameters.

For example:

```
[LanguageTypes]  
DefaultLanguageType=englishASCII  
DefaultEncoding=UTF8  
LanguageDirectory=C:\IDOLserver\IDOL\langfiles  
GenericTransliteration=true  
StopWordIndex=1  
0=english
```

## LangDetectUTF8

Type: Boolean

Default: false

Description

Set to true if you want files that contain 7-bit ASCII to be detected as UTF-8, rather than ISO-8859-1 (ASCII).

When Automatic Language Detection is used to detect the language of documents that are being indexed, it uses the Index fields by default. If these fields contain only 7-bit ASCII, the document will be detected as ASCII. If additional fields in the document contain UTF-8, these might be converted incorrectly. If you know that documents are generally in UTF-8, then set LangDetectUTF8 to true so that these documents will be detected as UTF-8.

## Example

```
LangDetectUTF8=true
```

## DefaultLanguageType

Type: String

Description

Default language type used for documents of unspecified language.

This parameter specifies the language type (language and encoding) IDOL server assumes documents to be when it cannot read the language type of a document from a specified field, the LanguageType parameter is not specified in an action command, or automatic language detection is not enabled.

The parameter also specifies the default for the LanguageType parameter in action commands such as Query and Summarize.

If you do not set a `DefaultLanguageType` or specify the language type in the action command, IDOL server uses the first language type defined in the `[MyLanguage]` section.

If you do not set the `DefaultEncoding` configuration parameter or the `OutputEncoding` parameter in an action command, IDOL server uses the encoding specified in the `LanguageType` parameter sent with the query, or in the `DefaultLanguageType` configuration parameter if the `LanguageType` parameter is not sent with the query.

**NOTE:** Documents whose encoding is identified but not their language can be placed into a `[General]` language category, if it is defined, and if Automatic Language Detection is activated.

Example

```
DefaultLanguageType=englishASCII
```

### DefaultEncoding

Type: String

Description

The default encoding used for the `OutputEncoding` parameter in action commands. The `OutputEncoding` parameter specifies the encoding to which you want the output resulting from actions commands to be converted before the output is returned.

If you do not set `DefaultEncoding` or `OutputEncoding`, IDOL server uses the encoding specified in the `LanguageType` parameter sent with the query, or in the `DefaultLanguageType` configuration parameter if the `LanguageType` parameter is not sent with the query.

Example

```
DefaultEncoding=UTF8
```

### LanguageDirectory

Type: String

Required: yes

Description

Path to the language-files directory. The directory in which the files are stored that the languages you are using require.

Example

```
LanguageDirectory=./langfiles
```

### GenericStemming

Type: Boolean

Default: false

Description

Use language-sensitive generic stemming algorithm. Normally, queries to which you have added `&AnyLanguage=true` only return multi-language documents if they contain terms that stem the same way as terms in your query. Since different stemming algorithms are applied to different languages, only terms that are common to multiple languages tend to stem the same way (for example, many international words like "computer" are stemmed differently in certain languages).

If you want to relax the way queries are matched against multiple languages, you can set `GenericStemming` to true. This instructs IDOL server to use a generic stemming algorithm for all your language types, which is automatically modified according to which languages you have configured.

You can use the `Stemming` configuration parameter for individual languages to specify whether that language's stemming rules should be used. For example, if you want to use only the English stemming rules for your generic stemming, you can set `Stemming` to true for English, and set `Stemming` to false for all other languages. This does not apply if you have set `GenericFullStemming` to true.

**NOTE:** If you want to configure an additional language type with `Stemming` set to true after you have indexed content into IDOL server, you must re-index all your content. If you add a new language with `Stemming` set to false, or if you have set `GenericFullStemming` to true, you do not need to re-index.

#### Example

```
GenericStemming=true
```

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.

### GenericTransliteration

Type: Boolean

#### Description

Enter true to apply transliteration (accent removal) to all languages and use a single generic transliteration scheme. Transliteration is applied to Japanese, Chinese, Greek, Spanish and Portuguese by default, and is optional for Western European, German, Scandinavian, and Russian.

Transliteration is the process of converting accented characters such as øù into equivalent non-accented characters. This is useful in environments where accented keyboards are not available.

You can alter transliteration settings for individual languages using the `Transliteration` configuration parameter in the individual language configuration sections.

#### Example

```
GenericTransliteration=true
```

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.



## **IDOL content service configuration parameters**

### **Service section**

The Service section contains parameters that determine which machines are permitted to use and control a service.

For example:

```
[Service]
ServicePort=9102
ServiceStatusClients=*. *.*.*
ServiceControlClients=*. *.*.*
```

### **ServicePort**

Type: Long

Default: 40010

Allowed range

Minimum: 1

Maximum: 65535

Recommended range

Minimum: 1024

Maximum: 65535

Required: yes

Description

The port on the host server on which the service listens for service status and control requests. To display the available service commands, click the online help's Service commands tab.

Example

```
ServicePort=40010
```

### **ServiceStatusClients**

Type: String

Required: yes

Description

Addresses of clients that can query a service's status. The IP addresses or names of clients that can request status information from a service. These clients cannot control the service.

To enter multiple addresses, separate the individual addresses with commas (there must be no space before or after the comma).

Alternatively, you can use wildcards in the IP address. Enter for example 187.\*.\* to permit any machine whose IP address begins with 187 to access the service's status.

Example

```
ServiceStatusClients=localhost,127.0.0.1
```

### **ServiceControlClients**

Type: String

Required: yes

Description

IP addresses or names of clients that can issue service control commands to the service. To enter multiple addresses, separate the individual addresses with commas (there must be no space before or after the comma).

Alternatively, you can use wildcards in the IP address. Enter for example 187.\*.\* to permit any machine whose IP address begins with 187 to control the connector.

Example

```
ServiceControlClients=localhost,127.0.0.1
```

### **Server section**

The [Server] section contains general parameters.

For example:

```
Port=9100
IndexPort=9101
Threads=4
AdminClients=*. *.*.*
IndexClients=*. *.*.*
MaxInputString=-1
MaxQueryTerms=-1
MaxResults=100000
DelayedSync=true
MaxSyncDelay=600
KillDuplicates=*/DREREFERENCE
MinFreeSpaceMB=1024
```

### **Port**

Type: Integer

Default: 9000

Allowed range

Minimum: 0

Maximum: 65536

Recommended range

Minimum: 1025

Maximum: 65536

Required: yes

Description

Action (ACI) port number. The port by which action commands are sent to IDOL server. You can display the available action commands by clicking on the online help's Action commands tab.

Example

Port=9000

### **IndexPort**

Type: Long

Default: 9001

Allowed range

Minimum: 0

Maximum: 65536

Recommended range

Minimum: 1025

Maximum: 65536

Description

Port number by which indexing commands are sent to IDOL server.

Example

IndexPort=59001

### **Threads**

Type: Long

Default: 10

Allowed range

Minimum: 1

Maximum: 32

Recommended range

Minimum: 4

Maximum: 32

Description

Maximum number of parallel actions. The number of action commands that the server can process in parallel at any one time. You can calculate the optimum number of threads as follows:

$(1 \times \text{Num CPU's}) + 1$  spare thread.

Example

Threads=2

### **QueryClients**

Type: String

Default: \*

Required: yes

Description

Addresses of hosts that can send action commands. Enter the IP address (or name) of one or more machines that are permitted to send action commands to the Port. If you want to enter multiple addresses, you must separate them with commas (there must be no space before or after a comma).

Example

QueryClients=10.1.1.\*,127.0.0.1

### **AdminClients**

Type: String

Default: localhost

Required: yes

Description

Addresses of administrative hosts. Enter the IP address (or name) of one or more machines that are permitted to send administrative commands to the Port. If you want to enter multiple addresses, you must separate them with commas (there must be no space before or after a comma).

Example

AdminClients=localhost,127.0.0.1

### **IndexClients**

Type: String

Description

Addresses of allowed indexing clients. One or more IP addresses or hostnames for machines that are permitted to index data into IDOL server (using the specified IndexPort).

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

Example

IndexClients=10.1.1.10,127.0.0.1

### **MaxInputString**

Type: Long

Default: 64000

Recommended range

Minimum: 100

Maximum: 100000

Description

Maximum characters in a request string. The maximum number of characters that can be sent in an HTTP request string to the server. Enter -1 if you do not want to restrict this number.

Example

```
MaxInputString=100
```

In this example, an HTTP request that is sent to the server can comprise a maximum of 100 characters.

### **MaxQueryTerms**

Type: Long

Default: 300

Recommended range

Minimum: 40

Maximum: 5000

Description

Maximum number of terms in a query. Allows you to specify the maximum number of words in the query text that can be used to query IDOL server (excluding stoplisted words, which are not used to query IDOL server). Any query text that exceeds the specified number is cut off (in case of Boolean or phrase strings this can result in query failure).

If an action's MaxQueryTerms parameter conflicts with the configuration file's MaxQueryTerms setting, the action's setting overrides the configuration file setting.

Example

```
MaxQueryTerms=200
```

### **MaxFileUploadSize**

Type: Long

Default: 10000000

Description

The maximum number of bytes to upload. Specify the maximum size (in bytes) of files that can upload to the server. To specify an unlimited upload size, set MaxFileUploadSize to -1.

**NOTE:** This setting applies to all actions sent using HTTP POST to the ACI port, even if they are not file uploads.

Example

```
MaxFileUploadSize=5000000
```

### **MaxResults**

Type: Long

Default: 10000

Recommended range

Minimum: 0

Maximum: 1000000

Description

Maximum number of query results. Specify the maximum number of result documents that can be returned by a query.

The MaxResults action parameter specifies the number of results you want returned for a query from the total number of results returned.

Example

```
MaxResults=300
```

### **QueryTimeoutInMilliseconds**

Type: Long

Description

Period of time (in milliseconds) to timeout query commands. Enter the number of milliseconds after which you want query commands to time out if IDOL server has not finished processing them. By default, query commands do not time out.

The TimeoutMS parameter in a query command overrides the setting in the QueryTimeoutInMilliseconds parameter.

Example

```
QueryTimeoutInMilliseconds=10000
```

### **RefHashes**

Type: Long

Default: 500000

Allowed range

Minimum: 10000

Maximum: 50000000

#### Recommended range

Minimum: 100000

Maximum: 20000000

#### Description

Number of hash slots in document-reference hash table. Specify the number of hash slots that are used when the hash table of references is created. For optimal speed, you should configure this to be the same as the expected number of documents within IDOL server.

**NOTE:** This value is fixed when you initialize IDOL server and in most cases, should not be changed.

#### Example

```
RefHashes=1000000
```

### TermSize

Type: Long

Default: 20

#### Recommended range

Minimum: 5

Maximum: 60

#### Description

Maximum permitted length (5-60) of a term in bytes.

#### Example

```
TermSize=10
```

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.

### DelayedSync

Type: Boolean

Default: false

#### Description

Delay storing indexed data in database. By default IDOL server stores index data every time it finishes indexing an IDX or XML file or when the specified MaxSyncDelay is reached.

Enter true if you want IDOL server to only store index data on disk when the index cache is full or when the specified MaxSyncDelay is reached. This allows you to increase performance considerably if you are indexing a lot of small files (files that are smaller than 100MB).

#### Example

DelayedSync=true

### **MaxSyncDelay**

Type: Long

Default: 120

Recommended range

Minimum: 5

Maximum: unlimited

Description

Maximum seconds to delay before writing indexed data to disk. Enter the maximum time (in seconds) that can elapse before IDOL server is forced to store index data on disk.

Example

```
MaxSyncDelay=180
```

### **AdvancedSearch**

Type: Boolean

Default: false

Description

Enable exact keyword search. Enter true if you want IDOL server to use an advanced weighting algorithm that improves conceptual querying.

Enabling AdvancedSearch also allows you to search for an exact word or phrase match. IDOL server matches any word or phrase that is enclosed in quotation marks in its exact pre-stemmed form when it is queried (query text that is not enclosed in quotation marks is still matched conceptually).

**NOTE:** AdvancedSearch is not affected by enabling or disabling stemming.

Enabling AdvancedSearch has no notable effect on IDOL server's index size.

Enabling AdvancedSearch does not prevent IDOL server from discarding stopwords at index and at query time.

If you are enabling AdvancedSearch, it is recommended that you set ProperNames to 7 in the appropriate language type sections of the configuration file.

If you are enabling AdvancedSearch, you can still execute a conceptual phrase search that uses stemming by using DNEAR1.

Examples

Word matching

The following query is executed:

```
action=Query&Text="lovely"
```

IDOL server does not apply stemming.



When it matches the query, IDOL server only returns documents that contain a word that exactly matches the query word. The query "lovely" can only return documents that contain the word lovely.

#### Phrase matching

The following query is executed:

```
action=Query&Text="fresh and lovely"
```

IDOL server removes any stopwords that the query contains (the example query above contains the stopword and) but does not apply stemming:

```
action=Query&Text="fresh lovely"
```

When it matches the query, IDOL server only returns documents that contain a phrase that matches the phrase in the query string. The query "fresh and lovely" can only return documents that contain a phrase which matches the phrase fresh lovely (for example, fresh lovely, fresh and lovely, fresh or lovely and so on).

#### Example

```
AdvancedSearch=true
```

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.

### SplitNumbers

Type: Boolean

Default: true

#### Description

Whether to split alphanumeric terms. By default, IDOL server splits alphanumeric terms (such as Y2K or CBF4) and stores them as separate terms, to reduce the number of indexed terms (when you have set IndexNumbers to 1 or 2). Enter false if you do not want to split alphanumerical terms.

**NOTE:** If you want to use wildcards to search for numbers, set SplitNumbers to false.

When you set SplitNumbers to true, you can also set either AlphaNumericTermChunkSize or NumericTermChunkSize to -1, to prevent splitting for either alphanumeric or pure numeric terms.

#### Example

```
SplitNumbers=true
```

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.

### DateFormatCSVs

Type: String

Default: YYYY/MM/DD

#### Description

List of supported date formats. One or more date formats that you want IDOL server to be able to read from the files that it contains. If you want to enter multiple date formats, you must separate them with commas (there must be no space before or after a comma). You can enter one of the following:

AUTNDATE

a string that contains one or more of the following:

String	Description
<b>YY</b>	Year (2 digits). For example, 99, 00, 01 and so on.
<b>YYYY</b>	Year (4 digits). For example, 1999, 2000, 2001 and so on.
<b>#YY+</b>	Year (2 or 4 digits). If 2 digits are provided, then the YY format is used. If 4 digits are provided, then the YYYY format is used. For example, 07 is interpreted as 2007 AD and 1007 is interpreted as 1007 AD.
<b>#Y</b>	Year (1 to a maximum of 16 digits) and may be followed by "AD" or "BC". An apostrophe (') immediately before the year denotes a truncated year. For example, 2008, '97 (interpreted as 1997), 97 (interpreted as 97 AD), '08 (interpreted as 2008), 2008 AD and 200 BC. A truncated year with a BC identifier is invalid ('08 BC).
<b>#FULLYEAR</b>	Year (1 to a maximum of 16 digits). For example 8, 98, 108, 2008, each of which is taken literally. The year is taken relative to the common EPOCH (0AD).
<b>#ADBC</b>	Time Period. For example, AD, CD, BC, BCE or any predefined list of EPOCH indicators. Typically, the year specified using the above Year formats is interpreted as untruncated and relative to the EPOCH. For example, 84 AD is interpreted as 1984 AD and 84 BC is interpreted as 84 BC.  The only exception to this is when #YY+ and #ADBC are both used. In this case, the format is interpreted as untruncated even if the year was set to truncated by #YY+. For example, 99 AD is interpreted as the year 99 AD.  It is recommended you only use YY, YYYY or #FULLYEAR with #ADBC.
<b>LONGMONTH</b>	A long month, for example, January, February and so on.
<b>SHORTMONTH</b>	A short month, for example, Jan, Feb and so on.
<b>MM</b>	Month (2 digits). For example, 01, 10, 12 and so on.
<b>M+</b>	Month (1 or 2 digits). For example, 1,2,3,10 and so on.
<b>DD</b>	Day (2 digits). For example, 01, 02, 03, 12, 23 and so on.
<b>D+</b>	Day (1 or 2 digits). For example, 1, 2, 12, 13, 31 and so on.
<b>LONGDAY</b>	2 digits with a postfix. For example, 1st, 2nd and so on.
<b>HH</b>	Hour (2 digits). For example, 01, 12, 13 and so on.

String	Description
<b>H+</b>	Hour (1 or 2 digits).
<b>NN</b>	Minute (2 digits).
<b>N+</b>	Minute (1 or 2 digits).
<b>SS</b>	Second (2 digits).
<b>S+</b>	Second (1 or 2 digits).
<b>ZZZ</b>	Time Zone, for example, GMT, EST, PST, and so on.
<b>ZZZZZ</b>	Time Difference (1 to 9 digits). For example, +04 denotes 4 hours ahead of UTC. Other examples include +4, +04, +0400, +0400 MSD (the string MSD is ignored). A further example is +030, in this case the time differences is interpreted as 30 minutes.
<b>#PM</b>	AM or PM indicator (2 characters). For example, 2001/09/09 02:46:40 pm
<b>#S</b>	A space.

**NOTE:** Format strings are matched in the order in which they are listed. You should therefore put the format in order of length (starting with the longest). This prevents the matching of, for example, 19/10/2002 with DD/MM/YY (if this format has been listed before DD/MM/YYYY) and the document from being indexed with the date 19/10/20.

**NOTE:** To enter one or more formats, separate them with commas (there must be no space before or after a comma). To specify a format that contains a space, put the format in quotation marks.

**NOTE:** When documents containing fields with partial dates are indexed into IDOL, the document is assigned a metadata field specifying a full date. This will be the first day of the month if the day is not specified, and the first of January if neither the day or the month are specified. The partial date in the document must still be specified using DateFormatCSVs.

#### Example

DateFormatCSVs=D+/SHORTMONTH/YYYY,DDMMYY

In this example, only document dates with the format D+/SHORTMONTH/YYYY (for example, 2/Jan/2001) or DDMMYY (for example, 020101) can be indexed with a date.

DateFormatCSVs="D+SHORTMONTH YYYY", "Date: D+ LONGMONTH, YYYY"

In this example, only document dates with the format D+ SHORTMONTH YYYY (for example, 2 Jan 2001) or Date: D+ LONGMONTH, YYYY (for example, Date: 2 January, 2001) can be indexed with a date.

DateFormatCSVs=LONGMONTH/D+/#Y H+:N+:S+

In this example, only document dates with the format LONGMONTH/D+/#Y H+:N+:S+ (for example, September/9/2001 AD 2:46:40 or September/9/2001 2:46:40 or September/9/'01 2:46:40 or September/9/'01 AD 2:46:40) can be indexed with a date.

```
DateFormatCSVs=LONGMONTH/D+/#FULLYEAR #ADBC H+:N+:S+
```

In this example, only document dates with the format LONGMONTH/D+/#FULLYEAR #ADBC H+:N+:S+ (for example, September/9/8 AD 2:46:40 or September/9/2008 AD 2:46:40 or September/9/'01 2:46:40 or September/9/'01 AD 2:46:40) can be indexed with a date.

```
DateFormatCSVs=D+ LONGMONTH YYYY HH:NN:SS ZZZZ
```

In this example, only document dates with the format D+ LONGMONTH YYYY HH:NN:SS ZZZZZ (for example, 17 May 2003 10:41:07 -0400) can be indexed with a date.

## KillDuplicates

Type: String

Default: Reference

Description

Determines how IDOL server handles duplicate documents. This parameter allows you to prevent the same document or document content from being stored in IDOL server more than once. Enter one of the following options to determine how IDOL server handles duplicate text. Note that if you postfix any of these options with =2, the KillDuplicates process is applied to all IDOL server databases (rather than just the database into which the current IDX or XML file is being indexed):

NONE

Duplicate documents are allowed in IDOL server and are not replaced or deleted.

REFERENCE

If the document being indexed has the same DREREFERENCE field value as a document that already exists in the IDOL server, IDOL server deletes the existing document and replaces it with the new document.

REFERENCEMATCHN

If the content of the document being indexed is more than N percent similar to the content of a document that already exists in the IDOL server database, IDOL server deletes the existing document and replaces it with the new document.

FieldName

If the document being indexed contains a FieldName Reference field with the same value as the FieldName Reference field in a document that already exists in IDOL server, IDOL server deletes the existing document and replaces it with the new document.

To specify multiple Reference fields, separate the fields with a plus sign (+) or a space. IDOL server deletes documents containing any of the specified fields with identical content.

**NOTE:** Fields are identified as Reference fields through field processes in the IDOL server configuration file. If you use a FieldName Reference field to eliminate duplicate documents, IDOL server automatically reads any fields listed alongside this field for the PropertyFieldCSVs

parameter in the field process, and also uses these fields to eliminate duplicate documents. If you want to define multiple reference fields but do not want them all to be used for document elimination, you must set up multiple field processes (see the IDOL Server Administrator Guide).

If documents are indexed into IDOL server using a DREADD or DREADDDATA command, and the command uses a KillDuplicates setting that conflicts with the KillDuplicates parameter in IDOL server's configuration file, the command's setting overrides the IDOL server configuration file's parameter.

#### Example

```
KillDuplicates=REFERENCE
```

In this example, if the document being indexed has the same DRREFERENCE as a document that already exists in IDOL server, IDOL server deletes the existing document and replaces it with the new document.

### **DocumentDelimiterCSVs**

Type: String

Default: \*/Document

#### Description

File fields (tags) marking start and end of document. Allows you to specify the fields in a file that indicate the beginning and end of a document, so that the documents are indexed individually. You should only have one document level per XML schema.

If documents are indexed into IDOL server using a DREADD or DREADDDATA command, and the command uses a DocumentDelimiters parameter that conflicts with the DocumentDelimiterCSVs setting in IDOL server's configuration file, the command's parameter overrides the IDOL server configuration file's setting.

#### Example

```
DocumentDelimiterCSVs=*/DOCUMENT,*/SPEECH
```

In this example, the beginning and end of individual documents in a file is marked by opening and closing DOCUMENT and SPEECH tags (or fields if you are running IDOL server in DRE3 mode).

### **UnstemmedMinDocOccs**

Type: Long

Default: 5

#### Description

Minimum number of documents a term must appear in to be a spellcheck suggestion or to be matched by a wildcard term.

#### Example

```
UnstemmedMinDocOccs=10
```

## RepositoryStorage

Type: Boolean

Default: false

Description

When set to TRUE, data is stored into a single nodetable. This allows IDOL to use "IndexCacheFlushPause", and the index is updated atomically (lower risk on index inconsistency). However, indexing takes longer as the nodetable grows.

When set to FALSE, data is stored into separate nodetables. This allows for faster indexing, but "IndexCacheFlushPause" is unavailable, and there is a higher risk of index inconsistencies

Example

```
RepositoryStorage=FALSE
```

## MinFreeSpaceMB

Type: Long

Default: 1053

Allowed range

Minimum: 50

Maximum: 2000000000

Recommended range

Minimum: 1053

Maximum: 4212

Description

Minimum disk space (in MB) required at startup. Enter the amount of free disk space (in MB) that IDOL server requires in the nodetable, dynterm and numeric directories in order to start up.

If the available disk space falls below the MinFreeSpaceMB value, and IDOL server has an indexing job to process, the job is paused. The job does not resume automatically, and the indexing queue remains paused until one of the following occurs:

The engine is shut down with the Stop action, or by stopping the service. This generates a warning message in the index log.

The index job is cancelled manually using the IndexerGetStatus action with the IndexAction parameter set to Cancel.

The index job is resumed manually using the IndexerGetStatus action with the IndexAction parameter set to Restart. If the index job is resumed in this way, it is processed as normal if sufficient disk space is available. Otherwise it is paused again.

Example

```
MinFreeSpaceMB=1053
```

### **CombineIgnoreMissingValue**

Type: Boolean

Default: False

Description

Determines how documents are combined in query results if they do not have the specified combine field. If the CombineIgnoreMissingValue parameter is set to true, documents in which the combine field (either FieldCheckType field or ReferenceType field) is absent are treated as distinct. When this parameter is set to false, the combined results include only one document with no such field.

Example

```
CombineIgnoreMissingValue=true
```

### **Term Cache section**

The [TermCache] section contains parameters that determine which query terms are stored in memory and how much memory is allocated for cached query terms. Storing terms in memory that appear frequently in documents prevent the terms from being read from disk whenever the term appears in a query.

For example:

```
[TermCache]  
TermCacheMaxSize=102400
```

### **TermCacheMaxSize**

Type: Long

Default: 102400

Allowed range

Minimum: 0

Maximum: 1024000

Recommended range

Minimum: 10240

Maximum: 1024000

Description

Size of the term cache (in Kb). The term cache stores query terms. Increasing the size of the cache can improve query performance, however, you should ensure the host machine has sufficient available memory. To enable term caching, set the TermCachePersistentMinOccs or TermCachePersistentKB parameter.

Example

```
TermCacheMaxSize=102400f
```

### Index Cache section

The [IndexCache] section contains parameters that determine how much memory IDOL server uses to cache data for indexing.

For example:

```
[IndexCache]
IndexCacheMaxSize=102400
```

#### IndexCacheMaxSize

Type: Long

Default: 102400

Description

Initial maximum size of index cache (in KB). The index cache stores data for indexing. Increasing the size of the cache can improve indexing performance. However, you should ensure that the host machine has sufficient available memory.

**NOTE:** You can dynamically grow or shrink the indexing cache after IDOL server startup, by using the DRERESIZEINDEXCACHE index command.

Example

```
IndexCacheMaxSize=102400
```

### Section Breaking section

The [SectionBreaking] section contains parameters that determine the size of the sections that documents are divided into before they are indexed.

For example:

```
[SectionBreaking]
MinFieldLength=80
MaxSectionLength=2000
```

#### MinFieldLength

Type: Long

Default: 100

Allowed range:

Minimum: 0

Maximum: 2000000000

Recommended range:

Minimum: 50

Maximum: 500



#### Description

The maximum number of characters an XML field can contain before IDOL server breaks it into sections. If a field contains fewer than the specified number of characters it is not broken into sections.

**NOTE:** This configuration parameter applies only to XML documents.

#### Example

```
MinFieldLength=150
```

### **MaxSectionLength**

Type: Long

Default: 1000

Allowed range

Minimum: 200

Maximum: 2000000000

Recommended range:

Minimum: 500

Maximum: 5000

#### Description

The maximum number of characters to include in each XML section. Enter the maximum number of characters that a document can contain before IDOL server breaks it into sections.

**NOTE:** This configuration parameter applies only to XML documents.

#### Example

```
MaxSectionLength=1000
```

### **Paths section**

The [Paths] section contains parameters that allow you to split the database into multiple partitions and parameters that indicate the location of files that IDOL server uses.

For example:

```
DyntermPath=./dynterm  
NodetablePath=./nodetable  
RefIndexPath=./refindex  
MainPath=./main  
StatusPath=./status
```

## Logging section

The Logging section lists the logging streams used to create separate log files for different log message types (such as, query, index, and application). It also contains a subsection for each of the listed logging streams in which you can configure the parameters that determine how each stream is logged.

For example:

```
[Logging]
LogDirectory=${Folder}\logs
LogTime=TRUE
LogOutputLogLevel=TRUE
LogLevel=Normal
LogExpireAction=Compress
LogOldAction=delete
LogMaxOldFiles=10
MaxLogSizeKbs=20480
```

```
0=ApplicationLogStream
1=QueryLogStream
2=IndexLogStream
3=ActionLogStream
```

```
[ApplicationLogStream]
LogFile=application.log
LogTypeCSVs=application
```

```
[QueryLogStream]
LogFile=query.log
LogTypeCSVs=query
```

```
[IndexLogStream]
LogFile=index.log
LogTypeCSVs=index
```

**NOTE:** All queries are truncated to 4000 characters in query logs.

## LogDirectory

Type: String

Default: ./logs

Description

Path to log directory. Enter the directory in which you want the application to store the log files it creates.

Example

```
LogDirectory=./logs
```

### **LogTime**

Type: Boolean

Default: true

Description

Display time with each log entry. Enable this parameter to display the current time next to each log entry in the log file.

Example

LogTime=false

### **LogOutputLogLevel**

Type: Boolean

Default: false

Description

Determines whether the log level is included in the log. Enable this parameter to include the log level of a message in the log entry.

Example

LogLevel=Always

LogOutputLogLevel=TRUE

In this example, Always is added to the log message:

```
21/12/2006 12:34:56 [10] Always: ACI Server attached to port 1622
```

### **LogExpireAction**

Type: String

Default: Datestamp

Description

Determines how log files are handled when they exceed the maximum size. Enter one of the following to determine how log files are handled when they exceed the MaxLogSizeKBs size:

Compress

The log file's name is appended with a time stamp, compressed and saved in the log directory. By default, this is a Zip file. Use the LogCompressionMode parameter to specify another compression format.

Consecutive

The log file's name is appended with a number and saved in the log directory. When the next log file reaches its LogMaxSizeKBs size, it is appended with the next consecutive number.

Datestamp

The log file's name is appended with a time stamp and saved in the log directory.

#### Previous

The log file's name is appended with `.previous` and saved in the log directory. Every time a log file reaches its `LogMaxSizeKBs` size, it is given the same postfix so it overwrites the old log file.

#### Day

Only one log file is created per day and is appended with the current time stamp. Log files are archived once they reach the `LogMaxSizeKBs` size.

**NOTE:** The `LogMaxSizeKBs` parameter takes precedence over the `LogExpireAction` parameter. Therefore, if you set `LogExpireAction` to `Day`, and the value for `LogMaxSizeKBs` results in more than one log file, multiple log files will be generated per day.

#### Example

```
LogExpireAction=Compress
```

### **LogOldAction**

Type: String

Default: Delete

#### Description

Determines how log files are handled when the maximum number of log files is exceeded. Enter one of the following to determine how log files are handled when the `LogDirectory` has reached the maximum number of log files, as determined by the `LogMaxOldFiles` parameter:

#### Delete

The log files are deleted.

#### Move

The log files are moved to the specified `LogArchiveDirectory`.

#### Example

```
LogOldAction=Move
```

### **LogMaxOldFiles**

Type: Long

Default: -1 (unlimited)

#### Description

Maximum number of log files in the log directory. The maximum number of log files the specified `LogDirectory` can store before the application executes the specified `LogOldAction`. If you do not want to restrict how many log files the `LogDirectory` can store, enter -1.

#### Example

```
LogMaxOldFiles=1000
```

### **LogMaxSizeKBs**

Type: Long

Default: 1024

Description

Maximum log file size (in KB). If you do not want to restrict the log file size, enter -1. The LogExpireAction parameter determines how a log file is handled once it has reached its maximum size.

Example

LogMaxSizeKBs=1000

### **LogFile**

Type: String

Required: yes

Description

Name of the log file. The name of the log file the application creates in the specified LogDirectory.

Example

LogFile=query.log

### **LogTypeCSVs**

Type: String

Required: yes

Description

List of message types to log. Enter one or more of the following message types to specify the type of messages written to the associated log file. If you want to enter multiple message types, separate them with commas (there must be no space before or after a comma):

All Components.

Action

Logs actions and related messages.

Application

Logs application-related occurrences.

IDOL Server.

Agent

Logs agent actions and related messages.

Category

Logs category actions and related messages.

#### Cluster

Logs cluster actions and related messages.

#### Community

Logs community actions and related messages.

#### ExtendedIndex

Logs index actions as well as index commands that are issued after IDOL server has routed incoming data through other processes.

#### Index

Logs index actions and related messages.

#### Mailer

Logs mailer actions and related messages.

#### Profile

Logs profile actions and related messages.

#### Query

Logs query actions and related messages.

#### QueryTerms

Logs each query term, after stemming, conversion to UTF8, capitalization and punctuation removal. This is mainly used by the DiSH server for statistical reports.

#### Role

Logs role actions and related messages.

#### Schedule

Logs schedule actions and related messages.

#### Taxonomy

Logs taxonomy actions and related messages.

#### User

Logs user actions and related messages.

#### User\_Audit

Logs USERADD and USERDELETE actions and related messages.

#### UserTerm

Logs terms that IDOL server uses to form a user's agents and profiles.

#### DIH.

#### Index

Logs index actions and related messages.

#### Query

Logs query actions and related messages.

DAH.

Security

Logs security action results.

Connectors.

FailureList

Logs details of files that were not imported successfully.

Import

Logs import actions and related messages.

Index

Logs index actions and related messages.

Spider

Logs spider actions and related messages. (HTTP Connector only)

CFS.

Indexer

Logs the status of indexing into IDOL.

Import

Logs import actions and related messages.

CFS Connectors.

Collect

Logs document collection for Legal Hold application.

Delete

Logs the deletion of documents from the repository.

Hold

Logs details of documents that are put on hold in Legal Hold applications.

Identifiers

Logs details of requests for document lists from repositories.

Synchronize

Logs data synchronization when indexing into IDOL.

Update

Logs details of documents whose metadata is updated in the repository.

View

Logs details of documents that are viewed from the repository.

### Example

```
LogTypeCSVs=Application,Index
```

### Field Processing section

The [FieldProcessing] section lists the processes you want to apply to fields and contains a subsection for each of the processes. Within each subsection, you specify the process and to which fields the process applies.

For example:

```
[FieldProcessing]
```

```
0=SetIndexFields
```

```
1=SetDateFields
```

```
2=SetDatabaseFields
```

```
[SetIndexFields]
```

```
Property=IndexFields
```

```
PropertyFieldCSVs=*/DRECONTENT,*/DRETITLE,*/TS_TITLE,*/TS_OWNERLOCATION,*/TS_EXTENSION,*/TS_CONTACT,*/TS_NOTES
```

```
[SetDateFields]
```

```
Property=DateFields
```

```
PropertyFieldCSVs=*/DREDATE
```

```
[SetDatabaseFields]
```

```
Property=DatabaseFields
```

```
PropertyFieldCSVs=*/DREDBNAME
```

### Property

Type: String

Description

Name of a property of this process. Enter a property name, which describes the process that you want to apply to documents that contain the specified PropertyFieldCSVs fields.

Example

```
Property=IndexingFields
```

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.

### PropertyFieldCSVs

Type: String

Description

Names of fields to associate with this process. Enter one or more fields that you want to associate with the processes that you have set up in the [FieldProcessing] section. Depending on its nature the



process is applied to the specified fields or the documents that contain the specified fields during the indexing process.

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

Example

```
PropertyFieldCSVs=*/DRECONTENT,*/DRETITLE
```

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.

### Properties section

The Properties section contains a section for each of the properties you create for the processes listed in the [FieldProcessing] section. Within the properties sections, you set the configuration parameters to apply to associated fields.

For example:

```
[IndexFields]  
Index=TRUE
```

```
[DateFields]  
DateType=TRUE
```

```
[DatabaseFields]  
DatabaseType=TRUE
```

```
[TitleFields]  
TitleType=TRUE
```

```
[ReferenceFields]  
ReferenceType=TRUE  
TrimSpaces=TRUE
```

```
[HighlightFields]  
HighlightType=TRUE
```

```
[RelatedDocsReferenceFields]  
ReferenceType=TRUE  
TrimSpaces=TRUE
```

```
[NumericDateFields]  
NumericDateType=TRUE
```

```
[OtherReferenceFields]  
ReferenceType=TRUE  
TrimSpaces=TRUE
```

## Index

Type: Boolean

Default: false

Description

The fields with this property are index fields. Type true to store the associated PropertyFieldCSVs as Index fields.

Store fields that contain text that you query frequently with conceptual queries as Index fields. IDOL server processes index fields linguistically when it stores them. IDOL server applies stemming and stoplists to text in Index fields before it stores them, which allows it to process queries for these fields more quickly (typically you store DRETITLE and DRECONTENT as Index fields).

It is recommended you do not store URLs or content that you are unlikely to use as Index fields. Also, do not store fields as Index fields if you query it frequently, but only query the value in its entirety. Indexing all fields in documents can potentially slow down the indexing process, and increase disk usage and requirements.

Example

Index=true

**NOTE:** If you change this setting after you have indexed content into IDOL server, you must re-index the content.

## DateType

Type: Boolean

Default: false

Description

The fields with this property hold document dates. Type true to indicate that the associated PropertyFieldCSVs fields hold the document date.

each document that you store in IDOL server must contain only one DateType field. If IDOL server receives a document that contains more than one DateType field, it acknowledges only the first DateType field and ignores the rest.

Example

DateType=true

**NOTE:** If you change this setting after you have indexed content into IDOL server, you must re-index the content.

## DatabaseType

Type: Boolean

Default: false

### Description

The fields with this property hold IDOL database names. Type true to indicate that the associated PropertyFieldCSVs fields contain the name of the database that IDOL server must store the document in.

If you specify an IDOL server database in the index action when you index a document, the database in the action overrides the database in the DatabaseType field.

**NOTE:** Each document that you store in IDOL server must contain only one DatabaseType field. If IDOL server receives a document that contains more than one DatabaseType field, it acknowledges only the first DatabaseType field and ignores the rest.

### Example

```
DatabaseType=true
```

### TitleType

Type: Boolean

Default: false

### Description

The fields with this property hold document title. Type true to use the associated PropertyFieldCSVs fields as the document title.

**NOTE:** Each document that you store in IDOL server must only contain one TitleType field. If IDOL server receives a document that contains more than one TitleType field, it acknowledges only the first TitleType field and ignores the rest.

### Example

```
TitleType=true
```

**NOTE:** If you change this setting after you have indexed content into IDOL server, you must re-index the content.

### ReferenceType

Type: Boolean

Default: false

### Description

The fields with this property contain document references. Type true to indicate that the associated PropertyFieldCSVs fields contain references.

### Example

```
ReferenceType=true
```

**NOTE:** If you change this setting after you have indexed content into IDOL server, you must re-index the content.

### TrimSpaces

Type: Boolean

Default: false

Description

Whether to remove extra spaces from the values of the fields with this property. Type true to remove multiple, leading or trailing spaces from the associated PropertyFieldCSVs fields.

Example

TrimSpaces=true

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.

### HighlightType

Type: Boolean

Default: false

Description

The fields with this property allow highlighting. Type true to allow highlighting of terms in the associated PropertyFieldCSVs when you send a query action with the Highlight parameter set to true.

Example

HighlightType=true

### NumericDateType

Type: Boolean

Default: false

Description

Optimize the speed of matching for the fields with this property. Type true to enable IDOL server to store the associated PropertyFieldCSVs fields, which contain numeric dates, in a fast-look-up table in memory to optimize matching of the fields when using FieldText specifiers.

**NOTE:** You cannot simultaneously configure a field as NumericDateType and MatchType.

Example

NumericDateType=true

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content, unless you have set `RegenerateNumericDateIndex` to true.

## Language Types section

The [LanguageTypes] section lists the languages you want to use and defines some generic parameters.

For example:

```
[LanguageTypes]
DefaultLanguageType=englishASCII
DefaultEncoding=UTF8
LanguageDirectory=C:\IDOLserver\IDOL\langfiles
GenericTransliteration=true
StopWordIndex=1
0=English
```

## DefaultEncoding

Type: String

Description

The default encoding used for the `OutputEncoding` parameter in action commands. The `OutputEncoding` parameter specifies the encoding to which you want the output resulting from actions commands to be converted before the output is returned.

If you do not set `DefaultEncoding` or `OutputEncoding`, IDOL server uses the encoding specified in the `LanguageType` parameter sent with the query, or in the `DefaultLanguageType` configuration parameter if the `LanguageType` parameter is not sent with the query.

Example

```
DefaultEncoding=UTF8
```

## DefaultLanguageType

Type: String

Description

Default language type used for documents of unspecified language.

This parameter specifies the language type (language and encoding) IDOL server assumes documents to be when it cannot read the language type of a document from a specified field, the `LanguageType` parameter is not specified in an action command, or automatic language detection is not enabled.

The parameter also specifies the default for the `LanguageType` parameter in action commands such as `Query` and `Summarize`.

If you do not set a `DefaultLanguageType` or specify the language type in the action command, IDOL server uses the first language type defined in the [MyLanguage] section.

If you do not set the `DefaultEncoding` configuration parameter or the `OutputEncoding` parameter in an action command, IDOL server uses the encoding specified in the `LanguageType` parameter sent with the query, or in the `DefaultLanguageType` configuration parameter if the `LanguageType` parameter is not sent with the query.

**NOTE:** Documents whose encoding is identified but not their language can be placed into a [General] language category, if it is defined, and if Automatic Language Detection is activated.

Example

```
DefaultLanguageType=englishASCII
```

### GenericFullStemming

Type: Boolean

Default: false

Description

Use language-insensitive generic stemming algorithm. If you have set `GenericStemming` to true, you can set `GenericFullStemming` to true to use a generic stemming algorithm for all your language types and not automatically modify this algorithm according to which languages you have configured.

Because this stemming algorithm is static and doesn't need to change according to which languages you have configured, you can configure additional language types after you have indexed content into IDOL server without having to re-index all your content.

Example

```
GenericFullStemming=true
```

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.

### GenericStemming

Type: Boolean

Default: false

Description

Use language-sensitive generic stemming algorithm. Normally, queries to which you have added `&AnyLanguage=true` only return multi-language documents if they contain terms that stem the same way as terms in your query. Since different stemming algorithms are applied to different languages, only terms that are common to multiple languages tend to stem the same way (for example, many international words like "computer" are stemmed differently in certain languages).

If you want to relax the way queries are matched against multiple languages, you can set `GenericStemming` to true. This instructs IDOL server to use a generic stemming algorithm for all your language types, which is automatically modified according to which languages you have configured.

You can use the `Stemming` configuration parameter for individual languages to specify whether that language's stemming rules should be used. For example, if you want to use only the English

stemming rules for your generic stemming, you can set Stemming to true for English, and set Stemming to false for all other languages. This does not apply if you have set GenericFullStemming to true.

**NOTE:** If you want to configure an additional language type with Stemming set to true after you have indexed content into IDOL server, you must re-index all your content. If you add a new language with Stemming set to false, or if you have set GenericFullStemming to true, you do not need to re-index.

Example

```
GenericStemming=true
```

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.

### GenericTransliteration

Type: Boolean

Description

Enter true to apply transliteration (accent removal) to all languages and use a single generic transliteration scheme. Transliteration is applied to Japanese, Chinese, Greek, Spanish and Portuguese by default, and is optional for Western European, German, Scandinavian, and Russian.

Transliteration is the process of converting accented characters such as øù into equivalent non-accented characters. This is useful in environments where accented keyboards are not available.

You can alter transliteration settings for individual languages using the Transliteration configuration parameter in the individual language configuration sections.

Example

```
GenericTransliteration=true
```

**NOTE:** If you want to change this setting after you have indexed content into IDOL server, you must re-index the content.

### LangDetectUTF8

Type: Boolean

Default: false

Description

Set to true if you want files that contain 7-bit ASCII to be detected as UTF-8, rather than ISO-8859-1 (ASCII).

When Automatic Language Detection is used to detect the language of documents that are being indexed, it uses the Index fields by default. If these fields contain only 7-bit ASCII, the document will be detected as ASCII. If additional fields in the document contain UTF-8, these might be converted

incorrectly. If you know that documents are generally in UTF-8, then set `LangDetectUTF8` to true so that these documents will be detected as UTF-8.

Example

```
LangDetectUTF8=true
```

### LanguageDirectory

Type: String

Required: yes

Description

Path to the language-files directory. The directory in which the files are stored that the languages you are using require.

Example

```
LanguageDirectory=./langfiles
```

### StopWordIndex

Type: Boolean

Default: 0

Description

Set to 1 to index stopwords.

Set to 2 to index stopwords, but to prevent IDOL server from expanding wildcards to stopwords.

Example

```
StopWordIndex=1
```

**NOTE:** In queries, IDOL server matches stopwords only within a phrase. Sample queries:  
winnie the bear. IDOL ignores the stopword "the".  
"winnie the bear". IDOL includes the stopword "the" in the phrase query.  
copy of "winnie the bear". IDOL ignores the stopword "of", but it includes "the" in the phrase query.

If you change this setting after you have indexed content into IDOL server, you must re-index the content.

### Databases section

The [Databases] section summarizes the number of databases that are created for IDOL server, and contains a subsection for each of these databases. Within a subsection, you specify configuration parameters for the database.

For example:

```
[Databases]  
NumDBs=2
```



[Database0]

Name=News

[Database1]

Name=Archive

### NumDBs

Type: Long

Description

Total number of databases. Enter the total number of databases you are creating in IDOL server (you need to create an [DatabaseN] section for each database).

Example

NumDBs=2

### DatabaseReadOnly

Type: Boolean

Default: false

Description

Make this database read-only. Enter true if you want the database to be read only. This means that while the database's content can be retrieved, no new content can be added to it. The database's content also cannot be expired or deleted.

**NOTE:** You should only set DatabaseReadOnly to true for a database after you have stored data in the database.

Example

DatabaseReadOnly=true

### XML document handling

To index XML tags in text files, you must add the content index server to the registry key.

**NOTE:** Make sure that the user has permission to add or edit the registry key.

Open **Regedit**, go to the following path and add the new parameter:

<b>For content index event processor</b>	Path	<b>HKEY_LOCAL_MACHINE\SOFTWARE\Micro Focus\Content Manager\Datasets\&lt;Dataset ID&gt;</b>
	Parameter name	<b>IndexXMLFormatToIDOL</b>
	Type	<b>DWORD (32 bit) Value (REG_DWORD)</b>

	Value	<b>1</b>
	<b>NOTE:</b> After adding the new registry parameter, re-start the event processor.	
<b>For Re-indexing from client</b>	Path	<b>HKEY_CURRENT_USER\SOFTWARE\Micro Focus\Content Manager\Datasets\&lt;Dataset ID&gt;</b>
	Parameter name	<b>IndexXMLFormatToIDOL</b>
	Type	<b>DWORD (32 bit) Value (REG_DWORD)</b>
	Value	<b>1</b>

## **Appendix B**

### **Default Ports Used By IDOL Components**

#### **Main IDOL Service**

- 9000
- 9001
- 9002

#### **AgentStore**

- 9050
- 9051
- 9052
- 9053

#### **Category**

- 9020
- 9021

#### **DAH**

- 9060
- 9061

#### **DIH**

- 9070
- 9071
- 9072

#### **Content Service 1**

- 9100
- 9101
- 9102

## **Content Service 2**

- 9200
- 9201
- 9202

## **Appendix C**

### **Supported File Types for IDOL Content Indexing**

Content Manager uses KeyView to extract text from documents which is then passed across to IDOL. Any file type can be content indexed if the version of KeyView used by Content Manager can extract text from that file type.

## Appendix D

### Content Manager Fields Available in IDOL

The following fields are indexed into IDOL

Content Manager Field	IDOL Field Name	Notes
URN	DRREFERENCE	The URN for a Content Manager record in IDOL. See <a href="#">Content Manager URN</a> for more info
Title	DRETITLE	The title of the record in IDOL. This will normally be the title of the associated record in Content Manager
-	AUTN_GROUP	The autonomy group. When indexed by Content Manager, this value will be "Trim"
-	AUTN_IDENTIFIER	The autonomy identifier
URN	DREROOTFAMILYREFERENCE	The URN of the associated Content Manager record. Either the same as the DRREFERENCE, or the IDOL record is a subdocument, then the DRREFERENCE of the parent record
-	SECURITYTYPE	When indexed by Content Manager, this value will be "TrimExt"
Unique identifier	TN_URI	

<b>Content Manager Field</b>	<b>IDOL Field Name</b>	<b>Notes</b>
Record number	TS_NUMBER	
Full title	TS_TITLE	
Date registered	TD_DATEREGISTERED	
Date created	TD_DATECREATED	
Classification	TS_CLASSIFICATION	
Owner location	TS_OWNERLOCATION	
Document extension	TS_EXTENSION	
Notes	TS_NOTES	
Document content	DRECONTENT	

## Appendix E

### FieldText Methods

#### MATCH

The MATCH field specifier (case sensitive) allows you to find documents in which the entirety of a specified field contains a value that exactly matches a specified string.

##### **Format**

FieldText=MATCH{yourStrings}:yourFields

*yourStrings*

Enter one or more strings. A document is returned only if one of these strings is matched by one of yourFields exactly. The matching is case-insensitive. You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if the value in this field exactly matches one of yourStrings. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

##### **Example**

FieldText=MATCH{DOC1234}:TS\_NUMBER

A record with a record number of DOC1234

#### EQUAL

The EQUAL field specifier (case sensitive) allows you to find documents in which a specified field contains a number that matches one of the numbers specified by you.

##### **Format**

FieldText=EQUAL{yourNumbers}:yourFields

*yourNumbers*

Enter one or more numbers. A document is only returned if one of yourFields contains one of these numbers. If you want to specify multiple numbers you must separate them with commas (there must be no space before or after a comma).

*yourFields*



Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains one of yourNumbers. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### **GREATER**

The GREATER field specifier (case sensitive) allows you to find documents in which a specified field contains a number that is greater than a number specified by you.

#### **Format**

FieldText=GREATER{yourNumber}:yourFields

*yourNumber*

Enter a number. A document is only returned if one of yourFields contains a number that is greater than this number.

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if the number in this field is greater than yourNumber. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### **LESS**

The LESS field specifier (case sensitive) allows you to find documents in which a specified field contains a number that is smaller than a number specified by you.

#### **Format**

FieldText=LESS{yourNumber}:yourFields

*yourNumber*

Enter a number. A document is only returned if one of yourFields contains a number that is smaller than this number.

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if the number in this field is smaller than yourNumber. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### **NRANGE**

The NRANGE field specifier (case sensitive) allows you to find documents in which a specified field contains a number that falls within the inclusive range of two numbers specified by you.

#### **Format**

FieldText=NRANGE{yourNumbers}:yourFields

*yourNumbers*

Enter two numbers separated by a comma (there must be no space before or after the comma). A document is only returned if one of yourFields contains a number that falls within the inclusive range of the specified numbers (including decimal numbers).

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains a number that falls within the inclusive range of *yourNumbers*. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

#### **NOTEQUAL**

The NOTEQUAL field specifier (case sensitive) returns documents in which at least one instance of a specified field contains a number that does NOT match the number you specify.

##### **Format**

FieldText=NOTEQUAL{*yourNumber*};*yourFields*

*yourNumber*

Enter one number. A document returns only if at least one instance of one of *yourFields* contain a number that is not *yourNumber*.

*yourFields*

Enter one or more fields. A document returns only if it contains at least one of these fields, and if at least one instance of one of these fields does not contain *yourNumber*. If you want to specify multiple fields, you must separate them with colons (:). There must be no space before or after a colon.

#### **GTNOW**

The GTNOW field specifier (case sensitive) allows you to find documents in which a specified field contains a date that is greater than the AUTNDATE (that is all documents that were indexed with dates after the current time).

##### **Format**

FieldText=GTNOW{};*yourFields*

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains a date that is greater than the AUTNDATE (that is all documents that were indexed with dates after the current time). If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

#### **LTNOW**

The LTNOW field specifier (case sensitive) allows you to find documents in which a specified field contains a date that is smaller than the AUTNDATE (that is all documents that were indexed with dates before the current time).

##### **Format**

FieldText=LTNOW{};*yourFields*

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains a date that is smaller than the AUTNDATE (that is all documents that were indexed with

dates before the current time). If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

## RANGE

The RANGE field specifier (case sensitive) allows you to find documents in which a specified field contains a date that falls within the inclusive range of two dates specified by you.

### Format

FieldText=RANGE{yourDates}:yourFields

#### *yourDates*

Enter two dates separated by a comma (there must be no space before or after the comma). A document is only returned if one of yourFields contains a date that falls within the inclusive time span specified by the specified dates.

You can use the following formats to specify each date.

Format	Type	Notes
DD/MM/YY	A date.	For example, 1/3/05, 23/12/99 or 10/07/40.  If the year is a number less than 40, it is read as a year in the 2000s. If the year is a number between 40 and 99, it is read as a year in the 1900s. For example, 1/02/1 is read as February 1st 2001, while 01/3/40 is read as March 1st 1940.
DD/MM/YYYY	A date.	For example, 1/3/2005, 23/12/1999 or 10/07/1940.
N	A positive or negative number of days from the current date.	For example, -1 specifies yesterday's date, 0 specifies today's date, 1 specifies tomorrow's date, 2 specifies two days from now (the current date plus two) and so on.
Ns	A positive or negative number of seconds from now.	For example, -60s specifies 1 minute ago, -900s specifies 15 minutes ago, -3600s specifies 1 hour ago and so on. 60s specifies 1 minute from now, 900s specifies 15 minutes from now, 3600s specifies 1 hour from now and so on.
Ne	Epoch seconds (seconds since January 1st 1970).	For example, 1012345000e specifies 22:56:40 on January 29th 2002.

#### *yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains a date that falls within the inclusive range of yourDates. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### Example

FieldText=RANGE{01/01/90,1/1/01}:TD\_DATECREATED

A record's date created must be a date between 01/01/1990 and 1/1/2001 for this record to be returned.

## **WILD**

The WILD field specifier (case sensitive) allows you to find documents in which a specified field contains a string that matches a specified wildcard string.

If the query does not contain any wildcard characters (? or \*) then the WILD field specifier acts in the same way as the MATCH field specifier.

### **Format**

FieldText=WILD{yourStrings}:yourFields

#### *yourStrings*

Enter one or more strings containing wildcards. A document is only returned if one of yourFields matches one of these strings. You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

#### *yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains one of yourStrings. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### **Example**

FieldText=WILD{\*.html,\*.htm}:TS\_TITLE

A title must end with html or htm for this record to be returned as a result.

## **ARANGE**

The ARANGE field specifier (case sensitive) allows you to find documents in which a specified field contains a term that falls within the inclusive alphabetical range of two terms specified by you.

### **Format**

FieldText=ARANGE{yourTerms}:yourFields

#### *yourTerms*

Enter two terms separated by a comma (there must be no space before or after the comma). A document is only returned if one of yourFields contains a term that falls within the inclusive alphabetical range of the specified terms.

A full stop (.) can be used in place of one of the terms, to represent an unrestricted value. If the full stop is used in place of the first term, all values up to the second term are included. If the full stop is used in place of the second term, all values after the first term are included.

**NOTE:** Unicode tables are used to determine alphabetical order. This means that non-7-bit ASCII characters (a, ä, å, ç, d, ê, ë, ø, ö, ü, û, ß, ÿ etc.) come after z in the alphabet.

### *yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains a term that falls within the inclusive alphabetical range of yourTerms. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### **Example**

**FieldText=ARANGE{aardvark,alligator}:TS\_ANIMAL**

A record's "animal" additional field must contain a value that alphabetically falls between aardvark and alligator. If a record's "animal" field contains the value aardvark, ant, anteater, antelope or alligator, the document is returned. If a record's "animal" field contains the value armadillo, it is not returned.

### **EMPTY**

The EMPTY field specifier (case sensitive) allows you to find documents in which a specified field doesn't exist or contains no value.

### **Format**

FieldText=EMPTY{}:yourFields

### *yourFields*

Enter one or more fields. A record is only returned if it doesn't contain any of these fields or if these fields are empty. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### **Example**

**FieldText=EMPTY{}:TS\_NOTES**

A record must nothing in its notes to be returned.

### **EXISTS**

The EXISTS field specifier (case sensitive) allows you to find documents that contain a specified field even if this field contains no value.

### **Format**

FieldText=EXISTS{}:yourFields

### *yourFields*

Enter one or more fields. A record returns only if it contains one of these fields (even if the field is empty). To specify multiple fields, separate them with colons (there must be no space before or after a colon).

### **Example**

**FieldText=EXISTS{}:TS\_NOTES**

A record must have notes to be returned.

## **FUZZY**

The FUZZY field specifier (case sensitive) allows you to find documents in which a specified field contains a term that is similar to a specified term or phrase.

### **Format**

FieldText=FUZZY{yourTerms}:yourFields

#### *yourTerms*

Enter one or more terms (or phrases). A document is only returned if one of these terms (or phrases) is similar to a string in one of yourFields. If you want to specify multiple terms (or phrases) you must separate them with commas (there must be no space before or after a comma).

#### *yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if the value in this field is similar to one of yourTerms. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### **Example**

FieldText=FieldText=FUZZY{Business News,Arkive}:TS\_TITLE

A record's title must be similar to the term Business News or Arkive for this record to be returned. (A record whose title contains Business News would be returned, while a record whose title contains Document Arkive would not).

## **LENGTH**

The LENGTH field specifier (case sensitive) allows you to find documents in which a specified field contains a number of characters that falls within the inclusive range of two numbers specified by you.

### **Format**

FieldText=LENGTH{yourNumber}:yourFields

#### *yourNumbers*

Enter two numbers separated by a comma (there must be no space before or after the comma). A document is only returned if the number of characters contained in one of yourFields falls within the inclusive range of the specified numbers.

#### *yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if the number of characters that this field contains falls within the inclusive range of yourNumbers. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### **Example**

FieldText=LENGTH{10,20}:TS\_TITLE

A record's title must be between 10 and 20 characters (inclusive) long for this record to be returned.

## MATCHALL

The MATCHALL field specifier (case sensitive) allows you to find documents in which a specified field occurs in multiple instances, and in which there is at least one match among those instances for each of a set of strings that you specify.

### **Format**

FieldText=MATCHALL{yourStrings}:yourField

*yourStrings*

Enter one or more strings. A document is returned only if all of these strings have exact matches among the instances of yourField. The matching is case-independent. You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

*yourField*

Enter the name of the field to match against. A document is returned only if it contains the field and only if all of yourStrings are matched at least once in various instances of the field.

Example

FieldText=MATCHALL{Archive,Web,docs}:TS\_TITLE

A record's title must include at least the values Archive and Web and docs for the record to be returned as a result.

## EQUALALL

The EQUALALL field specifier (case sensitive) allows you to find documents in which a specified field occurs in multiple instances, and in which there is at least one value among those instances that is equal to each of a set of numeric values that you specify.

### **Format**

FieldText=EQUALALL{yourValues}:yourNumericField

*yourValues*

Enter one or more numeric values. A document is returned only if all of these values occur among the instances of yourNumericField. Separate the numbers with commas (there must be no space before or after a comma).

*yourNumericField*

Enter the name of the field to match against. A document is returned only if it contains the field and only if all of yourValues are matched at least once in various instances of the field.

**Example**

FieldText=EQUALALL{32,98.6,212}:TN\_TEMPERATURE

The additional field Temperature on a record must include at least the values 32, 98.6, and 212 for the record to be returned as a result.

## **MATCHCOVER**

The MATCHCOVER field specifier (case sensitive) allows you to find documents in which the values in all instances of a specified field have matches in the set of values provided in the specifier. In other words, the specifier must "cover" all instances of the field. a search using MATCHCOVER is slower than one using MATCH.

### **Format**

FieldText=MATCHCOVER{yourStrings}:yourField

*yourStrings*

Enter one or more strings. A document is returned only if the value in each of its instances of yourField matches one of the strings in yourStrings. The matching is case-independent. You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

*yourField*

Enter the name of the field to match against. A document is returned only if:

- it contains one or more instances of the field and the value of each instance is found in yourStrings

or

- it does not contain the field at all.

### **Example**

FieldText=MATCHCOVER{Australia,India,China,Russia}:TS\_PLACEOFBIRTH

For a record to be returned as a result, its PlaceOf Birth additional field must have no value that is not in the specified list. For example, if a record's PlaceOfBirth field is "New Zealand", it would not be returned.

(If a record has no TS\_PLACEOFBIRTH field at all, it is returned.)

## **EQUALCOVER**

The EQUALCOVER field specifier (case sensitive) allows you to find documents in which the values in all instances of a specified field are found in the set of values provided in the specifier. In other words, the specifier must "cover" all instances of the field.

### **Format**

FieldText=EQUALCOVER{yourValues}:yourField

*yourValues*



Enter one or more numeric values. A document is returned only if the value in each of its instances of `yourField` equals one of the values in `yourValues`. Separate the numbers with commas (there must be no space before or after a comma).

*yourField*

Enter the name of the field to match against. A document is returned only if (1) it contains one or more instances of the field and the value of each instance equals a value in `yourValues`, or (2) it does not contain the field at all.

### **Example**

```
FieldText=EQUALCOVER{9,10,11,12}:TN_GRADELEVEL
```

For a record to be returned as a result, its `Gradelevel` additional field must have no values that are not in the specified list. For example, if a record includes a `gradelevel` field with the value 8, it is not returned.

(If a record has no `Gradelevel` field at all, it is returned.)

### **NOTMATCH**

The `NOTMATCH` field specifier (case sensitive) allows you to find documents in which at least one instance of the specified fields contains a value that does not match the specified string.

If there are one or more instances of a particular field in the document, the document will be returned as long as at least one instance does not contain any of the specified strings, even if another instance of the field does match. The document will not be returned if all instances of the specified fields contain an exact match of one of the specified strings.

### **Format**

```
FieldText=NOTMATCH{yourStrings}:yourFields
```

*yourStrings*

Enter one or more strings. A document is only returned if at least one instance of one of `yourFields` contains a value that is not an exact match for these strings. The matching is case insensitive. You can specify strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if the value in at least one instance of the field does not exactly match any of `yourStrings`. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### **Example**

```
FieldText=NOTMATCH{cat}:TS_ANIMAL
```

At least one instance of a record's `Animal` additional field must have a value other than `cat` for this document to be returned as a result.

## STRING

The STRING field specifier (case sensitive) allows you to specify one or more strings of which one must be contained as a substring in a specified field.

### Format

FieldText=STRING{yourStrings}:yourFields

*yourStrings*

Enter one or more strings. A document is only returned if one of these strings is a substring of the value in one of yourFields. You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if the value in this field is contains one of yourStrings as a substring. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### Example

FieldText=STRING{cat,dog}:TS\_NOTES

A record's notes must contain the substring cat or dog for this record to be returned.

## STRINGALL

The STRINGALL field specifier (case sensitive) allows you to specify one or more strings, which all must be contained as a substring in a specified field.

### Format

FieldText=STRING{yourStrings}:yourFields

*yourStrings*

Enter one or more strings. A document is only returned if all of these strings are substrings of the value in one of yourFields. You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if the value in this field is contains yourStrings as substrings. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### Example

#### **FieldText=STRINGALL{cat,dog}:TS\_NOTES**

A record's notes must contain the substrings cat and dog for this record to be returned. If a record's notes, for example, has the value "grooming cats and dogs" or "doggedly scattering seeds", this record will be returned.

### **SUBSTRING**

The SUBSTRING field specifier (case sensitive) allows you to return documents whose field value is a substring of a specified string (or equal to a specified strings).

#### **Format**

FieldText=SUBSTRING{yourStrings}:yourFields

#### *yourStrings*

Enter one or more strings. A document is only returned if one of yourFields contains a substring of one of the specified strings. You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

#### *yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if the value in this field is a substring of yourStrings. To specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

#### **Example**

**FieldText=SUBSTRING{Telecommunications,Technology}:SECTOR**

A document's SECTOR field must contain a string that is a substring of Telecommunications or Technology. If a document's SECTOR field, for example, has the value Telecom or Technology, the document will be returned. If a document's SECTOR field has the value Latest Technology, the document will not be returned.

### **TERM**

The TERM field specifier (case sensitive) allows you to find documents with a specified field whose value contains a conceptual match of one or more terms specified by you. A conceptual match exists if a term you specify matches a term in a specified field after it has been stemmed.

#### **Format**

FieldText=TERM{yourTerms}:yourFields

#### *yourTerms*

Enter one or more terms. A document is only returned if one of yourFields contains a value that includes a term which conceptually matches one of the specified terms. You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if a term in this field conceptually matches one of yourTerms. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

**Example**

**FieldText=TERM{shopping,centers}:TS\_TITLE**

A record's title must contain a term that conceptually matches shopping or centers for this document to be returned. If a record's title, for example, has the value shop this record will be returned, while if it has the value bookshopping, it will not be returned.

**TERMALL**

The TERMALL field specifier (case sensitive) allows you to find documents with a specified field whose value contains conceptual matches of several terms specified by you. A conceptual match exists if the terms you specify match terms in a specified field after they have been stemmed.

**Format**

**FieldText=TERMALL{yourTerms}:yourFields**

*yourTerms*

Enter multiple terms. A document is only returned if one of yourFields contains a value that includes terms which conceptually match the specified terms.

You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if a term in this field conceptually matches one of yourTerms. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

**Example**

**FieldText=TERMALL{shopping,centers}:TS\_TITLE**

A record's title must contain a term that conceptually matches shopping or centers for this document to be returned. If a record's title, for example, has the value town center shop this record will be returned.

## TERMEXACT

The TERMEXACT field specifier (case sensitive) allows you to find documents with a specified field that contains an exact match of any of the terms specified by you.

### **Format**

FieldText=TERMEXACT{yourTerms}:yourFields

#### *yourTerms*

Enter one or more terms. A document is only returned if one of yourFields contains a value that exactly matches one of the specified terms. You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

#### *yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains an exact match of one of yourTerms. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### **Example**

FieldText=TERMEXACT{help,helped}:TS\_TITLE

A record's title must contain the term help or helped for this document to be returned. If a title, for example, has the value helps or helping, the record will not be returned.

## TERMEXACTALL

The TERMEXACTALL field specifier (case sensitive) allows you to find documents with a specified field that contains an exact match of all terms specified by you.

### **Format**

FieldText=TERMEXACTALL{yourTerms}:yourFields

#### *yourTerms*

Enter multiple terms. A document is only returned if one of yourFields contains exact matches of the specified terms. You can match strings that contain punctuation (but see Note below) or consist of several words.

**NOTE:** To distinguish query syntax punctuation from punctuation within strings, double-percent-encode commas and curly braces within strings. Query syntax punctuation should be left unencoded. There should be no space before or after a separator comma.

#### *yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains an exact match of all yourTerms. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

## Example

**FieldText=TERMEXACTALL{rabbits,eating,carrots}:TS\_TITLE**

This query returns only records whose title contains all the specified terms (in their specified form). For example, a record whose title is “Rabbits like eating carrots” or “The carrots were there but the rabbits ate all the cabbage” will be returned as a result, while a record with a title that contains “Rabbits like to eat a carrot each day” will not be returned.

## TERMEXACTPHRASE

The TERMEXACTPHRASE field specifier (case sensitive) allows you to return documents in which a specified field contains an exact match of a phrase specified by you. Your phrase is matched before stemming is applied (stopwords are not removed). Any punctuation in the specifier or field is ignored.

### Format

**FieldText=TERMEXACTPHRASE{yourPhrase}:yourFields**

*yourPhrase*

Enter a phrase. A document is only returned if one of yourFields contains an exact match of the specified phrase.

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains an exact match of yourPhrase. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

### Example

**FieldText=TERMEXACTPHRASE{Batman! and Robins}:FILM**

A document whose FILM field contains Showing now, Batman and Robin's film, will be returned as a result, while a document whose FILM field contains Showing now, 'Batman and Robin' the movie will not be returned.

## TERMPHRASE

The TERMPHRASE field specifier (case sensitive) allows you to return documents in which a specified field contains a conceptual match of a phrase specified by you. Your phrase is matched after stemming is applied (stopwords are not removed). Any punctuation in the specifier or field is ignored.

### Format

**FieldText=TERMPHRASE{yourPhrase}:yourFields**

*yourPhrase*

Enter a phrase. A document is only returned if one of yourFields contains a conceptual match of the specified phrase.

*yourFields*

Enter one or more fields. A document is only returned if it contains one of these fields, and if this field contains a conceptual match of yourPhrase. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

**Example**

**FieldText=TERMPHRASE{Batman! and Robins}:FILM**

A record's whose FILM field contains Showing now: 'Batman and Robin', will be returned as a result.

**BIAS**

The BIAS field specifier (case sensitive) allows you to bias the score of results at query time according to the numerical proximity of the specified field to a given value.

**Format**

FieldText=BIAS{optimum,range,percentage}:yourfield

*optimum*

The value that the specified field must contain to increase or decrease the result's weight by the maximum percentage.

*range*

A positive number that determines the range of the specified optimum. If the specified field contains a value that is in the range of (optimum - range) to (optimum + range), the result's weight is increased or decreased according to the specified percentage.

*percentage*

A percentage in the range -100 to 100. If the value of the specified field is within the specified range, the score of the result is increased or decreased according to how close the value is to the specified optimum. When Absweight is set to true, this is the absolute value by which to boost the weight and it is then not limited by +/- 100.

*yourFields*

Enter one or more fields. A document has its score boosted if it contains one of these fields, and if this field contains a number within the specified range of the optimum value. To specify multiple fields, separate them with colons (there must be no space before or after a colon).

**Example**

**FieldText=BIAS{100,50,10}:\*/TN\_PRICE**

A record whose PRICE additional field value is within the range 50 either side of 100 will have its weight increased on a linear scale from 10% if the price is 100, to 0% if the price is 50 or 150.

**BIASDATE**

The BIASDATE field specifier (case sensitive) allows you to boost the score of result documents by a specified percentage, based on how close the date in a specified field is to a specified date.

**Format**

FieldText=BIASDATE{optimumDate,range,percentage}:yourFields

*optimumDate*

The date that the specified field must contain to increase or decrease the result's weight by the maximum percentage.

You can use the following formats to specify the date.

Format	Type	Notes
DD/MM/YY	A date.	For example, 1/3/05, 23/12/99 or 10/07/40.  If the year is a number less than 40, it is read as a year in the 2000s. If the year is a number between 40 and 99, it is read as a year in the 1900s. For example, 1/02/1 is read as February 1st 2001, while 01/3/40 is read as March 1st 1940.
DD/MM/YYYY	A date.	For example, 1/3/2005, 23/12/1999 or 10/07/1940.
N	A positive or negative number of days from the current date.	For example, -1 specifies yesterday's date, 0 specifies today's date, 1 specifies tomorrow's date, 2 specifies two days from now (the current date plus two) and so on.
Ns	A positive or negative number of seconds from now.	For example, -60s specifies 1 minute ago, -900s specifies 15 minutes ago, -3600s specifies 1 hour ago and so on. 60s specifies 1 minute from now, 900s specifies 15 minutes from now, 3600s specifies 1 hour from now and so on.
Ne	Epoch seconds (seconds since January 1st 1970).	For example, 1012345000e specifies 22:56:40 on January 29th 2002.

**range**

A positive value that determines the range in seconds of the specified optimum. If the specified field contains a value that is in the range of (optimum - range) to (optimum + range), the result's weight is increased or decreased according to the specified percentage.

**percentage**

A percentage in the range -100 to 100. If the value of the specified field is within the specified range, the score of the result is increased or decreased according to how close the value is to the specified optimum. When Absweight is set to true, this is the absolute value by which to boost the weight and it is then not limited by +/- 100.

**yourFields**

Enter one or more fields. A document will have its score boosted if it contains one of these fields, and if this field contains a date that falls within the specified range of the optimum date. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

**Example**



**FieldText=BIASDATE{1/12/2008,864000,10}:TD\_DATEREGISTERED**

A record's whose date registered value is within a range of 10 days (864000 seconds) either side of 1/12/2008 will have its weight increased on a linear scale from 10% if the date is 1/12/2008, to 0% if the date is 20/11/2008 or 11/12/2008.

## **BIASVAL**

The BIASVAL field specifier (case sensitive) allows you to bias the score of result documents by a specified percentage, based on whether they include a specific value in the specified field.

### **Format**

FieldText=BIASVAL{fieldValue,percentBoost}:fieldName

#### *fieldValue*

The value that the specified field must contain to increase or decrease the result's weight by the percentBoost.

#### *percentBoost*

A percentage in the range -100 to 100. The percentage (positive or negative) by which to boost the relevance of documents that have the value fieldValue in the field fieldName. (The boosted relevance score cannot be greater than 100% or less than -100%.) When Absweight is set to true, this is the absolute value by which to boost the weight and it is then not limited by +/- 100.

#### *fieldName*

The name of the field that must be present (and must contain the proper value) for a document's relevance to be boosted.

### **Example**

**FieldText=BIASVAL{Cambridge,10}:TS\_TOWN**

A record's whose TOWN additional field value is Cambridge will have its weighting increased by 10% (although the weighting cannot go over 100%).