Connector Framework Server

Software Version 12.7

Administration Guide

Document Release Date: October 2020
Software Release Date: October 2020

Legal notices

Copyright notice

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

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 Micro Focus 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:

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

Many areas of the portal require you to sign in. If you need an account, you can create one when prompted to sign in. To learn about the different access levels the portal uses, see the Access Levels descriptions.

About this PDF version of online Help

This document is a PDF version of the online Help.

This PDF file is provided so you can easily print multiple topics or read the online Help.

Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online Help.

Contents

Chapter 1: Introduction

This section provides an overview of Connector Framework Server.

Connector Framework Server

Connector Framework Server (CFS) processes the information that is retrieved by connectors, and then indexes the information into one or more indexes, such as IDOL Server.

Connectors send information to CFS in the form of documents. A *document* is a collection of metadata and, usually, an associated source file. The metadata describes the location of the file or record that was retrieved, and other information that was extracted by the connector. For example, a document sent for ingestion by a Web Connector includes the URL of the page and the links that were extracted from the page when it was crawled. The Web Connector provides the downloaded HTML in an associated file so that it can be processed by CFS.

Sometimes a document does not have an associated source file. For example, if you retrieve information from a database using the ODBC Connector, the documents sent for ingestion contain the information extracted by your chosen query, and might not have an associated file. These documents are referred to as having *metadata only*.

CFS uses KeyView to extract information from the source file. Some source files are container files, such as zip archives, and these are extracted. CFS then uses KeyView to obtain text and file-specific metadata from the file, and adds it to the document. The original source file is discarded before the document is indexed. This allows IDOL to search and categorize documents, and perform other operations, without needing to process the information from a repository in its native format.

CFS provides features to manipulate and enrich documents. For example, you can send media files to an IDOL Media Server and perform tasks such as optical character recognition and face recognition. This adds additional information to the IDOL document, so that when a user queries IDOL the results include relevant images, audio, and video files. CFS also supports the Lua scripting language so that you can write your own tasks and develop custom processing rules.

A single CFS can process information from any number of connectors. For example, a CFS might process files retrieved by a File System Connector, web pages retrieved by a Web Connector, and e-mail messages retrieved by an Exchange Connector. Alternatively, you or an application can send information to CFS directly.

Filter Documents and Extract Subfiles

CFS uses KeyView to extract meaningful information from the files retrieved by a connector. KeyView can extract the file content, metadata, and subfiles from over 1,000 different file types.

- *File content* is the main content of a file, for example the body of an e-mail message.
- *Metadata* is information about a file itself, for example the sender of an e-mail message or the date and time when it was received.
- *Subfiles* are files that are contained within the main file. For example, an e-mail message might contain embedded images or attachments that you want to index.

Manipulate and Enrich Documents

CFS provides features to manipulate and enrich documents. *Enriching* a document means adding additional information, or improving the quality and usefulness of the information, before the document is indexed into IDOL. For example, you can:

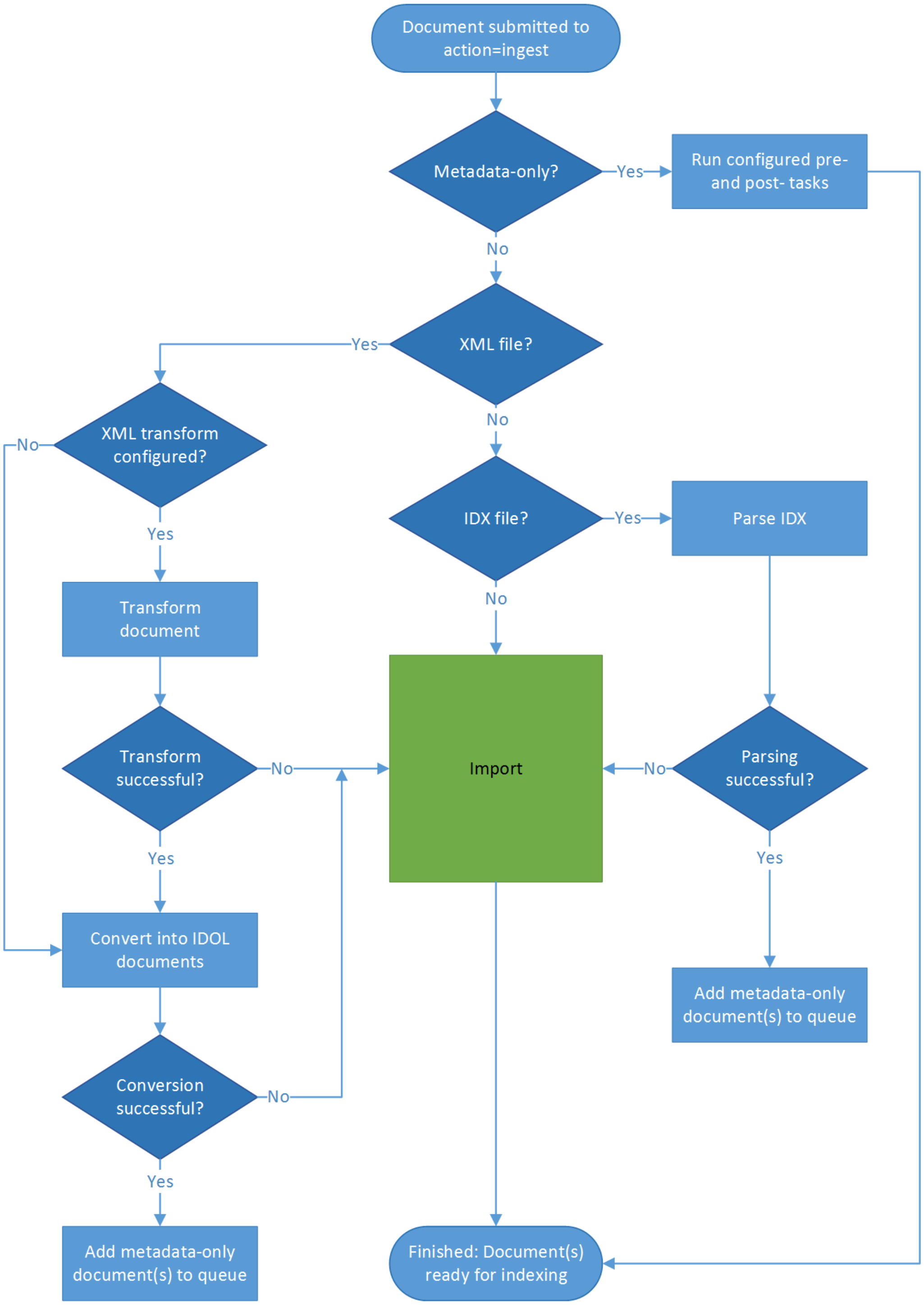
- Add additional fields to a document.
- Extract content from HTML pages, discarding irrelevant content such as headers, sidebars, advertisements, and scripts.
- Split long documents into multiple sections. This can improve performance when you query IDOL, because IDOL can return a specific part of a document in response to a query.
- Standardize field names, so that documents that originated from different repositories use the same fields to store the same type of information.
- Perform *Eduction* on document fields. Eduction extracts *entities* from a document, and writes them to specific document fields. An entity can be a word, phrase, or block of information - for example an address or telephone number.
- Perform analysis on image and video files and add the results to the document. Examples of media analysis include optical character recognition (OCR), face detection and recognition, and object recognition. To analyze media you must have an IDOL Media Server.
- Extract speech from audio and video files, and add the transcription to the document content. To analyze speech you must have an IDOL Speech Server.
- Reject documents that do not contain content in a specific language.

The simplest way to manipulate documents is to use the *import tasks* that are included with CFS. For information about the tasks that are available, see Manipulate and Enrich Documents, on page 48. You can configure these tasks by modifying configuration parameters in the CFS configuration file.

CFS also supports Lua, an embedded scripting language. You can write Lua scripts to manipulate documents and define custom processing rules. For information about the Lua functions that are provided with CFS, refer to the *Connector Framework Server Reference*.

The Ingestion Process

The following chart provides a summary of the ingestion process.


Document submitted to action=ingest
Metadata-only?
Yes
Run configured pre- and post- tasks
No
XML file?
Yes
No
XML transform configured?
No
Yes
IDX file?
Yes
Parse IDX
No
Transform document
Transform successful?
No
Import
No
Parsing successful?
Yes
Yes
Convert into IDOL documents
Add metadata-only document(s) to queue
Conversion successful?
No
Yes
Add metadata-only document(s) to queue
Finished: Document(s) ready for indexing

Documents are submitted to Connector Framework Server through the `ingest` action. If the document has metadata only, CFS runs any processing tasks that have been configured and the document is then ready for indexing. If the document has an associated file then the ingestion process depends on the file format.

- **All files apart from IDOL IDX and XML**. Most documents that have an associated file are added to the import queue so that the information in the file can be extracted by KeyView or other processing tasks. For information about the import process, see The Import Process, on the next page.
- **IDOL IDX files**. An IDX file contains one or more documents in IDOL IDX format, so CFS attempts to parse the file. If parsing is successful then the IDOL documents are returned to the ingest queue as metadata-only documents. If parsing is not successful then CFS adds the document to the import queue so that the IDX file is processed by KeyView. Parsing an IDX file is preferable to processing it with KeyView, because although KeyView can extract the text, it cannot extract the structure information that divides the text into separate documents, content sections, and metadata fields.
- **XML files**. Many systems export information in XML format and CFS has features to help you convert XML into IDOL documents.

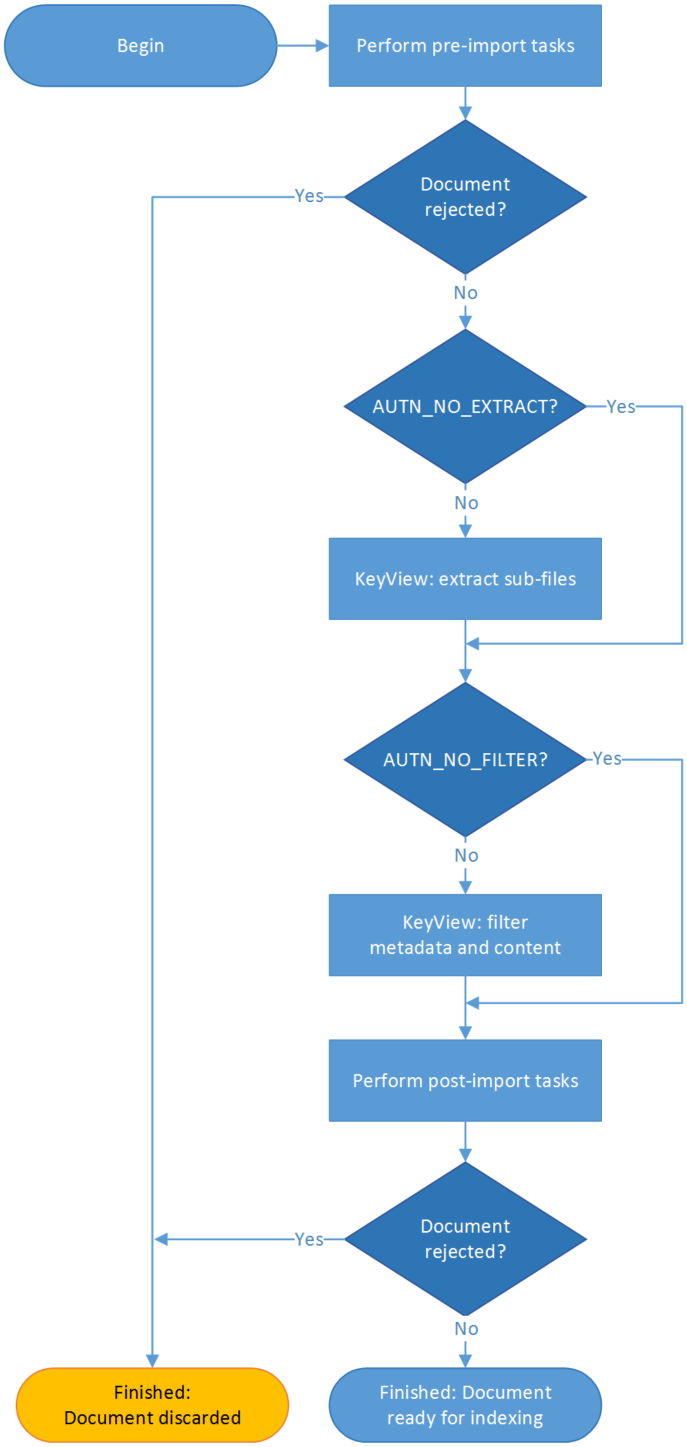
  CFS can run a transformation on an ingested XML file. This is an optional step but can be useful in cases where your XML files do not resemble IDOL documents or you are processing XML from many sources and the files have different schemas. You can configure any number of transformations and CFS runs the first transformation where the ingested XML matches the specified schema. You can also configure a default transformation that CFS runs when an XML file does not match any of your schemas. When a transformation is configured but is not successful, CFS adds the document to the import queue so that the XML is processed by KeyView.

  After an XML transformation is successful or when transformation is not configured, CFS attempts to convert the XML into IDOL documents. The conversion is performed by mapping elements in the XML to IDOL documents and document fields. If the conversion is successful the resulting documents are returned to the ingest queue as metadata-only documents. If the conversion does not result in any IDOL documents but the XML was transformed after matching a schema, CFS does not consider this as a failure and does not index any documents. Otherwise, CFS adds the document to the import queue so that the XML is processed by KeyView.

  Parsing an XML file is usually preferable to processing it with KeyView, because although KeyView can extract the text it does not preserve the structure information (the XML tags are discarded).

The Import Process

The following chart provides a summary of the import process.


Begin
Perform pre-import tasks
Document rejected?
Yes
No
AUTN_NO_EXTRACT?
Yes
No
KeyView: extract sub-files
AUTN_NO_FILTER?
Yes
No
KeyView: filter metadata and content
Perform post-import tasks
Document rejected?
Yes
No
Finished: Document discarded
Finished: Document ready for indexing

1. CFS takes a document from the import queue.
2. CFS performs the pre-import tasks that are configured in its configuration file. Pre-import tasks occur before files are processed by KeyView. You can use pre-import tasks to manipulate and enrich documents (see Manipulate and Enrich Documents, on page 10). Sometimes it is important to run tasks before KeyView processing. For example, if you send an audio file to Media Server for analysis, you might not want to process it with KeyView.

   > **TIP:** Both pre- and post-import tasks can reject a document, so that it is discarded and not indexed. You might configure CFS to reject a document if the associated file does not contain useful content. Documents are not rejected when an import task fails - in that case CFS continues processing the document.

3. Unless the document contains the metadata field `AUTN_NO_EXTRACT`, CFS uses KeyView to extract sub-files. Examples of files that have sub-files include e-mail messages (which have attachments) and zip files (which contain other files). CFS creates a new document for each sub-file and adds the new documents to the import queue to be processed separately.
4. Unless the document contains the metadata field `AUTN_NO_FILTER`, CFS uses KeyView to filter the associated source file. Filtering extracts the text from a file. An office document is likely to contain useful text, while an archive file (for example a zip file) or a media file is unlikely to have textual content.

   > **TIP:** Although media files (images, audio, and video) do not contain text, you can extract useful information by sending the files to an IDOL Media Server.

5. CFS performs the post-import tasks that are configured in its configuration file.
6. Processing is complete and the document is ready to be indexed.

Index Documents

After CFS finishes processing documents, it automatically indexes them into one or more indexes. You can index documents into:

- **IDOL Server** (or send them to a *Distributed Index Handler*, so that they can be distributed across multiple IDOL servers).
- **Vertica**.

The IDOL Platform

At the core of Connector Framework Server is the *Intelligent Data Operating Layer* (IDOL).

IDOL gathers and processes unstructured, semi-structured, and structured information in any format from multiple repositories using IDOL connectors and a global relational index. It can automatically form a contextual understanding of the information in real time, linking disparate data sources together based on the concepts contained within them. For example, IDOL can automatically link concepts contained in an email message to a recorded phone conversation, that can be associated with a stock trade. This information is then imported into a format that is easily searchable, adding advanced retrieval, collaboration, and personalization to an application that integrates the technology.

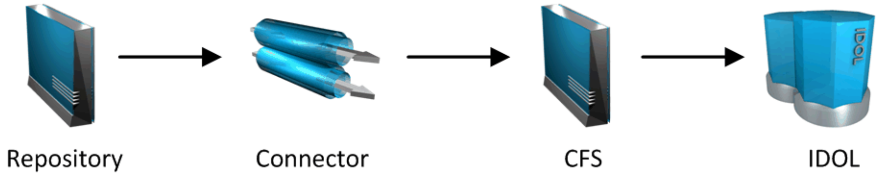
For more information on IDOL, see the *IDOL Getting Started Guide*.

System Architecture

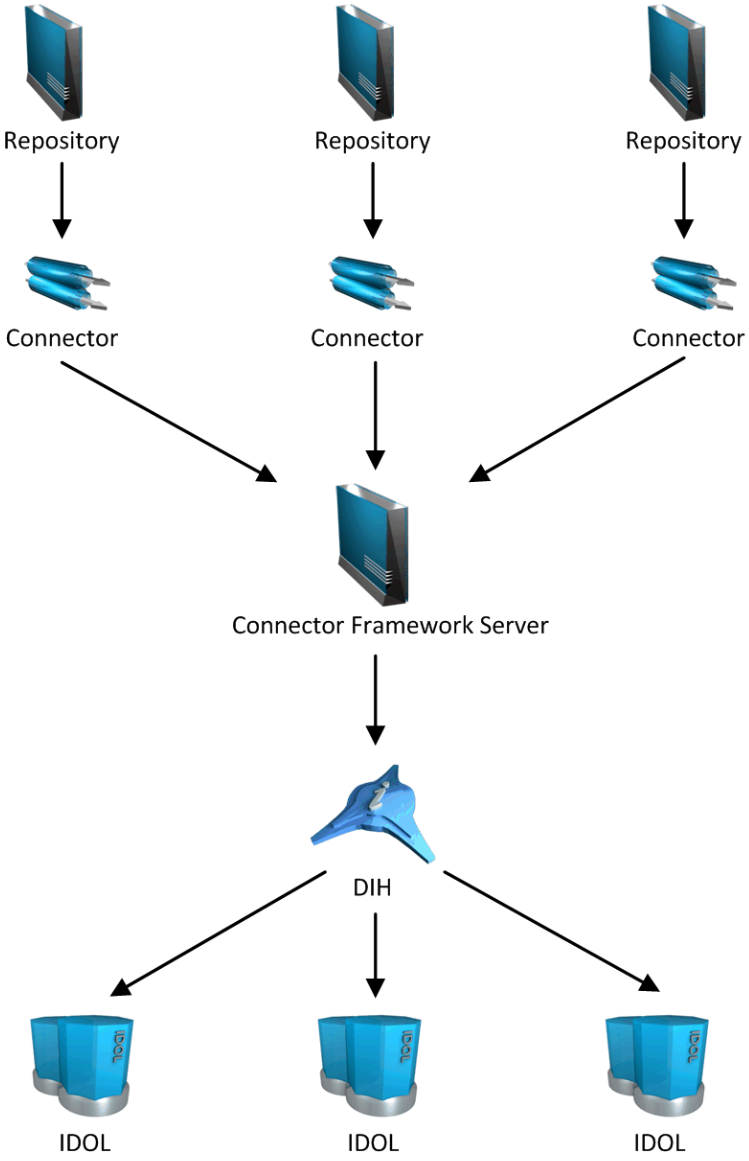
An IDOL infrastructure can include the following components:

- **Connectors**. Connectors extract data from repositories and send the data to CFS.
- **Connector Framework Server**.
- **IDOL Server**. IDOL Server provides features to analyze unstructured information and extract meaning from that information.
- **Distributed Index Handler (DIH)**. The Distributed Index Handler distributes data across multiple IDOL servers. Using multiple IDOL servers can increase the availability and scalability of the system.

These components can be installed in many different configurations. The simplest installation consists of a single connector, a single CFS, and a single IDOL server.



A more complex configuration might include more than one connector, or use a Distributed Index Handler (DIH) to index content across multiple IDOL servers.



OEM Certification

Connector Framework Server works in OEM licensed environments.

Related Documentation

The following documents provide more details on Connector Framework Server.

- *Connector Framework Server Reference*

  The *Connector Framework Server Reference* describes the configuration parameters and actions that are supported by CFS.

- *IDOL Server Administration Guide*

  The *IDOL Server Administration Guide* describes the operations that IDOL Server can perform, and describes how to set them up.

- *Distributed Index Handler (DIH) Administration Guide*

  This guide describes how you can use a DIH to distribute aggregated documents across multiple IDOL Servers.

- *License Server Administration Guide*

  This guide describes how to use a License Server to license multiple IDOL services.

Display Online Help

You can display the Connector Framework Server Reference by sending an action from your web browser. The Connector Framework Server Reference describes the actions and configuration parameters that you can use with Connector Framework Server.

For Connector Framework Server to display help, the help data file (`help.dat`) must be available in the installation folder.

To display help for Connector Framework Server

1. Start Connector Framework Server.
2. Send the following action from your web browser:

   ```
   http://host:port/action=Help
   ```

   where:

   host is the IP address or name of the machine on which Connector Framework Server is installed.

   port is the ACI port by which you send actions to Connector Framework Server (set by the `Port` parameter in the `[Server]` section of the configuration file).

   For example:

   ```
   http://12.3.4.56:9000/action=help
   ```

Chapter 2: Configure Connector Framework Server

This section describes how to configure CFS.

Connector Framework Server Configuration File

To configure CFS, modify the configuration file. The file is located in the CFS installation folder and can be modified with a text editor.

The parameters in the configuration file are divided into sections that represent CFS functionality. CFS supports standard Server, Service, Logging, and License parameters.

Service Section

The `[Service]` section specifies the service port used by CFS.

Server Section

The `[Server]` section specifies the ACI port of the Connector Framework Server. When you configure connectors, the `IngestPort` parameter in the connector configuration file should point to this port.

Actions Section

The `[Actions]` section specifies how CFS processes actions that are sent to the ACI port.

Logging Section

The `[Logging]` section contains configuration parameters that determine how messages are logged. You can create separate log streams for different message types. The configuration file also contains a section to configure each of the log streams.

Indexing Section

The `[Indexing]` section specifies the host name or IP address, and port, of machines where data is sent after it has been processed by CFS. This is usually the IP address and ACI port of an IDOL Server. You can use other indexing parameters to specify how data is indexed.

ImportService Section

The `[ImportService]` section specifies details for KeyView.

ImportTasks Section

The `[ImportTasks]` section is used to set up custom import tasks. CFS performs these tasks on data before it is indexed into IDOL Server. For more information about Import Tasks, see Manipulate and Enrich Documents, on page 48.

IndexTasks Section

The `[IndexTasks]` section is used to set up custom index tasks. IDOL connectors detect when documents are updated or removed from a repository. The connectors pass this information to CFS so that the documents can be updated or removed from IDOL Server. When CFS receives this information, it can perform custom Index tasks before the information is sent to IDOL. For more information about Index tasks, see Manipulate and Enrich Documents, on page 48.

Related Topics

- Customize Logging, on page 109

Modify Configuration Parameter Values

You modify Connector Framework Server configuration parameters by directly editing the parameters in the configuration file. When you set configuration parameter values, you must use UTF-8.

> **CAUTION:** You must stop and restart Connector Framework Server for new configuration settings to take effect.

This section describes how to enter parameter values in the configuration file.

Enter Boolean Values

The following settings for Boolean parameters are interchangeable:

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

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

Enter String Values

To enter a comma-separated list of strings when one of the strings contains a comma, you can indicate the start and the end of the string with quotation marks, for example:

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

Alternatively, you can escape the comma with a backslash:

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

If any string in a comma-separated list contains quotation marks, you must put this string into quotation marks and escape each quotation mark in the string by inserting a backslash before it. For example:

```
ParameterName="<font face=\"arial\" size=\"+1\"><b>","<p>"
```

Here, quotation marks indicate the beginning and end of the string. All quotation marks that are contained in the string are escaped.

Configure Connector Framework Server

This section describes how to configure CFS.

To configure CFS

1. Stop CFS, if it is running.
2. Open the CFS configuration file.
3. In the `[Service]` section, specify the service port:

`ServicePort`	The port for CFS to use as the service port.

4. In the `[Server]` section, set the ACI port:

`Port`	The port for CFS to use as the ACI port.

5. (Optional) In the `[ImportService]` section, you can set parameters to configure KeyView. You can choose the number of threads to use, specify the folders to use for extracting files, and customize how documents are imported.

`ThreadCount`	The number of threads to use for importing documents.

   For information about the configuration parameters that you can set, refer to the *Connector Framework Server Reference*.
6. Save the configuration file.

Related Topics

- Connector Framework Server Configuration File, on page 18
- Customize Logging, on page 109

Include an External Configuration File

You can share configuration sections or parameters between ACI server configuration files. The following sections describe different ways to include content from an external configuration file.

You can include a configuration file in its entirety, specified configuration sections, or a single parameter.

When you include content from an external configuration file, the `GetConfig` and `ValidateConfig` actions operate on the combined configuration, after any external content is merged in.

In the procedures in the following sections, you can specify external configuration file locations by using absolute paths, relative paths, and network locations. For example:

```
../sharedconfig.cfg
K:\sharedconfig\sharedsettings.cfg
\\example.com\shared\idol.cfg
file://example.com/shared/idol.cfg
```

Relative paths are relative to the primary configuration file.

> **NOTE:** You can use nested inclusions, for example, you can refer to a shared configuration file that references a third file. However, the external configuration files must not refer back to your original configuration file. These circular references result in an error, and Connector Framework Server does not start.
>
> Similarly, you cannot use any of these methods to refer to a different section in your primary configuration file.

Include the Whole External Configuration File

This method allows you to import the whole external configuration file at a specified point in your configuration file.

To include the whole external configuration file

1. Open your configuration file in a text editor.
2. Find the place in the configuration file where you want to add the external configuration file.
3. On a new line, type a left angle bracket (<), followed by the path to and name of the external configuration file, in quotation marks (""). You can use relative paths and network locations. For example:

```
< "K:\sharedconfig\sharedsettings.cfg"
```

4. Save and close the configuration file.

Include Sections of an External Configuration File

This method allows you to import one or more configuration sections (including the section headings) from an external configuration file at a specified point in your configuration file. You can include a whole configuration section in this way, but the configuration section name in the external file must exactly match what you want to use in your file. If you want to use a configuration section from the external file with a different name, see Merge a Section from an External Configuration File, on the next page.

To include sections of an external configuration file

1. Open your configuration file in a text editor.
2. Find the place in the configuration file where you want to add the external configuration file section.
3. On a new line, type a left angle bracket (<), followed by the path of the external configuration file, in quotation marks (""). You can use relative paths and network locations. After the configuration file path, add the configuration section name that you want to include. For example:

   ```
   < "K:\sharedconfig\extrasettings.cfg" [License]
   ```

   > **NOTE:** You cannot include a section that already exists in your configuration file.

4. Save and close the configuration file.

Include Parameters from an External Configuration File

This method allows you to import one or more parameters from an external configuration file at a specified point in your configuration file. You can import a single parameter or use wildcards to specify multiple parameters. The parameter values in the external file must match what you want to use in your file. This method does not import the section heading, such as `[License]` in the following examples.

To include parameters from an external configuration file

1. Open your configuration file in a text editor.
2. Find the place in the configuration file where you want to add the parameters from the external configuration file.
3. On a new line, type a left angle bracket (<), followed by the path of the external configuration file, in quotation marks (""). You can use relative paths and network locations. After the configuration file path, add the name of the section that contains the parameter, followed by the parameter name. For example:

   ```
   < "license.cfg" [License] LicenseServerHost
   ```

To specify a default value for the parameter, in case it does not exist in the external configuration file, specify the configuration section, parameter name, and then an equals sign (=) followed by the default value. For example:

```
< "license.cfg" [License] LicenseServerHost=localhost
```

You can use wildcards to import multiple parameters, but this method does not support default values. The * wildcard matches zero or more characters. The ? wildcard matches any single character. Use the pipe character | as a separator between wildcard strings. For example:

```
< "license.cfg" [License] LicenseServer*
```

4. Save and close the configuration file.

Merge a Section from an External Configuration File

This method allows you to include a configuration section from an external configuration file as part of your Connector Framework Server configuration file. For example, you might want to specify a standard SSL configuration section in an external file and share it between several servers. You can use this method if the configuration section that you want to import has a different name to the one you want to use.

To merge a configuration section from an external configuration file

1. Open your configuration file in a text editor.
2. Find or create the configuration section that you want to include from an external file. For example:

   ```
   [SSLOptions1]
   ```

3. After the configuration section name, type a left angle bracket (<), followed by the path to and name of the external configuration file, in quotation marks (""). You can use relative paths and network locations. For example:

   ```
   [SSLOptions1] < "../sharedconfig/ssloptions.cfg"
   ```

   If the configuration section name in the external configuration file does not match the name that you want to use in your configuration file, specify the section to import after the configuration file name. For example:

   ```
   [SSLOptions1] < "../sharedconfig/ssloptions.cfg" [SharedSSLOptions]
   ```

   In this example, Connector Framework Server uses the values in the [SharedSSLOptions] section of the external configuration file as the values in the [SSLOptions1] section of the Connector Framework Server configuration file.

   > **NOTE:** You can include additional configuration parameters in the section in your file. If these parameters also exist in the imported external configuration file, Connector Framework Server uses the values in the local configuration file. For example:

```
[SSLOptions1] < "ssloptions.cfg" [SharedSSLOptions]
SSLCACertificatesPath=C:\IDOL\HTTPConnector\CACERTS\
```

4. Save and close the configuration file.

Encrypt Passwords

Micro Focus recommends that you encrypt all passwords that you enter into a configuration file.

Create a Key File

A key file is required to use AES encryption.

To create a new key file

1. Open a command-line window and change directory to the Connector Framework Server installation folder.
2. At the command line, type:

   ```
   autpassword -x -tAES -oKeyFile=./MyKeyFile.ky
   ```

   A new key file is created with the name `MyKeyFile.ky`

> **CAUTION:** To keep your passwords secure, you must protect the key file. Set the permissions on the key file so that only authorized users and processes can read it. Connector Framework Server must be able to read the key file to decrypt passwords, so do not move or rename it.

Encrypt a Password

The following procedure describes how to encrypt a password.

To encrypt a password

1. Open a command-line window and change directory to the Connector Framework Server installation folder.
2. At the command line, type:

   ```
   autpassword -e -tEncryptionType [-oKeyFile] [-cFILE -sSECTION -pPARAMETER]
   PasswordString
   ```

   where:

Option	Description
`-t`	The type of encryption to use:

Option	Description
EncryptionType	- **Basic** - **AES** - AES256 For example: **-tAES** **NOTE:** AES is more secure than basic encryption.
-o*KeyFile*	AES encryption requires a key file. This option specifies the path and file name of a key file. The key file must contain 64 hexadecimal characters. For example: **-oKeyFile=./key.ky**
-cFILE -sSECTION -pPARAMETER	(Optional) You can use these options to write the password directly into a configuration file. You must specify all three options. - **-c**. The configuration file in which to write the encrypted password. - **-s**. The name of the section in the configuration file in which to write the password. - **-p**. The name of the parameter in which to write the encrypted password. For example: **-c./Config.cfg -sMyTask -pPassword**
PasswordString	The password to encrypt.

For example:

```
autpassword -e -tBASIC MyPassword
```

```
autpassword -e -tAES -oKeyFile=./key.ky MyPassword
```

```
autpassword -e -tAES -oKeyFile=./key.ky -c./Config.cfg -sDefault -pPassword
MyPassword
```

The password is returned, or written to the configuration file.

Decrypt a Password

The following procedure describes how to decrypt a password.

To decrypt a password

1. Open a command-line window and change directory to the Connector Framework Server installation folder.

2. At the command line, type:

```
autpassword -d -tEncryptionType [-oKeyFile] PasswordString
```

where:

Option	Description
`-t` *`EncryptionType`*	The type of encryption: • **`Basic`** • **`AES`** - AES256 For example: **`-tAES`**
`-o`*`KeyFile`*	AES encryption and decryption requires a key file. This option specifies the path and file name of the key file used to decrypt the password. For example: **`-oKeyFile=./key.ky`**
`PasswordString`	The password to decrypt.

For example:

```
autpassword -d -tBASIC 9t3M3t7awt/J8A
```

```
autpassword -d -tAES -oKeyFile=./key.ky 9t3M3t7awt/J8A
```

The password is returned in plain text.

Configure Client Authorization

You can configure Connector Framework Server to authorize different operations for different connections.

Authorization roles define a set of operations for a set of users. You define the operations by using the `StandardRoles` configuration parameter, or by explicitly defining a list of allowed actions in the `Actions` and `ServiceActions` parameters. You define the authorized users by using a client IP address, SSL identities, and GSS principals, depending on your security and system configuration.

For more information about the available parameters, see the *Connector Framework Server Reference*.

> **IMPORTANT:** To ensure that Connector Framework Server allows only the options that you configure in `[AuthorizationRoles]`, make sure that you delete any deprecated *`Role`*`Clients` parameters from your configuration (where *`Role`* corresponds to a standard role name, for example `AdminClients`).

To configure authorization roles

1. Open your configuration file in a text editor.
2. Find the `[AuthorizationRoles]` section, or create one if it does not exist.

3. In the [AuthorizationRoles] section, list the user authorization roles that you want to create. For example:

   ```
   [AuthorizationRoles]
   0=AdminRole
   1=UserRole
   ```

4. Create a section for each authorization role that you listed. The section name must match the name that you set in the [AuthorizationRoles] list. For example:

   ```
   [AdminRole]
   ```

5. In the section for each role, define the operations that you want the role to be able to perform. You can set StandardRoles to a list of appropriate values, or specify an explicit list of allowed actions by using Actions, and ServiceActions. For example:

   ```
   [AdminRole]
   StandardRoles=Admin,ServiceControl,ServiceStatus

   [UserRole]
   Actions=GetVersion
   ServiceActions=GetStatus
   ```

   > **NOTE:** The standard roles do not overlap. If you want a particular role to be able to perform all actions, you must include all the standard roles, or ensure that the clients, SSL identities, and so on, are assigned to all relevant roles.

6. In the section for each role, define the access permissions for the role, by setting Clients, SSLIdentities, and GSSPrincipals, as appropriate. If an incoming connection matches one of the allowed clients, principals, or SSL identities, the user has permission to perform the operations allowed by the role. For example:

   ```
   [AdminRole]
   StandardRoles=Admin,ServiceControl,ServiceStatus
   Clients=localhost
   SSLIdentities=admin.example.com
   ```

7. Save and close the configuration file.

8. Restart Connector Framework Server for your changes to take effect.

> **IMPORTANT:** If you do not provide any authorization roles for a standard role, Connector Framework Server uses the default client authorization for the role (localhost for Admin and ServiceControl, all clients for Query and ServiceStatus). If you define authorization only by actions, Micro Focus recommends that you configure an authorization role that disallows all users for all roles by default. For example:
>
> ```
> [ForbidAllRoles]
> StandardRoles=*
> Clients=""
> ```
>
> This configuration ensures that Connector Framework Server uses only your action-based authorizations.

Chapter 3: Send Actions to Connector Framework Server

This section describes how to send actions to Connector Framework Server.

Send Actions to Connector Framework Server

Connector Framework Server actions are HTTP requests, which you can send, for example, from your web browser. The general syntax of these actions is:

```
http://host:port/action=action&parameters
```

where:

host	is the IP address or name of the machine where Connector Framework Server is installed.
port	is the Connector Framework Server ACI port. The ACI port is specified by the `Port` parameter in the `[Server]` section of the Connector Framework Server configuration file. For more information about the `Port` parameter, see the *Connector Framework Server Reference*.
action	is the name of the action you want to run.
parameters	are the required and optional parameters for the action.

> **NOTE:** Separate individual parameters with an ampersand (&). Separate parameter names from values with an equals sign (=). You must percent-encode all parameter values.

For more information about actions, see the *Connector Framework Server Reference*.

Asynchronous Actions

When you send an asynchronous action to Connector Framework Server, the CFS adds the task to a queue and returns a token. Connector Framework Server performs the task when a thread becomes available. You can use the token with the `QueueInfo` action to check the status of the action and retrieve the results of the action.

Most of the actions sent to CFS are `ingest` actions, so when you use the `QueueInfo` action, query the `ingest` action queue, for example:

```
/action=QueueInfo&QueueName=ingest&QueueAction=GetStatus
```

Check the Status of an Asynchronous Action

To check the status of an asynchronous action, use the token that was returned by Connector Framework Server with the `QueueInfo` action. For more information about the `QueueInfo` action, refer to the *Connector Framework Server Reference*.

To check the status of an asynchronous action

- Send the `QueueInfo` action to Connector Framework Server with the following parameters.

`QueueName`	The name of the action queue that you want to check.
`QueueAction`	The action to perform. Set this parameter to **`GetStatus`**.
`Token`	(Optional) The token that the asynchronous action returned. If you do not specify a token, Connector Framework Server returns the status of every action in the queue.

For example:

```
/action=QueueInfo&QueueName=ingest&QueueAction=getstatus&Token=...
```

Cancel an Asynchronous Action that is Queued

To cancel an asynchronous action that is waiting in a queue, use the following procedure.

To cancel an asynchronous action that is queued

- Send the `QueueInfo` action to Connector Framework Server with the following parameters.

`QueueName`	The name of the action queue that contains the action to cancel.
`QueueAction`	The action to perform . Set this parameter to **`Cancel`**.
`Token`	The token that the asynchronous action returned.

For example:

```
/action=QueueInfo&QueueName=ingest&QueueAction=Cancel&Token=...
```

Stop an Asynchronous Action that is Running

You can stop an asynchronous action at any point.

To stop an asynchronous action that is running

- Send the `QueueInfo` action to Connector Framework Server with the following parameters.

`QueueName`	The name of the action queue that contains the action to stop.
`QueueAction`	The action to perform. Set this parameter to **`Stop`**.
`Token`	The token that the asynchronous action returned.

For example:

```
/action=QueueInfo&QueueName=ingest&QueueAction=Stop&Token=...
```

Store Action Queues in an External Database

Connector Framework Server provides asynchronous actions. Each asynchronous action has a queue to store requests until threads become available to process them. You can configure Connector Framework Server to store these queues either in an internal database file, or in an external database hosted on a database server.

The default configuration stores queues in an internal database. Using this type of database does not require any additional configuration.

You might want to store the action queues in an external database so that several servers can share the same queues. In this configuration, sending a request to any of the servers adds the request to the shared queue. Whenever a server is ready to start processing a new request, it takes the next request from the shared queue, runs the action, and adds the results of the action back to the shared database so that they can be retrieved by any of the servers. You can therefore distribute requests between components without configuring a Distributed Action Handler (DAH).

> **NOTE:** You cannot use multiple servers to process a single request. Each request is processed by one server.

Prerequisites

- Supported databases:
  - PostgreSQL 9.0 or later.
  - MySQL 5.0 or later.

- On each machine that hosts Connector Framework Server, you must install an ODBC driver for your chosen database. On Linux you must also install the unixODBC driver manager and configure the name and path of the ODBC driver in the unixODBC `odbcinst.ini` configuration file.
- If you use PostgreSQL, you must set the PostgreSQL ODBC driver setting `MaxVarChar` to **`0`** (zero). If you use a DSN, you can configure this parameter when you create the DSN. Otherwise, you can set the `MaxVarcharSize` parameter in the connection string.

Configure Connector Framework Server

To configure Connector Framework Server to use a shared action queue, follow these steps.

To store action queues in an external database

1. Stop Connector Framework Server, if it is running.
2. Open the Connector Framework Server configuration file.
3. Find the relevant section in the configuration file:
    - To store queues for all asynchronous actions in the external database, find the `[Actions]` section.
    - To store the queue for a single asynchronous action in the external database, find the section that configures that action.
4. Set the following configuration parameters.

    `AsyncStoreLibraryDirectory`

    The path of the directory that contains the library to use to connect to the database. Specify either an absolute path, or a path relative to the server executable file.

    `AsyncStoreLibraryName`

    The name of the library to use to connect to the database. You can omit the file extension. The following libraries are available:

    - **`postgresAsyncStoreLibrary`** - for connecting to a PostgreSQL database.
    - **`mysqlAsyncStoreLibrary`** - for connecting to a MySQL database.

    `ConnectionString`

    The connection string to use to connect to the database. The user that you specify must have permission to create tables in the database. For example:

    ```
    ConnectionString=DSN=ActionStore
    ```

    or

    ```
    ConnectionString=Driver={PostgreSQL};
    ```

```
Server=10.0.0.1; Port=9876;
Database=SharedActions; Uid=user; Pwd=password;
MaxVarcharSize=0;
```

If your connection string includes a password, Micro Focus recommends encrypting the value of the parameter before entering it into the configuration file. Encrypt the entire connection string. For information about how to encrypt parameter values, see Encrypt Passwords, on page 24.

For example:

```
[Actions]
AsyncStoreLibraryDirectory=acidlls
AsyncStoreLibraryName=postgresAsyncStoreLibrary
ConnectionString=DSN=ActionStore
```

5. You can use the same database to store action queues for more than one type of IDOL component (for example, a group of File System Connectors and a group of Media Servers). To use a database for more than one type of component, set the following parameter in the `[Actions]` section of the configuration file.

`DatastoreSharingGroupName`	The group of components to share actions with. You can set this parameter to any string, but the value must be the same for each server in the group. For example, to configure several Connector Framework Servers to share their action queues, set this parameter to the same value in every Connector Framework Server configuration. Micro Focus recommends setting this parameter to the name of the component. **CAUTION:** Do not configure different components (for example, two different types of connector) to share the same action queues. This will result in unexpected behavior.

```
For example:

[Actions]
...
DatastoreSharingGroupName=MediaServer
```

6. Save and close the configuration file.

   When you start Connector Framework Server it connects to the shared database.

Store Action Queues in Memory

Connector Framework Server provides asynchronous actions. Each asynchronous action has a queue to store requests until threads become available to process them. These queues are usually stored in a datastore file or in a database hosted on a database server, but in some cases you can increase performance by storing these queues in memory.

> **NOTE:** Storing action queues in memory improves performance only when the server receives large numbers of actions that complete quickly. Before storing queues in memory, you should also consider the following:
>
> - The queues (including queued actions and the results of finished actions) are lost if Connector Framework Server stops unexpectedly, for example due to a power failure or the component being forcibly stopped. This could result in some requests being lost, and if the queues are restored to a previous state some actions could run more than once.
> - Storing action queues in memory prevents multiple instances of a component being able to share the same queues.
> - Storing action queues in memory increases memory use, so please ensure that the server has sufficient memory to complete actions and store the action queues.

If you stop Connector Framework Server cleanly, Connector Framework Server writes the action queues from memory to disk so that it can resume processing when it is next started.

To configure Connector Framework Server to store asynchronous action queues in memory, follow these steps.

To store action queues in memory

1. Stop Connector Framework Server, if it is running.
2. Open the Connector Framework Server configuration file and find the `[Actions]` section.
3. If you have set any of the following parameters, remove them:
   - `AsyncStoreLibraryDirectory`
   - `AsyncStoreLibraryName`
   - `ConnectionString`
   - `UseStringentDatastore`
4. Set the following configuration parameters.

`UseInMemoryDatastore`	A Boolean value that specifies whether to keep the queues for asynchronous actions in memory. Set this parameter to **`TRUE`**.
`InMemoryDatastoreBackupIntervalMins`	(Optional) The time interval (in minutes) at which the action queues are written to disk. Writing the queues to disk can reduce the number of queued actions that would be lost if Connector Framework Server stops unexpectedly, but configuring a frequent backup will increase the load on the datastore and might reduce performance.

For example:

```
[Actions]
UseInMemoryDatastore=TRUE
InMemoryDatastoreBackupIntervalMins=30
```

5. Save and close the configuration file.

   When you start Connector Framework Server, it stores action queues in memory.

Use XSL Templates to Transform Action Responses

You can transform the action responses returned by Connector Framework Server using XSL templates. You must write your own XSL templates and save them with either an `.xsl` or `.tmpl` file extension.

After creating the templates, you must configure Connector Framework Server to use them, and then apply them to the relevant actions.

To enable XSL transformations

1. Ensure that the `autnxslt` library is located in the same directory as Connector Framework Server. If the library is not included in your installation, you can obtain it from Micro Focus Support.
2. Open the Connector Framework Server configuration file in a text editor.
3. In the `[Server]` section, ensure that the `XSLTemplates` parameter is set to **`true`**.

   > **CAUTION:** If `XSLTemplates` is set to **`true`** and the `autnxslt` library is not present in the same directory as the configuration file, the server will not start.

4. (Optional) In the `[Paths]` section, set the `TemplateDirectory` parameter to the path to the directory that contains your XSL templates. The default directory is `acitemplates`.
5. Save and close the configuration file.
6. Restart Connector Framework Server for your changes to take effect.

To apply a template to action output

- Add the following parameters to the action:

`Template`	The name of the template to use to transform the action output. Exclude the folder path and file extension.
`ForceTemplateRefresh`	(Optional) If you modified the template after the server started, set this parameter to **`true`** to force the ACI server to reload the template from disk rather than from the cache.

For example:

```
action=QueueInfo&QueueName=Ingest
                &QueueAction=GetStatus
                &Token=...
                &Template=myTemplate
```

In this example, Connector Framework Server applies the XSL template `myTemplate` to the response from an `Ingest` action.

> **NOTE:** If the action returns an error response, Connector Framework Server does not apply the XSL template.

Example XSL Templates

Connector Framework Server includes the following sample XSL templates, in the `acitemplates` folder:

XSL Template	Description
`LuaDebug`	Transforms the output from the `LuaDebug` action, to assist with debugging Lua scripts.

Chapter 4: Ingest Data

This section describes how to send data to CFS.

Ingest Data using Connectors

To configure a connector to send data to CFS, follow these steps.

To configure a connector to send data to CFS

1. Stop the connector, if it is running. For information about how to stop a connector, refer to the connector’s documentation.
2. Open the connector’s configuration file in a text editor.
3. In the `[Ingestion]` section, set the following parameters:

`EnableIngestion`	To enable ingestion, set this parameter to **`true`**.
`IngesterType`	To send data to CFS, set this parameter to **`CFS`**.
`IngestHost`	The host name or IP address of the CFS.
`IngestPort`	The ACI port of the CFS.

   For example:

   ```
   [Ingestion]
   EnableIngestion=True
   IngesterType=CFS
   IngestHost=localhost
   IngestPort=7000
   ```

4. Save and close the configuration file.

   You can now start the connector.

Ingest an IDX File

You can ingest an IDX file using the `Ingest` action.

Use the `adds` parameter to specify the document that you want to ingest. This parameter takes XML like the following example which ingests `c:\data.idx`:

```
<adds>
   <add>
      <source filename="c:\data.idx" />
   </add>
</adds>
```

The XML must be URL encoded:

```
http://server:port/action=ingest&adds=[URL encoded XML]
```

For more information about the `Ingest` action, refer to the *Connector Framework Server Reference*.

Ingest XML

Many systems export information in XML format and CFS has features to help you convert XML into IDOL documents.

> **NOTE:** The XML must be encoded in UTF-8.

You can configure CFS to transform XML files, with an XSL transformation, before they are processed. This is an optional step but can be useful in cases where your XML files do not resemble IDOL documents or you are processing XML from many sources and the files have different schemas. You can configure any number of transformations and CFS runs the first transformation where the ingested XML matches the specified schema. You can also configure a default transformation that CFS runs when an XML file does not match any of your schemas.

After an XML file has been transformed, or when transformation is not configured, CFS attempts to convert the XML into IDOL documents. The XML is parsed according to the rules that you configure in the `[XmlParsing]` section of the CFS configuration file. If the conversion is successful, the resulting metadata-only documents are added to the ingest queue (for more information about the ingestion process, see The Ingestion Process, on page 11). If the conversion does not result in any IDOL documents but the XML was transformed after matching a schema, CFS does not consider this as a failure and does not index any documents. Otherwise, for example if the XML is invalid, the XML file is added to the import queue so that it is processed by KeyView along with other file types.

Transform XML Files

CFS can transform XML files before attempting to parse them. XSL transformations are configured in the `[XmlTransformation]` section of the CFS configuration file.

To run a single transformation, you can specify the settings in the `[XmlTransformation]` section:

```
[XmlTransformation]
ValidationSchema=schema.xsd
TransformationStylesheet=transform.xslt
```

In this example, CFS uses the stylesheet `transform.xslt` to transform any XML file that matches `schema.xsd`.

If you are processing XML files that have more than one schema, you might want to configure several transformations. To do this, use the `Sections` parameter to specify the names of sections that configure the transformations:

```
[XmlTransformation]
Sections=XmlTransform1,XmlTransform2

[XmlTransform1]
ValidationSchema=schema1.xsd
TransformationStylesheet=transform1.xslt

[XmlTransform2]
TransformationStylesheet=transform2.xslt
```

In this example, any XML file that matches `schema1.xsd` is transformed by `transform1.xslt`. These files are then parsed. The parameter `ValidationSchema` is not set in the section `XmlTransform2`, so any files that do not match `schema1.xsd` are transformed by `transform2.xslt`.

You can configure as many different transformations as you require. If you set the parameter `ValidationSchema` in every section and an XML file does not match any of the schemas, it is not transformed.

Parse XML into Documents

CFS attempts to parse any XML file that it receives according to rules that are specified in the `[XMLParsing]` section of its configuration file. The parameters in the `[XMLParsing]` section specify:

- How to divide the XML into documents.
- How to populate each document's `DREREFERENCE` field.
- How to populate each document's `DRECONTENT` field.

To configure settings for parsing XML

1. Open the CFS configuration file.
2. In the `[XMLParsing]` section, set the following parameters:

`DocumentRootPaths`	A comma-separated list of paths to nodes that contain a single document. Specify the paths relative to the root of the XML. Use a forward slash (/) to represent levels in the XML hierarchy. Any elements

contained within the specified node are added to the document as metadata.

`IncludeRootPath` A Boolean value (default `false`) that specifies whether to include the node specified by `DocumentRootPaths` in the document. You might set this parameter to **`TRUE`** if the root node has attributes that you need to include in the document.

`ReferencePaths` A comma-separated list of possible paths to a node that contains the document reference. Specify the paths relative to the node identified by `DocumentRootPaths`. Use a forward slash (/) to represent levels in the XML hierarchy. The XML for each document must contain exactly one node that matches the specified path(s).

`ContentPaths` A comma-separated list of possible paths to a node that contains the document content. Specify the paths relative to the node identified by `DocumentRootPaths`. Use a forward slash (/) to represent levels in the XML hierarchy. If multiple content nodes are identified for a single document, a document is produced with multiple sections.

3. Save and close the configuration file.

Example

Consider the following XML:

```
<xml>
   <documents>
      <document>
         <metadata>
           <name>This is the name of the document</name>
           <created>28/02/15 11:01:17</created>
           <modified>28/02/15 15:23:00</modified>
         </metadata>
         <content>Here is some content</content>
       </document>
       <document>
         <metadata>
           <name>This is another document</name>
           <created>01/03/15 12:21:13</created>
           <modified>02/03/15 13:23:03</modified>
         </metadata>
         <different_content>Here is some content</different_content>
       </document>
     </documents>
 </xml>
```

To ingest this XML file, you might use the following configuration:

```
[XMLParsing]
DocumentRootPaths=documents/document
ReferencePaths=metadata/name
ContentPaths=content,different_content
```

To ingest the XML, send the ingest action to CFS:

```
http://localhost:7000/action=ingest&adds=%3Cadds%3E%3Cadd%3E%3Csource%20
                                         filename%3D%22xmlfile.xml%22%20
                                         lifetime%3D%22permanent%22%20%2F%3E
                                         %3C%2Fadd%3E%3C%2Fadds%3E
```

This would produce the following documents:

```
#DREREFERENCE This is the name of the document
#DREFIELD UUID="bfa1a8aac0b772d1ee467d830fa179bc"
#DREFIELD DocTrackingId="3cd0e5cf3160163adf7445d013ef10b1"
#DREFIELD ImportVersion="1207655"
#DREFIELD KeyviewVersion="10220"
#DREFIELD metadata/created="28/02/15 11:01:17"
#DREFIELD metadata/modified="28/02/15 15:23:00"
#DRECONTENT
Here is some content
#DREENDDOC

#DREREFERENCE This is another document
#DREFIELD UUID="aadf6628fccd0c6b885a79e2e39f4357"
#DREFIELD DocTrackingId="66a63287d85b500159c5b5fb099b99a5"
#DREFIELD ImportVersion="1207655"
#DREFIELD KeyviewVersion="10220"
#DREFIELD metadata/created="01/03/15 12:21:13"
#DREFIELD metadata/modified="02/03/15 13:23:03"
#DRECONTENT
Here is some content
#DREENDDOC
```

Ingest JSON

Many systems export information in JSON format and CFS has features to help you convert JSON into IDOL documents.

JSON files are parsed as follows:

- Any document that has the metadata field `AUTN_JSON_PARSING_SECTION` is parsed using the settings specified in the section of the CFS configuration file that matches the field value.
- All other JSON documents are parsed using the settings specified in the `[JSONParsing]` section of the CFS configuration file.

For information about the JSON parsing configuration parameters, refer to the *Connector Framework Server Reference*.

The CFS JSON reader accepts a stream of JSON document objects:

```
{"DREREFERENCE": "one"}{"DREREFERENCE": "two"}{"DREREFERENCE": "three"}
```

Or an array of JSON document objects:

```
[{"DREREFERENCE": "one"},{"DREREFERENCE": "two"},{"DREREFERENCE": "three"}]
```

Example

Consider the following JSON:

```
{
    "ref": "12345abcd",
    "someField": [
        "value1",
        "value2"
    ],
    "otherfield": {
        "field_name": "myfield",
        "@attribute": true,
        "attrs": {
            "attr1": "fish",
            "attr2": "chips",
            "attr3": "peas"
        },
        "field_value": {
            "attr_name": "key",
            "attr_value": "value",
            "interesting": 7
        },
        "someField": [
            12.9
        ]
    },
    "content_stuff": {
        "1": "page1",
        "2": [
            "page2",
            "page3"
        ]
    }
}
```

To ingest this JSON, you might use the following configuration:

```
[JSONParsing]
ReferencePath=ref
ContentPaths=content_stuff/^.*$
FieldSections=SomeField,OtherField
ExpandArrays=true

[SomeField]
Path=someField/^.*$

[OtherField]
Path=otherfield
NamePath=field_name
ValuePath=field_value/interesting
FieldSections=SomeField
AttributeSections=OtherFieldAttr,OtherFieldAttr2,OtherFieldAttr3

[OtherFieldAttr]
Path=^@(attribute)$

[OtherFieldAttr2]
NamePath=field_value/attr_name
ValuePath=field_value/attr_value

[OtherFieldAttr3]
Path=attrs/^.*$
```

To ingest the JSON, send the ingest action to CFS:

```
http://localhost:7000/action=ingest&adds=%3Cadds%3E%3Cadd%3E%3Csource%20
                                         filename%3D%22file.json%22%20
                                         lifetime%3D%22permanent%22%20%2F%3E
                                         %3C%2Fadd%3E%3C%2Fadds%3E
```

This would produce the following document:

```
#DREREFERENCE 12345abcd
#DREFIELD myfield="7"
#DREFIELD myfield/@attribute="true"
#DREFIELD myfield/@key="value"
#DREFIELD myfield/@attr1="fish"
#DREFIELD myfield/@attr2="chips"
#DREFIELD myfield/@attr3="peas"
#DREFIELD myfield/someField="12.9"
#DREFIELD someField="value1"
#DREFIELD someField="value2"
#DRESECTION 0
#DRECONTENT
page1
```

```
#DREENDDOC

#DREREFERENCE 12345abcd
#DRESECTION 1
#DRECONTENT
page2
#DREENDDOC

#DREREFERENCE 12345abcd
#DRESECTION 2
#DRECONTENT
page3
#DREENDDOC
```

Ingest PST Files

Consider the following points before ingesting Microsoft Outlook Personal Folders (PST) files:

- The best results are usually obtained when KeyView uses MAPI to extract and filter PST files. To use MAPI, you must:
  - Run CFS on Windows.
  - Install Microsoft Outlook on the same machine as CFS. If you are using 64-bit CFS, install 64-bit Outlook. If you are using 32-bit CFS, install 32-bit Outlook.
  - Ensure that MAPI has write access to the PST files. Set the `WorkingDirectory` parameter in the `[ImportService]` section of the CFS configuration file so that CFS copies files to a working directory and processes the copies, rather than processing the files in their original location.
- PST files can contain a large amount of data and KeyView might not finish processing them within the default time limit allowed by CFS. Consider increasing the value of the `KeyviewTimeout` parameter, in the `[ImportService]` section of the CFS configuration file.

Ingest Password-Protected Files

To process password-protected files you must provide CFS with the passwords.

To specify the passwords for password-protected files

1. Create a credentials file to contain the passwords for your password-protected files:
   a. Open a text editor and create a new text file.
   b. Create an `[ImportService]` section in the file.
   c. In the `ImportService` section, set the following parameter:

`ImportCredentialCount` The total number of file name and password combinations specified in the credentials file.

For example:

```
[ImportService]
ImportCredentialCount=1
```

d. Create a new section in the file, named `[Credential`*`N`*`]`, where *N* is the number of the file name/password combination, starting from 0.

In the new section, set the following parameters:

`FileSpec` The name of the password protected file(s). You can use the `*` wildcard to match the file name(s).

`Password` The password for the file(s). You can encrypt the password using the password encryption utility. For information about how to do this, see Encrypt Passwords, on page 24.

`UserName` The user name to use to open the file(s). Set this parameter if a user name is required to access the file.

`NotesIDFile` The path of the ID file. Set this parameter for `.nsf` files only.

For example, the following settings could be used to specify a single password for all ZIP files:

```
[ImportService]
ImportCredentialCount=1

[Credential0]
FileSpec=*.zip
Password=9t3M3t7awt/J8A
```

e. To specify further file name and password combinations, repeat steps c and d.

f. Save the file to a suitable location.

2. Specify the location of the credentials file. There are several ways to do this:

   - To use the credentials file you created for all ingested documents, set the CFS configuration parameter `ImportCredentialFile` to the path of the file. For more information about this parameter, refer to the *Connector Framework Server Reference*.
   - To use the credentials file that you created to process a single document, set the document field `AUTN_CREDENTIALS`. This field accepts either the path to the credentials file, or the credentials file content. You can encrypt the text using the password encryption utility. The `AUTN_CREDENTIALS` field is removed from all documents before they are indexed. When you send an `ingest` action to CFS, you can set this field using the `xmlmetadata` element in the `adds` or `updates` action parameter. For more information about the ingest action, refer to the *Connector Framework Server Reference*.

Ingest Data for Testing

To ingest data for testing purposes, use the IngestTest action. You can use this action to view the output of the ingestion process for a small amount of data, without the data being indexed into IDOL.

> **TIP:** CFS includes an XSL template to help you send IngestTest actions. Open a web browser and navigate to http://*host*:7000/action=IngestTest&Template=IngestTest (where *host* is the machine where CFS is running and 7000 is the CFS ACI port).
>
> Micro Focus does not support the XSL template, it is provided only as an example of a template that you could build.

The IngestTest action has the following parameters:

```
/action=IngestTest
             &config=[base64_encoded_config]
             &adds=[URL_encoded_adds_xml]
```

IngestTest is similar to the Ingest action, but has the following differences which make it suitable for testing:

- IngestTest is a synchronous action, and the document data is returned in the ACI response.
- Indexing, whether as a result of ingestion or as a result of an import task, is disabled.
- Update and Delete commands are disabled (you cannot use the updates and removes action parameters like you can with the Ingest action).
- Any writer tasks that have been configured (IdxWriter, XmlWriter, JsonWriter, CsvWriter, SqlWriter) are disabled.
- Logging to the import log stream is disabled. The log messages are redirected to the action response.
- The global Lua variable is_test is set to true. You can use this variable in your Lua scripts to prevent certain parts of your scripts from running when you use the IngestTest action.

For more information about the IngestTest action and its parameters, refer to the *Connector Framework Server Reference*.

Chapter 5: Filter Documents and Extract Subfiles

CFS automatically extracts metadata, content, and sub-files from all files that are ingested. KeyView does not need to be configured, but this section describes how to customize the filtering and extraction process.

Customize KeyView Filtering

If necessary, you can customize the filtering and extraction process. For example, you can choose whether to extract comments added by reviewers to a Microsoft Word document.

To customize filtering, use the Import Service parameters, in the `[ImportService]` section of the CFS configuration file. For information about the parameters that you can set, refer to the *Connector Framework Server Reference*.

You can also customize KeyView filtering by modifying the configuration parameters in the KeyView `filters\formats.ini` configuration file. For more information about customizing KeyView filtering by modifying `formats.ini`, refer to the KeyView documentation.

Disable Filtering or Extraction for Specific Documents

To prevent KeyView from processing specific documents, you can add the following fields to documents. You can add the fields with any value.

AUTN_FILTER_META_ONLY
Prevents CFS extracting content from a file. CFS only extracts metadata and adds this information to the document.

AUTN_NO_FILTER
Prevents CFS extracting any text (metadata or content) from a file. This can be useful if you do not want to extract text from certain file types.

AUTN_NO_EXTRACT
Prevents CFS from extracting sub-files. This can be useful if you want to avoid extracting items from ZIP files and other container files.

> **NOTE:** To add a field to a document, use a Lua script. You must run the Lua script using a *Pre* import task. This is because *Post* import tasks run after KeyView filtering.

Related Topics

- Write and Run Lua Scripts, on page 53
- Add a Field to a Document, on page 60

Chapter 6: Manipulate and Enrich Documents

This section describes how to manipulate and enrich documents using CFS.

Introduction

The documents produced by connectors and CFS contain information extracted from the source repository. In many cases you might want to add additional information to documents, or modify the structure of the documents, before they are indexed.

To modify documents before they are indexed, use *Import Tasks* and *Index Tasks*. These are customizable processing tasks that you can run on documents. You can use these tasks to write documents to disk, manipulate documents, reject documents, and run custom Lua scripts.

Write documents to disk

You can write documents to disk in IDX or XML format. This allows you to view the information that is being indexed, so that you can check the information is being indexed as you expected. If necessary, you can then use other import tasks to manipulate and enrich the information.

Manipulate and enrich documents

You can use import tasks to enrich documents. For example, you can:

- extract the meaningful content from HTML, and discard advertisements, headers, and sidebars.
- divide document content into sections. Dividing a document can result in more relevant query results, because IDOL can return a specific part of a document in response to a query.
- extract speech from audio and video files, and write a transcription of the speech to the document content. IDOL Server can then use the speech for retrieval, clustering, and other operations.

Validate and reject documents

You can reject documents that you do not want to index, for example those that do not appear to contain valid content. When a document is rejected, it is not processed further and is not indexed. However, you can index the document into an IDOL Server that has been configured to handle failed documents.

Run a Lua Script

Lua is an embedded scripting language that you can use to manipulate documents and define custom processing rules. CFS includes Lua functions for manipulating documents and running other tasks.

Choose When to Run a Task

Import Tasks run when new documents are processed by CFS, before the documents are indexed. You can run Import Tasks before and/or after KeyView filtering.

- *Pre*-import tasks run before KeyView filtering. At this point the document only contains metadata extracted from the repository by the connector.
- *Post*-import tasks run after KeyView filtering. At this point the document also contains any content and metadata that was extracted from the file associated with the document.

Index Tasks run when a document’s metadata (but not its content) is updated, or when a document is deleted. When a connector detects that document metadata has been updated or that a document has been deleted from a repository, it sends this information to CFS so that the document can be updated or removed from indexes such as IDOL Server.

- *Update* index tasks run when a document’s metadata (but not its content) is updated.
- *Delete* index tasks run when a document is deleted from a repository.

You can run some tasks, such as the `Lua` task, at any point during the import or indexing process.

You can run other tasks only at specific points within the import or indexing process. For example, to validate the content of documents you must use a post-import task. You cannot use a pre-import task because pre-import tasks occur before KeyView filtering, when documents do not contain any content.

The following table shows when you can run each type of task.

Task	Import Tasks		Index Tasks	
	Pre	Post	Update	Delete
Run a Lua script				
Lua	✓	✓	✓	✓
Write documents to disk				
CsvWriter	✓	✓	✓	✓
IdxWriter	✓	✓	✓	✓
JsonWriter	✓	✓	✓	✓
SqlWriter	✓	✓	✓	✓
XmlWriter	✓	✓	✓	✓
Manipulate and enrich documents				
Eduction		✓		
EmailAddressNormalisation	✓	✓		
ExtractMetadata	✓			
HtmlExtraction	✓			
ImportFile	✓	✓		
Sectioner		✓		

Task	Import Tasks		Index Tasks	
	Pre	Post	Update	Delete
Standardizer	✓	✓		
TextToDocs	✓			
Validate and reject documents				
BadFilesFilter		✓		
BinaryFileFilter		✓		
ImportErrorFilter		✓		
SymbolicContentFilter		✓		
WordLengthFilter		✓		
Media analysis				
MediaServerAnalysis	✓	✓		

You can also call many of the tasks from a Lua script, which allows more advanced processing. For example, you might want to run a task only on selected documents. For information about the Lua functions that are provided by CFS, refer to the *Connector Framework Server Reference*.

Related Topics

- The Import Process, on page 13.

Create Import and Index Tasks

Import tasks are configured in the `[ImportTasks]` section of the CFS configuration file. Import tasks run when files are imported, for example when a new item is retrieved from a repository or when the content of a file in a repository is updated. Use the `Pre` parameter to specify a list of tasks to run before KeyView filtering, and the `Post` parameter to specify a list of tasks to run after KeyView filtering.

Index tasks are configured in the `[IndexTasks]` section of the CFS configuration file. Use the `Update` parameter to specify a list of tasks to run when a connector instructs CFS to update the metadata of a document. Use the `Delete` parameter to specify a list of tasks to run when a connector instructs CFS to delete a document from indexes such as IDOL Server.

The tasks that you define run in sequence. In the following example, CFS creates an IDX file, then runs a Lua script, and then creates another IDX file:

```
[ImportTasks]
Post0=IdxWriter:C:\IDXArchive\before_script.idx
Post1=Lua:C:\Scripts\my_script.lua
Post2=IdxWriter:C:\IDXArchive\after_script.idx
```

To create an import task

1. Stop CFS.
2. Open the CFS configuration file.
3. Find the `[ImportTasks]` section of the configuration file, or create it if it does not exist.
4. Add the task by setting the `Pre` or `Post` parameter.

   The value of the `Pre` or `Post` parameter must be the name of the task that you want to run. Some tasks also require further information, such as the name of a file or the name of a section in the configuration file.

   For example, to run a Lua script before KeyView filtering:

   ```
   [ImportTasks]
   Pre0=Lua:myscript.lua
   ```

5. Some import tasks require you to identify the documents to process by adding a field to the documents. For example, the `IdolSpeech` task only runs on documents that have the `AUTN_NEEDS_TRANSCRIPTION` field. For more information about the document fields that are used with import tasks, see Document Fields for Import Tasks, on the next page.

   To add a field to the documents that you want to process, use a Lua script. In the following example, a Lua script named `Filter.Lua` runs before an `IdolSpeech` import task, to identify suitable documents and add the field `AUTN_NEEDS_TRANSCRIPTION`.

   ```
   [ImportTasks]
   Pre0=Lua:Filter.lua
   Pre1=IdolSpeech:IdolSpeechSettings
   ```

6. Save the configuration file and restart CFS.

To create an index task

1. Stop CFS.
2. Open the CFS configuration file.
3. Find the `[IndexTasks]` section of the configuration file, or create it if it does not exist.
4. Add the task by setting the `Update` or `Delete` parameter.

   The value of the `Update` or `Delete` parameter must be the name of the task that you want to run. Some tasks also require further information, such as the name of a file or the name of a section in the configuration file.

For example:

```
[IndexTasks]
Update0=Lua:myscript.lua
```

5. Save the configuration file and restart CFS.

Document Fields for Import Tasks

You can customize how documents are processed by import tasks, by adding the following fields to your documents.

> **NOTE:** The Lua script that adds the document fields must run before the import tasks.

AUTN_NEEDS_MEDIA_SERVER_ANALYSIS

To perform analysis on media files using the `MediaServerAnalysis` task, you must add this field to every document that you want to analyze. The field can have any value.

Write and Run Lua Scripts

Connector Framework Server supports Lua, an embedded scripting language. CFS supports all standard Lua functions. For more information about Lua, refer to http://www.lua.org/.

You can use a Lua script to:

- Add or modify document fields.
- Run built-in processing tasks, such as Eduction or image analysis.
- Call out to an external service, for example to alert a user.
- Interface with other libraries.

Write a Lua Script

Your Lua script must have the following structure:

```
function handler(document)
   ...
end
```

The `handler` function is called for each document and is passed a document object. The document object is an internal representation of the document being processed. Modifying this object changes the document.

For CFS to continue processing the document, the function must return **`true`**. If the function returns **`false`**, the document is discarded.

The script can also terminate due to an error, for example if you use the Lua `error` function or call a Lua function that causes an error. In this case CFS continues to process the document, but places an error message in the `ImportErrorDescription` field.

> **TIP:** You can write a library of useful functions to share between multiple scripts, which you can then include in the scripts by adding **`dofile("library.lua")`** to the top of the lua script outside of the `handler` function.

Run a Lua Script

To run a Lua script, create a `Lua` import or index task, and specify the path to your script. You can run Lua scripts using *pre* and *post* Import Tasks, and using *update* and *delete* index tasks. For example:

```
[ImportTasks]
Post0=Lua:c:\scripts\script1.lua
```

Debug a Lua Script

When you run a Lua script and the script fails due to an error, CFS writes the error to the import log stream, and to the `ImportErrorDescription` field of any documents that are affected.

To debug your Lua scripts, you can use the `LuaDebug` action. You can use this action to pause and resume scripts, and set and remove breakpoints. When a script is paused you can view the values of variables, view a stack trace, and step over single lines.

Sessions

CFS can have more than one import thread, and might run multiple Lua scripts concurrently. This means that you can have multiple Lua Debugging sessions. You might want to pause or continue running scripts on one thread but not others. Some of the commands available through the `LuaDebug` action allow or require you to specify a `session` action parameter. If the `session` parameter is optional and you do not specify a session, the command applies to all sessions. To view open sessions and obtain the values you can set for the `session` action parameter, use the command `/action=LuaDebug&command=get-status`.

Example

The following procedure demonstrates how to set a breakpoint in a script, view the values of Lua variables when the script is paused, and step over single lines. The actions in this procedure assume that your CFS is running on the local machine and is listening for actions on port 7000. For more information about the `LuaDebug` action, refer to the *Connector Framework Server Reference*.

To debug a Lua script

1. In the CFS configuration file, configure the script to run. This example uses the `AddLanguageDetectionFields` script that is included with CFS. For example:

   ```
   [ImportTasks]
   Post0=Lua:scripts/AddLanguageDetectionFields.lua
   ```

2. Start CFS.

3. To pause the script before a specific line is executed, set a breakpoint on that line. Line 33 of the `AddLanguageDetectionFields` script sends an action to IDOL Server and stores the response in a variable named `response`. To stop the script before this happens, use the following action:

   ```
   http://localhost:7000/action=luadebug
                              &command=set-breakpoint
                              &file=scripts/AddLanguageDetectionFields.lua
                              &line=33
   ```

4. (Optional) Confirm the breakpoint has been set using the `get-breakpoints` command:

   ```
   http://localhost:7000/action=luadebug&command=get-breakpoints
   ```

   CFS returns the response.

   ```
   <autnresponse xmlns:autn="http://schemas.autonomy.com/aci/">
      <action>LUADEBUG</action>
      <response>SUCCESS</response>
      <responsedata>
         <data>
            <command>get-breakpoints</command>
            <breakpoints>
               <breakpoint
   source="C:\Autonomy\ConnectorFramework\scripts\AddLanguageDetectionFields.lua"
   line="33"/>
            </breakpoints>
         </data>
      </responsedata>
   </autnresponse>
   ```

5. Send CFS an `IngestTest` action so that CFS ingests a document and runs the script:

   ```
   http://localhost:7000/action=IngestTest&adds=...
   ```

   > **TIP:** Use an `IngestTest` action, rather than an `Ingest` action, because the `IngestTest` action does not index any information into IDOL Server. For more information about the `IngestTest` action, see Ingest Data for Testing, on page 45.

   CFS runs the script. The `IngestTest` action does not finish (because the script is paused at the breakpoint) and therefore does not return a response.

6. Retrieve a token for the debugging session by sending CFS the `LuaDebug` command `get-status`:

```
http://localhost:7000/action=LuaDebug&command=get-status
```

CFS returns the response. You can see that there is a single debugging session and the Lua script has stopped at the breakpoint.

```
<autnresponse xmlns:autn="http://schemas.autonomy.com/aci/">
   <action>LUADEBUG</action>
   <response>SUCCESS</response>
   <responsedata>
      <data>
         <command>get-status</command>
         <session id="e4f7c45f561930cd17a5aed0fe1481d8">
            <status>AtBreak</status>
         </session>
      </data>
   </responsedata>
</autnresponse>
```

7. To retrieve the values of the Lua variables at the breakpoint, run the `LuaDebug` command `get-locals`. Use the session token that you retrieved with the `get-status` command:

   ```
   http://localhost:7000/action=LuaDebug
                              &command=get-locals
                              &session=e4f7c45f561930cd17a5aed0fe1481d8
   ```

   CFS returns the response.

   ```
   <autnresponse xmlns:autn="http://schemas.autonomy.com/aci/">
     <action>LUADEBUG</action>
     <response>SUCCESS</response>
     <responsedata>
       <data>
         <command>get-locals</command>
         <session id="e4f7c45f561930cd17a5aed0fe1481d8">
           <locals>
             ...
             <local name="idolHost" type="string">localhost</local>
             <local name="idolACIPort" type="number">9000</local>
             <local name="timeout" type="number">30000</local>
             ...
             ...
             <local type="string" name="detectionString">This is a document that
                  contains text in English. Automatic Language Detection will
                  detect the language and add the information to the document
             ...</local>
           </locals>
         </session>
       </data>
     </responsedata>
   </autnresponse>
   ```

8. Use the `step` command to run line 33. CFS does not run subsequent lines (the script will remain

paused). Use the session token you retrieved with the `get-status` command:

```
http://localhost:7000/action=LuaDebug
                           &command=step
                           &session=e4f7c45f561930cd17a5aed0fe1481d8
```

CFS returns the response.

```
<autnresponse xmlns:autn="http://schemas.autonomy.com/aci/">
  <action>LUADEBUG</action>
  <response>SUCCESS</response>
  <responsedata>
    <data>
      <command>step</command>
      <session id="e4f7c45f561930cd17a5aed0fe1481d8"/>
    </data>
  </responsedata>
</autnresponse>
```

9. To see what effect the step had on the variables, run the `get-locals` command again. You should see a new variable named `response` that contains the response from the `DetectLanguage` action.

```
http://localhost:7000/action=LuaDebug
                           &command=get-locals
                           &session=e4f7c45f561930cd17a5aed0fe1481d8
```

10. After examining the variables, you might want to remove the breakpoint. To remove the breakpoint, send CFS the following action:

```
http://localhost:7000/action=LuaDebug
                           &command=remove-breakpoint
                           &file=scripts/AddLanguageDetectionFields.lua
                           &line=33
```

CFS returns the response. You can also use `/action=LuaDebug&command=get-breakpoints` to confirm that the breakpoint has been removed.

11. To continue running the Lua script, use the `continue` command:

```
http://localhost:7000/action=LuaDebug
                           &command=continue
                           &session=e4f7c45f561930cd17a5aed0fe1481d8
```

CFS continues to run the script. The `IngestTest` action finishes and returns a response.

Lua Scripts Included With CFS

The CFS installation directory includes a `scripts` folder that includes the following Lua scripts:

Script	Description
`AddLanguageDetectionFields.lua`	Detects the language of a document’s content, using the

Script	Description
	IDOL Server action `DetectLanguage`. The script then adds fields describing the language and encoding to the document's metadata. The script demonstrates how to: • send an action to an ACI server. • parse the action response to a `LuaXmlDocument`. • use the methods of `LuaXmlDocument` to extract data from the document. The script assumes that an IDOL Server is installed on the local machine with an ACI port of 9000. You might need to modify these values. If you use this script, run it as a *post* import task so that it runs after KeyView has extracted document content.
`CategorySuggestFromText.lua`	Sends a document to IDOL for categorization, and adds information about the matching categories to the document's metadata. For information about how to use this script, see Categorize Documents, on page 69.
`filterdodgyfiles.lua`	An example script that demonstrates various ways to reject unwanted files, for example by file extension, by detecting the file format, or by analyzing the file content.
`identifiers.lua`	Adds sub-file indexes to the `AUTN_IDENTIFIER` document field of sub-files. This allows a connector to retrieve the sub-file, rather than the whole container, when the `collect` or `view` actions are used to retrieve the original file. If you use this script, you must run it as a *post* import task (so that it runs after KeyView processes the documents). For more information about the `AUTN_IDENTIFIER` field, see AUTN_IDENTIFIER, on page 212.
`mediaserver/*.lua`	These scripts run analysis on images, audio files, and video files by sending them to Media Server. For information about configuring media analysis, see Analyze Media, on page 62.

NOTE: CFS also includes scripts for use with Eduction. Some of these scripts are Eduction post processing scripts, which modify the output from an Eduction import task. The post processing scripts have the entry point `function processmatch (edkmatch)`, rather than `function handler (document)`. You must run a post processing script using the Eduction import task. Do not run an Eduction post processing script using a Lua task. For more information about Eduction Lua Post Processing, see Lua Post Processing, on page 74. For information about the Eduction scripts that are included with CFS, refer to the *Eduction User Guide.*

Use Named Parameters

Some Lua functions have an argument that takes named parameters. This argument is a table in which you can specify values for various parameters that affect the operation of the function.

You can specify a value for every parameter, or just those that you need. If you do not specify a value for a parameter, the function uses a default value. You can also specify the name of a configuration section and the function will read settings from that section in the CFS configuration file.

For example, when you call the function `looks_like_language`, you can set only the `term_file` named parameter, and use default values for the other settings:

```
looks_like_language(document, { term_file = "english.ocr" })
```

You might choose to set the `stop_list` parameter as well:

```
looks_like_language(document, { term_file = "english.ocr",
                       stop_list = "englishstoplist.dat" })
```

Alternatively, you can specify the name of a section in the CFS configuration file:

```
looks_like_language(document, { term_file = "english.ocr",
                       section = "LanguageSettings" })
```

In this example, the function uses the `english.ocr` term file. The settings for the remaining parameters are read from the `LanguageSettings` section of the CFS configuration file.

If you specify the name of a configuration section and use named parameters, the named parameters override any values set in the configuration file. In the following example, the `threshold` is set to 100, while other parameters (like `term_file`) are read from the `LanguageSettings` section:

```
looks_like_language(document, { section = "LanguageSettings",
                       threshold=100 })
```

For information about individual named parameters and corresponding configuration parameters, refer to the *Connector Framework Server Reference*.

Enable or Disable Lua Scripts During Testing

Lua scripts run by CFS can read a global Lua variable, `is_test`.

- When a script runs as part of an `Ingest` action, this variable is `false`.
- When a script runs as part of an `IngestTest` action, this variable is `true`.

You can use the `is_test` variable to enable or disable parts of a script. For example:

```
if is_test then
   -- The part of the script to enable for IngestTest
   -- (or disable for Ingest)
end

if not is_test then
```

```
   -- The part of the script to disable for IngestTest
   -- (or enable for Ingest)
end
```

Example Lua Scripts

This section contains example Lua scripts.

Add a Field to a Document

The following script demonstrates how to add a field named “MyField” to a document, with a value of “MyValue”.

```
function handler(document)
   document:addField("MyField", "MyValue");
   return true;
end
```

The following script demonstrates how to add the field `AUTN_NEEDS_IMAGE_SERVER_ANALYSIS` to all JPEG, TIFF and BMP documents. This field specifies that the documents can be processed using an `ImageServerAnalysis` import task (you must also define the task in the CFS configuration file).

The script finds the file type using the `DREREFERENCE` document field, so this field must contain the file extension for the script to work correctly.

```
function handler(document)
   local extensions_for_ocr = { jpg = 1 , tif = 1, bmp = 1 };
   local filename = document:getFieldValue("DREREFERENCE");
   local extension, extension_found =
      filename:gsub("^.*%.(%w+)$", "%1", 1);

   if extension_found > 0 then
      if extensions_for_ocr[extension:lower()] ~= nil then
         document:addField("AUTN_NEEDS_IMAGE_SERVER_ANALYSIS", "");
      end
   end

   return true;
end
```

Count Sections

For each document, this Lua script adds a total sections count to the title, and replaces the content of each section with the section number.

```
function handler(document)
   local section_count = 0;
   local section = document;
   while section do
      section_count = section_count + 1;
      section:setContent("Section "..section_count);
      section = section:getNextSection();
   end

   local title = document:getFieldValue("TITLE");

   if title == nil then title = "" end
   document:setFieldValue("TITLE", title .." Total Sections "
      ..section_count);

   return true;
end
```

Merge Document Fields

This script demonstrates how to merge the values of document fields.

When you extract data from a repository, CFS can produce documents that have multiple values for a single field, for example:

```
#DREFIELD ATTACHMENT="attachment.txt"
#DREFIELD ATTACHMENT="image.jpg"
#DREFIELD ATTACHMENT="document.pdf"
```

This script shows how to merge the values of these fields, so that the values are contained in a single field, for example:

```
#DREFIELD ATTACHMENTS="attachment.txt, image.jpg, document.pdf"
```

Example Script

```
function handler(document)
   onefield(document,"ATTACHMENT","ATTACHMENTS")
   return true;
end

function onefield(document,existingfield,newfield)
   if document:hasField(existingfield) then
      local values = { document:getFieldValues(existingfield) }
      local newfieldvalue=""

      for i,v in ipairs(values) do
         if i>1 then
            newfieldvalue = newfieldvalue ..", "
         end

         newfieldvalue = newfieldvalue..v
      end
      document:addField(newfield,newfieldvalue)
   end

   return true;
end
```

Add Titles to Documents

IDOL documents have a field named `DRETITLE` that can contain a title for the document. Front end applications might use the value of this field to present a title to users when displaying query results.

You should not rely on a connector to add a document title, because the connector might not be able to obtain this information. A suitable title for an e-mail message could be the subject of the e-mail, but this is not extracted until the e-mail is processed by CFS.

You can therefore use a Lua script to add a title to documents that do not have one, and, if necessary, ensure that all documents have suitable titles.

CFS includes a Lua script that adds titles to documents. The script is named `ExtractDreTitles.lua`, and is located in the `scripts` folder, in the CFS installation directory. You can use this script or modify it to suit your requirements.

The unmodified script ensures that all documents have a title. If a title has already been added to the document, that title is respected. If the document does not have a title, the script attempts to extract one from metadata fields that are added by KeyView and often contain titles. If none of these fields are present, the script adds a title by extracting the original file name from the field `DREORIGINALNAME`.

To add titles to documents using the `ExtractDreTitles` Lua script

1. Open the CFS configuration file.
2. Find the `[ImportTasks]` section of the configuration file, or create this section if it does not exist.
3. In the `[ImportTasks]` section, configure a `Post` import task to run the Lua script `scripts/ExtractDreTitles.Lua`.

   For example:

   ```
   [ImportTasks]
   Post0=Lua:scripts/ExtractDreTitles.lua
   ```

   > **TIP:** You must use a `Post` task so that the script runs after KeyView filtering.

4. Save and close the configuration file.

Analyze Media

Images, audio, and video are examples of unstructured information that represent a vast quantity of data. CFS extracts metadata from these files but cannot process their content, so by default documents that represent these files are indexed without any content.

To enrich documents that represent rich media files, you can send the files to an IDOL Media Server for analysis. Media Server can:

- extract text from scanned documents, and subtitles and scrolling text from video.
- identify people that appear by matching faces to a database of known faces.
- identify known logos and objects.
- detect and read barcodes, including QR codes.
- determine the language of speech in a video file, convert the speech into text, and identify any known speakers.

For more information about the types of analysis that you can run, refer to the *Media Server Administration Guide*.

> **NOTE:** Some types of analysis require you to train Media Server before you start processing.

Create a Media Server Configuration

To run analysis on media, you must create a Media Server session configuration file that instructs Media Server how to process the media. Micro Focus recommends that you save the configuration in a location accessible to CFS, and configure CFS to send the configuration to Media Server with each request.

Example Media Server configurations are provided with CFS in the `script_resources/mediaserver` directory.

The following example demonstrates the basic structure of a Media Server session configuration.

```
[Session]
IngestRate=0
Engine0=AudioVideo
//Engine0=Image
Engine1=FaceDetect
Engine2=OCR
Engine3=Response

[AudioVideo]
Type=Video

[Image]
Type=Image

[FaceDetect]
Type=FaceDetect
MinSize=70

[OCR]
Type=OCR
```

```
[Response]
Type=response
```

Ingestion

There is no single configuration that can process both images and video, so you must configure Media Server to ingest the correct type of media.

The example, above, demonstrates how to configure ingestion. To process audio or video files, set the parameter `Engine0=AudioVideo` so that Media Server uses the settings in the `[AudioVideo]` section of the configuration. To process image files (including PDF files and office documents that contain embedded images), set the parameter `Engine0=Image`. The Lua functions `analyze_media_in_document` and `analyze_media_in_file` allow you to override individual parameters, so if you request analysis from a Lua script you can first determine the file type and then set the value of the parameter when you call the function.

For more information about configuring ingestion, and the file types that are supported, refer to the *Media Server Administration Guide*.

Analysis

To add the analysis tasks that you want to run, list the task names using the `Engine` parameter in the `[Session]` section, and then create a matching section for each task.

Output

CFS expects Media Server to return the results of analysis in the `process` action response. You must create an output task to do this.

For more information about configuring a processing session in Media Server, refer to the *Media Server Administration Guide*.

Configure the Media Analysis Task

You can run media analysis on documents by using the `MediaServerAnalysis` import task. This task only processes documents that have the document field `AUTN_NEEDS_MEDIA_SERVER_ANALYSIS`, so you must add this field to any document that you want to process.

To configure the Media Server Analysis task

1. Write a Lua script to add the document field `AUTN_NEEDS_MEDIA_SERVER_ANALYSIS` to the documents that you want to analyze. For an example script that adds a field to a document, see Add a Field to a Document, on page 60.

2. Open the CFS configuration file.

3. In the `[ImportTasks]` section, configure a `Pre` or `Post` import task to run your Lua script. For

example:

```
[ImportTasks]
Pre0=Lua:scripts/TagVideoFiles.lua
```

4. Add another `Pre` or `Post` task to run the `MediaServerAnalysis` task. Set the `Pre` or `Post` parameter to `MediaServerAnalysis`, followed by a colon (`:`), followed by the name of the section in the CFS configuration file that contains the task settings. For example:

   ```
   Pre1=MediaServerAnalysis:MediaServerSettings
   ```

5. Create a new section in the configuration file, using the name you specified in Step 4.
6. In the new section, set the following parameters:

`MediaServerHost`	The host name and ACI port of your Media Server. To distribute requests between several servers, specify a comma-separated list of servers.
`MediaAnalysisTransform`	(Optional) To transform the metadata produced by Media Server, before CFS adds the data to your documents, set this parameter to the path of the XSL transformation to use. By default, CFS adds the information to your documents in a document field named `MediaServerAnalysis`, in the same structure that is returned from Media Server.

7. Specify the Media Server configuration file that you want to use for running analysis:
   - If you saved your configuration file in the directory specified by the `ConfigDirectory` parameter, in the `[Paths]` section of the Media Server configuration file, set `MediaServerConfigurationName` to the name of the configuration.
   - If you saved your configuration file in a location accessible by CFS, set the parameter `MediaServerConfigurationFileName` to the path of the configuration file. If you set a relative path, specify the path relative to CFS, not relative to Media Server.
8. Specify how to send media to Media Server:
   - If your Media Server can read files directly from the CFS working directory, set `ReadFromOriginalLocation=TRUE`.
   - To copy files to a shared folder, set the configuration parameter `MediaServerSharedPath`. This folder must be accessible to both CFS and Media Server. CFS copies files to the shared folder so that Media Server can read them. Micro Focus recommends that you use a shared folder for sending large files.
   - To send files to Media Server using HTTP POST requests, set neither `ReadFromOriginalLocation` nor `MediaServerSharedPath`.
9. Save and close the configuration file.

Examples

The following example shows how to configure the `MediaServerAnalysis` task. This example runs analysis using a configuration named `RecognizeFacesInVideo` that exists on the Media Server machine:

```
[ImportTasks]
Pre0=Lua:TagVideoFiles.lua
Pre1=MediaServerAnalysis:MediaServerSettings

[MediaServerSettings]
MediaServerHost=localhost:14000
MediaServerConfigurationName=RecognizeFacesInVideo
ReadFromOriginalLocation=TRUE
```

The following example is similar but configures CFS to send a configuration file to Media Server:

```
[ImportTasks]
Pre0=Lua:TagVideoFiles.lua
Pre1=MediaServerAnalysis:MediaServerSettings

[MediaServerSettings]
MediaServerHost=localhost:14000
MediaServerConfigurationFileName=./script_resources/mediaserver/facerecognition.cfg
ReadFromOriginalLocation=TRUE
```

If your CFS and Media Server are running on separate machines, you can configure CFS to copy media files to a shared folder:

```
[ImportTasks]
Pre0=Lua:TagVideoFiles.lua
Pre1=MediaServerAnalysis:MediaServerSettings

[MediaServerSettings]
MediaServerHost=media1:14000,media2:14000
MediaServerConfigurationName=RecognizeFacesInVideo
MediaServerSharedPath=\\server\videofiles
```

CFS adds the results of analysis to your documents. By default, the information is added in the same structure that is returned from Media Server, in a document field named `MediaServerAnalysis`. Using the configuration parameter `MediaAnalysisTransform`, you can configure CFS to run an XSL transformation to transform the information before adding it a document:

```
[ImportTasks]
Pre0=Lua:TagVideoFiles.lua
Pre1=MediaServerAnalysis:MediaServerSettings

[MediaServerSettings]
MediaServerHost=media1:14000,media2:14000
MediaServerConfigurationName=RecognizeFacesInVideo
```

```
MediaServerSharedPath=\\server\videofiles
MediaAnalysisTransform=./xslt/transform.xsl
```

For more information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference*.

Run Analysis From Lua

CFS provides Lua functions to run media analysis from a Lua script. These functions are named `analyze_media_in_document` and `analyze_media_in_file`. There are several advantages to running media analysis from a Lua script, instead of using the `MediaServerAnalysis` import task.

Firstly, you can use a single configuration to process audio, video, and image files. Your Lua script can identify the type of content that is associated with a document, and choose the correct Media Server engine to use for ingesting that content. The only way to process audio, video, and image files using the `MediaServerAnalysis` import task is to configure several tasks.

Secondly, you can configure more complex operations. For example, you can write a Lua script that sends audio to Media Server for language identification, and then uses the results of language identification to run speech-to-text with the correct language pack.

Finally, you can run analysis from Lua by configuring a single import task. To use the `MediaServerAnalysis` import task, you run a Lua script that identifies the documents to process, followed by the `MediaServerAnalysis` task itself.

CFS is supplied with example scripts that run media analysis. The scripts are in the `scripts/mediaserver` folder, in the CFS installation directory.

The following procedure demonstrates how to configure media analysis from a Lua script, in this case language detection followed by speech-to-text.

To run media analysis from Lua

1. Write a Lua script that identifies the documents that you want to process and calls the function `analyze_media_in_document` (or `analyze_media_in_file`).

   An example Lua script for running language detection and speech-to-text is located at `./scripts/mediaserver/LangDetectAndSpeechToText.lua`.

2. Create one or more configurations for Media Server that specify the tasks to perform. The Lua script `LangDetectAndSpeechToText.lua` uses two configurations, one for language detection and another for speech-to-text:

   - `./script_resources/mediaserver/langdetect.cfg`
   - `./script_resources/mediaserver/speechtotext.cfg`

   If you are using the example configuration files, check that the details are correct for your environment.

3. In the CFS configuration file, create an import task to run the Lua script. For example:

```
[ImportTasks]
Pre0=Lua:./scripts/mediaserver/LangDetectAndSpeechToText.lua

[MediaServerSettings]
MediaServerHost=mediaserver:14000
ReadFromOriginalLocation=true
// MediaServerSharedPath=<Share Directory UNC path>
```

The example script passes the `[MediaServerSettings]` section to the Lua function `analyze_media_in_document`. In the example configuration, above, this section provides the host name and ACI port of the Media Server and specifies how Media Server can access the media.

You can provide files to Media Server in several ways:

- If your Media Server can read files directly from the CFS working directory, set `ReadFromOriginalLocation=TRUE`.
- To copy files to a shared folder, set the configuration parameter `MediaServerSharedPath`. This folder must be accessible to both CFS and Media Server. CFS copies files to the shared folder so that Media Server can read them. Micro Focus recommends that you use a shared folder for sending large files.
- To send files to Media Server using HTTP POST requests, set neither `ReadFromOriginalLocation` nor `MediaServerSharedPath`.

4. Save and close the configuration file.

Examples

The following example configuration runs OCR on all supported image and video files ingested by CFS:

```
[ImportTasks]
Pre0=Lua:scripts/mediaserver/OCR.lua

[MediaServerSettings]
MediaServerHost=localhost:14000
ReadFromOriginalLocation=TRUE
```

If your CFS and Media Server are running on separate machines, you can configure CFS to copy the files to a shared folder:

```
[ImportTasks]
Pre0=Lua:scripts/mediaserver/OCR.lua

[MediaServerSettings]
MediaServerHost=mediaserver:14000
MediaServerSharedPath=\\server\videofiles
```

Troubleshoot Media Analysis

This section describes how to troubleshoot problems that might occur when you configure media analysis.

Error: Failed to find output node in response

```
Task (type: POST) failed with error: MediaServerAnalysis task failed: Failed to
find output node in response
```

Analysis will fail if CFS cannot retrieve the results of analysis from Media Server. If your analysis task fails with this error, check that your Media Server configuration includes an output task to add the results of analysis to the `process` action response. For example:

```
[Output]
OutputEngine0=response

[response]
Type=response
```

Categorize Documents

Categorization analyzes the concepts that exist in a document and, if those concepts match categories in IDOL Server, adds category information to the document. Categorizing documents is useful because you can alert IDOL users to new content that matches their interests, help them find information through taxonomies, and help them to identify similar documents.

To use categorization, you must have created and trained categories in IDOL Server. CFS queries IDOL by sending the `CategorySuggestFromText` action for each document, and IDOL returns information about any categories that match. If a document does not match any of the categories in IDOL Server, the document is not categorized. For information about how to create and train categories, refer to the *IDOL Server Administration Guide*.

To categorize documents

1. Stop CFS.
2. Open the CFS configuration file.
3. Create an import task to run the `CategorySuggestFromText` Lua script that is supplied with CFS. For example:

   ```
   [ImportTasks]
   Post0=Lua:./scripts/CategorySuggestFromText.lua
   ```

4. Open the script in a text editor.
5. Modify the variables in the script so that the script sends actions to your IDOL Server:

Line	Variable name	Value
178	`idolCategorizeHost`	The host name or IP address of your IDOL Server.
179	`idolCategorizePort`	The ACI port of your IDOL Server. The `port` argument in the function `send_aci_action` expects a number, so do not surround the port number with quotation marks.

184	`timeoutMilliseconds`	The amount of time, in milliseconds, that CFS waits for a response from your IDOL Server. If CFS does not receive a response within this time limit and the number of retries is reached, the document is not categorized. You should not need to modify the default value, which is 60 seconds.
185	`retries`	The number of times that CFS retries a request to your IDOL Server, if the first attempt is not successful.
186-192	`sslParameters`	A table of SSL parameters for connecting to your IDOL Server. For more information about the SSL parameters that you can set, refer to the *Connector Framework Server Reference*.

For example:

```
local idolCategorizeHost = "10.0.0.1"
local idolCategorizePort = 9000

...

local timeoutMilliseconds = 30000
local retries = 3
local sslParameters =
   {
       SSLMethod = "SSLV23",
       --SSLCertificate = "host1.crt",
       --SSLPrivateKey = "host1.key",
       --SSLCACertificate = "trusted.crt"
   }
```

6. Save and close the script.

Customize the Query

The `CategorySuggestFromText` Lua script sends an entire document (metadata and content) to IDOL for categorization. The document is converted to a string using the `to_idx` method and then passed to the `QueryText` parameter of the `CategorySuggestFromText` action:

```
local categorySuggestFromTextParameters = { QueryText = document:to_idx() }

...

local output = send_aci_action(
               idolCategorizeHost,
               idolCategorizePort,
               "categorysuggestfromtext",
               categorySuggestFromTextParameters,
               timeoutMilliseconds,
               retries,
               sslParameters
               )
```

You can modify the script to categorize the document based on a specific field. For example, to use only the document content:

```
local categorySuggestFromTextParameters = {
                      QueryText = document:getContent()
                      }
```

Alternatively, to use the value of a single document field:

```
local categorySuggestFromTextParameters = {
                      QueryText = document:getFieldValue("MyFieldName")
                      }
```

You can also add additional parameters to the action. For example, the `CategorySuggestFromText` Lua script does not limit the number of categories that are added to the document. To add only the most relevant category to a document, add the `CategorySuggestFromText` action parameter `NumResults=1` by modifying the script as follows:

```
local categorySuggestFromTextParameters = {
                      QueryText = document:getContent(),
                      NumResults = 1
                      }
```

For more information about the `CategorySuggestFromText` action and the parameters that it supports, refer to the *IDOL Server Reference*.

Customize the Output

The `CategorySuggestFromText` Lua script creates the following document fields by default:

Field name	Value
`category_title`	The name of the category.
`category_id`	The ID of the category in IDOL Server.
`category_reference`	The `DREREFERENCE` of the category, stored as a document in the Agentstore.

The script adds one value to each field for each category that matches the document. For example:

```
#DREFIELD category_id="200"
#DREFIELD category_id="100"
#DREFIELD category_reference="200"
#DREFIELD category_reference="100"
#DREFIELD category_title="Science"
#DREFIELD category_title="BusinessNews"
```

To modify how the information is added to the document, customize the Lua script. For example, to change the names of the fields, modify the first argument of the `addField` method on lines 211 to 213:

```
document:addField("category_name", category["title"])
document:addField("category_ref", category["reference"])
document:addField("category_id", category["id"])
```

To add only the category names, remove lines 212 and 213:

```
document:addField("category_title", category["title"])
-- document:addField("category_reference", category["reference"])
-- document:addField("category_id", category["id"])
```

To add all of the category information to a single field, using subfields, you could modify the script as follows (replacing lines 207-219):

```
if(suggestWasSuccessful) then
   local suggestedCategories = parseCategories(output)

   document:addField("category", "category information")
   local field = document:getField("category")
   for i, category in ipairs(suggestedCategories) do
       field:addField("title",category["title"])
       field:addField("reference",category["reference"])
       field:addField("id",category["id"])
   end

   document:setFieldValue("result", output)

   return true
end
```

To add all of the category names to a single field as a comma-separated list, you could modify the script as follows (replacing lines 207-219):

```
if(suggestWasSuccessful) then
   local suggestedCategories = parseCategories(output)

   local names=""
   for i, category in ipairs(suggestedCategories) do
       if i==1 then
          names = category["title"]
       else
          names = names .. "," .. category["title"]
       end
   end

   if names~="" then
      document:addField("category_names_CSV", names)
   end

   document:setFieldValue("result", output)
   return true
end
```

Run Eduction

Eduction identifies and extracts entities from text, based on a pattern that you define. An *entity* is a word, phrase, or block of information. A *pattern* might be a dictionary, for example a list of people or places. Alternatively, the pattern can describe what the entity looks like without having to list it explicitly, for example a regular expression that describes an address or telephone number. After entities are extracted the text is written to the document fields that you specify. For more information about Eduction, refer to the *IDOL Eduction User Guide*.

You can run Eduction on document fields using the `Eduction` task.

> **NOTE:** To use the `Eduction` task you must have a license for Eduction and the relevant grammar files, and specify the host name and ACI port of your License Server in the CFS configuration file.

You can run the `Eduction` task using the `Post` parameter. The parameters that are passed to the task are specified in a named section of the configuration file. For example:

```
[ImportTasks]
Post0=Eduction:EductionSettings

[EductionSettings]
ResourceFiles=C:\MyGrammar\gram1.ecr
SearchFields=DRECONTENT
Entity0=edk_common_entities/postal_address
EntityField0=SHIPPING_ADDRESS
```

Redact Documents

You can use the `Eduction` task to redact information in documents.

To enable redaction, set the configuration parameter `RedactedOutput=True`. If you want to specify the value or characters that replace the redacted text, use the configuration parameter `RedactionOutputString` or `RedactionReplacementCharacter`.

For example, the following configuration redacts addresses contained in a document's `DRECONTENT` or `ADDRESS` fields:

```
[ImportTasks]
Post0=Eduction:EductionSettings

[EductionSettings]
ResourceFiles=C:\Autonomy\IDOLServer\Eduction\address_gb.ecr
SearchFields=DRECONTENT,ADDRESS
RedactedOutput=True
```

The fields specified by `SearchFields` are not modified. CFS places the redacted text in fields with a _ `REDACTED` suffix. For example:

```
#DREFIELD ADDRESS="Cambridge Business Park, Cowley Road, Cambridge, CB4 0WZ"
#DREFIELD ADDRESS_REDACTED="[redacted]"
```

The `Eduction` task also adds the value, offset, and score for any matched entities to the document. For example:

```
#DREFIELD /offset="298"
#DREFIELD /score="1"
#DREFIELD /value="Cambridge Business Park, Cowley Road, Cambridge, CB4 0WZ"
```

Lua Post Processing

An *Eduction Lua Post Processing task* runs a Lua script that modifies the output from the Eduction module. For example, you might want to increase the score for a match if it is found near similar matches.

> **NOTE:** The Lua script is run by the Eduction module, not by CFS. The Eduction module expects the script to start with `function processmatch (edkmatch)`. You cannot modify the document being processed by CFS, or use the Lua methods that are available to CFS Lua scripts. For information about the Lua methods that are available in the Eduction module, refer to the *Eduction User Guide*.

To create an Eduction Lua Post Processing task, set the `PostProcessingTaskN` parameter. This specifies the name of a section in the CFS configuration file that contains parameters to configure the task. For example:

```
[ImportTasks]
Post0=Eduction:EductionSettings

[EductionSettings]
ResourceFiles=C:\MyGrammar\gram1.ecr
SearchFields=DRECONTENT
Entity0=edk_common_entities/postal_address
EntityField0=SHIPPING_ADDRESS
PostProcessingTask0=EductionLuaPostProcessing

[EductionLuaPostProcessing]
Script=scripts/eduction_post_process.lua
ProcessEnMasse=False
```

The Eduction Lua module will call `function processmatch (edkmatch)`. For example:

```
function processmatch(edkmatch)
   if edkmatch then
      local text = edkmatch:getOutputText()
      -- modify the match
      edkmatch:setOutputText(text)
      return true
   end
   return false -- return false to drop the match
end
```

The `edkmatch` argument represents a single eduction match, or the complete set of matches if you set the configuration parameter `ProcessEnMasse` to `true`.

If the `processmatch` function returns `true`, the match is returned to CFS. If the function returns `false`, the match is discarded.

For more information about writing Eduction post-processing scripts, and information about the Lua methods that you can use, refer to the *Eduction User Guide*.

Process HTML

Connectors, including the IDOL Web Connector, can send documents to CFS that have associated HTML files.

CFS can send the HTML files to KeyView, which discards the HTML markup and extracts the text contained in the file. However, HTML pages often contain irrelevant content such as invalid HTML, headers, sidebars, advertisements, and scripts. This text does not contain any useful information and could pollute the IDOL index, degrading performance. KeyView does not remove this irrelevant content, so Connector Framework Server provides features to process HTML files.

- **HTML processing with WKOOP**. CFS can use an embedded browser (WKOOP) to process HTML in a similar way to the IDOL Web Connector. There are many reasons to use WKOOP over other methods of processing HTML:

  - The browser allows scripts to run before the page is processed, so CFS can extract content and links that are added by JavaScript.
  - Links are resolved before a document is ingested, so that indexed documents contain absolute URLs.
  - You can remove unwanted content using the automatic clipping algorithm, or by selecting parts of the page with CSS selectors.
  - You can extract metadata or divide pages into multiple documents using CSS selectors rather than regular expressions.

  > **NOTE:** To use WKOOP you must also install the IDOL Web Connector, because WKOOP is not provided with CFS. You must install a version of WKOOP that is the same as, or later than, the version of CFS that you are using.

- **HTMLExtraction**. HTML extraction extracts the useful information from the page and discards the irrelevant content. It automatically determines which content is relevant, so there are no configuration parameters for customizing this operation. If HTML extraction does not produce good results for your use case, you might want to use the clipping features provided by WKOOP, instead.

HTML Processing with WKOOP

The WKOOPHtmlExtraction task processes an HTML file that is associated with a document. It extracts links and metadata and adds these to the document in a metadata field named HTML_PROCESSING. The task appends a page to the document content that contains the plain text extracted from the HTML source. It also sets the field AUTN_NO_FILTER, to prevent the document being processed by KeyView.

This section describes how to configure HTML processing with WKOOP.

You can configure WKOOP HTML extraction as a pre-import task (Pre0 in the following example). The Pre0 parameter also specifies the name of a section that contains the settings for the task. In the following example the section is named HtmlProcessingSettings.

```
[ImportTasks]
Pre0=WKOOPHtmlExtraction:HtmlProcessingSettings

[HtmlProcessingSettings]
WKOOPPath=F:\IDOL\WebConnector\WKOOP.exe
ProxyHost=proxy.domain.com
ProxyPort=8080
SSLMethod=NEGOTIATE
ExtractLinks=TRUE
ResolveLinks=TRUE
Url=https://www.example.com/
```

The WKOOPPath parameter specifies the path to WKOOP. WKOOP is not included with CFS, so you must install the IDOL Web Connector and specify the path to the WKOOP executable file. You must install a version of WKOOP that is the same as, or later than, the version of CFS that you are using.

If you are running CFS on a machine that is behind a proxy server, set the ProxyHost and ProxyPort parameters to specify the proxy server to use to access the web. The SSLMethod parameter specifies the version of SSL or TLS to use when connecting to the web site, and is necessary to retrieve resources over HTTPS. Setting this parameter to NEGOTIATE uses the latest version that is supported by both CFS and the web server.

The ExtractLinks parameter accepts a Boolean value that specifies whether to extract links from HTML pages and add the links to the document metadata. When ResolveLinks=TRUE the links are resolved so that indexed documents contain absolute URLs. The Url parameter specifies the source URL so that links can be resolved. You do not need to specify the exact URL of the page being processed, as long as all URLs in the document being processed are relative to the web server.

For a full list of configuration parameters that you can use to configure WKOOP HTML extraction, refer to the *Connector Framework Server Reference*.

Remove Irrelevant Content

To remove irrelevant content from HTML pages using the automatic clipping algorithm, add the parameter Clipped=TRUE to your task configuration. CFS decides which parts of the page to keep and which to discard.

The automatic clipping algorithm has been designed to work with many different pages, but this means that automatic clipping might not give the best results for every page. Alternatively, you can use CSS selectors to choose which parts of the page to keep and which to discard. To clip pages with CSS selectors, add `Clipped=TRUE` to your task configuration, and then set `ClipPageUsingCssSelect` to specify the parts of the page to keep and `ClipPageUsingCssUnselect` to specify the parts of the page to remove. These parameters accept standard CSS2 selectors.

You can also remove scripts and hidden content from the HTML page:

- Remove all scripts from the HTML page by setting `RemoveScripts=TRUE`.
- Remove "noframes" content by setting `RemoveNoframes=TRUE`. When web developers use frames they might include content in a `<noframes></noframes>` element, for web browsers that do not support frames. This content might duplicate content elsewhere in the HTML page or simply contain a message that the browser does not support frames.

Extract Metadata

This section demonstrates how to extract metadata from an HTML page and add it to a document field.

Consider the following HTML:

```
<h1>This is a title</h1>
<h2>This is a sub-title</h2>
<p class="important">This is <strong>important</strong> text</p>
```

From this HTML you could extract all of the headings and add them to a metadata field named `heading`. You could also extract the important text and add that to a separate document field.

The configuration parameters `MetadataSelector` and `MetadataFieldName` select the information to extract and provide the name of the destination document field. These parameters must be set in numbered pairs (so that each `MetadataSelector` parameter has a matching `MetadataFieldName`). The `MetadataSelector` parameter accepts standard CSS2 selectors.

The following configuration would extract the information described above:

```
MetadataSelector0=h1,h2,h3
MetadataFieldName0=heading
MetadataSelector1=p.important
MetadataFieldName1=important_paragraph
MetadataSelectorExtractPlainText=TRUE
```

The parameter `MetadataSelectorExtractPlainText` specifies whether to extract as plain text (removing HTML markup, for example).

The configuration above would produce the following metadata fields:

```
#DREFIELD heading="This is a title"
#DREFIELD heading="This is a sub-title"
#DREFIELD important_paragraph="This is important text"
```

Split Web Pages into Multiple Documents

You might want to split pages into multiple documents. For example, if you ingest pages from a discussion board you might want to ingest one document for each message on the page.

Connector Framework Server can create documents for sections of a Web page identified using CSS selectors. CFS creates a child document for each section of the page that is identified. Metadata fields (named `CHILD_DOCUMENT`) are added to the parent document, to refer to the child documents.

To split pages into multiple documents, add the following parameters to your `WKOOPHtmlExtraction` task:

`ChildDocumentSelector` A CSS2 selector that identifies the root element of each child document in the page source.

`ChildReferenceSelector` (Optional) An element in the child document that contains a value to use as the document reference. The value you extract should be unique for each child document, because it is used as part of the `DREREFERENCE` field in the child document. If you do not set this parameter, the connector uses a GUID. Specify the element using a CSS2 selector, relative to the element identified by `ChildDocumentSelector`.

`ChildMetadataSelector` (Optional) A list of elements in the child document that contain metadata. The metadata in these elements are extracted and added to the metadata fields of child documents. Specify the elements as a list of CSS2 selectors, relative to the element identified by `ChildDocumentSelector`.

To specify the name(s) of the document field(s) to contain the extracted information, set the configuration parameter `ChildMetadataFieldName`. Both parameters must have the same number of values.

`ChildMetadataFieldName` (Optional) The names to use for document fields (in child documents) that contain information extracted using the parameter `ChildMetadataSelector`. This parameter must have the same number of values as `ChildMetadataSelector`.

For example, consider the following example page which represents messages on a page of a discussion board:

```
<html>
    <head>
        <title>Example Page</title>
        <meta charset="utf-8">
    </head>
    <body>
        <div>
            <h1>Example Page</h1>
            <div class="content">
                <p>content</p>
            </div>
            <div class="message">
                <h1>Message 1</h1>
                <p class="meta">some metadata</p>
                <p>some content</p>
            </div>
            <div class="message">
                <h1>Message 2</h1>
                <p class="meta">some metadata</p>
                <p>some content</p>
            </div>
            ...
        </div>
    </body>
</html>
```

To create separate documents for the messages contained on this page, you could use the following configuration:

```
[MyTask]
...
ChildDocumentSelector=div.message
ChildReferenceSelector=h1
ChildMetadataFieldName0=my_metadata
ChildMetadataSelector0=p.meta
```

This example would produce the following child document (and a similar document for the second message):

```
#DREREFERENCE <current_document_reference>:<child_reference>
#DREFIELD my_metadata="some metadata"
...
#DRECONTENT
Message 1
some metadata
some content
...
```

The value of the `DREREFERENCE` field is constructed from the reference of the original document and the value of the element identified by the `ChildReferenceSelector` configuration parameter. If you don't set this configuration parameter or the element is not found, CFS uses a GUID instead.

CFS adds the reference of the original document to the fields `DREPARENTREFERENCE` and `DREROOTPARENTREFERENCE`. It also adds an `HTML_PROCESSING` metadata field that contains any metadata and links that are extracted from the child document.

The `DRECONTENT` field is populated with text extracted from the HTML elements that you identified as belonging to the child document.

Connector Framework Server automatically adds fields to the parent document, named `CHILD_DOCUMENT`, that contain the references of associated child documents.

HTML Extraction

HTML pages often contain irrelevant content such as invalid HTML, headers, sidebars, advertisements, and scripts. CFS can extract the useful information from the page and discard the irrelevant content.

To extract the useful information from an HTML page, use the `HtmlExtraction` import task. This task works only on HTML files and ignores other file types.

CFS reads the HTML document, and discards data such as invalid HTML, headers, sidebars, advertisements, and scripts. In the remaining content, CFS then extracts blocks of text that contain a large number of stopwords and a low proportion of links. This text is likely to be the most important content. Because CFS automatically determines which content is relevant, there are no configuration parameters for customizing this task.

Micro Focus recommends that you configure the `HtmlExtraction` task as a *Pre* import task. For example:

```
[ImportTasks]
Pre0=HtmlExtraction
```

After extracting the useful information, the HTML Extraction task sets the document field AUTN_NO_FILTER, so that the HTML file is not processed by KeyView.

Extract Metadata from Files

The `ExtractMetadata` task extracts metadata from the file associated with a document. This task extracts a subset of the metadata obtained by standard KeyView filtering. It is faster than standard KeyView filtering and does not extract the file content.

> **TIP:** When documents are ingested, CFS automatically extracts metadata. Do not use this task unless you have set the fields `AUTN_NO_FILTER` and `AUTN_NO_EXTRACT` on a document and want to extract basic metadata only.

The `ExtractMetadata` task is configured as a *Pre* task. Specify the name of the section that contains settings for the task. For example:

```
[ExtractMetadata]
Pre0=Lua:scripts/nofilter.lua
Pre1=ExtractMetadata:ExtractMetadataSettings

[ExtractMetadataSettings]
FieldnamePrefix=FIELD_
ReservedFieldnames=Reserved1,Reserved2
```

The `Pre0` task runs a Lua script that adds the fields `AUTN_NO_FILTER` and `AUTN_NO_EXTRACT` to documents. Adding these fields prevents KeyView from filtering the documents and extracting subfiles.

The `Pre1` task runs the `ExtractMetadata` task using the settings contained in the `[ExtractMetadataSettings]` section of the CFS configuration file.

The `FieldnamePrefix` parameter specifies a prefix for the names of the metadata fields that are added to the document. The `ReservedFieldnames` parameter specifies a comma-separated list of field names that the task must not use. If the task needs to add a metadata field with one of the specified names, it prefixes the name with an underscore. For example, with the settings specified above, the task would not add a field named `FIELD_Reserved1`. Instead, the task would add `_FIELD_Reserved1`.

Import Content Into a Document

The `ImportFile` task imports a file and adds its content to the document being processed. CFS does not extract sub files from the file.

The `ImportFile` task can be configured as a *Pre* or *Post* task. When you create the task, specify the name of a document field that contains the path or URL of the file to import, for example:

```
Pre0=ImportFile:fieldname
Post0=ImportFile:fieldname
```

where **`fieldname`** is the name of the document field.

Alternatively, specify the name of a section in the configuration file that contains the settings for the task:

```
[ImportTasks]
Post0=ImportFile:MySettings

[MySettings]
Fieldname=field_containing_file_path_or_url
ProxyHost=10.0.0.1
ProxyPort=8080
SSLMethod=TLSV1
```

If the field contains a URL, CFS downloads the file and adds its content to the document.

Reject Invalid Documents

You can configure CFS to reject documents based on several criteria.

When documents are rejected, they are not processed by further tasks. You can index rejected documents or discard them:

- To index the documents into one or more indexes, such as an IDOL Error Server, set the parameters `OnErrorIndexerSections` and `IndexDatabase`. `OnErrorIndexerSections` specifies a list of configuration file sections to use to index a document. These sections must contain indexing parameters, such as the host name and ACI port of your IDOL Server. `IndexDatabase` specifies the name of the IDOL database into which the rejected documents are indexed. Before indexing a document, CFS writes the name of the filter that caused the document to be rejected to a field named `MATCHEDFILEFILTERS`.
- If you do not specify any indexing details, the documents are discarded. CFS writes a message to the import log showing that the document was rejected, and showing which filter caused the rejection.

Reject Documents with Binary Content

The `BinaryFileFilter` task rejects any documents that have been filtered as binary. This can occur when KeyView filtering fails, for example due to corrupt files.

When CFS detects a non-UTF8 character, it replaces the character with a hexadecimal character code. The `BinaryFileFilter` task detects these character codes and rejects documents where the proportion exceeds the limit set by the `ThresholdPercent` parameter.

The `BinaryFileFilter` task can be configured as a *Post* task. The parameters that are passed to the task are specified in a named section of the configuration file. For example:

```
[ImportTasks]
Post0=BinaryFileFilter:BinaryFileFilterSettings

[BinaryFileFilterSettings]
ThresholdPercent=10
OnErrorIndexerSections=IdolErrorServer
IndexDatabase=IdolErrorReview
```

For information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference*.

Reject Documents with Import Errors

The `ImportErrorFilter` task rejects any documents for which errors have occurred. Errors can occur during KeyView filtering or during *pre* and *post* import tasks.

The `ImportErrorFilter` task can be configured as a *Post* task. The parameters that are passed to the task are specified in a named section of the configuration file. For example:

```
[ImportTasks]
Post0=ImportErrorFilter:ImportErrorFilterSettings

[ImportErrorFilterSettings]
OnErrorIndexerSections=IdolErrorServer
IndexDatabase=IdolErrorReview
```

For information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference*.

Reject Documents with Symbolic Content

The `SymbolicContentFilter` task calculates the proportion of symbolic characters in a document. If the proportion of symbolic characters in the document content exceeds the limit specified by the `MaxSymbolicCharactersPercent` parameter, the document is rejected.

Symbolic characters are defined as any character between U+2000 and U+2FFF.

The `SymbolicContentFilter` task can be configured as a *Post* task. The parameters that are passed to the task are specified in a named section of the configuration file. For example:

```
[ImportTasks]
Post0=SymbolicContentFilter:SymbolicContentFilterSettings

[SymbolicContentFilterSettings]
MaxSymbolicCharactersPercent=8
OnErrorIndexerSections=IdolErrorServer
IndexDatabase=IdolErrorReview
```

For information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference*.

Reject Documents by Word Length

The `WordLengthFilter` task calculates the average length of words in a document. If the average length of words in the document content (`DRECONTENT`) falls outside the limits specified by the `MinimumAverage` or `MaximumAverage` parameters, the document is rejected.

The `WordLengthFilter` task can be configured as a *Post* task. The parameters that are passed to the task are specified in a named section of the configuration file. For example:

```
[ImportTasks]
Post0=WordLengthFilter:WordLengthFilterSettings

[WordLengthFilterSettings]
MinimumAverage=3.0
MaximumAverage=10.0
OnErrorIndexerSections=IdolErrorServer
IndexDatabase=IdolErrorReview
```

For information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference*.

Reject All Invalid Documents

The `BadFilesFilter` task rejects all documents that are considered to be invalid:

- Documents that have binary content.
- Documents for which import errors have occurred.
- Documents that have too high a proportion of symbolic content.
- Documents where the average word length is too long or too short.

`BadFilesFilter` must be configured as a *Post* task.

BadFilesFilter reads configuration parameters from the section of the configuration file that you specify in the Post parameter. In this section you can set parameters for each filter. In the example below, two parameters have been set to configure the word length filter:

```
[ImportTasks]
Post0=BadFilesFilter:BadFilesFilterSettings

[BadFilesFilterSettings]
MinimumAverage=3.0
MaximumAverage=10.0
OnErrorIndexerSections=IdolErrorServer
IndexDatabase=IdolErrorReview
```

For information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference.*

Split Document Content into Sections

Dividing the content of long documents into sections can result in more relevant search results, because IDOL Server can return a specific part of a document in response to a query.

To divide document content into sections, use the Sectioner task.

The Sectioner import task must be configured as a *Post* task. The parameters that are passed to the task are specified in a named section of the configuration file. For example:

```
[ImportTasks]
Post0=Sectioner:Sectioning

[Sectioning]
SectionerMaxBytes=3000
SectionerMinBytes=1500
```

For information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference*.

Split Files into Multiple Documents

Sometimes you might retrieve files from a repository that you would prefer to ingest as multiple documents.

You can use the TextToDocs task to split a file containing text (for example an HTML file or XML file) into multiple documents. To divide a file, you specify regular expressions that match the relevant parts of the document. The task creates a main document and one or more child documents, which can all have metadata and content. When you run TextToDocs on a document, the original document is discarded. The documents created by TextToDocs are metadata-only documents, which means that they do not have an associated file and are not filtered by KeyView.

The TextToDocs task should be configured as a *Pre* task. The parameters that are passed to the task are specified in a named section of the configuration file. For example:

```
[ImportTasks]
Pre0=TextToDocs:MyTextToDocs

[MyTextToDocs]
...
```

For information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference*.

The `TextToDocs` task expects documents to use UTF-8 character encoding. If your documents are not encoded in UTF-8 you can use the configuration parameter `SourceEncoding` to specify the character set encoding of the source documents, so that they can be converted to UTF-8. If conversion fails, the original encoding is used and CFS adds an error message to the `ImportErrorCode` and `ImportErrorDescription` document fields.

Example

The following HTML is an example file that you might want to ingest as separate documents. There are clear sections which could represent different topics:

```
<html>
  <body>
    <p class="main">Main content</p>

    <div class="section">
      <h1>First document</h1>
      <p class="metadata">Extract as metadata</p>
      <p>Some text</p>
    </div>

    <div class="section">
       <h1>Second document</h1>
       <p class="metadata">Extract as metadata</p>
       <p>Some text</p>
    </div>

    <div class="section">
       <h1>Third document</h1>
       <p class="metadata">Extract as metadata</p>
       <p>Some text</p>
    </div>

  </body>
</html>
```

You might want to split this file into a main document and three child documents, one of which might look like this:

```
#DREREFERENCE C:\MyFiles\TextToDocs\textToDocs.html:0
#DREDBNAME FileSystem
```

```
#DREFIELD MyMetadataField="Extract as metadata"
#DRECONTENT
First document
Some text

#DREENDDOC
```

To do this, you could use the following configuration:

```
[ImportTasks]
Pre0=TextToDocs:MyTextToDocs

[MyTextToDocs]
FilenameMatchesRegex0=.*\.html

MainRangeRegex0=<html>(.*)</html>
MainContentRegex0=<p class="main">(.*?)</p>

ChildrenRangeRegex0=<html>(.*)</html>
ChildRangeRegex=<div class="section">(.*?)</div>
ChildContentRegex0=<h1>(.*?)</h1>
ChildContentRegex1=<p>(.*?)</p>
ChildFieldName0=MyMetadataField
ChildFieldRegex0=<p class="metadata">(.*?)</p>
ChildInheritFields=DREDBNAME
```

In this example, the `FilenameMatchesRegex` parameter has been set to process only those files that have the extension `.html`.

The `MainContentRegex` parameter identifies parts of the original document to add to the `DRECONTENT` field of the main document.

The `ChildRangeRegex` parameter identifies the parts of the original document that should become child documents. The sub-match `(.*?)` matches all of the content between a `<div class="section">` tag and a `</div>` tag. When this regular expression is applied to the example document above, there are three matches and therefore three child documents are created. It is important to make the regular expression lazy, because otherwise it would match everything between the first `<div class="section">` and the final `</div>`, resulting in a single child document.

The `ChildContentRegex` parameter identifies the content to add to the `DRECONTENT` field of a child document. In this example two regular expressions are used to extract content. The `ChildFieldName` and `ChildFieldRegex` parameters populate metadata fields. In this example a single field named `MyMetadataField` is created.

Setting the parameter `ChildInheritFields=DREDBNAME` specifies that the child documents inherit the field `DREDBNAME` from the original document. If you are indexing documents into IDOL Server it is important to set this parameter, because (depending on how your system is configured) documents without a `DREDBNAME` field might not be indexed.

Write Documents to Disk

CFS can write documents to disk in several formats. You might want to write documents to disk for the following reasons:

- so that you can see the data that is being indexed. You can then set up further processing tasks to manipulate and enrich the data.
- so that you can debug your Lua scripts or other processing tasks.
- so that you can export the data from documents to other systems.

Write Documents to Disk in IDX Format

To write documents to disk in IDX format, configure an `IdxWriter` processing task by setting the `Pre`, `Post`, `Update`, or `Delete` configuration parameter.

To run the `IdxWriter` task with default settings, use the `Pre`, `Post`, `Update`, or `Delete` parameter to specify the file name for the IDX file:

```
[ImportTasks]
Pre0=IdxWriter:pre.idx
Post0=IdxWriter:post.idx
```

Alternatively, set the parameter to `IdxWriter`, followed by a colon (`:`), followed by the name of the section in the configuration file that contains custom settings for the task. For example:

```
[ImportTasks]
Pre0=IdxWriter:PreIDX
Post0=IdxWriter:PostIDX

[PreIDX]
IdxWriterFilename=pre.idx
IdxWriterMaxSizeKBs=100
IdxWriterArchiveDirectory=./IDXArchive

[PostIDX]
IdxWriterFilename=post.idx
IdxWriterMaxSizeKBs=100
IdxWriterArchiveDirectory=./IDXArchive
```

For information about the configuration parameters you can use to configure this task, refer to the *Connector Framework Server Reference*.

Write Documents to Disk in XML Format

To write documents to disk in XML format, configure an `XmlWriter` processing task by setting the `Pre`, `Post`, `Update`, or `Delete` configuration parameter.

When you create the `XmlWriter` task, specify the file name of the XML file. For example:

```
[ImportTasks]
Pre0=XmlWriter:C:\ConnectorFrameworkServer\pre.xml
Post0=XmlWriter:C:\ConnectorFrameworkServer\post.xml
```

Write Documents to Disk in JSON Format

To write documents to disk in JSON format, configure a `JsonWriter` processing task by setting the `Pre`, `Post`, `Update`, or `Delete` configuration parameter.

To configure the task with default settings, specify the file name for the output file:

```
[ImportTasks]
Pre0=JsonWriter:pre.jsn
Post0=JsonWriter:post.jsn
```

Alternatively, set the parameter to `JsonWriter`, followed by a colon (`:`), followed by the name of the section in the configuration file that contains custom settings for the task. For example:

```
[ImportTasks]
Post0=JsonWriter:PostJsonWriting

[PostJsonWriting]
JsonWriterFilename=post.jsn
JsonWriterMaxSizeKBs=1000
JsonWriterArchiveDirectory=./JSONarchive
```

For information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference*.

Write Documents to Disk in CSV Format

The `CsvWriter` task writes document metadata and content to a comma-separated values (CSV) file. This allows you to export the data to other systems.

The task always writes the document reference (`DREREFERENCE`) and content (`DRECONTENT`) fields, and you can choose the other fields that you want to include. The task writes the field names, followed by one line of values for each document that is ingested.

The `CsvWriter` task can be configured as a *Pre*, *Post*, *Update* or *Delete* task.

To run the task with default settings, specify the file name for the output file:

```
[ImportTasks]
Post0=CsvWriter:MyTask.csv
```

Alternatively, specify the name of a section in the configuration file that contains the settings for the task:

```
[ImportTasks]
Post0=CsvWriter:CsvWriting

[CsvWriting]
```

```
CsvWriterFilename=MyTask.csv
CsvWriterMaxSizeKBs=1000
CsvWriterArchiveDirectory=./CSVarchive
CsvWriterFieldNames0=A_FIELD
CsvWriterFieldNames1=A_FIELD/subfield
CsvWriterFieldNames2=A_FIELD/@attribute
```

For information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference*.

Write Documents to Disk as SQL INSERT Statements

The `SqlWriter` task writes document metadata and content to a file in the form of SQL “INSERT” statements. You can use the SQL to insert the data from the documents into a database.

The task always writes the document reference (`DREREFERENCE`) and content (`DRECONTENT`) fields, and you can choose the other fields that you want to include. The task writes one INSERT statement for each document that is processed.

The `SqlWriter` task can be configured as a *Pre*, *Post*, *Update* or *Delete* task.

To configure the task, specify the name of a section that contains the settings, for example:

```
[ImportTasks]
Post0=SqlWriter:SqlWriting

[SqlWriting]
SqlWriterFileName=MyTask.sql
SqlWriterTableName=table
SqlWriterDreReferenceColumnName=REFERENCE
SqlWriterDreContentColumnName=CONTENT

SqlWriterFieldNames0=MODIFIED_DATE
SqlWriterColumnNames0=DATE
SqlWriterDataTypes0=DATE_TIME

SqlWriterUseNullForMissingFields=true

SqlWriterDateFormats0=DD/MM/YYYY
SqlWriterDateFormats1=YYYY/MM/DD

SqlWriterMaxSizeKBs=1024
SqlWriterArchiveDirectory=./SQLarchive
```

For information about the parameters that you can use to configure this task, refer to the *Connector Framework Server Reference*.

Standardize Document Fields

The documents created by your connectors might not have consistent field names. For example, documents created by the File System Connector can have a field named `FILEOWNER`. Documents created by the Documentum Connector can have a field named `owner_name`. Both of these fields store the name of the person who owns a file.

You might want to rename document fields so that documents use the same field names to store the same type of information. CFS includes the `standardizer` task to do this.

You can configure the `Standardizer` task as a *Pre* or *Post* task. For example:

```
[ImportTasks]
Post0=Standardizer
```

To use the `Standardizer` task, you must set the `EnableFieldNameStandardization` and `FieldNameDictionaryPath` configuration parameters in the `[ImportService]` section of the CFS configuration file. For more information about these parameters, refer to the *Connector Framework Server Reference*.

Customize Field Standardization

Field standardization modifies documents so that they have a consistent structure and consistent field names. You can use field standardization so that documents indexed into IDOL through different connectors use the same fields to store the same type of information. Field standardization only modifies fields that are specified in a dictionary, which is defined in XML format. A standard dictionary, named `dictionary.xml`, is supplied in the CFS installation folder.

In most cases you should not need to modify the standard dictionary, but you can modify it to suit your requirements or create dictionaries for different purposes. By modifying the dictionary, you can configure CFS to apply rules that modify documents before they are ingested. For example, you can move fields, delete fields, or change the format of field values.

The following examples demonstrate how to perform some operations with field standardization.

The following rule renames the field `Author` to `DOCUMENT_METADATA_AUTHOR_STRING`. This rule applies to all components that run field standardization and applies to all documents.

```
<FieldStandardization>
    <Field name="Author">
        <Move name="DOCUMENT_METADATA_AUTHOR_STRING"/>
    </Field>
</FieldStandardization>
```

The following rule demonstrates how to use the `Delete` operation. This rule instructs CFS to remove the field `KeyviewVersion` from all documents. The `Product` element ensures that this rule is run only by CFS.

```
<FieldStandardization>
    <Product key="ConnectorFrameWork">
        <Field name="KeyviewVersion">
```

```
            <Delete/>
        </Field>
    </Product>
</FieldStandardization>
```

There are several ways to select fields to process using the `Field` element.

Field element attribute	Description	Example

`name`	Select a field where the field name matches a fixed value.	Select the field `MyField`:
		`<Field name="MyField">`
		`   ...`
		`</Field>`
		Select the field `Subfield`, which is a subfield of `MyField`:
		`<Field name="MyField">`
		`   <Field name="Subfield">`
		`      ...`
		`   </Field>`
		`</Field>`

`path`	Select a field where its path matches a fixed value.	Select the field `Subfield`, which is a subfield of `MyField`.
		`<Field path="MyField/Subfield">`
		`   ...`
		`</Field>`

`nameRegex`	Select all fields at the current depth where the field name matches a regular expression.	In this case the field name must begin with the word `File`:
		`<Field nameRegex="File.*">`
		`   ...`
		`</Field>`

`pathRegex`	Select all fields where the path of the field matches a regular expression.
	This operation can be inefficient because every metadata field must be checked. If possible, select the fields to process another way.
		This example selects all subfields of `MyField`.
		`<Field pathRegex="MyField/[^/]*">`
		`   ...`
		`</Field>`
		This approach would be more efficient:
		`<Field name="MyField">`
		`   <Field nameRegex=".*">`
		`      ...`
		`   </Field>`
		`</Field>`

You can also limit the fields that are processed based on their value, by using one of the following:

Field element attribute	Description	Example

`matches`	Process a field if its value matches a fixed value.	Process a field named `MyField`, if its value matches `abc`.
		`<Field name="MyField" matches="abc">`
		`   ...`
		`</Field>`

`matchesRegex`	Process a field if its entire value matches a regular expression.	Process a field named `MyField`, if its value matches one or more digits.
		`<Field name="MyField" matchesRegex="\d+">`
		`   ...`
		`</Field>`

`containsRegex`	Process a field if its value contains a match to a regular expression.	Process a field named `MyField` if its value contains three consecutive digits.
		`<Field name="MyField" containsRegex="\d{3}">`
		`   ...`
		`</Field>`

The following rule deletes every field or subfield where the name of the field or subfield begins with `temp`.

```
<FieldStandardization>
    <Field pathRegex="(.*/)?temp[^/]*">
        <Delete/>
    </Field>
</FieldStandardization>
```

The following rule instructs CFS to rename the field `Author` to `DOCUMENT_METADATA_AUTHOR_STRING`, but only when the document contains a field named `DocumentType` with the value `230` (the KeyView format code for a PDF file).

```
<FieldStandardization>
    <Product key="ConnectorFrameWork">
        <IfField name="DocumentType" matches="230"> <!-- PDF -->
            <Field name="Author">
                <Move name="DOCUMENT_METADATA_AUTHOR_STRING"/>
            </Field>
        </IfField>
    </Product>
</FieldStandardization>
```

> **TIP:** In this example, the `IfField` element is used to check the value of the `DocumentType` field. The `IfField` element does not change the current position in the document. If you used the `Field` element, field standardization would attempt to find an `Author` field that is a subfield of `DocumentType`, instead of finding the `Author` field at the root of the document.

The following rules demonstrate how to use the `ValueFormat` operation to change the format of dates. The only format that you can convert date values into is the IDOL AUTNDATE format. The first rule transforms the value of a field named `CreatedDate`. The second rule transforms the value of an attribute named `Created`, on a field named `Date`.

```
<FieldStandardization>
    <Field name="CreatedDate">
        <ValueFormat type="autndate" format="YYYY-SHORTMONTH-DD HH:NN:SS"/>
    </Field>
    <Field name="Date">
        <Attribute name="Created">
            <ValueFormat type="autndate" format="YYYY-SHORTMONTH-DD HH:NN:SS"/>
        </Attribute>
    </Field>
</FieldStandardization>
```

As demonstrated by this example, you can select field attributes to process in a similar way to selecting fields.

You must select attributes using either a fixed name or a regular expression:

Select a field attribute by name	`<Attribute name="MyAttribute">`
Select attributes that match a regular expression	`<Attribute nameRegex=".*">`

You can then add a restriction to limit the attributes that are processed:

Process an attribute only if its value matches a fixed value	`<Attribute name="MyAttribute" matches="abc">`
Process an attribute only if its value matches a regular expression	`<Attribute name="MyAttribute" matchesRegex=".*">`
Process an attribute only if its value contains a match to a regular expression	`<Attribute name="MyAttribute" containsRegex="\w+">`

The following rule moves all of the attributes of a field to sub fields, if the parent field has no value. The `id` attribute on the first `Field` element provides a name to a matching field so that it can be referred to by later operations. The `GetName` and `GetValue` operations save the name and value of a selected field or attribute (in this case an attribute) into variables (in this case `$'name'` and `$'value'`) which can be used by later operations. The `AddField` operation uses the variables to add a new field at the selected location (the field identified by `id="parent"`).

```
<FieldStandardization>
    <Field pathRegex=".*" matches="" id="parent">
        <Attribute nameRegex=".*">
            <GetName var="name"/>
            <GetValue var="value"/>
            <Field fieldId="parent">
                <AddField name="$'name'" value="$'value'"/>
            </Field>
            <Delete/>
        </Attribute>
    </Field>
</FieldStandardization>
```

The following rule demonstrates how to move all of the subfields of `UnwantedParentField` to the root of the document, and then delete the field `UnwantedParentField`.

```
<FieldStandardization id="root">
    <Product key="ConnectorFrameWork">
        <Field name="UnwantedParentField">
            <Field nameRegex=".*">
                <Move destId="root"/>
            </Field>
            <Delete/>
        </Field>
    </Product>
</FieldStandardization>
```

Normalize E-mail Addresses

Documents can contain e-mail addresses in many formats, and often the name of the sender or recipient is contained in the same metadata field as their e-mail address.

The `EmailAddressNormalisation` task searches metadata fields for the names and e-mail addresses of e-mail senders and recipients. It then writes the information back to the document in a standard format. For named e-mail addresses (`"Name" <name@domain.com>`), the task separates the name from the address. The task also converts all e-mail addresses to lower-case.

For example, a document might include the following field:

```
<To>"One, Some" <Someone@Somewhere.com>, <user.name@domain.com>, "Else, Someone" <
SomeoneElse@Somewhere.com ></To>
```

The `EmailAddressNormalisation` task reads this information and adds the following fields to the document:

```
<to_email>someone@somewhere.com</to_email>
<to_email>user.name@domain.com</to_email>
<to_email>someoneelse@somewhere.com</to_email>
<to_name>One, Some</to_name>
<to_name/>
<to_name>Else, Someone</to_name>
```

As shown in the previous example, when an e-mail address does not have an associated name, an empty name field is added to the document. This is necessary because the order of the fields in the document is the only way to determine which name belongs with which e-mail address. The first e-mail address is associated with the first name, the second e-mail address with the second name, and so on.

This means that if your source field does not contain any names:

```
<To>Someone@Somewhere.com, SomeoneElse@Somewhere.com</To>
```

The task writes the following fields to the document:

```
<to_email>someone@somewhere.com</to_email>
<to_email>someoneelse@somewhere.com</to_email>
<to_name/>
<to_name/>
```

You can configure `EmailAddressNormalisation` as a *Pre* or *Post* task. For example:

```
[ImportTasks]
Post0=EmailAddressNormalisation:EmailAddressNormalisationSettings

[EmailAddressNormalisationSettings]
FieldNameRegex="To","From","Cc","Bcc"
AddresseeFieldName="to_name","from_name","cc_name","bcc_name"
EmailFieldName="to_email","from_email","cc_email","bcc_email"
```

The `Post0` task runs e-mail address normalisation using the settings in the `[EmailAddressNormalisationSettings]` section. The `FieldNameRegex` parameter specifies a list of regular expressions that identify the fields to process. The `AddresseeFieldName` and `EmailFieldName` parameters specify the names of the fields to add to the document. CFS adds the name of the sender or recipient to the addressee field and their e-mail address to the e-mail field.

Language Detection

CFS can identify the language of a document, and write the name of the language to a document field. A front-end application could use this field to provide a way to filter documents by language. You can also use language detection to reject invalid documents (when a language cannot be detected).

Language detection can be configured as a post-import task. Set the `Post` parameter to `LangDetect` and specify the name of a configuration file section that contains the task settings. For example:

```
[ImportTasks]
Post0=LangDetect:LangDetectSettings

[LangDetectSettings]
LanguageDetectionDirectory=./filters/datafiles/
OutputField=DetectedLanguage
FailIfLanguageUnknown=TRUE
```

You must set the parameter `LanguageDetectionDirectory` to the path of the folder that contains the file `langdetect.dat`. The remaining parameters are optional. The parameter `OutputField` specifies the name of the document field to write the name of detected language to. By default, CFS rejects documents where it cannot detect a language but you can configure this by setting `FailIfLanguageUnknown`. To continue processing documents when a language cannot be detected, set `FailIfLanguageUnknown=FALSE`.

Translate Documents

CFS can use third-party translation services to translate documents into other languages. This can be useful when you have documents that are not in your users' native language.

> **IMPORTANT:** To perform translation, you must have a license for one of the supported translation services. CFS relies on third-party services to translate text between languages. The language translation library that CFS uses to communicate with third-party translation services is not included in the standard CFS installation but can be obtained through Micro Focus software support.

Micro Focus recommends that you configure the `LanguageTranslation` task as a post-import task. Use the `Post` parameter to specify the name of a section that contains the task settings.

The following is an example task that uses the translation services provided by an SDL Enterprise Translation Server:

```
[ImportTasks]
Post0=Library:LanguageTranslation

[LanguageTranslation]
Library=LanguageTranslation
TranslatorType=SdlEts
ApiBaseUrl=https://host:port
ApiKey=0a12b34c56d78e9
SourceField=DRECONTENT
TargetField=DRECONTENT
TargetLanguage=ENG
Quality=medium-high
```

The following is an example task that uses the translation services provided by sdlbeglobal.com:

```
[ImportTasks]
Post0=Library:LanguageTranslation

[LanguageTranslation]
Library=LanguageTranslation
TranslatorType=SdlBeGlobal
AccountId=12345
ApiKey=0a12b34c56d78e9
UserId=23456
TouchpointId=34567
SourceField=DRECONTENT
```

```
TargetField=DRECONTENT
TargetLanguage=ENG
```

In both cases, the `TranslatorType` configuration parameter specifies the type of translation service to use. To use an SDL Enterprise Translation Server, set this parameter to `SdlEts`. To use the services that are provided by `sdlbeglobal.com`, set this parameter to `SdlBeGlobal`. You must then set the relevant parameters that are required to use the API:

- To use an SDL Enterprise Translation Server, set the configuration parameter `ApiBaseUrl` to the base URL of the API to use for translation, and the `ApiKey` parameter to the API key.
- To use the services provided by sdlbeglobal.com, set the parameters `AccountId`, `ApiKey`, `UserId` and `TouchpointId` to the values provided by SDL.

The remaining parameters are optional but you can set these to customize the task to your use case.

The `SourceField` configuration parameter specifies the name of the document field to translate, and the `TargetField` parameter specifies the name of the field to contain the results. If you specify the same field name for both parameters, CFS reads the text from the field, sends it for translation, and then writes the result back to the same field (which overwrites the original value).

The source language is detected automatically, but if automatic detection is not successful you can specify the source language by setting the configuration parameter `SourceLanguage`, which accepts the name of a language or a three-letter language code. Alternatively, you can configure the task to read the source language from a document field, by setting the parameter `SourceLanguageField`. For information about the languages that are supported, refer to the SDL documentation or translation service user interface.

The target language is specified by the configuration parameter `TargetLanguage`, which has a default value of `ENG` (English).

If you are using an SDL Enterprise Translation Server, you can also choose how to balance translation quality and speed by setting the `Quality` parameter.

For more information about the configuration parameters that you can use to configure the language translation task, refer to the *Connector Framework Server Reference*.

Identify Files in a NIST RDS Hash Set

The `NistRdsFilter` task calculates the checksum of the file associated with a document. If the checksum is present in a NIST RDS hash set, the function adds the fields `AUTN_NIST_RDS_LIST`, `AUTN_NO_FILTER`, and `AUTN_NO_EXTRACT` to the document.

The NIST RDS hash sets contain the checksums of millions of files that originate from operating systems and application software, so you can use this task to filter out documents that represent those files.

> **NOTE:** Before using this task you must populate a datastore with the NIST RDS hash set(s). For information about how to do this, see the following procedure.

To configure a NIST RDS Filter task

1. Configure the NIST RDS filter task in the CFS configuration file. For example:

   ```
   [ImportTasks]
   Pre0=NistRdsFilter:NistRdsFilterSettings

   [NistRdsFilterSettings]
   NistRdsStoreType=BTREE
   ```

2. Download the NIST RDS hash sets that you want to use.

3. Use the NIST RDS tool, provided in the `tools` directory of your CFS installation, to populate the datastore that you have configured.

   ```
   NistRdsTool.exe CFS.cfg NistRdsFilterSettings NistRdsFile.zip
   ```

   where,

   CFS.cfg is the path to the CFS configuration file.

   NistRdsFilterSettings is the section in the configuration file that contains the NIST RDS Filter task settings.

   NistRdsFile.zip is the list of NIST RDS hash sets that you downloaded. If you want to use multiple hash sets, include all of the file names separated by spaces.

Chapter 7: Index Documents

This section describes how to configure indexing.

Introduction

The final step in the ingestion process is to index information into an index, such as an IDOL Content component. After CFS finishes processing documents, it automatically indexes them into the indexes that you have configured.

CFS can index documents into:

- **IDOL Content**.

  Index documents into IDOL Server to search, analyze, and find patterns in unstructured information. You can index documents directly into an IDOL Server, or send them to a Distributed Index Handler (DIH) to be distributed between multiple IDOL Servers in a distributed architecture.

- **Vertica**.

  Index documents into a Vertica database to analyze the structured information contained in your data repositories. Much of the metadata extracted by connectors and by KeyView is structured information held in structured fields, so you can use Vertica to gain insight into this information.

By default, CFS indexes each document into all of the indexes specified by the `IndexerSections` parameter in the `[Indexing]` section of its configuration file. However, if the document field `AUTN_INDEXER_SECTIONS` is set, CFS routes the document to the indexes specified in the field. The field accepts a comma-separated list of index names that must match the names of the sections in the CFS configuration file.

Configure the Batch Size and Time Interval

CFS indexes documents in batches. This is more efficient because fewer requests are made to the server.

Documents wait in the index queue until there are enough documents to create a batch, or until the maximum time interval for indexing is reached. If the time interval is reached, CFS indexes all of the documents in the queue regardless of the batch size.

To configure indexing settings

1. Stop CFS.
2. Open the CFS configuration file.
3. In the `[Indexing]` section, set the following configuration parameters:

`IndexBatchSize`	The number of documents to index in a single batch. CFS waits until this number of documents are ready for indexing, unless the `IndexTimeInterval` is reached.
`IndexTimeInterval`	The maximum amount of time, in seconds, that a document can wait in the index queue.

   For example:

   ```
   [Indexing]
   IndexBatchSize=1000
   IndexTimeInterval=600
   ```
4. Save the configuration file.

Index Documents into an IDOL Server

To index documents into an IDOL Server

1. Check that CFS is authorized to index documents into the IDOL index. In the IDOL Content configuration file, CFS must belong to an authorization role that includes the `admin`, `query`, and `index` standard roles.
2. Stop CFS.
3. Open the CFS configuration file.
4. In the `[Indexing]` section, use the `IndexerSections` parameter to specify the names of the sections that contain indexing settings. If this parameter is already set, add the name of the new indexer to the list. For example:

   ```
   [Indexing]
   IndexerSections=IdolServer
   ```
5. Create a new section in the CFS configuration file, with the same name that you specified in the `IndexerSections` parameter. In the new section, set the following parameters:

`Host`	The host name or IP address of the IDOL Server.

`Port`	The ACI Port of the IDOL Server.
`DefaultDatabaseName`	The name of the IDOL database to index a document into when the `DREDBNAME` document field is not set.
`SSLConfig`	(Optional) The name of a section in the CFS configuration file that contains SSL settings for connecting to IDOL. Set this parameter if your IDOL Server is configured to accept connections over SSL. For more information about the configuration parameters you can use to configure SSL connections, refer to the *Connector Framework Server Reference*.
`CreateDatabase`	(Optional, default `false`) Specifies whether IDOL should create databases that do not already exist. For example, if you set this parameter to `TRUE` and the database specified in a `DREDBNAME` document field does not exist, IDOL Server will create it.

For example:

```
[IdolServer]
Host=idol
Port=9000
DefaultDatabaseName=News
SSLConfig=SSLOptions

[SSLOptions]
SSLMethod=NEGOTIATE
```

For more information about these parameters and other parameters that you can set to customize the indexing process, refer to the *Connector Framework Server Reference*.

6. Save and close the configuration file.

Index Documents into Vertica

CFS can index documents into Vertica, so that you can run queries on structured fields (document metadata).

Depending on the metadata contained in your documents, you could:

- Investigate the average age of documents in a repository. You might want to answer questions such as: How much time has passed since the documents were last updated? How many files are regularly updated? Does this represent a small proportion of the total number of documents? Who are the most active users?

- Find the number of e-mail messages sent to your sales or support teams each week, and calculate the average response time to customer queries.

Prerequisites

- CFS supports indexing into Vertica 7.1 and later.
- You must install the appropriate Vertica ODBC drivers (version 7.1 or later) on the machine that hosts Connector Framework Server. If you want to use an ODBC Data Source Name (DSN) in your connection string, you will also need to create the DSN. For more information about installing Vertica ODBC drivers and creating the DSN, refer to the Vertica documentation.

New, Updated and Deleted Documents

When documents are indexed into Vertica, CFS adds a timestamp that contains the time when the document was indexed. The field is named VERTICA_INDEXER_TIMESTAMP and the timestamp is in the format YYYY-MM-DD HH:NN:SS.

When a document in a data repository is modified, CFS adds a new record to the database with a new timestamp. All of the fields are populated with the latest data. The record describing the older version of the document is not deleted. You can create a projection to make sure your queries only return the latest record for a document.

When a connector detects that a document has been deleted from a repository, CFS inserts a new record into the database. The record contains only the DREREFERENCE and the field VERTICA_INDEXER_DELETED set to TRUE.

Fields, Sub-Fields, and Field Attributes

Documents that are created by connectors and processed by CFS can have multiple levels of fields, and field attributes. A database table has a flat structure, so this information is indexed into Vertica as follows:

- Document fields become columns in the flex table. An IDOL document field and the corresponding database column have the same name.
- Sub-fields become columns in the flex table. A document field named my_field with a sub-field named subfield results in two columns, my_field and my_field.subfield.
- Field attributes become columns in the flex table. A document field named my_field, with an attribute named my_attribute results in two columns, my_field holding the field value and my_field.my_attribute holding the attribute value.

Prepare the Vertica Database

Indexing documents into a standard database is problematic, because documents do not have a fixed schema. A document that represents an image has different metadata fields to a document that represents an e-mail message. Vertica databases solve this problem with *flex tables*. You can create a flex table without any column definitions, and you can insert a record regardless of whether a referenced column exists.

You must create a flex table before you index data into Vertica.

When creating the table, consider the following:

- Flex tables store entire records in a single column named `__raw__`. The default maximum size of the `__raw__` column is 128K. You might need to increase the maximum size if you are indexing documents with large amounts of content. Alternatively, you could configure CFS to remove content from documents before they are indexed.
- Documents are identified by their `DREREFERENCE`. Micro Focus recommends that you do not restrict the size of any column that holds this value, because this could result in values being truncated. As a result, rows that represent different documents might appear to represent the same document. If you do restrict the size of the `DREREFERENCE` column, ensure that the length is sufficient to hold the longest `DREREFERENCE` that might be indexed.

To create a flex table without any column definitions, run the following query:

```
create flex table my_table();
```

To improve query performance, create real columns for the fields that you query frequently. For documents indexed by CFS, this is likely to include the `DREREFERENCE`:

```
create flex table my_table(DREREFERENCE varchar NOT NULL);
```

You can add new column definitions to a flex table at any time. Vertica automatically populates new columns with values for existing records. The values for existing records are extracted from the `__raw__` column.

For more information about creating and using flex tables, refer to the Vertica Documentation or contact Micro Focus Vertica technical support.

Configure CFS to Index into Vertica

The following procedure demonstrates a basic configuration that indexes all documents into a Vertica database.

However, you can customize the indexing process. For example, your CFS might be importing files from a File System Connector, e-mail messages from Exchange, and social media content. You might want to index these items into separate flex tables. To do this, you could run a Lua script to set the `AUTN_INDEXER_SECTIONS` field in each document, and create a separate indexing operation for each type of content.

To configure CFS to index documents into Vertica

1. Stop CFS.
2. Open the CFS configuration file.
3. In the `[Indexing]` section, use the `IndexerSections` parameter to specify the names of the sections that contain indexing settings. If this parameter is already set, add the name of the new indexer to the list. For example:

   ```
   [Indexing]
   IndexerSections=IdolServer,Vertica
   ```

4. Create a new section in the CFS configuration file, with the same name that you specified in the `IndexerSections` parameter. In the new section, set the following parameters:

`IndexerType`	The type of index to index documents into. Set this parameter to **`Library`**.
`LibraryDirectory`	The directory that contains the library to use to index data.
`LibraryName`	The name of the library to use to index data. You can omit the `.dll` or `.so` file extension. Set this parameter to **`verticaIndexer`**.
`ConnectionString`	The connection string to use to connect to the Vertica database.
`TableName`	The name of the table in the Vertica database to index the documents into. The table must be a flex table and must exist before you start indexing documents. For more information, see Prepare the Vertica Database, on page 102.

   For example:

```
[Vertica]
IndexerType=Library
LibraryDirectory=indexerdlls
LibraryName=verticaIndexer
ConnectionString=DSN=VERTICA
TableName=my_flex_table
```

   For more information about these parameters and other parameters that you can set to customize the indexing process, refer to the *Connector Framework Server Reference*.

5. Save and close the configuration file.

Troubleshooting

This section describes how to troubleshoot problems that might occur when you index data into Vertica.

Documents are not indexed into Vertica

Documents cannot be indexed when the Vertica database server is unavailable, or cannot be reached by CFS. To see whether an indexing error has occurred, check the CFS indexer log. The default location for this log file is `logs/indexer.log`. If documents were not indexed successfully, you will need to ingest these documents again.

Index Documents into another CFS

You can index documents into another CFS. You might want to do this if you want to perform further processing on them.

To index documents into another CFS

1. Stop CFS.
2. Open the CFS configuration file.
3. In the `[Indexing]` section, use the `IndexerSections` parameter to specify the names of the sections in the configuration file that contain indexing settings. If this parameter is already set, add the name of the new indexer to the list. For example:

   ```
   [Indexing]
   IndexerSections=IndexCFS
   ```

4. Create a new section in the CFS configuration file, with the same name that you specified in the `IndexerSections` parameter. In the new section, set the following parameters:

`IndexerType`	The type of index that you want to index documents into. Set this parameter to **CFS**.
`Host`	The host name or IP address of the CFS.
`Port`	The ACI Port of the CFS.
`SSLConfig`	(Optional) The name of a section in the CFS configuration file that contains SSL settings for connecting to the other CFS. Set this parameter if the CFS is configured to accept connections over SSL. For more information about the configuration parameters you can use to configure SSL connections, refer to the *Connector Framework Server Reference*.

   For example:

   ```
   [IndexCFS]
   IndexerType=CFS
   Host=cfs.domain.com
   Port=7000
   SSLConfig=SSLOptions

   [SSLOptions]
   SSLMethod=TLSV1.2
   ```

   For more information about these parameters and other parameters that you can set to customize the indexing process, refer to the *Connector Framework Server Reference*.

5. Save and close the configuration file.

Index Documents into MetaStore

To index documents into MetaStore

1. Stop CFS.
2. Open the CFS configuration file.
3. In the `[Indexing]` section, use the `IndexerSections` parameter to specify the names of the sections that contain indexing settings. If this parameter is already set, add the name of the new indexer to the list. For example:

   ```
   [Indexing]
   IndexerSections=IdolServer,MetaStore
   ```

4. Create a new section in the CFS configuration file, with the same name that you specified in the `IndexerSections` parameter. In the new section, set the following parameters:

`IndexerType`	Set this parameter to **`MetaStore`**.
`Host`	The host name or IP address of the MetaStore.
`Port`	The port of the MetaStore.
`SSLConfig`	(Optional) The name of the section in the CFS configuration file that contains the SSL settings to use for communicating with the MetaStore.

   For example:

   ```
   [MetaStore]
   IndexerType=MetaStore
   Host=localhost
   Port=4500
   ```

   For more information about these parameters and other parameters that you can set to customize the indexing process, refer to the *Connector Framework Server Reference*.

5. Save and close the configuration file.

Document Fields for Indexing

To customize the way that documents are indexed, set the following document fields.

AUTN_NO_INDEX

Documents that have this field are not indexed. You can use this field when you want to troubleshoot the ingestion process without indexing documents.

AUTN_INDEXER_SECTIONS

A comma-separated list of sections in the CFS configuration file to use to index the document. CFS indexes the document into all of the indexes that you specify. If this field is not set, CFS indexes the document into all of the indexes specified by the configuration parameter `IndexerSections`.

AUTN_INDEXPRIORITY

This field can be used to increase the priority of an index action sent to IDOL Server to index a batch of documents. You can specify a priority from 0 to 100, where 0 is the lowest priority and 100 is the highest. This means that you can configure some documents to be indexed before others, or before documents from sources other than CFS.

> **CAUTION:** Use this field with care. Modifying the index priority for documents changes the order of the index commands processed by IDOL. For example, in the case of an ingest-replace, the add command could be processed before the delete, resulting in a loss of data. Micro Focus recommends that you configure the indexing priority for batches of documents indexed into IDOL using the `IndexPriority` configuration parameter. This ensures that all of the batches indexed by CFS have the same priority.

If documents in a batch contain the field `INDEXPRIORITY`, and the value of this field is greater than that specified by the `IndexPriority` configuration parameter, the priority of the batch is increased to the highest `INDEXPRIORITY` field value present in the batch.

Manipulate Documents Before Indexing

CFS can index documents into multiple indexes. Normally, CFS indexes identical data into every index, but you might want to manipulate documents depending on the index that they are sent to. For example, if you are using Vertica to analyze structured information, you might want to remove the content from the documents indexed into Vertica, but keep the content in documents that are indexed into IDOL.

You cannot use import and index tasks to manipulate documents in this way, because those tasks affect documents sent to all of the indexes. To manipulate the documents sent to a single index, you can run a Lua script during the indexing process.

The script must define a `handler` function:

```
function handler(document, operation)
     -- do something, for example
     document:deleteField("UNINTERESTING_FIELD")
     return true
 end
```

The operation argument specifies the documents that you want to run the script on. This argument is a string and can be set to add, update, or remove:

- add - manipulate documents that are being added to the index. Ingest-adds are sent when a connector finds new documents in a repository, or when a document's content is changed (the old document is removed, and the new document added).
- update - manipulate documents that represent metadata updates.
- remove - manipulate documents that represent information deleted from the source repository.

To index the document the handler function must return true. To discard the document, return false.

To manipulate documents before indexing

1. Open the CFS configuration file.
2. In a section of the configuration file specified by the IndexerSections configuration parameter, set the IndexLuaScript parameter. This parameter specifies the path to the script that you want to run. For example:

   ```
   [Indexing]
   IndexerSections=IdolServer,Vertica

   [Vertica]
   IndexerType=Library
   LibraryDirectory=indexerdlls
   LibraryName=verticaIndexer
   ConnectionString=DSN=VERTICA
   TableName=my_flex_table
   IndexLuaScript=./scripts/remove_content.lua
   ```

3. Save and close the configuration file.

Chapter 8: Monitor Connector Framework Server

This section describes how to monitor CFS.

Use the Logs

As Connector Framework Server runs, it outputs messages to log files. Most log messages occur due to normal operation, for example when CFS starts, receives actions, or processes documents. If CFS encounters an error, the logs are the first place to look for information to help troubleshoot the problem.

CFS separates log messages into the following message types, each of which relates to a specific feature:

Log Message Type	Description
`Action`	Logs actions that are received by CFS, and related messages.
`Application`	Logs application-related occurrences, such as when the CFS starts.
`Import`	Information about the import process. When you set `LogLevel` to `FULL`, CFS can log a significant amount of information about the import process, which might reduce performance.
`Indexer`	Information about the indexing process.

Customize Logging

You can customize logging by setting up your own *log streams*. Each log stream creates a separate log file in which specific types of message are logged.

To set up log streams

1. Open the CFS configuration file in a text editor.
2. Find the `[Logging]` section. If the configuration file does not contain a `[Logging]` section, create it.
3. In the `[Logging]` section, create a list of the log streams you want to set up using the format *`N=LogStreamName`*. List the log streams in consecutive order, starting from 0 (zero). For example:

   ```
   [Logging]
   LogLevel=NORMAL
   0=ApplicationLogStream
   1=ActionLogStream
   2=ImportLogStream
   3=IndexLogStream
   ```

   You can also use the `[Logging]` section to configure any default values for logging configuration parameters, such as `LogLevel`. For more information, refer to the *Connector Framework Server Reference*.
4. Create a new section for each of the log streams. Each section must have the same name as the log stream. For example:

   ```
   [ApplicationLogStream]
   [ActionLogStream]
   ...
   ```
5. Specify the settings for each log stream in the appropriate section. You can specify the type of logging to perform (for example, full logging), the maximum size of log files, and so on. For example:

   ```
   [ApplicationLogStream]
   LogTypeCSVs=application
   LogFile=application.log
   LogHistorySize=50
   LogTime=True
   LogEcho=False
   LogMaxSizeKBs=1024

   [ActionLogStream]
   LogTypeCSVs=action
   LogFile=action.log
   LogHistorySize=50
   LogTime=true
   LogEcho=false
   LogMaxSizeKBs=1024
   ```
6. Save and close the configuration file.
7. Restart CFS for your changes to take effect. For information about how to start and stop CFS, refer to the *IDOL Getting Started Guide*.

Monitor Asynchronous Actions using Event Handlers

Some of the actions that you can send to Connector Framework Server are asynchronous. Asynchronous actions do not run immediately, but are added to a queue. This means that the person or application that sends the action does not receive an immediate response. However, you can configure Connector Framework Server to call an event handler when an asynchronous action starts, finishes, or encounters an error.

You might use an event handler to:

- Return data about an event back to the application that sent the action.
- Write event data to a text file, to log any errors that occur.

You can also use event handlers to monitor the size of asynchronous action queues. If a queue becomes full this might indicate a problem, or that applications are making requests to Connector Framework Server faster than they can be processed.

Connector Framework Server can call an event handler for the following events.

OnStart The `OnStart` event handler is called when Connector Framework Server starts processing an asynchronous action.

OnFinish The `OnFinish` event handler is called when Connector Framework Server successfully finishes processing an asynchronous action.

OnError The `OnError` event handler is called when an asynchronous action fails and cannot continue.

OnQueueEvent The `OnQueueEvent` handler is called when an asynchronous action queue becomes full, becomes empty, or the queue size passes certain thresholds.

- A `QueueFull` event occurs when the action queue becomes full.
- A `QueueFilling` event occurs when the queue size exceeds a configurable threshold (`QueueFillingThreshold`) and the last event was a `QueueEmpty` or `QueueEmptying` event.
- A `QueueEmptying` event occurs when the queue size falls below a configurable threshold (`QueueEmptyingThreshold`) and the last event was a `QueueFull` or `QueueFilling` event.
- A `QueueEmpty` event occurs when the action queue becomes empty.

Connector Framework Server supports the following types of event handler:

- The `TextFileHandler` writes event data to a text file.
- The `HttpHandler` sends event data to a URL.
- The `LuaHandler` runs a Lua script. The event data is passed into the script.

Configure an Event Handler

To configure an event handler, follow these steps.

To configure an event handler

1. Stop Connector Framework Server.
2. Open the Connector Framework Server configuration file in a text editor.
3. Set the `OnStart`, `OnFinish`, `OnError`, or `OnQueueEvent` parameter to specify the name of a section in the configuration file that contains the event handler settings.
   - To run an event handler for all asynchronous actions, set these parameters in the `[Actions]` section. For example:

     ```
     [Actions]
     OnStart=NormalEvents
     OnFinish=NormalEvents
     OnError=ErrorEvents
     ```

   - To run an event handler for a specific action, set these parameters in the `[`*`ActionName`*`]` section, where *`ActionName`* is the name of the action. The following example calls an event handler when the *`Example`* action starts and finishes successfully, and uses a different event handler to monitor the queue size:

     ```
     [Example]
     OnStart=NormalEvents
     OnFinish=NormalEvents
     OnQueueEvent=QueueSizeEvents
     ```

4. Create a new section in the configuration file to contain the settings for your event handler. You must name the section using the name you specified with the `OnStart`, `OnFinish`, `OnError`, or `OnQueueEvent` parameter.
5. In the new section, set the `LibraryName` parameter.

`LibraryName` The type of event handler to use to handle the event.

- To write event data to a text file, set this parameter to **`TextFileHandler`**, and then set the `FilePath` parameter to specify the path of the file.
- To send event data to a URL, set this parameter to **`HttpHandler`**, and then use the HTTP event handler parameters to specify the URL, proxy server settings, credentials, and so on.
- To run a Lua script, set this parameter to **`LuaHandler`**, and then use the `LuaScript` parameter to specify the script to run. For information about writing the script, see Write a Lua Script to Handle Events, on the next page.

For example:

```
[NormalEvents]
LibraryName=TextFileHandler
FilePath=./events.txt

[ErrorEvents]
LibraryName=HTTPHandler
URL=http://handlers:8080/lo-proxy/callback.htm?

[QueueSizeEvents]
LibraryName=LuaHandler
LuaScript=./handle_queue_events.lua
```

6. Save and close the configuration file. You must restart Connector Framework Server for your changes to take effect.

Write a Lua Script to Handle Events

The Lua event handler runs a Lua script to handle events. The Lua script must contain a function named `handler` with the arguments `request` and `xml`, as shown below:

```
function handler(request, xml)
      ...
end
```

- `request` is a table holding the request parameters. For example, if the request was `action=Example&MyParam=Value`, the table will contain a key `MyParam` with the value `Value`. Some events, for example queue size events, are not related to a specific action and so the table might be empty.
- `xml` is a string of XML that contains information about the event.

Monitor the size of the Import and Index Queues

CFS generates events when the import queue and the outgoing (indexing) queue become full, become empty, or the queue size passes certain thresholds. If a queue approaches its maximum size, this might indicate a problem, or that applications are making requests to Connector Framework Server faster than they can be processed.

CFS generates the following events for each queue that is monitored:

- A `QueueFull` event occurs when the queue becomes full.
- A `QueueFilling` event occurs when the queue size exceeds a configurable threshold (`QueueFillingThreshold`) and the last event was a `QueueEmpty` or `QueueEmptying` event.
- A `QueueEmptying` event occurs when the queue size falls below a configurable threshold (`QueueEmptyingThreshold`) and the last event was a `QueueFull` or `QueueFilling` event.
- A `QueueEmpty` event occurs when the queue becomes empty.

You can configure event handlers to process these events. For example, you might want to notify an administrator if the size of a queue reaches 80 percent of the maximum.

To monitor queue sizes

1. Stop CFS.
2. Open the CFS configuration file in a text editor.
3. Set the `OnQueueEvent` parameter to the name of a section that configures the event handler.
   - To monitor the size of the import queue, set this parameter in the `[ImportService]` section. For example:

     ```
     [ImportService]
     OnQueueEvent=MyEventHandler
     ```

   - To monitor the size of the outgoing (indexing) queue, set this parameter in the `[Indexing]` section. For example:

     ```
     [Indexing]
     OnQueueEvent=MyEventHandler
     ```

4. Create a new section in the configuration file to contain the settings for your event handler. You must name the section using the name you specified with the `OnQueueEvent` parameter.
5. In the new section, set the `LibraryName` parameter.

`LibraryName`	The type of event handler to use to handle the event. • To write event data to a text file, set this parameter to **`TextFileHandler`**, and then set the `FilePath` parameter to specify the path of the file. • To send event data to a URL, set this parameter to **`HttpHandler`**, and then use the HTTP event handler parameters to specify the URL, proxy server settings, credentials, and so on. • To run a Lua script, set this parameter to **`LuaHandler`**, and then use the `LuaScript` parameter to specify the script to run. For information about writing the script, see Write a Lua Script to Handle Events, on the previous page.

   For example:

   ```
   [MyEventHandler]
   LibraryName=LuaHandler
   LuaScript=./handle_queue_event.lua
   ```

6. Save and close the configuration file. You must restart CFS for your changes to take effect.

Set Up Document Tracking

Document tracking reports metadata about documents when they pass through various stages in the ingestion and indexing process. Document tracking can help you detect problems with the indexing process.

You can write document tracking events to a database, log file, or IDOL Server. For information about how to set up a database to store document tracking events, refer to the *IDOL Server Administration Guide*.

To enable Document Tracking

1. Open the CFS configuration file.
2. Create a new section in the configuration file, named `[DocumentTracking]`.
3. In the `[DocumentTracking]` section, specify where the document tracking events are sent.
   - To send document tracking events to a database through ODBC, set the following parameters:

`Backend`	To send document tracking events to a database, set this parameter to **`Library`**.
`LibraryPath`	Specify the location of the ODBC document tracking library. This is included with IDOL Server.
`ConnectionString`	The ODBC connection string for the database.

     For example:

     ```
     [DocumentTracking]
     Backend=Library
     LibraryPath=C:\Autonomy\IDOLServer\IDOL\modules\dt_odbc.dll
     ConnectionString=DSN=MyDatabase
     ```

   - To send document tracking events to the CFS import log, set the following parameters:

`Backend`	To send document tracking events to the logs, set this parameter to **`Log`**.
`DatabaseName`	The name of the log stream to send the document tracking events to. Set this parameter to **`import`**.

     For example:

     ```
     [DocumentTracking]
     Backend=Log
     DatabaseName=import
     ```

   - To send document tracking events to an IDOL Server, set the following parameters:

`Backend`	To send document tracking events to an IDOL Server, set this parameter to **`IDOL`**.
`TargetHost`	The host name or IP address of the IDOL Server.
`TargetPort`	The index port of the IDOL Server.

For example:

```
[DocumentTracking]
Backend=IDOL
TargetHost=idol
TargetPort=9001
```

For more information about the parameters you can use to configure document tracking, refer to the *Connector Framework Server Reference*.

4. Save and close the configuration file.

Appendix A: KeyView Supported Formats

This section lists the file formats that KeyView can detect.

Key to Supported Formats Table

The supported formats table includes the following information:

Column	Description
Format Name	The format name that is returned by KeyView format detection.
Number	The format number that is returned by KeyView format detection.
Category	This value is used in the KeyView configuration file `formats.ini` to specify the reader to use to filter, export, or view the format. Several formats might have the same category value.
Description	A short description of the file format.
MIME Type	The MIME type (if any).
Extension	A list of common file extensions for the file format. **NOTE:** This is not a complete list of file extensions. KeyView does not distinguish between file types based on their extension. Instead, it detects the file format based on the file content. This is more reliable because content cannot always be predicted from the file extension, and because some file extensions are associated with multiple formats.

Supported Formats

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Reserved__Fmt	-1	-1				
Unknown_Fmt	0	0				
AES_Multiplus_Comm_Fmt	1	1	Multiplus (AES)		PTF	
ASCII_Text_Fmt	2	2	Plain Text file	text/plain	TXT	afsr
MSDOS_Batch_File_Fmt	3	2	MS-DOS Batch File	application/x-bat	BAT	afsr
Applix_Alis_Fmt	4	3	Applix Asterix		AX	axsr
BMP_Fmt	5	4	Windows Bitmap Image (BMP)	image/bmp	BMP	bmpsr, kpbmprdr
CT_DEF_Fmt	6	5	Convergent Technologies DEF Comm. Format			cdsr
Corel_Draw_Fmt	7	6	CorelDRAW (up to version 13/X3)	application/coreldraw	CDR	kpcdrrdr
CGM_ClearText_Fmt	8	8	Computer Graphics Metafile (CGM)		CGM	kpcgmrdr
CGM_Binary_Fmt	9	8	Computer Graphics Metafile (CGM)	image/cgm	CGM	kpcgmrdr
CGM_Character_Fmt	10	8	Computer Graphics Metafile (CGM)		CGM	kpcgmrdr
Word_Connection_Fmt	11	9	Word Connection		CN	stringssr
COMET_TOP_Word_Fmt	12	10	Nixdorf COMET TOP Financial Accounting software			
CEOwrite_Fmt	13	11	CEOwrite		CW	stringssr
DSA101_Fmt	14	12	DSA101 (Honeywell Bull)			stringssr
DCA_RFT_Fmt	15	13	IBM DCA-RFT (Revisable Form)	application/dca-rft	RFT, DC	dcasr
CDA_DDIF_Fmt	16	14	CDA / DDIF		DDIF	
DG_CDS_Fmt	17	16	DG Common Data Stream (CDS)		CDS	stringssr
Micrografx_Draw_Fmt	18	18	Windows Draw (Micrografx)		DRW	
Data_Point_VistaWord_Fmt	19	19	Vistaword		DV	stringssr
DECdx_Fmt	20	20	DEC WPS Plus DX format		DX	
Enable_WP_Fmt	21	21	Enable Word Processing		WPF	stringssr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
EPSF_Fmt	22	22	Encapsulated PostScript	application/postscript	EPS	kpepsrdr
Preview_EPSF_Fmt	23	22	Encapsulated PostScript	application/postscript		kpepsrdr
MS_Executable_Fmt	24	23	MSDOS/Windows executable	application/x-msdownload	EXE	exesr
G31D_Fmt	25	24	CCITT G3 1D			
GIF_87a_Fmt	26	25	Graphics Interchange Format (GIF87a)	image/gif	GIF	gifsr, kpgifrdr
GIF_89a_Fmt	27	25	Graphics Interchange Format (GIF89a)	image/gif	GIF	gifsr, kpgifrdr
HP_Word_PC_Fmt	28	26	HP Word PC		HW	stringssr
IBM_1403_LinePrinter_Fmt	29	27	IBM 1403 Line Printer		I4	
IBM_DCF_Script_Fmt	30	28	DCF Script		IC	stringssr
IBM_DCA_FFT_Fmt	31	29	DCA-FFT (IBM Final Form)		IF, FFT	
Interleaf_Fmt	32	30	Interleaf			
GEM_Image_Fmt	33	31	GEM Bit Image		IMG	
IBM_Display_Write_Fmt	34	32	IBM DisplayWrite		IP	dw4sr
Sun_Raster_Fmt	35	33	Sun Raster image	image/x-cmu-raster	RAS, RS, SUN	kpsunrdr
Ami_Pro_Fmt	36	35	Lotus Ami Pro	application/x-lotus-amipro	SAM	lasr
Ami_Pro_StyleSheet_Fmt	37	35	Lotus Ami Pro Style Sheet			lasr
MORE_Fmt	38	36	MORE Database MAC			
Lyrix_Fmt	39	37	Lyrix Word Processing			stringssr
MASS_11_Fmt	40	38	MASS-11		M1	stringssr
MacPaint_Fmt	41	39	MacPaint		PNTG	kpmacrdr
MS_Word_Mac_Fmt	42	40	Microsoft Word for Macintosh (up to version 3)	application/msword	DOC	mbsr
SmartWare_II_Comm_Fmt	43	41	SmartWare II			
MS_Word_Win_Fmt	44	42	Microsoft Word for Windows (up to version 6)	application/msword	DOC, WPS	misr
Multimate_Fmt	45	43	MultiMate		MM	stringssr
Multimate_Fnote_Fmt	46	43	MultiMate Footnote File		MMFN	stringssr
Multimate_Adv_Fmt	47	43	MultiMate Advantage			stringssr
Multimate_Adv_Fnote_Fmt	48	43	MultiMate Advantage Footnote File			stringssr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Multimate_Adv_II_Fmt	49	43	MultiMate Advantage II			stringssr
Multimate_Adv_II_Fnote_Fmt	50	43	MultiMate Advantage II Footnote File		FBX, FNX	stringssr
Multiplan_PC_Fmt	51	44	Multiplan (PC)			
Multiplan_Mac_Fmt	52	44	Multiplan (Mac)			
MS_RTF_Fmt	53	45	Rich Text Format (RTF)	application/rtf	RTF	rtfsr
MS_Word_PC_Fmt	54	46	Microsoft Word for PC (up to version 6)	application/x-ms-wordpc	MW	mwsr
MS_Word_PC_StyleSheet_Fmt	55	46	Microsoft Word for PC (up to version 6) Style Sheet			mwsr
MS_Word_PC_Glossary_Fmt	56	46	Microsoft Word for PC (up to version 6) Glossary			mwsr
MS_Word_PC_Driver_Fmt	57	46	Microsoft Word for PC (up to version 6) Driver			mwsr
MS_Word_PC_Misc_Fmt	58	46	Microsoft Word for PC (up to version 6) Miscellaneous File			mwsr
NBI_Async_Archive_Fmt	59	47	NBI Async Archive Format			
Navy_DIF_Fmt	60	48	Navy DIF (document interchange format)		ND	stringssr
NBI_Net_Archive_Fmt	61	49	NBI OASys Net Archive Format		NN	nnsr
NIOS_TOP_Fmt	62	50	NIOS TOP			
FileMaker_Mac_Fmt	63	51	Filemaker MAC		FP5, FP7	
ODA_Q1_11_Fmt	64	52	ODA / ODIF Q1 11		OD	stringssr
ODA_Q1_12_Fmt	65	52	ODA / ODIF Q1 12		OD	stringssr
OLIDIF_Fmt	66	53	OLIDIF (Olivetti)			
Office_Writer_Fmt	67	55	Office Writer		OW	stringssr
PC_Paintbrush_Fmt	68	56	PC Paintbrush Graphics (PCX)	image/vnd.zbrush.pcx	PCX	kppcxrdr
CPT_Comm_Fmt	69	57	CPT Corporation word processor		PF	stringssr
Lotus_PIC_Fmt	70	58	Lotus PIC	image/x-pict	PIC	kppicrdr
Mac_PICT_Fmt	71	59	Macintosh Raster / QuickDraw Picture	image/x-pict	PCT	kppctrdr
Philips_Script_Word_Fmt	72	60	Philips Script			
PostScript_Fmt	73	61	PostScript	application/postscript	PS	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
PRIMEWORD_Fmt	74	62	PRIMEWORD			pwsr
Quadratron_Q_One_v1_Fmt	75	63	Q-One V1.93J		Q1, QX	stringssr
Quadratron_Q_One_v2_Fmt	76	64	Q-One V2.0		Q1, QX	stringssr
SAMNA_Word_IV_Fmt	77	65	SAMNA Word		SAM	stringssr
Ami_Pro_Draw_Fmt	78	66	Lotus Ami Pro Draw		SDW	kpsdwrdr
SYLK_Spreadsheet_Fmt	79	67	SYmbolic LinK (SYLK) format		SLK	
SmartWare_II_WP_Fmt	80	68	Informix SmartWare II word processor		DOC, SMT	swsr
Symphony_Fmt	81	69	Lotus Symphony spreadsheet		WR1	
Targa_Fmt	82	70	Truevision Targa image	image/x-tga	TGA	kpTGArdr
TIFF_Fmt	83	71	Tagged Image File Format (TIFF)	image/tiff	TIF, TIFF	kptifrdr, tifsr
Targon_Word_Fmt	84	72	Targon Word		TW	stringssr
Uniplex_Ucalc_Fmt	85	73	Uniplex Ucalc		SS	
Uniplex_WP_Fmt	86	74	Uniplex word processor		UP	stringssr
MS_Word_UNIX_Fmt	87	75	Microsoft Word UNIX	application/msword		
WANG_PC_Fmt	88	76	WANG PC			
WordERA_Fmt	89	77	WordERA		DC, GL, FR	stringssr
WANG_WPS_Comm_Fmt	90	78	WANG WPS		WF	stringssr
WordPerfect_Mac_Fmt	91	79	WordPerfect MAC	application/x-corel-wordperfect		wpmsr
WordPerfect_Fmt	92	86	WordPerfect version 4	application/x-corel-wordperfect	WP, WP4	stringssr
WordPerfect_VAX_Fmt	93	139	WordPerfect VAX	application/x-corel-wordperfect		
WordPerfect_Macro_Fmt	94	139	WordPerfect Macro	application/vnd.wordperfect	MRS	
WordPerfect_Dictionary_Fmt	95	139	WordPerfect Spelling Dictionary	application/vnd.wordperfect	SPW	
WordPerfect_Thesaurus_Fmt	96	139	WordPerfect Thesaurus	application/vnd.wordperfect		
WordPerfect_Resource_Fmt	97	139	WordPerfect Resource File	application/vnd.wordperfect	WWK, PRS	
WordPerfect_Driver_Fmt	98	139	WordPerfect Driver	application/vnd.wordperfect	IRS, VRS	
WordPerfect_Cfg_Fmt	99	139	WordPerfect Configuration File	application/vnd.wordperfect	PFX	
WordPerfect_Hyphenation_Fmt	100	139	WordPerfect Hyphenation Dictionary	application/vnd.wordperfect	HYC	
WordPerfect_Misc_Fmt	101	139	WordPerfect Miscellaneous File	application/vnd.wordperfect		

Format Name	Number	Category	Description	MIME Type	Extension	Readers
WordMARC_Fmt	102	82	WordMARC Composer	video/x-ms-wm	WM, PW	stringssr
Windows_Metafile_Fmt	103	83	Windows Metafile	image/wmf	WMF	kpwmfrdr
Windows_Metafile_NoHdr_Fmt	104	83	Windows Metafile (no header)	image/wmf	WMF	kpwmfrdr
SmartWare_II_DB_Fmt	105	84	Informix SmartWare II database			
WordPerfect_Graphics_Fmt	106	195	WordPerfect Graphics (version 2 and higher)	application/vnd.wordperfect	WPG, QPG	kpwg2rdr, kpwpgrdr
WordStar_Fmt	107	87	WordStar		WS, WSD	stringssr
WANG_WITA_Fmt	108	88	WANG WITA		WT	stringssr
Xerox_860_Comm_Fmt	109	89	Xerox 860			stringssr
Xerox_Writer_Fmt	110	91	Xerox Writer			stringssr
DIF_SpreadSheet_Fmt	111	92	Data Interchange Format (DIF)	application/dif+xml	DIF	difsr
Enable_Spreadsheet_Fmt	112	93	Enable Spreadsheet	application/vnd.epson.ssf	SSF	
SuperCalc_Fmt	113	94	Sorcim SuperCalc spreadsheet		CAL	
UltraCalc_Fmt	114	95	UltraCalc spreadsheet			
SmartWare_II_SS_Fmt	115	96	Informix SmartWare II spreadsheet			
SOF_Encapsulation_Fmt	116	97	Serialized Object Format (SOF)	application/java-serialized-object	SOF	
PowerPoint_Win_Fmt	117	98	Microsoft PowerPoint PC (up to version 4)	application/x-ms-powerpoint	PPT	kpp40rdr
PowerPoint_Mac_Fmt	118	99	Microsoft PowerPoint MAC (up to version 4)	application/x-ms-powerpoint	PPT	
PowerPoint_95_Fmt	119	212	Microsoft PowerPoint 95	application/x-ms-powerpoint	PPT	kpp95rdr
PowerPoint_97_Fmt	120	272	Microsoft PowerPoint 97	application/x-ms-powerpoint	PPT	kpp97rdr
PageMaker_Mac_Fmt	121	100	PageMaker for Macintosh			
PageMaker_Win_Fmt	122	101	PageMaker for Windows			
MS_Works_Mac_WP_Fmt	123	103	Microsoft Works Word Processor for MAC	application/x-msworks	MWK	stringssr
MS_Works_Mac_DB_Fmt	124	104	Microsoft Works Database for MAC	application/x-msworks		
MS_Works_Mac_SS_Fmt	125	105	Microsoft Works Spreadsheet for MAC	application/x-msworks		mwssr
MS_Works_Mac_Comm_Fmt	126	106	Microsoft Works Communication for MAC	application/x-msworks		

Format Name	Number	Category	Description	MIME Type	Extension	Readers
MS_Works_DOS_WP_Fmt	127	107	Microsoft Works Word Processor for DOS	application/x-msworks	WPS	stringssr
MS_Works_DOS_DB_Fmt	128	108	Microsoft Works Database for DOS	application/x-msworks	WDB	
MS_Works_DOS_SS_Fmt	129	109	Microsoft Works Spreadsheet for DOS	application/x-msworks		mwssr
MS_Works_Win_WP_Fmt	130	227	Microsoft Works Word Processor for Windows (up to 2000)	application/x-msworks	WPS, W40	msw6sr, mswsr
MS_Works_Win_DB_Fmt	131	231	Microsoft Works Database for Windows	application/x-msworks		
MS_Works_Win_SS_Fmt	132	228	Microsoft Works Spreadsheet for Windows	application/x-msworks	WKS, S30, S40	mwssr
PC_Library_Fmt	133	111	DOS/Windows Object Library	application/x-archive	LIB, A	
MacWrite_Fmt	134	112	MacWrite	application/macwriteii		stringssr
MacWrite_II_Fmt	135	113	MacWrite II	application/macwriteii		stringssr
Freehand_Fmt	136	114	Freehand MAC	image/x-freehand		
Disk_Doubler_Fmt	137	115	Disk Doubler			
HP_GL_Fmt	138	116	HP Graphics Language	vector/x-hpgl	HPGL, HPG	
FrameMaker_Fmt	139	136	FrameMaker	application/vnd.framemaker	FM, FRM	
FrameMaker_Book_Fmt	140	136	FrameMaker Book	application/vnd.framemaker	BOOK	
Maker_Markup_Language_Fmt	141	174	Maker Markup Language	application/vnd.mif		
Maker_Interchange_Fmt	142	117	Adobe FrameMaker Interchange Format (MIF)	application/x-mif	MIF	mifsr
JPEG_File_Interchange_Fmt	143	118	JPEG Interchange Format	image/jpeg	JPG, JPEG	jpgsr, kpjpgrdr
Reflex_Fmt	144	119	Borland Reflex database			
Framework_Fmt	145	276	Framework office suite			
Framework_II_Fmt	146	120	Framework II office suite		FW3	
Paradox_Fmt	147	121	Borland Paradox database		DB	
MS_Windows_Write_Fmt	148	123	Microsoft Windows Write	application/x-ms-write	WRI	mwsr
Quattro_Pro_DOS_Fmt	149	124	Corel Quattro Pro for DOS	application/x-quattropro	WQ1	
Quattro_Pro_Win_Fmt	150	184	Corel Quattro Pro for Windows	application/x-quattro-win	WB1, WB2, WB3	qpssr
Persuasion_Fmt	151	126	Adobe Persuasion			

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Windows_Icon_Fmt	152	128	Windows Icon Format	image/vnd.microsoft.icoN	ICO	kpicordr
Windows_Cursor_Fmt	153	133	Windows Cursor	image/x-win-bitmap	CUR	
MS_Project_Activity_Fmt	154	129	Microsoft Project (up to version 3) activity file			
MS_Project_Resource_Fmt	155	129	Microsoft Project (up to version 3) resource file			
MS_Project_Calc_Fmt	156	129	Microsoft Project (up to version 3) calc file			
PKZIP_Fmt	157	132	ZIP Archive	application/zip	ZIP, ZIPX	unzip
Quark_Xpress_Fmt	158	134	Quark Xpress MAC			
ARC_PAK_Archive_Fmt	159	135	PAK/ARC Archive		ARC, PAK	
MS_Publisher_Fmt	160	137	Microsoft Publisher (up to version 3)	application/x-mspublisher	PUB	mspubsr
PlanPerfect_Fmt	161	138	PlanPerfect			
WordPerfect_Auxiliary_Fmt	162	139	Corel WordPerfect auxiliary file		WPW	
MS_WAVE_Audio_Fmt	163	141	Microsoft Wave audio	audio/wav	WAV	MCI, riffsr
MIDI_Audio_Fmt	164	142	MIDI audio	audio/mid	MID, MIDI	MCI
AutoCAD_DXF_Binary_Fmt	165	143	Autodesk AutoCAD DXF binary format	image/x-dxf	DXF	kpDXFrdr, kpODArdr
AutoCAD_DXF_Text_Fmt	166	143	Autodesk AutoCAD DXF text format	image/x-dxf	DXF	kpDXFrdr, kpODArdr
dBase_Fmt	167	144	dBase Database III+/IV	application/x-dbf	DBF, VCX	dbfsr
OS_2_PM_Metafile_Fmt	168	145	OS/2 PM Metafile		MET	
Lasergraphics_Language_Fmt	169	146	Lasergraphics Language			
AutoShade_Rendering_Fmt	170	147	AutoShade Rendering			
GEM_VDI_Fmt	171	148	GEM VDI Metafile image		GEM, GDI	
Windows_Help_Fmt	172	149	Windows Help File	application/winhlp	HLP	
Volkswriter_Fmt	173	150	Volkswriter word processor		VW4	stringssr
Ability_WP_Fmt	174	151	Ability Word Processor			
Ability_DB_Fmt	175	151	Ability Database			
Ability_SS_Fmt	176	151	Ability Spreadsheet			

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Ability_Comm_Fmt	177	151	Ability Presentation			
Ability_Image_Fmt	178	151	Ability Image			
XyWrite_Fmt	179	152	XYWrite / Nota Bene		XY4	xywsr
CSV_Fmt	180	153	CSV (Comma Separated Values)	text/csv	CSV	csvsr
IBM_Writing_Assistant_Fmt	181	154	IBM Writing Assistant		IWA	stringssr
WordStar_2000_Fmt	182	155	WordStar 2000		WS2	stringssr
HP_PCL_Fmt	183	157	HP Printer Control Language	application/pcl	PCL	
UNIX_Exe_PreSysV_VAX_Fmt	184	158	UNIX executable (PDP-11/pre-System V VAX)	application/octet-stream		
UNIX_Exe_Basic_16_Fmt	185	158	UNIX executable (Basic-16)	application/octet-stream		
UNIX_Exe_x86_Fmt	186	158	UNIX executable (x86)	application/octet-stream		
UNIX_Exe_iAPX_286_Fmt	187	158	UNIX executable (iAPX 286)	application/octet-stream		
UNIX_Exe_MC68k_Fmt	188	158	UNIX executable (MC680x0)	application/octet-stream		
UNIX_Exe_3B20_Fmt	189	158	UNIX executable (3B20)	application/octet-stream		
UNIX_Exe_WE32000_Fmt	190	158	UNIX executable (WE32000)	application/octet-stream		
UNIX_Exe_VAX_Fmt	191	158	UNIX executable (VAX)	application/octet-stream		
UNIX_Exe_Bell_5_Fmt	192	158	UNIX executable (Bell 5.0)	application/octet-stream		
UNIX_Obj_VAX_Demand_Fmt	193	159	UNIX object module (VAX Demand)			
UNIX_Obj_MS8086_Fmt	194	159	UNIX object module (old MS 8086)			
UNIX_Obj_Z8000_Fmt	195	159	UNIX object module (Z8000)			
AU_Audio_Fmt	196	161	NeXT/Sun Audio Data	audio/basic	AU, SND	MCI
NeWS_Font_Fmt	197	162	NeWS bitmap font			
cpio_Archive_CRChdr_Fmt	198	163	cpio archive (CRC Header)	application/x-cpio		
cpio_Archive_CHRhdr_Fmt	199	163	cpio archive (CHR Header)	application/x-cpio		
PEX_Binary_Archive_Fmt	200	164	SUN PEX Binary Archive			
Sun_vfont_Fmt	201	165	SUN vfont Definition			
Curses_Screen_Fmt	202	166	Curses Screen Image			
UUEncoded_Fmt	203	167	UU-encoded text	text/x-uuencode	UUE	uudsr
WriteNow_Fmt	204	168	WriteNow MAC			stringssr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
PC_Obj_Fmt	205	169	DOS/Windows Object Module	application/octet-stream	OBJ	
Windows_Group_Fmt	206	170	Windows Group			
TrueType_Font_Fmt	207	171	TrueType Font	application/x-font-ttf	TTF	
Windows_PIF_Fmt	208	172	Program Information File (PIF)	application/octet-stream	PIF	
MS_COM_Executable_Fmt	209	173	PC (.COM)	application/octet-stream	COM	
StuffIt_Fmt	210	175	StuffIt (MAC)	application/x-stuffit	HQX	
PeachCalc_Fmt	211	176	PeachCalc		CAL	
Wang_GDL_Fmt	212	177	WANG Office GDL Header			
Q_A_DOS_Fmt	213	179	Q & A for DOS			stringssr
Q_A_Win_Fmt	214	180	Q & A for Windows		JW	stringssr
WPS_PLUS_Fmt	215	181	WPS-PLUS	application/vnd.ms-wpl	WPL	stringssr
DCX_Fmt	216	182	DCX FAX Format(PCX images)	image/dcx	DCX	kpdcxrdr
OLE_Fmt	217	183	OLE Compound Document		OLE	olesr
EBCDIC_Fmt	218	186	EBCDIC Text			
DCS_Fmt	219	187	DCS			
UNIX_SHAR_Fmt	220	190	SHAR shell archive format	application/x-shar	SHAR	
Lotus_Notes_BitMap_Fmt	221	191	Lotus Notes Bitmap			
Lotus_Notes_CDF_Fmt	222	193	Lotus Notes CDF	application/cdf	CDF	stringssr
Compress_Fmt	223	192	UNIX Compress archive	application/x-compress	Z	kvzee, kvzeesr
GZ_Compress_Fmt	224	198	GZ Compress archive	application/gzip	GZ	kvgz, kvgzsr
TAR_Fmt	225	194	TAR (tape archive)	application/tar	TAR	tarsr
ODIF_FOD26_Fmt	226	196	Open Document Architecture (ODA / ODIF) FOD26	application/oda	F26	
ODIF_FOD36_Fmt	227	196	Open Document Architecture (ODA / ODIF) FOD36	application/oda	F36	
ALIS_Fmt	228	197	ALIS			
Envoy_Fmt	229	199	WordPerfect Envoy	application/envoy	EVY	
PDF_Fmt	230	200	Adobe PDF (Portable Document Format)	application/pdf	PDF	kppdf2rdr, kppdfrdr,

Format Name	Number	Category	Description	MIME Type	Extension	Readers
						pdf2sr, pdfsr
BinHex_Fmt	231	206	BinHex	application/mac-binhex40	HQX	kvhqxsr
SMTP_Fmt	232	207	SMTP (Text Mail / Outlook Express)	message/rfc822	SMTP	emlsr
MIME_Fmt	233	208	MIME (EML / MBX email)[1]	message/rfc822	EML, MBX	mbxsr
USENET_Fmt	234	264	USENET	message/news		
SGML_Fmt	235	209	SGML	text/sgml	SGML	afsr
HTML_Fmt	236	210	HTML	text/html	HTM, HTML	htmsr
ACT_Fmt	237	211	ACT! CRM software		ACT	
PNG_Fmt	238	213	Portable Network Graphics (PNG)	image/png	PNG	kppngrdr, pngsr
MS_Video_Fmt	239	214	Video for Windows (AVI)	video/avi	AVI	MCI
Windows_Animated_Cursor_Fmt	240	215	Windows Animated Cursor		ANI	kpanirdr
Windows_CPP_Obj_Storage_Fmt	241	216	Windows C++ Object Storage			
Windows_Palette_Fmt	242	217	Windows Palette		PAL	
RIFF_DIB_Fmt	243	218	RIFF Device Independent Bitmap			
RIFF_MIDI_Fmt	244	219	RIFF MIDI	audio/midi	RMI	
RIFF_Multimedia_Movie_Fmt	245	220	RIFF Multimedia Movie		MMM	
MPEG_Fmt	246	221	MPEG Movie	video/mpeg		
QuickTime_Fmt	247	222	QuickTime Movie, MPEG-4 audio	video/quicktime	MOV, QT, MP4	MCI, mpeg4sr
AIFF_Fmt	248	223	Audio Interchange File Format (AIFF)	audio/aiff	AIF, AIFF, AIFC	MCI, aiffsr
Amiga_MOD_Fmt	249	224	Amiga MOD		MOD	
Amiga_IFF_8SVX_Fmt	250	225	Amiga IFF (8SVX) Sound	audio/x-8svx	IFF	
Creative_Voice_Audio_Fmt	251	226	Creative Voice (VOC)		VOC	
AutoDesk_Animator_FLI_Fmt	252	229	AutoDesk Animator FLIC	video/x-fli	FLI	
AutoDesk_AnimatorPro_FLC_Fmt	253	230	AutoDesk Animator Pro FLIC	video/x-flc	FLC	
Compactor_Archive_Fmt	254	233	Compactor / Compact Pro	application/mac-compactpro		
VRML_Fmt	255	234	VRML	model/vrml	WRL	
QuickDraw_3D_Metafile_Fmt	256	235	QuickDraw 3D Metafile			
PGP_Secret_Keyring_Fmt	257	236	PGP Secret Keyring	application/pgp		

Format Name	Number	Category	Description	MIME Type	Extension	Readers
PGP_Public_Keyring_Fmt	258	237	PGP Public Keyring	application/pgp		
PGP_Encrypted_Data_Fmt	259	238	PGP Encrypted Data	application/pgp		
PGP_Signed_Data_Fmt	260	239	PGP Signed Data	application/pgp		
PGP_SignedEncrypted_Data_Fmt	261	240	PGP Signed and Encrypted Data	application/pgp		
PGP_Sign_Certificate_Fmt	262	241	PGP Signature Certificate	application/pgp-signature	SIG	
PGP_Compressed_Data_Fmt	263	246	PGP Compressed Data	application/pgp		
PGP_ASCII_Public_Keyring_Fmt	264	242	ASCII-armored PGP Public Keyring	application/pgp	PGP	
PGP_ASCII_Encoded_Fmt	265	243	ASCII-armored PGP encoded	application/pgp		
PGP_ASCII_Signed_Fmt	266	244	ASCII-armored PGP signed	application/pgp		
OLE_DIB_Fmt	267	245	OLE DIB object			
SGI_Image_Fmt	268	247	SGI RGB Image	image/sgi	RGB	kpsgirdr
Lotus_ScreenCam_Fmt	269	248	Lotus ScreenCam	application/vnd.lotus-screencam	SCM	
MPEG_Audio_Fmt	270	249	MPEG-1 Audio layer3 (MP3)	audio/mpeg	MPEGA, MPG, MP3	MCI, mp3sr
FTP_Software_Session_Fmt	271	250	FTP Session Data		STE	
Netscape_Bookmark_File_Fmt	272	210	Netscape Bookmark File	text/html		htmsr
Corel_Draw_CMX_Fmt	273	252	Corel CMX	application/cmx	CMX	
AutoDesk_DWG_Fmt	274	253	AutoDesk AutoCAD Drawing (DWG)	image/x-dwg	DWG	kpDWGrdr, kpODArdr
AutoDesk_WHIP_Fmt	275	254	AutoDesk WHIP		WHP	
Macromedia_Director_Fmt	276	255	Macromedia Shockwave/Adobe Director	application/x-director	DCR, DXR, DIR	
Real_Audio_Fmt	277	256	Real Audio	audio/x-pn-realaudio	RM, RA	
MSDOS_Device_Driver_Fmt	278	257	MSDOS Device Driver	application/octet-stream	SYS	
Micrografx_Designer_Fmt	279	258	Micrografx Designer		DSF	
SVF_Fmt	280	259	Simple Vector Format (SVF)	image/x-svf	SVF	
Applix_Words_Fmt	281	261	Applix Words	application/x-applix-word	AW	awsr
Applix_Graphics_Fmt	282	262	Applix Graphics		AG	kpagrdr
MS_Access_Fmt	283	263	Microsoft Access (versions 1 and 2)	application/x-msaccess	MDB	mdbsr
MS_Access_95_Fmt	284	263	Microsoft Access 95	application/msaccess	MDB	mdbsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
MS_Access_97_Fmt	285	263	Microsoft Access 97	application/msaccess	MDB	mdbsr
MacBinary_Fmt	286	265	MacBinary	application/x-macbinary	BIN	macbinsr
Apple_Single_Fmt	287	266	Apple Single			
Apple_Double_Fmt	288	267	Apple Double	multipart/appledouble	AD	
Enhanced_Metafile_Fmt	289	270	Enhanced Metafile	image/x-emf	EMF	kpemfrdr
MS_Office_Drawing_Fmt	290	271	Microsoft Office Drawing			kpmsordr
XML_Fmt	291	285	XML	text/xml	XML	xmlsr
DeVice_Independent_Fmt	292	274	DeVice Independent file (DVI)	application/x-dvi	DVI	
Unicode_Fmt	293	275	Unicode text file	text/plain	UNI	unisr
Lotus_123_Worksheet_Fmt	294	81	Lotus 1-2-3	application/x-lotus-123	WKS, WK1, WK3, WK4	wkssr
Lotus_123_Format_Fmt	295	81	Lotus 1-2-3 Formatting	application/x-123	FM3	l123sr
Lotus_123_97_Fmt	296	81	Lotus 1-2-3 97	application/x-lotus-123	123	l123sr
Lotus_Word_Pro_96_Fmt	297	268	Lotus Word Pro 96	application/vnd.lotus-wordpro	LWP, MWP	lwpsr
Lotus_Word_Pro_97_Fmt	298	268	Lotus Word Pro 97	application/vnd.lotus-wordpro	LWP, MWP	lwpsr
Freelance_DOS_Fmt	299	140	Lotus Freelance for DOS	application/x-freelance	PRZ	kpprzrdr
Freelance_Win_Fmt	300	140	Lotus Freelance for Windows	application/x-freelance	PRE, FLW	kpprerdr
Freelance_OS2_Fmt	301	140	Lotus Freelance for OS/2	application/x-freelance	PRS	kpprerdr
Freelance_96_Fmt	302	140	Lotus Freelance 96	application/x-freelance	PRZ	kpprzrdr
Freelance_97_Fmt	303	140	Lotus Freelance 97	application/x-freelance	PRZ	kpprzrdr
MS_Word_95_Fmt	304	189	Microsoft Word 95	application/msword	DOC	mw6sr
MS_Word_97_Fmt	305	269	Microsoft Word 97	application/msword	DOC, WPS, WBK	mw8sr
Excel_Fmt	306	90	Microsoft Excel (up to version 5)	application/x-ms-excel	XLS	xlssr
Excel_Chart_Fmt	307	90	Microsoft Excel (up to version 5) chart	application/x-ms-excel	XLC	xlssr
Excel_Macro_Fmt	308	90	Microsoft Excel (up to version 5) macro	application/vnd.ms-excel	XLM	xlssr
Excel_95_Fmt	309	188	Microsoft Excel 95	application/x-ms-excel	XLS	xlssr
Excel_97_Fmt	310	188	Microsoft Excel 97	application/x-ms-excel	XLS, XLR	xlssr
Corel_Presentations_Fmt	311	127	Corel Presentations	application/x-corelpresentations	XFD, XFDL	kpshwrdr
Harvard_Graphics_Fmt	312	131	Harvard Graphics		PR4	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Harvard_Graphics_Chart_Fmt	313	131	Harvard Graphics Chart		CH3, CHT	
Harvard_Graphics_Symbol_Fmt	314	131	Harvard Graphics Symbol File		SY3	
Harvard_Graphics_Cfg_Fmt	315	131	Harvard Graphics Configuration File			
Harvard_Graphics_Palette_Fmt	316	131	Harvard Graphics Palette			
Lotus_123_R9_Fmt	317	81	Lotus 1-2-3 Release 9	application/x-lotus-123	123	l123sr
Applix_Spreadsheets_Fmt	318	278	Applix Spreadsheets	application/x-applix-spreadsheet	AS	assr
MS_Pocket_Word_Fmt	319	45	Microsoft Pocket Word		PWD	rtfsr
MS_DIB_Fmt	320	279	Microsoft Device Independent Bitmap	image/bmp	DIB	
MS_Word_2000_Fmt	321	269	Microsoft Word 2000	application/msword	DOC	mw8sr
Excel_2000_Fmt	322	188	Microsoft Excel 2000	application/x-ms-excel	XLS	xlssr
PowerPoint_2000_Fmt	323	272	Microsoft PowerPoint 2000	application/x-ms-powerpoint	PPT	kpp97rdr
MS_Access_2000_Fmt	324	263	Microsoft Access 2000	application/x-msaccess	MDB	mdbsr
MS_Project_4_Fmt	325	281	Microsoft Project 4		MPP	mppsr
MS_Project_41_Fmt	326	281	Microsoft Project 4.1		MPP	mppsr
MS_Project_98_Fmt	327	281	Microsoft Project 98	application/vnd.ms-project	MPP	mppsr
Folio_Flat_Fmt	328	282	Folio Flat File		FFF	foliosr
HWP_Fmt	329	283	Haansoft Hangul HWP (Arae-Ah Hangul)	application/x-hwp	HWP	hwposr, hwpsr
ICHITARO_Fmt	330	284	ICHITARO (v4-10)		JTD	jtdsr
IS_XML_Fmt	331	273	Extended or Custom XML	text/xml	XML	
Oasys_Fmt	332	286	Fujitsu OASYS	application/vnd.fujitsu.oasys	OAS, OA2, OA3	oa2sr
PBM_ASC_Fmt	333	287	Portable Bitmap Utilities ASCII format (PBM)	image/pbm	PBM	
PBM_BIN_Fmt	334	287	Portable Bitmap Utilities BINARY format (PBM)	image/pbm	PBM	
PGM_ASC_Fmt	335	288	Portable Greymap Utilities ASCII format (PGM)	image/x-pgm	PGM	
PGM_BIN_Fmt	336	288	Portable Greymap Utilities BINARY format (PGM)	image/x-pgm	PGM	
PPM_ASC_Fmt	337	289	Portable Pixmap Utilities ASCII format (PPM)	image/x-portable-pixmap	PPM	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
PPM_BIN_Fmt	338	289	Portable Pixmap Utilities BINARY format (PPM)	image/x-portable-pixmap	PPM	
XBM_Fmt	339	290	X Bitmap format (XBM)	image/x-xbitmap	XBM	
XPM_Fmt	340	291	X Pixmap format (XPM)	image/xpm	XPM	
FPX_Fmt	341	292	Kodak FlashPix FPX Image format	image/fpx	FPX	
PCD_Fmt	342	293	PCD Image format	image/pcd	PCD	
MS_Visio_Fmt	343	294	Microsoft Visio (up to version 11)	image/x-vsd	VSD	kpVSD2rdr, vsdsr
MS_Project_2000_Fmt	344	281	Microsoft Project 2000	application/vnd.ms-project	MPP	mppsr
MS_Outlook_Fmt	345	295	Microsoft Outlook message	application/vnd.ms-outlook	MSG, OFT	msgsr
ELF_Relocatable_Fmt	346	159	ELF Relocatable	application/octet-stream	O	
ELF_Executable_Fmt	347	158	ELF Executable	application/octet-stream		
ELF_Dynamic_Lib_Fmt	348	160	ELF Dynamic Library	application/octet-stream	SO	
MS_Word_XML_Fmt	349	285	Microsoft Word 2003 XML	text/xml	XML	xmlsr
MS_Excel_XML_Fmt	350	285	Microsoft Excel 2003 XML	text/xml	XML	xmlsr
MS_Visio_XML_Fmt	351	285	Microsoft Visio 2003 XML	text/xml	VDX	xmlsr
SO_Text_XML_Fmt	352	314	OpenDocument format (OpenOffice 1/StarOffice 6,7) Text XML	application/vnd.sun.xml.writer	SXW	odfwpsr
SO_Spreadsheet_XML_Fmt	353	315	OpenDocument format (OpenOffice 1/StarOffice 6,7) Spreadsheet XML	application/vnd.sun.xml.calc	SXC, STC	sosr
SO_Presentation_XML_Fmt	354	316	OpenDocument format (OpenOffice 1/StarOffice 6,7) Presentation XML	application/vnd.sun.xml.impress	SXD, SXI	kpodfrdr
XHTML_Fmt	355	296	XHTML	text/xhtml	XML, XHTML, XHT	
MS_OutlookPST_Fmt	356	297	Microsoft Outlook Personal Folders File (.pst)	application/vnd.ms-outlook-pst	PST	pstnsr, pstsr, pstxsr
RAR_Fmt	357	298	RAR archive format	application/x-rar-compressed	RAR, REV, R00, R01	rarsr
Lotus_Notes_NSF_Fmt	358	299	IBM Lotus Notes Database NSF/NTF	application/x-lotus-notes	NSF	nsfsr
Macromedia_Flash_Fmt	359	300	Macromedia Flash (.swf)	application/x-shockwave-flash	SWF, SWD	swfsr
MS_Word_2007_Fmt	360	301	Microsoft Word 2007 XML - Docx	application/x-ms-word07	DOCX, DOTX	mwxsr
MS_Excel_2007_Fmt	361	302	Microsoft Excel 2007 XML	application/x-ms-excel07	XLSX, XLTX	xlsxsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
MS_PPT_2007_Fmt	362	303	Microsoft PowerPoint 2007 XML	application/x-ms-powerpoint07	PPTX, POTX, PPSX	kpppxrdr
OpenPGP_Fmt	363	304	OpenPGP Message Format (with new packet format)	application/pgp-encrypted	PGP	
Intergraph_V7_DGN_Fmt	364	305	Intergraph Standard File Format (ISFF) V7 DGN (non-OLE)		DGN	
MicroStation_V8_DGN_Fmt	365	306	MicroStation V8 DGN (OLE)		DGN	
MS_Word_Macro_2007_Fmt	366	307	Microsoft Word Macro 2007 XML	application/x-ms-word07m	DOCM, DOTM	mwxsr
MS_Excel_Macro_2007_Fmt	367	308	Microsoft Excel Macro 2007 XML	application/x-ms-excel07m	XLSM, XLTM, XLAM	xlsxsr
MS_PPT_Macro_2007_Fmt	368	309	Microsoft PPT Macro 2007 XML	application/x-ms-powerpoint07m	PPTM, POTM, PPSM, PPAM	kpppxrdr
LZH_Fmt	369	310	LZH Archive	application/x-lzh-compressed	LZH, LHA	lzhsr
Office_2007_Fmt	370	311	Office 2007 document		XLSB	
MS_XPS_Fmt	371	312	Microsoft Open XML Paper Specification (XPS/OXPS)	application/vnd.ms-xpsdocument	XPS, OXPS	xpssr
Lotus_Domino_DXL_Fmt	372	313	IBM Domino Data in XML format (.dxl)	text/xml	DXL	dxlsr
ODF_Text_Fmt	373	314	ODF Text	application/vnd.oasis.opendocument.text	ODT	odfwpsr
ODF_Spreadsheet_Fmt	374	315	ODF Spreadsheet	application/vnd.oasis.opendocument.spreadsheet	ODS	odfsssr
ODF_Presentation_Fmt	375	316	ODF Presentation	application/vnd.oasis.opendocument.presentation	ODP	kpodfrdr
Legato_Extender_ONM_Fmt	376	317	Legato Extender Native Message ONM	application/x-lotus-notes	ONM	onmsr
bin_Unknown_Fmt	377	318	Bin unknown format (.xxx)			
TNEF_Fmt	378	319	Transport Neutral Encapsulation Format (TNEF)	application/vnd.ms-tnef		tnefsr
CADAM_Drawing_Fmt	379	320	CADAM Drawing		CDD	
CADAM_Drawing_Overlay_Fmt	380	321	CADAM Drawing Overlay		CDO	
NURSTOR_Drawing_Fmt	381	322	NURSTOR Drawing		NUR	
HP_GLP_Fmt	382	323	HP Graphics Language (Plotter)	vector/x-hpgl2	HPG	
ASF_Fmt	383	324	Advanced Systems Format (ASF)	application/x-ms-asf	ASF	asfsr
WMA_Fmt	384	325	Windows Media Audio Format (WMA)	audio/x-ms-wma	WMA	asfsr
WMV_Fmt	385	326	Windows Media Video Format (WMV)	video/x-ms-wmv	WMV	asfsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
EMX_Fmt	386	327	Legato EMailXtender Archives Format (EMX)		EMX	emxsr
Z7Z_Fmt	387	328	7-Zip archive (7z)	application/7z	7Z	z7zsr
MS_Excel_Binary_2007_Fmt	388	329	Microsoft Excel Binary 2007	application/vnd.ms-excel.sheet.binary.macroenabled.12	XLSB	xlsbsr
CAB_Fmt	389	330	Microsoft Cabinet File (CAB)	application/vnd.ms-cab-compressed	CAB	cabsr
CATIA_Fmt	390	331	CATIA Formats (CAT*)		CATPART, CATPRODUCT[2]	kpCATrdr
YIM_Fmt	391	332	Yahoo! Instant Messenger History		DAT	yimsr
ODF_Drawing_Fmt	392	316	ODF Drawing/Graphics	application/vnd.oasis.opendocument.graphics	ODG	kpodfrdr
Founder_CEB_Fmt	393	333	Founder Chinese E-paper Basic (ceb)	application/ceb	CEB	cebsr
QPW_Fmt	394	334	Corel Quattro Pro 9+ for Windows	application/quattro-pro	QPW	qpwsr
MHT_Fmt	395	335	MIME HTML MHTML format (MHT)[1]	multipart/related	MHT, MHTML	mhtsr
MDI_Fmt	396	336	Microsoft Document Imaging Format	image/vnd.ms-modi	MDI	
GRV_Fmt	397	337	Microsoft Office Groove Format	application/vnd.groove-injector	GRV	
IWWP_Fmt	398	338	Apple iWork Pages format	application/vnd.apple.pages	PAGES	iwwpsr
IWSS_Fmt	399	339	Apple iWork Numbers format	application/vnd.apple.numbers	NUMBERS	iwsssr
IWPG_Fmt	400	340	Apple iWork Keynote format	application/vnd.apple.keynote	KEY	kpIWPGrdr
BKF_Fmt	401	341	Microsoft Windows Backup File		BKF	bkfsr
MS_Access_2007_Fmt	402	342	Microsoft Access 2007	application/msaccess	ACCDB	mdbsr
ENT_Fmt	403	343	Microsoft Entourage Database Format			entsr
DMG_Fmt	404	344	Mac Disk Copy Disk Image File	application/x-apple-diskimage	DMG	dmgsr
CWK_Fmt	405	345	AppleWorks (Claris Works) File	application/appleworks	CWK	stringssr
OO3_Fmt	406	346	Omni Outliner V3 File		OO3	oo3sr
OPML_Fmt	407	347	Omni Outliner OPML File		OPML	oo3sr
Omni_Graffle_XML_Fmt	408	348	Omni Graffle XML File		GRAFFLE	kpGFLrdr
PSD_Fmt	409	349	Adobe Photoshop Document	image/vnd.adobe.photoshop	PSD, PSB	psdsr
Apple_Binary_PList_Fmt	410	350	Apple Binary Property List format		PLIST	
Apple_iChat_Fmt	411	351	Apple iChat format		ICHAT	ichatsr
OOUTLINE_Fmt	412	352	OOutliner File		OOUTLINE	oo3sr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
BZIP2_Fmt	413	353	Bzip 2 Compressed File	application/x-bzip2	BZ2	bzip2sr
ISO_Fmt	414	354	ISO-9660 CD Disc Image Format	application/x-iso9660-image	ISO	isosr
DocuWorks_Fmt	415	355	DocuWorks Format	application/vnd.fujixerox.docuworks	XDW	
RealMedia_Fmt	416	356	RealMedia Streaming Media	application/vnd.rn-realmedia	RM, RA	
AC3Audio_Fmt	417	357	AC3 Audio File Format	audio/ac3	AC3	
NEF_Fmt	418	358	Nero Encrypted File		NEF	
SolidWorks_Fmt	419	359	SolidWorks Format Files		SLDASM, SLDPRT, SLDDRW, SLDDRT	
XFDL_Fmt	420	366	Extensible Forms Description Language	application/x-xfdl	XFDL, XFD	kpXFDLrdr
Apple_XML_PList_Fmt	421	367	Apple XML Property List format		PLIST	
OneNote_Fmt	422	368	Microsoft OneNote Note Format	application/onenote	ONE	kpONErdr
IFilter_Fmt	423	369	iFilter			
Dicom_Fmt	424	370	Digital Imaging and Communications in Medicine (Dicom)	application/dicom	DCM	dcmsr
EnCase_Fmt	425	371	Expert Witness Compression Format (EnCase)		E01, L01, Lx01	encase2sr, encasesr
Scrap_Fmt	426	372	Shell Scrap Object File		SHS	olesr
MS_Project_2007_Fmt	427	373	Microsoft Project 2007	application/vnd.ms-project	MPP	mppsr
MS_Publisher_98_Fmt	428	374	Microsoft Publisher from version 98	application/x-mspublisher	PUB	mspubsr
Skype_Fmt	429	375	Skype Log File		DBB	skypesr
Hl7_Fmt	430	377	Health level7 message		HL7	hl7sr
MS_OutlookOST_Fmt	431	378	Microsoft Outlook Offline Folders File (OST)	application/vnd.ms-outlook-pst	OST	pffsr
Epub_Fmt	432	379	Open Publication Structure electronic publication	application/epub+zip	EPUB	epubsr
MS_OEDBX_Fmt	433	380	Microsoft Outlook Express DBX Message Database		DBX	dbxsr
BB_Activ_Fmt	434	381	BlackBerry Activation File		DAT	
DiskImage_Fmt	435	382	Disk Image		DMG	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Milestone_Fmt	436	383	Milestone Document		MLS, ML3, ML4, ML5, ML6, ML7, ML8, ML9, MLA	
E_Transcript_Fmt	437	384	RealLegal E-Transcript File		PTX	
PostScript_Font_Fmt	438	385	PostScript Type 1 Font	application/x-font	PFB	
Ghost_DiskImage_Fmt	439	386	Ghost Disk Image File		GHO, GHS	
JPEG_2000_JP2_File_Fmt	440	387	JPEG-2000 JP2 File Format Syntax (ISO/IEC 15444-1)	image/jp2	JP2, JPF, J2K, JPWL, JPX, PGX	jp2000sr, kpjp2000rdr
Unicode_HTML_Fmt	441	388	Unicode HTML	text/html	HTM, HTML	unihtmsr
CHM_Fmt	442	389	Microsoft Compiled HTML Help	application/x-chm	CHM	chmsr
EMCMF_Fmt	443	390	Documentum EMCMF format		EMCMF	msgsr
MS_Access_2007_Tmpl_Fmt	444	391	Microsoft Access 2007 Template		ACCDT	
Jungum_Fmt	445	392	Samsung Electronics Jungum Global document		GUL	
JBIG2_Fmt	446	393	JBIG2 File Format	image/jbig2	JB2, JBIG2	kpJBIG2rdr
EFax_Fmt	447	394	eFax file		EFX	
AD1_Fmt	448	395	AD1 Evidence file		AD1	ad1sr
SketchUp_Fmt	449	396	Google SketchUp		SKP	
GWFS_Email_Fmt	450	397	GroupWise FileSurf email		GWFS	gwfssr
JNT_Fmt	451	398	Windows Journal format		JNT	
Yahoo_yChat_Fmt	452	399	Yahoo! Messenger chat log		YCHAT	
PaperPort_MAX_File_Fmt	453	400	PaperPort MAX image file	image/max	MAX	
ARJ_Fmt	454	402	ARJ (Archive by Robert Jung) file format	application/arj	ARJ	multiarcsr
RPMSG_Fmt	455	403	Microsoft Outlook Restricted Permission Message	application/x-microsoft-rpmsg-message	RPMSG	
MAT_Fmt	456	404	MATLAB file format	application/x-matlab-data	MAT, FIG	
SGY_Fmt	457	405	SEG-Y Seismic Data format		SGY, SEGY	
CDXA_MPEG_PS_Fmt	458	406	MPEG-PS container with CDXA stream	video/mpeg	MPG	
EVT_Fmt	459	407	Microsoft Windows NT Event Log		EVT	
EVTX_Fmt	460	408	Microsoft Windows Vista Event Log		EVTX	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
MS_OutlookOLM_Fmt	461	409	Microsoft Outlook for Macintosh format		OLM	olmsr
WARC_Fmt	462	410	Web ARChive	application/warc	WARC	
JAVACLASS_Fmt	463	411	Java Class format	application/x-java-class	CLASS	
VCF_Fmt	464	412	Microsoft Outlook vCard file format	text/vcard	VCF	vcfsr
EDB_Fmt	465	413	Microsoft Exchange Server Database file format		EDB	
ICS_Fmt	466	414	Microsoft Outlook iCalendar file format	text/calendar	ICS, VCS	icssr
MS_Visio_2013_Fmt	467	415	Microsoft Visio 2013	application/vnd.visio	VSDX, VSTX, VSSX	ActiveX components, kpVSDXrdr
MS_Visio_2013_Macro_Fmt	468	415	Microsoft Visio 2013 macro	application/vnd.visio	VSDM, VSTM, VSSM	kpVSDXrdr
ICHITARO_Compr_Fmt	469	417	ICHITARO Compressed format	application/x-js-taro	JTDC	jtdsr
IWWP13_Fmt	470	418	Apple iWork 2013 Pages format		IWA, PAGES	iwwp13sr
IWSS13_Fmt	471	419	Apple iWork 2013 Numbers format		IWA, NUMBERS	iwss13sr
IWPG13_Fmt	472	420	Apple iWork 2013 Keynote format		IWA, KEY	kpIWPG13rdr, kpIWPGrdr
XZ_Fmt	473	421	XZ archive format	application/x-xz	XZ	multiarcsr
Sony_WAVE64_Fmt	474	422	Sony Wave64 format	audio/wav64	W64	
Conifer_WAVPACK_Fmt	475	423	Conifer Wavpack format	audio/x-wavpack	WV	
Xiph_OGG_VORBIS_Fmt	476	424	Xiph Ogg Vorbis format	audio/ogg	OGG	
MS_Visio_2013_Stencil_Fmt	477	415	MS Visio 2013 stencil format	application/vnd.visio	VSSX	kpVSDXrdr
MS_Visio_2013_Stencil_Macro_Fmt	478	415	MS Visio 2013 stencil Macro format	application/vnd.visio	VSSM	kpVSDXrdr
MS_Visio_2013_Template_Fmt	479	415	MS Visio 2013 template format	application/vnd.visio	VSTX	kpVSDXrdr
MS_Visio_2013_Template_Macro_Fmt	480	415	MS Visio 2013 template Macro format	application/vnd.visio	VSTM	kpVSDXrdr
Borland_Reflex_2_Fmt	481	425	Borland Reflex 2 format		R2D	
PKCS_12_Fmt	482	426	PKCS #12 (p12) format	application/x-pkcs12	P12, PFX	
B1_Fmt	483	427	B1 format	application/x-b1	B1	b1sr
ISO_IEC_MPEG_4_Fmt	484	428	ISO/IEC MPEG-4 (ISO 14496) format	video/mp4	MP4	mpeg4sr
RAR5_Fmt	485	429	RAR5 Format	application/x-rar-compressed	RAR	multiarcsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Unigraphics_NX_Fmt	486	362	Unigraphics (UG) NX CAD Format		PRT	kpUGrdr
PTC_Creo_Fmt	487	430	PTC Creo CAD Format		ASM, PRT	
KML_Fmt	488	431	Keyhole Markup Language	application/vnd.google-earth.kml+xml	KML	xmlsr
KMZ_Fmt	489	432	Zipped Keyhole Markup Language	application/vnd.google-earth.kmz	KMZ	unzip
WML_Fmt	490	433	Wireless Markup Language	text/vnd.wap.wml	WML	xmlsr
ODF_Formula_Fmt	491	434	ODF Formula	application/vnd.oasis.opendocument.formula	ODF	unzip
SO_Text_Fmt	492	435	Star Office 4,5 Writer Text	application/vnd.stardivision.writer	SDW, SGL, VOR	kpsdwrdr, starwsr
SO_Spreadsheet_Fmt	493	436	Star Office 4,5 Calc Spreadsheet	application/vnd.stardivision.calc	SDC	starcsr
SO_Presentation_Fmt	494	437	Star Office 4,5 Impress Presentation	application/vnd.stardivision.draw	SDD, SDA	kpsddrdr
SO_Math_Fmt	495	438	Star Office 4,5 Math	application/vnd.stardivision.math	SMF	
STEP_Fmt	496	439	ISO 10303-21 STEP format			
STL_Fmt	497	364	3D Systems STL ASCII format			
AppleScript_Fmt	498	440	AppleScript Source Code[3]	text/x-applescript	APPLESCRIPT	afsr
Assembly_Fmt	499	441	Assembly Code[3]	text/x-assembly		afsr
C_Fmt	500	442	C Source Code[3]	text/x-c	C, H	afsr
Csharp_Fmt	501	443	C# Source Code[3]	text/x-csharp	CS	afsr
CPlusPlus_Fmt	502	444	C++ Source Code[3]	text/x-c++	CPP, HPP	afsr
Css_Fmt	503	445	Cascading Style Sheet[3]	text/css	CSS	afsr
Clojure_Fmt	504	446	Clojure Source Code[3]	text/x-clojure	CLJ, CL2	afsr
CoffeeScript_Fmt	505	447	CoffeeScript Source Code[3]	text/x-coffeescript	COFFEE, CAKE	afsr
Lisp_Fmt	506	448	Common Lisp Source Code[3]	text/x-common-lisp	EL	afsr
Dockerfile_Fmt	507	449	Dockerfile[3]	text/x-dockerfile		afsr
Eiffel_Fmt	508	450	Eiffel Source Code[3]	text/x-eiffel	E	afsr
Erlang_Fmt	509	451	Erlang Source Code[3]	text/x-erlang	ERL, ES	afsr
Fsharp_Fmt	510	452	F# Source Code[3]	text/x-fsharp	FS	afsr
Fortran_Fmt	511	453	Fortran Source Code[3]	text/x-fortran	F	afsr
Go_Fmt	512	454	Go Source Code[3]	text/x-go	GO	afsr
Groovy_Fmt	513	455	Groovy Source Code[3]	text/x-groovy	GRT, GVY	afsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Haskell_Fmt	514	456	Haskell Source Code[3]	text/x-haskell	HS	afsr
Ini_Fmt	515	457	Initialization (INI) file[3]	text/x-ini		afsr
Java_Fmt	516	458	Java Source Code[3]	text/x-java-source	JAVA	afsr
Javascript_Fmt	517	459	Javascript Source Code[3]	text/javascript	JS	afsr
Lua_Fmt	518	460	Lua Source Code[3]	text/x-lua	LUA	afsr
Makefile_Fmt	519	461	Makefile[3]	text/x-makefile	MAKE	afsr
Mathematica_Fmt	520	462	Wolfram Mathematica Source Code[3]	text/x-mathematica	M	afsr
ObjC_Fmt	521	464	Objective-C Source Code[3]	text/x-objc		afsr
ObjCpp_Fmt	522	465	Objective-C++ Source Code[3]	text/x-objectivec++		afsr
ObjJ_Fmt	523	466	Objective-J Source Code[3]	text/x-objectivej	J	afsr
PHP_Fmt	524	467	PHP Source Code[3]	text/x-php	PHP	afsr
PLSQL_Fmt	525	468	PLSQL Source Code[3]	text/x-plsql		afsr
Pascal_Fmt	526	469	Pascal Source Code[3]	text/x-pascal	PASCAL	afsr
Perl_Fmt	527	470	Perl Source Code[3]	text/x-perl	PL	afsr
Powershell_Fmt	528	471	PowerShell Source Code[3]	text/x-powershell	PS1	afsr
Prolog_Fmt	529	472	Prolog Source Code[3]	text/x-prolog	PRO, PROLOG	afsr
Puppet_Fmt	530	473	Puppet Source Code[3]	text/x-puppet	PP	afsr
Python_Fmt	531	474	Python Source Code[3]	text/x-python	PY	afsr
R_Fmt	532	475	R Source Code[3]	text/x-rsrc	R	afsr
Ruby_Fmt	533	476	Ruby Source Code[3]	text/x-ruby	RB	afsr
Rust_Fmt	534	477	Rust Source Code[3]	text/x-rust	RS	afsr
Scala_Fmt	535	478	Scala Source Code[3]	text/x-scala	SC	afsr
Shell_Fmt	536	479	Shell Script[3]	application/x-sh	SH	afsr
Smalltalk_Fmt	537	480	Smalltalk Source Code[3]	text/x-stsrc	ST	afsr
ML_Fmt	538	481	Standard ML Source Code[3]	text/x-ml	ML	afsr
Swift_Fmt	539	482	Swift Source Code[3]	text/x-swift	SWIFT	afsr
Tcl_Fmt	540	483	Tool Command Language (Tcl) Source Code[3]	text/x-tcl	TM	afsr
Tex_Fmt	541	484	TeX Typesetting File[3]	application/x-tex		afsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
TypeScript_Fmt	542	485	TypeScript Source Code[3]	text/x-typescript	TS	afsr
Verilog_Fmt	543	486	Verilog Source Code[3]	text/x-verilog	V	afsr
YAML_Fmt	544	487	YAML File[3]	text/x-yaml	YML	afsr
Wiki_Fmt	545	488	MediaWiki File	text/x-mediawiki		afsr
MS_Word_2007_Flat_XML_Fmt	546	301	Microsoft Word 2007 XML - Flat xml	text/xml	XML	mwxsr
Matroska_Fmt	547	489	Matroska video File	video/x-matroska	MKV	
SVG_Fmt	548	490	Scalable Vector Graphics image	image/svg+xml	SVG	xmlsr
Shapefile_Fmt	549	491	Shapefile	application/x-shapefile	SHP, SHX	
Flash_Video_Fmt	550	492	Flash video File	video/x-flv	FLV	
Embedded_OpenType_Fmt	551	493	Embedded OpenType font	application/vnd.ms-fontobject	EOT	
Web_Open_Font_Fmt	552	494	Web Open Font Format	font/woff	WOFF, WOFF2	
OpenType_Fmt	553	495	OpenType Font	font/otf	OTF	
MNG_Fmt	554	496	Multiple-image Network Graphics	video/x-mng	MNG	
JNG_Fmt	555	497	JPEG Network Graphics	image/x-jng	JNG	
AppleScript_Binary_Fmt	556	498	AppleScript Binary Source Code		SCPT	
Maya_Binary_Fmt	557	499	Autodesk Maya binary file		MB	
Jupiter_Tesselation_Fmt	558	363	UGS Jupiter Tesselation file		JT	
OGV_Fmt	559	500	Ogg Theora Video format	video/ogg	OGV	
OGG_Container_Fmt	560	501	General Ogg Container format	application/ogg	OGG	
GNU_Message_Catalog_Fmt	561	502	GNU Message Catalog format		MO	
Windows_Shortcut_Fmt	562	503	Windows shortcut file	application/x-ms-shortcut	LNK	
Apple_Typedstream_Fmt	563	504	Apple/NeXT typedstream data format			
XCF_Fmt	564	505	GIMP XCF image	image/x-xcf	XCF	
PaintShop_Pro_Fmt	565	506	PaintShop Pro image		PSP, PSPIMAGE	
SQLite_Database_Fmt	566	507	SQLite database format	application/x-sqlite3	QHC	
MySQL_Table_Fmt	567	508	MySQL table definition file		FRM	
Microsoft_Program_DB_Fmt	568	509	Microsoft Program Database format		PDB	
OpenEXR_Fmt	569	510	OpenEXR image format		EXR	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
XMV_Fmt	570	511	4X Movie File		4XM	
AMV_Fmt	571	512	AMV video file		AMV	
NIFF_Fmt	572	513	Notation Interchange File Format		NIF	
CuBase_Fmt	573	514	Steinberg CuBase file			
SoundFont_Fmt	574	515	SoundFont file			
WebP_Fmt	575	516	WebP image	image/webp	WEBP	
ICC_Fmt	576	517	International Color Consortium files	application/vnd.iccprofile	ICC, ICM	
PCF_Fmt	577	518	X11 Portable Compiled Font file	application/x-font-pcf	PCF	
WebM_Fmt	578	519	WebM video file	video/webm	WEBM	
AMFF_Fmt	579	520	Amiga Metafile		AMF	
ANBM_Fmt	580	521	IFF Animated Bitmap			
ANIM_Fmt	581	522	IFF Amiga animated raster graphics format			
DEEP_Fmt	582	523	IFF-DEEP TVPaint image		DEEP	
FAXX_Fmt	583	524	IFF-FAXX Facsimile image			
ICON_Fmt	584	525	IFF Glow Icon image			
ILBM_Fmt	585	526	Interleaved BitMap image		IFF	
LWOB_Fmt	586	527	LightWave Object format		LWOB	
MAUD_Fmt	587	528	IFF-MAUD MacroSystem audio format			
PBM_Fmt	588	529	IFF Planar BitMap			
TDDD_Fmt	589	530	IFF TDDD and Imagine Object animation format		TDD	
DjVu_Fmt	590	531	AT&T DjVu format	image/vnd.djvu	DJVU	
InDesign_Fmt	591	532	Adobe InDesign document	application/x-indesign	INDD	
Calamus_Fmt	592	533	Calamus Desktop Publishing			
Adaptive_MultiRate_Fmt	593	534	Adaptive Multi-Rate audio format	audio/amr	AMR	
FLAC_Fmt	594	535	Free Lossless Audio Codec format	audio/flac	FLAC	
Ogg_FLAC_Fmt	595	536	Ogg Container FLAC audio format		OGG	
SAS7BDAT_Fmt	596	537	SAS7BDAT database storage format		SAS7BDAT	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Design_Web_Format_Fmt	597	538	Autodesk Design Web Format	model/vnd.dwf	DWF	
Adobe_Flash_Audio_Book_Fmt	598	539	Adobe Flash Player audio book	audio/mp4	F4B	mpeg4sr
Adobe_Flash_Audio_Fmt	599	540	Adobe Flash Player audio	audio/mp4	F4A	mpeg4sr
Adobe_Flash_Protected_Video_Fmt	600	541	Adobe Flash Player protected video	video/mp4	F4P	mpeg4sr
Adobe_Flash_Video_Fmt	601	542	Adobe Flash Player video	video/x-f4v	F4V	mpeg4sr
Audible_Audiobook_Fmt	602	543	Audible Enhanced Audiobook	audio/vnd.audible.aax	AAX	mpeg4sr
Canon_Camera_Fmt	603	544	Canon Digital Camera image			
Canon_Raw_Fmt	604	545	Canon Raw image		CR3	
Casio_Camera_Fmt	605	546	Casio Digital Camera image			
Convergent_Design_Fmt	606	547	Convergent Design file			
DMB_MAF_Audio_Fmt	607	548	DMB MAF audio			
DMB_MAF_Video_Fmt	608	549	DMB MAF video			
DMP_Content_Fmt	609	550	Digital Media Project Content Format			
DVB_Fmt	610	551	Digital Video Broadcast format	video/vnd.dvb.file	DVB	
Dirac_Wavelet_Compression_Fmt	611	552	ISO-BMFF Dirac Wavelet compression			
HEICS_Image_Sequence_Fmt	612	553	High Efficiency Image Format HEVC image sequence	image/heic-sequence	HEICS	
HEIC_Image_Fmt	613	554	High Efficiency Image Format HEVC image	image/heic	HEIC	
HEIFS_Image_Sequence_Fmt	614	555	High Efficiency Image Format image sequence	image/heif-sequence	HEIFS	
HEIF_Image_Fmt	615	556	High Efficiency Image Format image	image/heif	HEIF	
ISMACryp_Fmt	616	557	ISMACryp 2.0 Encrypted format			
ISO_3GPP2_Fmt	617	558	3GPP2 video file	video/3gpp2	3G2	mpeg4sr
ISO_3GPP_Fmt	618	559	3GPP video file	video/3gpp	3GP	mpeg4sr
ISO_JPEG2000_JP2_Fmt	619	560	ISO-BMFF JPEG 2000 image	image/jp2	JP2	jp2000sr, kpjp2000rdr
ISO_JPEG2000_JPM_Fmt	620	561	ISO-BMFF JPEG 2000 compound image	image/jpm	JPM	jp2000sr, kpjp2000rdr
ISO_JPEG2000_JPX_Fmt	621	562	ISO-BMFF JPEG 2000 with extensions	image/jpx	JPX	jp2000sr, kpjp2000rdr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
ISO_QuickTime_Fmt	622	563	Apple ISO-BMFF QuickTime video	video/quicktime	QT, MOV	MCI
KDDI_Video_Fmt	623	564	KDDI Video file	video/3gpp2		mpeg4sr
MAF_Photo_Player_Fmt	624	565	MAF Photo Player			
MPEG4_AVC_Fmt	625	566	ISO-BMFF MPEG-4 with AVC extension	video/mp4		mpeg4sr
MPEG4_M4A_Fmt	626	567	Apple MPEG-4 Part 14 audio	audio/x-m4a	M4A	mpeg4sr
MPEG4_M4B_Fmt	627	568	Apple MPEG-4 Part 14 audio book	audio/mp4	M4B	mpeg4sr
MPEG4_M4P_Fmt	628	569	Apple MPEG-4 Part 14 protected audio	audio/mp4	M4P	mpeg4sr
MPEG4_M4V_Fmt	629	570	Apple MPEG-4 Part 14 video	video/x-m4v	M4V	mpeg4sr
MPEG4_Sony_PSP_Fmt	630	571	Sony PSP MPEG-4	audio/mp4	MP4	mpeg4sr
MPEG_21_Fmt	631	572	MPEG-21	audio/mp4		mpeg4sr
Mobile_QuickTime_Fmt	632	573	Mobile QuickTime video	video/quicktime	MQV	MCI
Motion_JPEG_2000_Fmt	633	574	Motion JPEG 2000	video/mj2	MJ2, MJP2	jp2000sr, kpjp2000rdr
NTT_MPEG4_Fmt	634	575	NTT MPEG-4	video/mp4		mpeg4sr
Nero_MPEG4_AVC_Profile	635	576	Nero MPEG-4 profile with AVC extension	video/mp4		
Nero_MPEG4_Audio_Fmt	636	577	Nero AAC audio	audio/mp4		mpeg4sr
Nero_MPEG4_Profile	637	578	Nero MPEG-4 profile	video/mp4		
OMA_DRM_Fmt	638	579	OMA DRM (ISOBMFF) Format			
Panasonic_Camera_Fmt	639	580	Panasonic Digital Camera image			
Ross_Video_Fmt	640	581	Ross video			
SDA_Video_Fmt	641	582	SDA SD Memory Card video			
Samsung_Stereoscopic_Fmt	642	583	Samsung stereoscopic stream			
Sony_XAVC_Fmt	643	584	Sony XAVC video			mpeg4sr
JPEG_2000_PGX_Fmt	644	585	JPEG 2000 PGX Verification Model image		PGX	jp2000sr, kpjp2000rdr
Apple_Desktop_Services_Store_Fmt	645	586	Apple Desktop Services Store file		DS_Store	
Core_Audio_Fmt	646	587	Apple Core Audio Format	audio/x-caf	CAF	
VICAR_Fmt	647	588	VICAR image format		IMG	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
FITS_Fmt	648	589	Flexible Image Transport System FITS image	image/fits	FIT	
DIF_Fmt	649	590	Digital Interface Format (DIF) DV video		DV	
MPEG_Transport_Stream_Fmt	650	591	MPEG Transport Stream data	video/MP2T	TS	
MPEG_Sequence_Fmt	651	592	MPEG Sequence format	video/mpeg		
Ogg_OGM_Fmt	652	593	Ogg OGM video format	video/ogg	OGM	
Ogg_Speex_Fmt	653	594	Ogg Speex audio format	audio/ogg	SPX	
Ogg_Opus_Fmt	654	595	Ogg Opus audio format	audio/ogg	OGG	
Musepack_Audio_Fmt	655	596	Musepack audio format	audio/x-musepack	MPC	
ART_Image_Fmt	656	597	ART image format		ART	
Vivo_Fmt	657	598	Vivo audio-video format	video/vnd.vivo	VIV	
QCP_Fmt	658	599	Qualcomm QCP audio	audio/qcelp	QCP	
CSP_Codec_Fmt	659	600	Creative Signal Processor codec		CSP	
TwinVQ_Fmt	660	601	NTT TwinVQ audio format		VQF	
Interplay_MVE_Fmt	661	602	Interplay MVE video format		MVE	
IRIX_Moviemaker_Fmt	662	603	IRIX Silicon Graphics moviemaker video file	video/x-sgi-movie	MV, MOVIE	
Sega_FILM_Fmt	663	604	Sega FILM video format		CPK, CAK	
SMAF_Fmt	664	605	Synthetic music Mobile Application Format	application/vnd.smaf	MMF	
NIST_SPHERE_Fmt	665	606	NIST SPeech HEader REsources format		NIST	
Chinese_AVS_Fmt	666	607	Chinese AVS video format			
VQA_Fmt	667	608	Westwood Studios Vector Quantized Animation video file		VQA	
YAFA_Fmt	668	609	Wildfire YAFA animation		YAFA	
Origin_MVE_Fmt	669	610	Origin Wing Commander III MVE movie format		MVE	
BBC_Dirac_Fmt	670	611	BBC Dirac video format	video/x-dirac	DRC	
Maya_ASCII_Fmt	671	612	Autodesk Maya ASCII file format		MA	
RenderMan_Fmt	672	613	Pixar RenderMan Interface Bytestream		RIB	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
			file			
NOFF_Binary_Fmt	673	614	NOFF 3D Object File Format		NOFF	
VTK_ASCII_Fmt	674	615	Visualization Toolkit VTK ASCII format		VTK	
VTK_Binary_Fmt	675	616	Visualization Toolkit VTK Binary format		VTK	
Wolfram_CDF_Fmt	676	617	Wolfram Mathematica Computable Document Format	application/cdf	CDF	
Wolfram_Notebook_Fmt	677	618	Wolfram Mathematica Notebook Format		NB	
HDF4_Fmt	678	619	Hierarchical Data Format HDF4	application/x-hdf	HDF, H4	
HDF5_Fmt	679	620	Hierarchical Data Format HDF5	application/x-hdf	HDF, H5	
ARMovie_Fmt	680	621	Acorn RISC ARMovie video format		RPL	
Windows_TV_DVR_Fmt	681	622	Windows Television DVR format		WTV	
InstallShield_Z_Fmt	682	623	InstallShield Z archive format	application/x-compress	Z	
MS_DirectDraw_Surface_Fmt	683	624	Microsoft DirectDraw Surface container format		DDS	
Bink_Fmt	684	625	Bink audio-video container format		BIK, BK2	
LZMA_Fmt	685	626	LZMA compressed data format	application/x-lzma	LZMA	
True_Audio_Fmt	686	627	True Audio format	audio/x-tta	TTA	
Keepass_Fmt	687	628	Keepass Password file		KDB, KDBX	
RPM_Fmt	688	629	RPM Package Manager file	application/x-rpm	RPM	
Printer_Font_Metrics_Fmt	689	630	Adobe Printer Font Metrics format	application/x-font-printer-metric	PFM	
Adobe_Font_Metrics_Fmt	690	631	Adobe Font Metrics ASCII format	application/x-font-adobe-metric	AFM	
Printer_Font_ASCII_Fmt	691	632	Adobe Printer Font ASCII format	application/x-font-type1	PFA	
Netware_Loadable_Module_Fmt	692	633	Netware Loadable Module format		NLM	
TCPdump_pcap_Fmt	693	634	TCPdump packet stream capture savefile format	application/vnd.tcpdump.pcap	PCAP	
Multiple_Master_Font_Fmt	694	635	Adobe Multiple master font format		MMM	
TrueType_Font_Collection_Fmt	695	636	TrueType font collection format	application/x-font-ttf	TTC	
Shapefile_Spatial_Index_Fmt	696	637	Shapefile binary spatial index format	application/x-shapefile	SBX, SBN	
Java_Key_Store_Fmt	697	638	Java Key Store format	application/x-java-keystore	KS	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Java_JCE_Key_Store_Fmt	698	639	Java JCE Key Store format	application/x-java-jce-keystore		
Quark_Xpress_Intel_Fmt	699	640	QuarkXPress Intel format	application/vnd.quark.quarkxpress	QXB	
Windows_Imaging_Fmt	700	641	Microsoft Windows Imaging Format WIM		WIM	
VMware_Virtual_Disk_Fmt	701	642	VMware Virtual Disk Format 5.0	application/x-vmdk	VMDK	
XPConnect_Typelib_Fmt	702	643	XPConnect Typelib Format		XPT	
MS_DOS_Compression_Fmt	703	644	Microsoft MS-DOS installation compression (SZDD, KWAJ)	application/x-ms-compress	EX_	
DLS_Fmt	704	645	DLS Downloadable Sounds format		DLS	
MS_Windows_Registry_Fmt	705	646	Microsoft Windows Registry format			
Microsoft_Help_2_Fmt	706	647	Microsoft Help 2.0 format	application/x-ms-reader	HXD, HXW, HXH	
Qt_Translation_Fmt	707	648	Qt binary translation file format		QM	
PEM_SSL_Certificate_Fmt	708	649	PEM-encoded SSL certificate	application/pkix-cert	CRT, PEM, CER, KEY	
PostScript_Printer_Description_Fmt	709	650	Adobe PostScript Printer Description file	application/vnd.cups-ppd	PPD	
Speedo_Font_Fmt	710	651	Speedo Font format		SPD	
InstallShield_Cabinet_Fmt	711	652	InstallShield Cabinet Archive format		CAB, HDR	
InstallShield_Uninstall_Fmt	712	653	InstallShield Uninstall format		ISU	
MS_OEDBX_Folder_Fmt	713	654	Outlook Express DBX folder database format		DBX	
LabVIEW_Fmt	714	655	National Instruments LabVIEW file format		VI	
SAP_Archive_SAR_Fmt	715	656	SAP compression archive SAR format		SAR	
Netscape_Address_Book_Fmt	716	657	Netscape Address Book format		NAB	
Universal_3D_Fmt	717	658	Universal 3D file format		U3D	
Open_Inventor_ASCII_Fmt	718	659	Open Inventor ASCII format		IV	
Open_Inventor_Binary_Fmt	719	660	Open Inventor Binary format		IV	
X_Window_Dump_Fmt	720	661	X Window Dump image	image/x-xwindowdump	XWD	
Git_Packfile_Fmt	721	662	Git Packfile format		PACK	
Xara_Xar_Fmt	722	663	Xara X Xar image format	application/vnd.xara	XAR	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Internet_Archive_ARC_Fmt	723	664	Internet Archive ARC format	application/x-ia-arc	ARC	
Applix_Builder_Fmt	724	665	Applix Builder format		AB	
Applix_Bitmap_Fmt	725	666	Applix Bitmap image format		IM	
PEM_RSA_Private_Key_Fmt	726	667	PEM-encoded RSA private key		PEM	
MIFF_Fmt	727	668	Magick Image File Format		MIFF	
Subversion_Dump_Fmt	728	669	Subversion Dump format			
Virtual_Hard_Disk_Fmt	729	670	Microsoft Virtual Hard Disk format	application/x-vhd	VHD	
Direct_Access_Archive_Fmt	730	671	PowerISO Direct Access Archive format		DAA	
Debian_Binary_Fmt	731	672	Debian binary package format	application/x-debian-package	DEB	
XUL_Fastload_Fmt	732	673	Mozilla XUL Fastload format		MFL	
Nastran_OP2_Fmt	733	674	Nastran OP2 format		OP2	
Binary_Logging_Fmt	734	675	CAD Binary Logging Format		BLF	
Measurement_Data_Fmt	735	676	CAD Measurement Data Format		MDF	
Abaqus_ODB_Fmt	736	677	Abaqus ODB Format		ODB	
Open_Diagnostic_Data_Exchange_Fmt	737	678	Vector Open Diagnostic Data Exchange format		ODX	xmlsr
Vector_ASCII_Fmt	738	679	Vector CAD ASCII ASC format		ASC	
LSDYNA_State_Database_Fmt	739	680	LS-DYNA State Database format			
LSDYNA_Binary_Output_Fmt	740	681	LS-DYNA binary output (binout) format			
MS_Power_BI_Fmt	741	682	Microsoft Power BI Desktop format		PBIX	pbixsr
Tableau_Workbook_Fmt	742	683	Tableau Workbook format		TWB	xmlsr
Tableau_Packaged_Workbook_Fmt	743	684	Tableau Packaged Workbook format		TWBX	unzip
Tableau_Extract_Fmt	744	685	Tableau Extract format		TDE	
Tableau_Data_Source_Fmt	745	686	Tableau Data Source format		TDS	xmlsr
Tableau_Packaged_Data_Source_Fmt	746	687	Tableau Packaged Data Source format		TDSX	unzip
Tableau_Preferences_Fmt	747	688	Tableau Preferences format		TPS	xmlsr
Tableau_Map_Source_Fmt	748	689	Tableau Map Source format		TMS	xmlsr
ABAP_Fmt	749	690	ABAP Source Code[4]	text/x-abap	ABAP	afsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
AMPL_Fmt	750	691	AMPL Source Code[4]		AMPL	afsr
APL_Fmt	751	692	APL Source Code[4]		APL	afsr
ASN1_Fmt	752	693	ASN.1 Source Code[4]		ASN	afsr
ATS_Fmt	753	694	ATS Source Code[4]			afsr
Agda_Fmt	754	695	Agda Source Code[4]	text/x-agda	AGDA	afsr
Alloy_Fmt	755	696	Alloy Source Code[4]	text/x-alloy	ALS	afsr
Apex_Fmt	756	697	Apex Source Code[4]		CLS	afsr
Arduino_Fmt	757	698	Arduino Source Code[4]	text/x-arduino	INO	afsr
AsciiDoc_Fmt	758	699	AsciiDoc Source Code[4]	text/x-asciidoc	ASC	afsr
AspectJ_Fmt	759	700	AspectJ Source Code[4]	text/x-aspectj	AJ	afsr
Awk_Fmt	760	701	Awk Source Code[4]	text/x-awk	AWK	afsr
BlitzMax_Fmt	761	702	BlitzMax Source Code[4]	text/x-bmx	BMX	afsr
Bluespec_Fmt	762	703	Bluespec Source Code[4]		BSV	afsr
Brainfuck_Fmt	763	704	Brainfuck Source Code[4]	text/x-brainfuck	B, BF	afsr
Brightscript_Fmt	764	705	Brightscript Source Code[4]		BRS	afsr
CLIPS_Fmt	765	706	CLIPS Source Code[4]		CLP	afsr
CMake_Fmt	766	707	CMake Source Code[4]	text/x-cmake	CMAKE	afsr
COBOL_Fmt	767	708	COBOL Source Code[4]	text/x-cobol	CBL, CCP, COB, CPY	afsr
CWeb_Fmt	768	709	CWeb Source Code[4]		W	afsr
CartoCSS_Fmt	769	710	CartoCSS Source Code[4]		MSS	afsr
Ceylon_Fmt	770	711	Ceylon Source Code[4]	text/x-ceylon	CEYLON	afsr
Chapel_Fmt	771	712	Chapel Source Code[4]		CHPL	afsr
Clarion_Fmt	772	713	Clarion Source Code[4]		CLW	afsr
Clean_Fmt	773	714	Clean Source Code[4]		DCL, ICL	afsr
Component_Pascal_Fmt	774	715	Component Pascal Source Code[4]	text/x-component-pascal	CP	afsr
Cool_Fmt	775	716	Cool Source Code[4]		CL	afsr
Coq_Fmt	776	717	Coq Source Code[4]	text/x-coq	V	afsr
Creole_Fmt	777	718	Creole Source Code[4]		CREOLE	afsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Crystal_Fmt	778	719	Crystal Source Code[4]		CR	afsr
Csound_Fmt	779	720	Csound Source Code[4]		ORC	afsr
Csound_Document_Fmt	780	721	Csound Document Source Code[4]		CSD	afsr
Cuda_Fmt	781	722	Cuda Source Code[4]	text/x-cuda	CU	afsr
D_Fmt	782	723	D Source Code[4]	text/x-d	DCL, ICL	afsr
DIGITAL_Command_Language_Fmt	783	724	DIGITAL Command Language Source Code[4]		COM	afsr
DTrace_Fmt	784	725	DTrace Source Code[4]		D	afsr
Dart_Fmt	785	726	Dart Source Code[4]	text/x-dart	DART	afsr
E_Fmt	786	727	E Source Code[4]		E	afsr
ECL_Fmt	787	728	ECL Source Code[4]	application/x-ecl	ECL	afsr
Elm_Fmt	788	729	Elm Source Code[4]	text/x-elm	ELM	afsr
Emacs_Lisp_Fmt	789	730	Emacs Lisp Source Code[4]	text/x-emacs-lisp	EL	afsr
EmberScript_Fmt	790	731	EmberScript Source Code[4]		EM	afsr
Fantom_Fmt	791	732	Fantom Source Code[4]	application/x-fantom	FAN	afsr
Forth_Fmt	792	733	Forth Source Code[4]	text/x-forth	FOR, FORTH	afsr
FreeMarker_Fmt	793	734	FreeMarker Source Code[4]		FTL	afsr
Frege_Fmt	794	735	Frege Source Code[4]		FR	afsr
G_code_Fmt	795	736	G-code Source Code[4]		G	afsr
GAMS_Fmt	796	737	GAMS Source Code[4]		GMS	afsr
GAP_Fmt	797	738	GAP Source Code[4]			afsr
GDScript_Fmt	798	739	GDScript Source Code[4]		GD	afsr
GLSL_Fmt	799	740	GLSL Source Code[4]	text/x-glslsrc	GLSL	afsr
Game_Maker_Language_Fmt	800	741	Game Maker Language Source Code[4]		GML	afsr
Gnuplot_Fmt	801	742	Gnuplot Source Code[4]	text/x-gnuplot	GNU, GP	afsr
Golo_Fmt	802	743	Golo Source Code[4]		GOLO	afsr
Gosu_Fmt	803	744	Gosu Source Code[4]	text/x-gosu	GS	afsr
Gradle_Fmt	804	745	Gradle Source Code[4]		GRADLE	afsr
GraphQL_Fmt	805	746	GraphQL Source Code[4]		GRAPHQL	afsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Graphviz_DOT_Fmt	806	747	Graphviz (DOT) Source Code[4]		DOT	afsr
HLSL_Fmt	807	748	HLSL Source Code[4]		HLSL	afsr
Hack_Fmt	808	749	Hack Source Code[4]			afsr
Haml_Fmt	809	750	Haml Source Code[4]	text/x-haml	HAML	afsr
Handlebars_Fmt	810	751	Handlebars Source Code[4]		HBS	afsr
Hy_Fmt	811	752	Hy Source Code[4]	text/x-hy	HY	afsr
IDL_Fmt	812	753	IDL Source Code[4]	text/x-idl	PRO	afsr
IGOR_Pro_Fmt	813	754	IGOR Pro Source Code[4]	text/ipf	IPF	afsr
Idris_Fmt	814	755	Idris Source Code[4]	text/x-idris	IDR	afsr
Inform_7_Fmt	815	756	Inform 7 Source Code[4]		I7X	afsr
Ioke_Fmt	816	757	Ioke Source Code[4]	text/x-iokesrc	IK	afsr
Isabelle_Fmt	817	758	Isabelle Source Code[4]	text/x-isabelle		afsr
J_Fmt	818	759	J Source Code[4]	text/x-j	IJS	afsr
JSONiq_Fmt	819	760	JSONiq Source Code[4]		JQ	afsr
JSX_Fmt	820	761	JSX Source Code[4]		JSX	afsr
Jasmin_Fmt	821	762	Jasmin Source Code[4]		J	afsr
Jolie_Fmt	822	763	Jolie Source Code[4]			afsr
Julia_Fmt	823	764	Julia Source Code[4]	text/x-julia	JL	afsr
KiCad_Layout_Fmt	824	765	KiCad Layout Source Code[4]			afsr
KiCad_Schematic_Fmt	825	766	KiCad Schematic Source Code[4]		SCH	afsr
Kotlin_Fmt	826	767	Kotlin Source Code[4]		KT	afsr
LFE_Fmt	827	768	LFE Source Code[4]	text/x-kotlin	LFE	afsr
LOLCODE_Fmt	828	769	LOLCODE Source Code[4]		LOL	afsr
Lasso_Fmt	829	770	Lasso Source Code[4]	text/x-lasso	LAS, LASSO	afsr
Limbo_Fmt	830	771	Limbo Source Code[4]	text/limbo		afsr
LiveScript_Fmt	831	772	LiveScript Source Code[4]	text/x-livescript	LS	afsr
M_Fmt	832	773	M Source Code[4]		M	afsr
MAXScript_Fmt	833	774	MAXScript Source Code[4]		MS	afsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Markdown_Fmt	834	775	Markdown Source Code[4]		MD	afsr
Matlab_Fmt	835	463	Matlab Source Code[4]	text/x-matlab	M	afsr
Max_Code_Fmt	836	776	Max Source Code[4]		MXT	afsr
Mercury_Fmt	837	777	Mercury Source Code[4]			afsr
Modelica_Fmt	838	778	Modelica Source Code[4]	text/x-modelica	MO	afsr
Modula_2_Fmt	839	779	Modula-2 Source Code[4]	text/x-modula2	MOD	afsr
Monkey_Fmt	840	780	Monkey Source Code[4]	text/x-monkey	MONKEY	afsr
Moocode_Fmt	841	781	Moocode Source Code[4]	text/x-moocode	MOO	afsr
NL_Fmt	842	782	NL Source Code[4]		NL	afsr
NSIS_Fmt	843	783	NSIS Source Code[4]	text/x-nsis	NSI	afsr
NetLogo_Fmt	844	784	NetLogo Source Code[4]		NLOGO	afsr
NewLisp_Fmt	845	785	NewLisp Source Code[4]	text/x-newlisp	NL	afsr
Nginx_Fmt	846	786	Nginx Source Code[4]	text/x-nginx-conf	VHOST	afsr
Nix_Fmt	847	787	Nix Source Code[4]	text/x-nix	NIX	afsr
Nu_Fmt	848	788	Nu Source Code[4]		NU	afsr
OCaml_Fmt	849	789	OCaml Source Code[4]	text/x-ocaml		afsr
OpenCL_Fmt	850	790	OpenCL Source Code[4]		CL	afsr
OpenEdge_ABL_Fmt	851	791	OpenEdge ABL Source Code[4]	text/x-openedge		afsr
OpenSCAD_Fmt	852	792	OpenSCAD Source Code[4]		SCAD	afsr
Ox_Fmt	853	793	Ox Source Code[4]		OX	afsr
Oxygene_Fmt	854	794	Oxygene Source Code[4]		OXYGENE	afsr
Oz_Fmt	855	795	Oz Source Code[4]		OZ	afsr
PAWN_Fmt	856	796	PAWN Source Code[4]	text/x-pawn	PWN	afsr
PLpgSQL_Fmt	857	797	PLpgSQL Source Code[4]	text/x-plpgsql	PLSQL	afsr
Pan_Fmt	858	798	Pan Source Code[4]		PAN	afsr
Parrot_Assembly_Fmt	859	799	Parrot Assembly Source Code[4]		PASM	afsr
PicoLisp_Fmt	860	800	PicoLisp Source Code[4]			afsr
Pike_Fmt	861	801	Pike Source Code[4]	text/x-pike	PIKE	afsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Pony_Fmt	862	802	Pony Source Code[4]		PONY	afsr
Processing_Fmt	863	803	Processing Source Code[4]		PDE	afsr
PureBasic_Fmt	864	804	PureBasic Source Code[4]		PB	afsr
QMake_Fmt	865	805	QMake File[4]			afsr
RAML_Fmt	866	806	RAML Source Code[4]		RAML	afsr
RDoc_Fmt	867	807	RDoc Source Code[4]		RDOC	afsr
REXX_Fmt	868	808	REXX Source Code[4]	text/x-rexx	REXX	afsr
Racket_Fmt	869	809	Racket Source Code[4]	text/x-racket		afsr
Ragel_Fmt	870	810	Ragel Source Code[4]			afsr
Rascal_Fmt	871	811	Rascal Source Code[4]		RSC	afsr
Rebol_Fmt	872	812	Rebol Source Code[4]	text/x-rebol	REB, REBOL	afsr
Red_Fmt	873	813	Red Source Code[4]	text/x-red	RED	afsr
RenPy_Fmt	874	814	Ren'Py Source Code[4]		RPY	afsr
RenderScript_Fmt	875	815	RenderScript Source Code[4]		RS	afsr
Ring_Fmt	876	816	Ring Source Code[4]		RING	afsr
RobotFramework_Fmt	877	817	RobotFramework Source Code[4]	text/x-robotframework	ROBOT	afsr
SAS_Fmt	878	818	SAS Source Code[4]		SAS	afsr
SPARQL_Fmt	879	819	SPARQL format[4]	application/sparql-query		afsr
SQL_Fmt	880	820	SQL format[4]	text/x-sql		afsr
SQLPL_Fmt	881	821	SQLPL Source Code[4]			afsr
SaltStack_Fmt	882	822	SaltStack Source Code[4]		SLS	afsr
Scheme_Fmt	883	823	Scheme Source Code[4]	text/x-scheme		afsr
Scilab_Fmt	884	824	Scilab Source Code[4]	text/scilab	SCI	afsr
Squirrel_Fmt	885	825	Squirrel Source Code[4]		NUT	afsr
Stan_Fmt	886	826	Stan Source Code[4]		STAN	afsr
Stata_Fmt	887	827	Stata Source Code[4]			afsr
Stylus_Fmt	888	828	Stylus Source Code[4]		STYL	afsr
SuperCollider_Fmt	889	829	SuperCollider Source Code[4]	text/supercollider	SC	afsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
SystemVerilog_Fmt	890	830	SystemVerilog Source Code[4]	text/x-systemverilog	SV	afsr
TXL_Fmt	891	831	TXL Source Code[4]		TXL	afsr
Turing_Fmt	892	832	Turing Source Code[4]		T	afsr
Turtle_Fmt	893	833	Turtle Source Code[4]	text/turtle	TTL	afsr
UrWeb_Fmt	894	834	UrWeb Source Code[4]		UR, URS	afsr
Vim_script_Fmt	895	835	Vim script File[4]	text/x-vim	VIM	afsr
Visual_Basic_Fmt	896	836	Visual Basic Source Code[4]	text/x-vbasic	VB	afsr
WebAssembly_Fmt	897	837	WebAssembly Source Code[4]		WAT	afsr
WebIDL_Fmt	898	838	WebIDL Source Code[4]		WEBIDL	afsr
X10_Fmt	899	839	X10 Source Code[4]	text/x-x10	X10	afsr
XQuery_Fmt	900	840	XQuery Source Code[4]	text/xquery	XQM	afsr
Xojo_Fmt	901	841	Xojo Source Code[4]			afsr
Xtend_Fmt	902	842	Xtend Source Code[4]	text/x-xtend	XTEND	afsr
YANG_Fmt	903	843	YANG Source Code[4]		YANG	afsr
Zephir_Fmt	904	844	Zephir Source Code[4]		ZEP	afsr
eC_Fmt	905	845	eC Source Code[4]	text/x-ecsrc	EC	afsr
reStructuredText_Fmt	906	846	reStructuredText Source Code[4]	text/x-rst		afsr
xBase_Fmt	907	847	xBase Source Code[4]			afsr
Windows_Installer_Fmt	908	848	MSI Windows Installer format	application/x-ole-storage	MSI	olesr
Autodesk_3ds_Max_Fmt	909	849	Autodesk 3ds Max format		MAX	
PhotoDraw_Mix_Fmt	910	850	PhotoDraw MIX image	image/vnd.mix	MIX	
Softimage_SCN_Fmt	911	851	Softimage Scene SCN format		SCN	
Parasolid_XT_Fmt	912	852	Parasolid ascii XT format		X_T	
Parasolid_XB_Fmt	913	853	Parasolid binary XB format		X_B	
IGES_Fmt	914	854	Initial Graphics Exchange Specification format	model/iges	IGS	
ACE_Archive_Fmt	915	855	ACE archive format	application/x-ace-compressed	ACE	
Grasshopper_GHX_Fmt	916	856	Grasshopper GHX format		GHX	xmlsr
MS_FrontPage_Macro_Fmt	917	857	Microsoft FrontPage macro file format		FPM	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
MS_AtWork_Fax_Fmt	918	858	Microsoft AtWork Fax format		AWD	
MS_Image_Composer_Fmt	919	859	Microsoft Image Composer format		MIC	
MS_Visual_InterDev_Fmt	920	860	Microsoft Visual InterDev web project items file		WDM	
Macromedia_Flash_FLA_OLE_Fmt	921	861	Macromedia Flash FLA Project File OLE format		FLA	
Corel_Draw_X4_Fmt	922	862	CorelDRAW version X4 onwards	application/x-vnd.corel.zcf.draw.document+zip	CDRX	
Ogg_Daala_Fmt	923	863	Ogg Daala video format	video/daala	OGV	
Ogg_BBC_Dirac_Fmt	924	864	Ogg BBC Dirac video format	video/x-dirac	OGV	
PKCS_7_Fmt	925	865	PKCS #7 cryptographic format	application/pkcs7-signature	P7S	
Time_Stamped_Data_Fmt	926	866	Time-stamped data format	application/timestamped-data	TSD	
Sereal_Fmt	927	867	Sereal data serialization format	application/sereal	SRL	
Associated_Signature_Simple_Fmt	928	868	Associated Signature Container Simple format	application/vnd.etsi.asic-s+zip	ASICS	
Associated_Signature_Extended_Fmt	929	869	Associated Signature Container Extended format	application/vnd.etsi.asic-e+zip	ASICE	
iBooks_Fmt	930	870	Apple iBooks format	application/x-ibooks+zip	IBOOKS	
PDF_Forms_Data_Fmt	931	871	PDF Forms Data Format	application/vnd.fdf	FDF	
PDF_XML_Forms_Data_Fmt	932	872	PDF XML Forms Data Format	application/vnd.adobe.xfdf	XFDF	xmlsr
AxCrypt_Fmt	933	873	AxCrypt encrypted document	application/x-axcrypt	AXX	
Unix_Archive_Fmt	934	874	Unix Archive ar format	application/x-archive	AR	
Berkeley_Btree_Database_Fmt	935	875	Berkeley DB btree database format	application/x-berkeley-db	DB	
Berkeley_Hash_Database_Fmt	936	876	Berkeley DB hash database format	application/x-berkeley-db	DB	
Berkeley_Log_Database_Fmt	937	877	Berkeley DB log database format	application/x-berkeley-db		
Berkeley_Queue_Database_Fmt	938	878	Berkeley DB queue database format	application/x-berkeley-db		
BitTorrent_Fmt	939	879	BitTorrent file format	application/x-bittorrent	TORRENT	
Chrome_Extension_Fmt	940	880	Google Chrome Extension format	application/x-chrome-package	CRX	
Dalvik_Executable_Fmt	941	881	Dalvik Executable dex format	application/x-dex	DEX	
Foxmail_Fmt	942	882	Foxmail email format	application/x-foxmail	BOX	
GRIB_Fmt	943	883	General Regularly-distributed	application/x-grib	GRB, GRIB2	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
			Information in Binary form GRIB format			
Zstandard_Fmt	944	884	Zstandard compression format	application/zstd	ZSTD	
LZ4_Fmt	945	885	LZ4 compressed file	application/x-lz4	LZ4	
MS_Money_Fmt	946	886	Microsoft Money format	application/x-msmoney	MNY	
NetCDF_Fmt	947	887	Network Common Data Form NetCDF format	application/x-netcdf	NC	
SAS6_Data_Fmt	948	888	SAS 6 Data storage format	application/x-sas-data-v6	SD2	
SAS_Transport_Fmt	949	889	SAS Transport File XPORT format	application/x-sas-xport	XPT, XPORT	
Snappy_Framed_Fmt	950	890	Snappy Framed compression format	application/x-snappy-framed	SZ	
Stata_Data_Fmt	951	891	Stata Data Format	application/x-stata-dta	DTA	
SPSS_SAV_Fmt	952	892	SPSS Statistics Data File Format		SAV	
Zoo_Archive_Fmt	953	893	Zoo Compressed Archive Format	application/x-zoo	ZOO	
CDX_Fmt	954	894	ChemDraw CDX format	chemical/x-cdx	CDX	
CDXML_Fmt	955	895	ChemDraw CDXML format	application/vnd.chemdraw+xml	CDXML	xmlsr
BPG_Fmt	956	896	Better Portable Graphics BPG format	image/x-bpg	BPG	
Apple_Icon_Fmt	957	897	Apple Icon image format	image/icns	ICNS	
NITF_Fmt	958	898	National Imagery Transmission Format NITF image	image/nitf	NTF, NITF	
ERDAS_Imagine_Fmt	959	899	ERDAS Imagine image format	application/x-erdas-hfa	HFA, RRD, AUX	
MS_Office_Temporary_Owner_Fmt	960	900	Microsoft Office temporary owner file	application/x-ms-owner		
EAC3_Audio_Fmt	961	901	Enhanced-AC3 (EAC3) Audio File format	audio/eac3	AC3	
COFF_Relocatable_Fmt	962	902	Common Object File Format (COFF) relocatable object	application/x-object-file	O	
COFF_Executable_Fmt	963	903	Common Object File Format (COFF) executable	application/x-executable-file		
COFF_Dynamic_Lib_Fmt	964	904	Common Object File Format (COFF) dynamic library	application/x-library-file		
ELF_Core_Fmt	965	905	ELF Core file	application/x-coredump		
Purify_Fmt	966	906	Rational Purify data file		PFY	
Kryptel_Fmt	967	907	Kryptel encrypted file		EDC	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Windows_Core_Dump_Fmt	968	908	Windows heap or mini core dump file	application/x-dmp	DMP	
Qt_Prerendered_Font_Fmt	969	909	Qt Prerendered Font format		QPF2	
AIX_Relocatable_Fmt	970	910	AIX/RISC COFF relocatable object	application/x-object-file		
AIX_Executable_Fmt	971	911	AIX/RISC COFF executable	application/x-executable-file		
AIX_Dynamic_Lib_Fmt	972	912	AIX/RISC COFF dynamic library	application/x-library-file	A	
HPUX_Relocatable_Fmt	973	913	HPUX/PA-RISC COFF relocatable object	application/x-object-file		
HPUX_Executable_Fmt	974	914	HPUX/PA-RISC COFF executable	application/x-executable-file		
HPUX_Dynamic_Lib_Fmt	975	915	HPUX/PA-RISC COFF dynamic library	application/x-library-file	SL	
XML_EBCDIC_Fmt	976	916	EBCDIC-encoded XML file	application/xml	XML	
MPEG_JVT_H264_Fmt	977	917	MPEG JVT-NAL sequence H264 video	video/h264	264	
Material_Exchange_Fmt	978	918	Material Exchange Format audio-video container format	application/mxf	MXF	
MS_Agent_Character_Fmt	979	919	Microsoft Agent Character file		ACS	
Quicken_Fmt	980	920	Quicken data file		QDF	
MS_Outlook_Address_Fmt	981	921	Microsoft Outlook address file		WAB	
MS_Answer_Wizard_Fmt	982	922	Microsoft Answer Wizard file			
ADX_Fmt	983	923	ADX audio file		ADX	
System_Deployment_Image_Fmt	984	924	Microsoft System Deployment Image SDI format		SDI	
Free_Lossless_Image_Fmt	985	925	Free Lossless Image Format (FLIF)	image/flif	FLIF	
DPX_Fmt	986	926	Digital Picture Exchange (DPX) image format	image/dpx	DPX	
Avro_Fmt	987	927	Apache Avro binary format		AVRO	
InstallShield_Archive_Fmt	988	928	InstallShield archive (early versions) format		EX_	
Mac_Executable_Fmt	989	929	Mac OS-X (Mach-O) executable format			
GDSII_Fmt	990	930	GDSII data format		GDS, GDS2	gdsiisr
ActiveMime_Fmt	991	931	Microsoft ActiveMime (mso) documents	application/x-mso	MSO	
SmartCharts_Fmt	992	932	BizInt SmartCharts data format		CHP, CHRR	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Webex_ARF_Fmt	993	933	Webex advanced network ARF recordings		ARF	
Webex_WRF_Fmt	994	934	Webex local WRF recordings		WRF	
PGP_NetShare_Fmt	995	935	Symantec PGP NetShare encrypted file			
Ability_WP_OLE_Fmt	996	936	Ability Write later versions format		AWW	
Ability_SS_OLE_Fmt	997	937	Ability Spreadsheet later versions format		AWS	
InDesign_IDML_Fmt	998	938	Adobe InDesign IDML format	application/vnd.adobe.indesign-idml-package	IDML	
Executable_JAR_Fmt	999	939	Executable Java Archive (jar) file	application/java-archive	JAR	unzip
IDOL_IDX_Fmt	1000	940	IDOL Server IDX file		IDX	
Android_Package_Kit_Fmt	1001	941	Android Package Kit (APK) format	application/vnd.android.package-archive	APK	
Android_Binary_XML_Fmt	1002	942	Android Binary XML (compressed by aapt) format	application/xml	XML	
Java_WAR_Fmt	1003	943	Java WAR file format		WAR	
Java_EAR_Fmt	1004	944	Java EAR file format		EAR	
Atom_Syndication_Fmt	1005	945	Atom Syndication Format	application/atom+xml	ATOM	xmlsr
RSS_Fmt	1006	946	RSS syndication XML format	application/rss+xml	RSS	xmlsr
SMIL_Fmt	1007	947	Synchonized Multimedia Integration Language (SMIL) XML format	application/smil+xml	SMIL	xmlsr
XSLT_Fmt	1008	948	Extensible Stylesheet Language Transformations (XSLT) format	application/xslt+xml	XSL, XSLT	xmlsr
XML_Shareable_Playlist_Fmt	1009	949	XML Shareable Playlist Format (XSPF)	application/xspf+xml	XSPF	xmlsr
FictionBook_Fmt	1010	950	FictionBook e-book XML format	application/x-fictionbook+xml	FB2	xmlsr
Adobe_Premiere_Project_Fmt	1011	951	Adobe Premiere project format	image/vnd.adobe.premiere	PPJ	
RDF_XML_Fmt	1012	952	RDF/XML format	application/rdf+xml	RDF	xmlsr
Really_Simple_Discovery_Fmt	1013	953	Really Simple Discovery (RSD) XML format	application/rsd+xml	RSD	xmlsr
SBML_Fmt	1014	954	Systems Biology Markul Language (SBML) XML format	application/sbml+xml	SBML	xmlsr
SRU_Fmt	1015	955	Search/Retrieve via URL (SRU) XML format	application/sru+xml	SRU	xmlsr
SSML_Fmt	1016	956	Speech Synthesis Markup Language	application/ssml+xml	SSML	xmlsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
			(SSML) XML format			
PLS_Fmt	1017	957	Pronunciation Lexicon Specification (PLS) XML format	application/pls+xml	PLS	xmlsr
TEI_Fmt	1018	958	Text Encoding Initiative (TEI) XML format	application/tei+xml	TEI	xmlsr
METS_Fmt	1019	959	Metadata Encoding and Transmission Standard (METS) XML format	application/mets+xml	METS	xmlsr
MODS_Fmt	1020	960	Metadata Object Description Schema (MODS) XML format	application/mods+xml	MODS	xmlsr
Metalink_Fmt	1021	961	Metalink XML format	application/metalink4+xml	METALINK	xmlsr
Open_eBook_Fmt	1022	962	Open eBook (OEBPS) XML format	application/oebps-package+xml	OPF	xmlsr
SRGS_Fmt	1023	963	Speech Recognition Grammar Specification (SRGS) XML format	application/srgs+xml	SRGS	xmlsr
SPARQL_Results_Fmt	1024	964	SPARQL Query Results XML format	application/sparql-results+xml	SRX	xmlsr
Adobe_XML_Data_Package_Fmt	1025	965	Adobe XML Data Package format	application/vnd.adobe.xdp+xml	XDP	xmlsr
ESzigno_Fmt	1026	966	e-Szigno signed xml document	application/vnd.eszigno3+xml	ES3	xmlsr
Mozilla_XUL_Fmt	1027	967	Mozilla XML User Interface Language (XUL) XML format	application/vnd.mozilla.xul+xml	XUL	xmlsr
SyncML_Fmt	1028	968	Synchronization Markup Language (SyncML) XML format	application/vnd.syncml+xml	XML	xmlsr
VoiceXML_Fmt	1029	969	VoiceXML (VXML) XML format	application/voicexml+xml	VXML	xmlsr
TI_Target_Configuration_Fmt	1030	970	Texas Instruments CCXML target configuration XML format		CCXML	
LZFSE_Fmt	1031	971	Lempel-Ziv Finite State Entropy (LZFSE) compression format		LZFSE	
Kindle_eBook_Fmt	1032	972	Amazon Kindle or Mobipocket eBook format	application/vnd.amazon.ebook	AZW, PRC	
Oasis_Stream_Fmt	1033	973	Open Artwork System Interchange Standard (OASIS) format		OAS	
Amazon_KFX_Fmt	1034	974	Amazon KFX eBook format		KFX	
KTX_Fmt	1035	975	KTX image format	image/ktx	KTX	
GMSH_Mesh_Fmt	1036	976	GMSH Mesh polygon format	model/mesh	MSH	
Collada_DAE_Fmt	1037	977	Collada Digital Asset Exchange (DAE)	model/vnd.collada+xml	DAE	xmlsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
			format			
YIN_Fmt	1038	978	YIN XML format	application/yin+xml	YIN	xmlsr
MPEG_Playlist_Fmt	1039	979	MPEG audio playlist format	audio/mpegurl	M3U	
Windows_Audio_Playlist_Fmt	1040	980	Windows Audio playlist format	audio/x-ms-wax	WAX	xmlsr
DTS_Audio_Fmt	1041	981	DTS Coherent Acoustics audio format	audio/vnd.dts	DTS	
Chemical_Markup_Language_Fmt	1042	982	Chemical Markup Language (CML) XML format	chemical/x-cml	CML	xmlsr
CrystalMaker_Fmt	1043	983	CrystalMaker chemical format	chemical/x-cmdf	CMDF	
VTK_XML_Fmt	1044	984	Visualization Toolkit VTK XML format	model/vnd.vtu	VTU	xmlsr
IPFIX_Fmt	1045	985	IP Flow Information Export (IPFIX) format	application/ipfix	IPFIX	
Portable_Font_Resource_Fmt	1046	986	Portable Font Resource font format	application/font-tdpfr	PFR	
MARC_Fmt	1047	987	Machine-Readable Cataloging (MARC21) format	application/marc	MARC	
MARC_XML_Fmt	1048	988	Machine-Readable Cataloging (MARC) XML format	application/marcxml+xml	XML	xmlsr
XAR_Fmt	1049	989	Extensible Archive (XAR) format			
Symbian_Installer_Fmt	1050	990	Symbian installer format	application/vnd.symbian.install	SIS	
SO_Drawing_XML_Fmt	1051	316	OpenDocument format (OpenOffice 1/StarOffice 6.7) Drawing XML	application/vnd.sun.xml.draw	SXD	kpodfrdr
SO_Text_Global_XML_Fmt	1052	991	OpenDocument format (OpenOffice 1/StarOffice 6.7) Writer Master document XML	application/vnd.sun.xml.writer.global	SXG	
ODF_Chart_Fmt	1053	992	ODF Chart	application/vnd.oasis.opendocument.chart	ODC	
ODF_Database_Fmt	1054	993	ODF Database	application/vnd.sun.xml.base	ODB	
ODF_Image_Fmt	1055	994	ODF Image	application/vnd.oasis.opendocument.image	ODI	
ODF_Text_Master_Fmt	1056	995	ODF Text Master	application/vnd.oasis.opendocument.text-master	ODM	
ODF_Text_Web_Fmt	1057	996	ODF Text Web	application/vnd.oasis.opendocument.text-web	OTH	
ODF_Chart_Template_Fmt	1058	997	ODF Chart Template	application/vnd.oasis.opendocument.chart-template	OTC	
ODF_Formula_Template_Fmt	1059	998	ODF Formula Template	application/vnd.oasis.opendocument.formula-template	OTF	unzip
ODF_Drawing_Template_Fmt	1060	316	ODF Drawing/Graphics Template	application/vnd.oasis.opendocument.graphics-	OTG	kpodfrdr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
				template		
ODF_Image_Template_Fmt	1061	999	ODF Image Template	application/vnd.oasis.opendocument.image-template	OTI	
ODF_Presentation_Template_Fmt	1062	316	ODF Presentation Template	application/vnd.oasis.opendocument.presentation-template	OTP	kpodfrdr
ODF_Spreadsheet_Template_Fmt	1063	315	ODF Spreadsheet Template	application/vnd.oasis.opendocument.spreadsheet-template	OTS	odfsssr
ODF_Text_Template_Fmt	1064	314	ODF Text Template	application/vnd.oasis.opendocument.text-template	OTT	odfwpsr
ODF_Chart_XML_Fmt	1065	1000	ODF Chart flat XML format	application/vnd.oasis.opendocument.chart.xml	FODC	
ODF_Drawing_XML_Fmt	1066	1001	ODF Drawing/Graphics flat XML format	application/vnd.oasis.opendocument.formula.xml	FODG	
ODF_Formula_XML_Fmt	1067	1002	ODF Formula flat XML format	application/vnd.oasis.opendocument.graphics.xml	FODF	
ODF_Image_XML_Fmt	1068	1003	ODF Image flat XML format	application/vnd.oasis.opendocument.image.xml	FODI	
ODF_Presentation_XML_Fmt	1069	1004	ODF Presentation flat XML format	application/vnd.oasis.opendocument.presentation.xml	FODP	
ODF_Spreadsheet_XML_Fmt	1070	1005	ODF Spreadsheet flat XML format	application/vnd.oasis.opendocument.spreadsheet.xml	FODS	
ODF_Text_XML_Fmt	1071	1006	ODF Text flat XML format	application/vnd.oasis.opendocument.text.xml	FODT	
ODF_Extension_Fmt	1072	1007	ODF Extension format	application/vnd.openofficeorg.extension	OXT	
StarView_Metafile_Fmt	1073	1008	OpenOffice StarView MetaFile format	image/x-svm	SVM	
BBeB_LRF_eBook_Fmt	1074	1009	Broad Band eBook (BBeB) in LRF format	application/x-ext-lrf	LRF	
GPG_Trust_DB_Fmt	1075	1010	GPG trust database format		GPG	
VICE_Emulator_Fmt	1076	1011	VICE (Versatile Commodore Emulator) format		VSF	
Portable_Game_Notation_Fmt	1077	1012	Portable Game Notation chess format	application/vnd.chess-pgn	PGN	
Doom_WAD_Fmt	1078	1013	Doom IWAD/PWAD format	application/x-doom	WAD	
Device_Tree_Blob_Fmt	1079	1014	Linux Device Tree Blob format		DTB	
BDF_Font_Fmt	1080	1015	Glyph Bitmap Distribution Format	application/x-font-bdf	BDF	
PC_Screen_Font_Fmt	1081	1016	PC Screen Font format	application/x-font-psf	PSF	
JNLP_Fmt	1082	1017	Java Network Launching Protocol	application/x-java-jnlp-file	JNLP	xmlsr
XAML_Browser_Application_Fmt	1083	1018	XAML Browser Application (XBAP) format	application/x-ms-xbap	XBAP	xmlsr
MS_Binder_Fmt	1084	1019	Microsoft Office Binder format	application/x-msbinder	OBP	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
XAP_Fmt	1085	1020	Microsoft Silverlight application (XAP) format	application/x-silverlight-app	XAP	
StuffIt_X_Fmt	1086	1021	StuffIt X (SITX) archive format	application/x-stuffitx	SITX	
FIG_Fmt	1087	1022	Facility for Interactive Generation of figures (FIG) image format	application/x-xfig	FIG	
XPInstall_Fmt	1088	1023	XPInstall Cross-Platform Installer Module (XPI) format	application/x-xpinstall	XPI	
XDF_Fmt	1089	1024	Extensible Data Format (XDF) XML format		XDF	xmlsr
MXML_Fmt	1090	1025	MXML UI markup language XML format		MXML	xmlsr
MusicXML_Fmt	1091	1026	MusicXML format	application/vnd.recordare.musicxml	MXL	xmlsr
Finale_Fmt	1092	1027	Finale audio format		MUS	
Spotfire_DXP_Fmt	1093	1028	TIBCO Spotfire DXP data format	application/vnd.spotfire.dxp	DXP	
MS_Office_Theme_2007_Fmt	1094	1029	Microsoft Office theme format	application/vnd.ms-officetheme	THMX	
Adobe_AIR_Installer_Fmt	1095	1030	Adobe AIR application installer package	application/vnd.adobe.air-application-installer-package+zip	AIR	
Flex_Project_Fmt	1096	1031	Adobe Flash Flex project file format	application/vnd.adobe.fxp	FXP	
FoxPro_Fmt	1097	1032	FoxPro compiled source format		FXP	
VST_Preset_Fmt	1098	1033	Virtual Studio Technology (VST) preset format		FXP	
Mischief_Image_Fmt	1099	1034	Mischief vector graphics image format		ART	
FreeArc_Fmt	1100	1035	FreeArc archive format	application/x-freearc	ARC	
Autodesk_3ds_Fmt	1101	1036	Autodesk 3ds format	application/x-3ds	3DS	
Monkeys_Audio_Fmt	1102	1037	Monkey’s Audio format		APE	
CALS_Fmt	1103	1038	CALS raster image format		CAL	
Dr_Halo_PAL_Fmt	1104	1039	Dr Halo raster image PAL file format		PAL	
DPG_Fmt	1105	1040	Nintendo DS DPG video format		DPG	
JPEG_XR_Fmt	1106	1041	JPEG XR (extended range) image format	image/vnd.ms-photo	JXR, HDP	
TCR_eBook_Fmt	1107	1042	TCR/ZVR (Text Compression for Reader) eBook format		TCR, ZVR	
IHEX_Fmt	1108	1043	Intel Hex format		IHEX	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
QCOW_Fmt	1109	1044	QEMU Copy On Write		QCOW	
VDI_Fmt	1110	1045	VirtualBox Disk Image		VDI	
OneNote_Alternate_Fmt	1111	1046	OneNote Alternative Packaging Format			onealtsr
RMS_Protected_Fmt	1112	1047	Rights Management Services (RMS)-protected format		PFILE, PPDF, PJPG, PTXT	pfilesr
Portfolio_PDF_Fmt	1113	1048	Portfolio PDF File	application/pdf	PDF	pdfsr
Crystal_Reports_Fmt	1114	1049	SAP Crystal Reports format	application/x-rpt	RPT	
Thumbs_db_Fmt	1115	1050	Microsoft Windows thumbs.db format		DB	
PagePlus_Fmt	1116	1051	Serif PagePlus format		PPP	
MS_Project_Exchange_Fmt	1117	1052	Microsoft Project Exchange format		MPX	
MS_Management_Pack_MPX_Fmt	1118	1053	Microsoft Systems Center Operation Manager (SCOM) management pack MPX format		MPX	xmlsr
AutoCAD_VBA_Project_Fmt	1119	1054	AutoCAD VBA project format		DVB	
PLY_ASCII_Fmt	1120	1055	Polygon File Format (PLY) ASCII format		PLY	
PLY_Binary_Fmt	1121	1056	Polygon File Format (PLY) binary format		PLY	
JavaView_JVX_Fmt	1122	1057	JavaView XML (JVX) format		JVX	xmlsr
X3D_Fmt	1123	1058	Extensible 3d Graphics (X3D) XML format	model/x3d+xml	X3D	
ZBrush_Project_Fmt	1124	1059	ZBrush ZProject (ZPR) format		ZPR	
ZBrush_Tool_Fmt	1125	1060	ZBrush ZTtool (ZTL) format		ZTL	
Windows_Installer_Patch_Fmt	1126	1061	Microsoft Windows Installer Patch Package (MSP) format		MSP	
Windows_Installer_Transform_Fmt	1127	1062	Microsoft Windows Installer Transform (MST) format		MST	
Lotus_Approach_Fmt	1128	1063	Lotus Approach format	application/vnd.lotus-approach	APR, MPR	
Outlook_SendRcv_Settings_Fmt	1129	1064	Microsoft Outlook 2002 Send-Receive Settings		SRS	
MS_Publisher_Scheme_Fmt	1130	1065	Microsoft Publisher colour scheme		SCM	
SO_Chart_Fmt	1131	1066	Star Office 4,5 Chart	application/vnd.stardivision.chart	SDS	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
SO_Database_Fmt	1132	1067	Star Office 4,5 Database	application/vnd.stardivision.base	SDB	
SO_Library_Fmt	1133	1068	Star Office 4,5 Library		SBL	
PageMaker_Document_Fmt	1134	1069	Adobe PageMaker document	application/pagemaker	PMD	
MS_DTS_Fmt	1135	1070	Microsoft Data Transformation Services (DTS) package file		DTS	
Cognos_PowerPlay_PPR_Fmt	1136	1071	Cognos PowerPlay up to version 7 (PPR) format		PPR	
Visual_Studio_SUO_Fmt	1137	1072	Microsoft Visual Studio solution user options (suo) file		SUO	
MS_GraphEdit_Fmt	1138	1073	Microsoft GraphEdit File format		GRF	
ArcGIS_Graph_Fmt	1139	1074	ArcGIS Graph format		GRF	
SID_Audio_Fmt	1140	1075	SID Audio format	audio/prs.sid	SID	
MrSID_Fmt	1141	1076	LizardTech MrSID image format	image/x-mrsid	SID	
Cardfile_Fmt	1142	1077	Microsoft Windows Cardfile address book format	application/x-mscardfile	CRD	
MS_Word_Mac_4_Fmt	1143	205	Microsoft Word for Macintosh (version 4,5)	application/msword	DOC	mbsr
WordPerfect_5_Fmt	1144	80	WordPerfect (version 5)	application/x-corel-wordperfect	WOP, DOC	wosr
WordPerfect_6_Fmt	1145	178	Corel WordPerfect (version 6 and higher)	application/x-corel-wordperfect	WPD	wp6sr
WordPerfect_Graphics_1_Fmt	1146	85	WordPerfect Graphics (version 1)	application/vnd.wordperfect	WPG, QPG	
Organization_Chart_Fmt	1147	1078	OrgPlus Organization Chart	application/orgplus	OPX	
Lotus_Organizer_Fmt	1148	1079	Lotus Organizer documents	application/vnd.lotus-organizer	OR2, OR3, OR4, OR5, OR6	
MS_DBML_Fmt	1149	1080	Microsoft Database Markup Language XML document		DBML	
XMind_Fmt	1150	1081	XMind document	application/xmind	XMIND	
MSI_Cerius_Fmt	1151	1082	MSI Cerius chemical formula document	chemical/x-cerius	MSI	
GenBank_Fmt	1152	1083	GenBank DNA character sequence document	chemical/x-genbank	GB	
GIS_World_File_Fmt	1153	1084	ESRI GIS World file		BPW, GFW, JGW, J2W, PGW, SDW, TFW, WLD	afsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
GIS_Projection_Metadata_Fmt	1154	1085	ESRI Projection Metadata (PRJ) file		PRJ	
PowerWorld_Binary_Fmt	1155	1086	PowerWorld Binary (PWB) file		PWB	
PowerWorld_Display_Fmt	1156	1087	PowerWorld Display (PWD) file		PWD	
ArcXML_Fmt	1157	1088	ESRI ArcIMS project XML file (ArcXML)		AXL	
GAMS_GDX_Fmt	1158	1089	General Algebraic Modeling System (GAMS) Data Exchange (GDX) format		GDX	
ArcMap_MXD_Fmt	1159	1090	ArcMap Map Exchange Document project (MXD)		MXD	
RRDtool_Fmt	1160	1091	RRDtool (Round Robin Database) data file		RRD	
HWPX_Fmt	1161	1092	Hangul HWPX document	application/hwp+zip	HWPX	
SolidWorks_2015_Fmt	1162	1093	SolidWorks (2015 onwards) file		SLDPRT, SLDDRW, SLDASM	
MS_Photo_Editor_Fmt	1163	1094	Microsoft Photo Editor ‘embedded GIF’ file	application/vnd.ms-photo-editor		
MS_Word_HTML_Fmt	1164	1095	Microsoft Word HTML format		DOC, HTM	
MS_Excel_HTML_Fmt	1165	1096	Microsoft Excel HTML format		XLS, HTM	
Portable_FloatMap_Fmt	1166	1097	Portable FloatMap (PFM) image	image/x-portable-floatmap	PFM	
RGBE_Fmt	1167	1098	Radiance RGBE (HDR) image	image/vnd.radiance	HDR, PIC, RGBE, XYZE	
APNG_Fmt	1168	1099	Animated Portable Network Graphics (Animated-PNG)	image/apng	APNG, PNG	kppngrdr
Enhanced_Compressed_Wavelet_Fmt	1169	1100	Enhanced Compressed Wavelet image	image/ecw	ECW	
Ensoniq_Waveset_Fmt	1170	1101	Ensoniq Waveset audio data file		ECW	
Corel_Photo_Paint_Fmt	1171	1102	Corel Photo Paint (version 7 and higher)	image/x-corelphotopaint	CPT	
OpenRaster_Fmt	1172	1103	OpenRaster image	image/openraster	ORA	
Krita_Fmt	1173	1104	Krita image	application/x-krita	KRA	
Gerber_Fmt	1174	1105	Gerber image format	application/vnd.gerber	GBR	
PGML_Fmt	1175	1106	Precision Graphics Markup Language		PGML	
Away3D_Fmt	1176	1107	Away3D scene file		AWD	
CAD_3MF_Fmt	1177	1108	3D Manufacturing Format document	application/vnd.ms-package.3dmanufacturing-	3MF	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
				3dmodel+xml		
AMF_Fmt	1178	1109	Additive manufacturing file format (AMF) document	application/x-amf	AMF	xmlsr
C3D_Fmt	1179	1110	Coordinate 3D (C3D) format		C3D	
CAD_3DSystems_BFF_Fmt	1180	1111	3D Sprint (3D Systems) SLA Build file		BFF	
NRRD_Fmt	1181	1112	NRRD (nearly raw raster data) image format		NRRD	
Cinema_4D_Fmt	1182	1113	Cinema 4D model		C4D	
FBX_ASCII_Fmt	1183	1114	Kaydara FBX project (ASCII)		FBX	
FBX_Binary_Fmt	1184	1115	Kaydara FBX project (binary)		FBX	
Wavefront_OBJ_Fmt	1185	1116	Wavefront OBJ geometry definition file		OBJ	
Wavefront_MTL_Fmt	1186	1117	Wavefront Material Template Library (MTL)		MTL	
MS_Power_BI_Template_Fmt	1187	1118	Microsoft Power BI Desktop template format		PBIT	
Windows_Sticky_Notes_Fmt	1188	1119	Microsoft Windows Sticky Notes format		SNT	
BlakHole_Fmt	1189	1120	BlakHole compression format		BH	
PowerArchiver_Fmt	1190	1121	PowerArchiver PA compression format		PA	
PageMagic_Fmt	1191	1122	NEBS PageMagic format		DTP	
PIM_Archiver_Fmt	1192	1123	PIM Archiver format		PIM	
Softdisk_Text_Compressor_Fmt	1193	1124	Softdisk Text Compressor format		CTX	
Ability_PhotoPaint_Fmt	1194	1125	Ability Office PhotoPaint image		APX	
Softlib_Fmt	1195	1126	Softdisk Softlib compression format		SLB	
Timeworks_Publisher_Fmt	1196	1127	Timeworks Publisher (Publish It) format		DTP	
Scribe_Fmt	1197	1128	Scribe markup language and word processing system		MSS	afsr
SQLite_Write_Ahead_Log_Fmt	1198	1129	SQLite Write-Ahead Log file		WAL	
SQLite_WAL_Index_Fmt	1199	1130	SQLite WAL-index (shm) file		SHM	
AutoForm_Design_Fmt	1200	1131	AutoForm Design file		AFD	
TSV_Fmt	1201	1132	Tab-separated values (TSV) file	text/tab-separated-values	TSV, TAB	afsr, afsr

Format Name	Number	Category	Description	MIME Type	Extension	Readers
OpenStreetMap_XML_Fmt	1202	1133	OpenStreetMap XML data		OSM	
OpenStreetMap_PBF_Fmt	1203	1134	OpenStreetMap Protocolbuffer Binary Format data file (.osm.pbf)		PBF	
Nero_Audio_Compilation_Fmt	1204	1135	Nero Audio-CD compilation file		NRA	
Nero_ISO_Compilation_Fmt	1205	1136	Nero ISO compilation file		NRI	
WordStar_for_Windows_Fmt	1206	1137	WordStar for Windows file		WSD	stringssr
MS_Outlook_PAB_Fmt	1207	1138	Microsoft Outlook Personal Address Book (PAB)		PAB	
HLSL_FXO_Fmt	1208	1139	DirectX High-Level Shader Language (HLSL) pre-compiled shader		FXO	
HLSL_CSO_Fmt	1209	1140	DirectX High-Level Shader Language (HLSL) compiled shader object		CSO	
Oberon_Document_Fmt	1210	1141	Component Pascal / Oberon Document file		ODC	
Oberon_Symbol_Fmt	1211	1142	Component Pascal / Oberon Symbol file		OSF	
Oberon_Code_Fmt	1212	1143	Component Pascal / Oberon Code (executable and loadable object) file		OCF	
Python_Bytecode_Fmt	1213	1144	Python compiled bytecode	application/x-bytecode.python	PYC	
PCPaint_Fmt	1214	1145	PCPaint / Pictor Paint image format		PIC	
PCRaster_Map_Fmt	1215	1146	PCRaster Map / Cross System Format geographical data		MAP, CSF	
COM_Type_Library_Fmt	1216	1147	Microsoft Component Object Model (COM) Type library		TLB	
MS_Visual_C_Export_Fmt	1217	1148	Microsoft Visual C++ Export file		EXP	
Lotus_Organizer_Report_Fmt	1218	1149	Lotus Organizer report document		REP	
Audible_Audiobook_AA_Fmt	1219	1150	Audible Audiobook (AA) file	audio/audible	AA	
DOS_RED_Fmt	1220	1151	MS-DOS RED installer library format		RED	
CA_ZIPXP_Fmt	1221	1152	CA Technologies ZIPXP compressed document		CAZ	
Kindle_Topaz_Fmt	1222	1153	Amazon Kindle Topaz eBook		AZW, AZW1, TPZ	
Windows_Shim_Database_Fmt	1223	1154	Microsoft Windows Shim Database file		SDB	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
MS_Incremental_Linker_Fmt	1224	1155	Microsoft Visual Studio incremental linker file		ILK	
Lotus_Smart_Icon_Fmt	1225	1156	Lotus Smart Icon image file		SMI	
Lotus_Organizer_Layout_Fmt	1226	1157	Lotus Organizer print/paper layout file		PLT	
CMZ_Fmt	1227	1158	CMZ compression format		CMZ	
RFFlow_Fmt	1228	1159	RFFlow flowchart document		FLO	
InstallShield_Script_Fmt	1229	1160	InstallShield script document		INS	
InstallShield_Rules_Fmt	1230	1161	InstallShield Compiled Rules file		INX	
Windows_FTS_Fmt	1231	1162	Microsoft Windows 95/NT help full-text-search file		FTS	
DVD_Info_Fmt	1232	1163	DVD Information (IFO) file	content/dvd	IFO	
Emacs_Lisp_Bytecode_Fmt	1233	1164	Byte-compiled Lisp (Emacs/XEmacs)	application/x-bytecode.elisp	ELC	
Windows_Resource_Fmt	1234	1165	Microsoft Windows binary resource file		RES	
MS_Precompiled_Header_Fmt	1235	1166	Microsoft Visual C/C++ binary pre-compiled header		PCH	
Borland_Turbo_Project_Fmt	1236	1167	Borland Turbo C project file		PRJ	
PS_Font_Descriptor_Fmt	1237	1168	PostScript binary Font Descriptor file		NTF	
MySQL_Index_Fmt	1238	1169	MySQL MyISAM Table index		MYI	
MS_SQL_Fmt	1239	1170	Microsoft SQL Server primary database file		MDF	
DNL_eBook_Fmt	1240	1171	DNAML DNL eBook		DNL	
GD_Image_Fmt	1241	1172	GD Library image		GD, GD2	
ITunes_Library_Fmt	1242	1173	Apple iTunes music library		ITL	
MS_SQM_Fmt	1243	1174	Microsoft Windows Live Messenger/Mail log file		SQM	
VIFF_Fmt	1244	1175	Khoros Visualization Image File Format (VIFF)	image/x-viff	XV, VIF, VIFF	
JBIG_Fmt	1245	1176	JBIG (JBIG1) image	image/jbig	JBG, JBIG, BIE	
CodeWarrior_Project_Fmt	1246	1177	CodeWarrior C/C++ project		MCP	
PaintShop_Pro_JBF_Fmt	1247	1178	PaintShop Pro JBF image cache file	image/jbf	JBF	
Delphi_Diagram_Portfolio_Fmt	1248	1179	Delphi Diagram Portfolio file		DDP	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Adobe_Swatch_Exchange_Fmt	1249	1180	Adobe Swatch Exchange Format		ASE, ASEF	
ASCII_Scene_Exporter_Fmt	1250	1181	Autodesk 3ds Max ASCII Scene Exporter file		ASE	
AVR_Fmt	1251	1182	AVR (Audio Visual Research) format		AVR	
Winamp_AVS_Fmt	1252	1183	Winamp AVS (Advanced Visualization Studio) plug-in file		AVS	
After_Effects_Project_Fmt	1253	1184	Adobe After Effects project		AEP	
Anfy_Applet_Generator_Fmt	1254	1185	Anfy (Java) Applet Generator file		AJP	
SmartCipher_Fmt	1255	1186	SmartCipher encrypted file			
General_Exchange_Fmt	1256	1187	General Exchange Format (GXF)	application/gxf	GXF	
Maxis_XA_Fmt	1257	1188	Maxis XA audio file		XA	
NUT_Fmt	1258	1189	NUT Open Container Format		NUT	
OpenMG_Audio_Fmt	1259	1190	Sony OpenMG Audio (OMA) container file		OMA, OMG	
TXD_Fmt	1260	1191	Renderware Texture Dictionary (TXD) file		TXD	
DFA_Fmt	1261	1192	DreamForge DFA FMV format		DFA	
FunCom_ISS_Fmt	1262	1193	FunCom ISS audio		ISS	
Sony_MSV_Fmt	1263	1194	Sony Compressed Audio (MSV/DVF)		DVF, ICS, MSV	
THP_Fmt	1264	1195	GameCube THP Video		THP	
Smush_Animation_Fmt	1265	1196	Smush Animation Format (SAN)		SAN, NUT	
SIFF_Audio_Fmt	1266	1197	Beam Software SIFF audio file		SON	
SNES_SPC_Fmt	1267	1198	SNES SPC700 audio file		SPC	
Sierra_VMD_Fmt	1268	1199	Sierra Video and Music Data format		VMD	
VTech_MJP_Fmt	1269	1200	VTech MHP video format		MJP	
Nullsoft_Video_Fmt	1270	1201	Nullsoft Video format (NSV)		NSV	
Shorten_Fmt	1271	1202	Shorten audio file		SHN	
Leitch_Video_Fmt	1272	1203	Leitch Exchange Format video (LXF)		LXF	
ETV_Fmt	1273	1204	ETV video file		ETV	
TAK_Audio_Fmt	1274	1205	TAK audio file		TAK	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Maelstrom_ANM_Fmt	1275	1206	Maelstrom ANM animation		ANM	
SW_ANM_Fmt	1276	1207	Savage Warriors ANM animation		ANM	
DeluxePaint_Animation_Fmt	1277	1208	DeluxePaint animation		ANM	
Crack_Art_Fmt	1278	1209	Crack Art image		CA1	
Time_Shift_Video_Fmt	1279	1210	Time Shift Video (TSV) format		TSV	
XBV_Fmt	1280	1211	XBV video		XBV	
HNM4_Fmt	1281	1212	CRYO HNM4 video		HNM	
HNM6_Fmt	1282	1213	CRYO HNM6 video		HNM, HNS	
NXV_Fmt	1283	1214	NXV video		NXV	
VP5_Fmt	1284	1215	On2 VP5 video		VP5	
FutureVision_FST_Fmt	1285	1216	FutureVision FST video		FST	
Electronic_Arts_Audio_Fmt	1286	1217	Electronic Arts audio file		STR	
YOP_Fmt	1287	1218	Psygnosis YOP video		YOP	
Matrox_Setup_Program_Fmt	1288	1219	Matrox Setup Program Archive MVA file		MVA	
Vivado_Design_Suite_Fmt	1289	1220	Xilinx Vivado Design Suite file		VDS	
Meridian_Lossless_Packing_Fmt	1290	1221	Meridian Lossless Packing Audio file		MLP	
Electronic_Arts_SEAD_Fmt	1291	1222	Electronic Arts SEAD audio		TGV	
Electronic_Arts_MPC_Fmt	1292	1223	Electronic Arts MPC video		MPC	
PMP_Fmt	1293	1224	PMP video		PMP	
DEGAS_Fmt	1294	1225	DEGAS (Design & Entertainment Graphic Arts System) image		PI1, PI2, PI3	
DEGAS_Compressed_Fmt	1295	1226	DEGAS (Design & Entertainment Graphic Arts System) compressed image		PC1, PC2, PC3	
AutoCAD_Plotter_Fmt	1296	1227	AutoCAD Plot Style and Configuration files		CTB, STB, PC3, PMP	
Tiny_Stuff_Fmt	1297	1228	Tiny Stuff image		TNY, TN1, TN2, TN3.TN4.TN5.TN6	
JV_Video_Fmt	1298	1229	Bitmap Brothers JV video		JV	
REDCode_Fmt	1299	1230	REDCode video format		R3D	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
SIFF_Video_Fmt	1300	1231	Beam Software SIFF video file		VB	
VP6_Fmt	1301	1232	On2 VP6 video		VP6	
MTV_Fmt	1302	1233	Chinese MP4/MTV video		MTV	
RSO_Fmt	1303	1234	Mindstorm RSO audio		RSO	
Star3_Fmt	1304	1235	Creative Labs Star 3 audio		ST3	
DXA_Fmt	1305	1236	Runesoft DXA video		DXA	
MTH_Fmt	1306	1237	Nintendo GameCube video file		MTH	
MAD_Fmt	1307	1238	Electronic Arts MAD video file		MAD	
Bink2_Fmt	1308	1239	Bink Video 2 audio-video container		BIK, BK2	
PVA_Fmt	1309	1240	TechnoTrend PVA video		PVA	
Interplay_ACMP_Fmt	1310	1241	Interplay ACMP audio			
Ipix_Fmt	1311	1242	Ipix spherical image		IPX	
IVR_Fmt	1312	1243	RealNetworks Internet Video Recording (IVR) file		IVR	
NuppelVideo_Fmt	1313	1244	NuppelVideo file		NUV	
VFlash_PTX_Fmt	1314	1245	VTech V.Flash VTX image		PTX	
PMD_Ringtone_Fmt	1315	1246	Polyphonic Ringtone PMD audio	application/x-pmd	PMD	
RoQ_Fmt	1316	1247	RoQ video		ROQ	
CRYO_APC_Fmt	1317	1248	CRYO Interactive APC audio		APC, HNM, BF, ZIK	
VGZ_Fmt	1318	1249	VGZ video		VGZ	
Novastorm_Video_Fmt	1319	1250	Novastorm Media video file		FA, FLM	
UTalk_Fmt	1320	1251	MicroTalk/UTalk audio		UTK	
Xbox_XMV_Fmt	1321	1252	Microsoft Xbox XMV video		XMV	
AbiWord_Fmt	1322	1253	AbiWord document	application/x-abiword	ABW	
AbiWord_Template_Fmt	1323	1254	AbiWord template		ABT	
Psion_Word_Fmt	1324	1255	Psion EPOC Word document		PSI, PSITEXT	stringssr
Psion_Sheet_Fmt	1325	1256	Psion EPOC Sheet spreadsheet		PSISHEET	
Psion_Sketch_Fmt	1326	1257	Psion EPOC Sketch image			

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Psion_Record_Fmt	1327	1258	Psion EPOC Record audio			
Psion_MBM_Fmt	1328	1259	Psion EPOC Multi-Bitmap (MBM) image		MBM	
Psion_TextEd_Fmt	1329	1260	Psion EPOC TextEd file			stringssr
Psion_AIF_Fmt	1330	1261	Psion EPOC Application Information File (AIF)		AIF	
Psion_PIC_Fmt	1331	1262	Psion 3 PIC bitmap		PIC	
Psion_Object_Fmt	1332	1263	Psion 3 OPL Object File		OPA, OPO	
Psion_Executable_Fmt	1333	1264	Psion 3 IMG/APP executable		IMG, APP	
Psion_Sound_Fmt	1334	1265	Psion 3 Sound file		WVE	
Psion_Database_Fmt	1335	1266	Psion EPOC Database			
Psion_Word_3_Fmt	1336	1267	Psion 3 Word document		WRD	stringssr
Psion_Sheet_3_Fmt	1337	1268	Psion 3 Sheet spreadsheet		SPR	
Zoner_Draw_Fmt	1338	1269	Zoner Draw / Zoner Callisto Metafile (ZMF)		ZMF	
Zoner_BMI_Fmt	1339	1270	Zoner BMI image		BMI	
TealDoc_Fmt	1340	1271	TealDoc PalmOS eBook		PDB	
TealPaint_Fmt	1341	1272	TealPaint PalmOS eBook		PDB	
PalmDOC_Fmt	1342	1273	PalmDOC / Aportis DOC eBook	application/x-aportisdoc	PRC, PDB	
QiOO_Fmt	1343	1274	QiOO mobile eBook		JAR	
Plucker_Fmt	1344	1275	Plucker eBook	application/prs.plucker	PDB	
eReader_Fmt	1345	1276	eReader (Palm Reader/ Peanut Reader) eBook		PDB	
Quickword_Fmt	1346	1277	PalmOS Quickword document		PRC	stringssr
Quicksheet_Fmt	1347	1278	PalmOS Quicksheet document		PRC	
Quickpoint_Fmt	1348	1279	PalmOS Quickpoint document		PRC	
TealMeal_Fmt	1349	1280	TealMeal PalmOS database		PDB	
zTXT_Fmt	1350	1281	zTXT eBook	application/x-pdb-ztxt-ebook	PDB	
TomeRaider_Fmt	1351	1282	TomeRaider eBook		TR	
TomeRaider_PDB_Fmt	1352	1283	TomeRaider PDB eBook		TR2, TR3	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
WordSmith_Fmt	1353	1284	PalmOS Wordsmith document			
iSilo_Fmt	1354	1285	PalmOS iSilo document	application/x-pdb-isilo-ebook	PDB	
SuperMemo_Fmt	1355	1286	PalmOS SuperMemo document		KNO, PDB	
BDicty_Fmt	1356	1287	PalmOS BDicty document		PDB	
PalmOS_Executable_Fmt	1357	1288	PalmOS executable	application/vnd.palm	PRC	
PalmOS_Library_Fmt	1358	1289	PalmOS dynamic library		PRC	
Shanda_Bambook_Fmt	1359	1290	Shanda Bambook eBook	application/x-snb-ebook	SNB	
PMLZ_Fmt	1360	1291	Palm Markup Language (PMLZ) eBook		PMLZ	
Rocket_eBook_Fmt	1361	1292	Rocket eBook	application/x-rocketbook	RB	
iBooks_Author_Fmt	1362	1293	Apple iBooks Author eBook	application/vnd.apple.ibauthor	IBA	
Statistica_Spreadsheet_Fmt	1363	1294	Statsoft Statistica Spreadsheet		STA	
Statistica_Graph_Fmt	1364	1295	Statsoft Statistica Graph File		STG	
Statistica_Scrollsheet_Fmt	1365	1296	Statsoft Statistica Scrollsheet		SCR	
Apple_Newton_Package_Fmt	1366	1297	Apple Newton executable/installer/file		PKG	
Adobe_Zip_Extension_Fmt	1367	1298	Adobe Zip Format Extension Package (ZXP)	application/vnd.adobe.air-ucf-package+zip	ZXP	
Uniform_Office_Fmt	1368	1299	Uniform Office Format document		UOF	
Uniform_Office_Text_Fmt	1369	1300	Uniform Office Format word processing document	application/vnd.uof.text	UOF, UOT	
Uniform_Office_Spreadsheet_Fmt	1370	1301	Uniform Office Format spreadsheet	application/vnd.uof.spreadsheet	UOF, UOS	
Uniform_Office_Presentation_Fmt	1371	1302	Uniform Office Format presentation	application/vnd.uof.presentation	UOF, UOP	
Uniform_Office_Zip_Fmt	1372	1303	Uniform Office Format document, zip format		UOF	
Uniform_Office_Text_Zip_Fmt	1373	1304	Uniform Office Format word processing document, zip format	application/vnd.uof.text+zip	UOF, UOT	
Uniform_Office_Spreadsheet_Zip_Fmt	1374	1305	Uniform Office Format spreadsheet, zip format	application/vnd.uof.spreadsheet+zip	UOF, UOS	
Uniform_Office_Presentation_Zip_Fmt	1375	1306	Uniform Office Format presentation, zip format	application/vnd.uof.presentation+zip	UOF, UOP	
MacDraft_Fmt	1376	1307	MacDraft drawing		DRW, MDD	
RagTime_Fmt	1377	1308	RagTime document		RAG, RTD	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
MacDraw_Fmt	1378	1309	MacDraw drawing			
Wingz_Fmt	1379	1310	Wingz spreadsheet		WKZ	
Claris_Draw_Fmt	1380	1311	Claris Draw document			
BeagleWorks_Word_Fmt	1381	1312	BeagleWorks (later WordPerfect Works) Word Processor document		BW, WPW	stringssr
BeagleWorks_Database_Fmt	1382	1313	BeagleWorks (later WordPerfect Works) Database document		BW, WPW	
BeagleWorks_Spreadsheet_Fmt	1383	1314	BeagleWorks (later WordPerfect Works) Spreadsheet document		BW, WPW	
BeagleWorks_Paint_Fmt	1384	1315	BeagleWorks (later WordPerfect Works) Paint document		BW, WPW	
BeagleWorks_Draw_Fmt	1385	1316	BeagleWorks (later WordPerfect Works) Draw document		BW, WPW	
GreatWorks_Word_Fmt	1386	1317	Symantec GreatWorks Word Processor document			stringssr
GreatWorks_Outline_Fmt	1387	1318	Symantec GreatWorks Outline document			
GreatWorks_Database_Fmt	1388	1319	Symantec GreatWorks Database document			
GreatWorks_Spreadsheet_Fmt	1389	1320	Symantec GreatWorks Spreadsheet document			
GreatWorks_Draw_Fmt	1390	1321	Symantec GreatWorks Draw document			
GreatWorks_Chart_Fmt	1391	1322	Symantec GreatWorks Chart document			
MS_Works_3_Mac_WP_Fmt	1392	1323	Microsoft Works for Mac, version 3 and 4, Word Processor document	application/x-msworks	MSW, WPS	
MS_Works_3_Mac_DB_Fmt	1393	1324	Microsoft Works for Mac, version 3 and 4, Database	application/x-msworks	WDB	
MS_Works_3_Mac_SS_Fmt	1394	1325	Microsoft Works for Mac, version 3 and 4, Spreadsheet	application/x-msworks	WKS	
MS_Works_3_Mac_Comm_Fmt	1395	1326	Microsoft Works for Mac, version 3 and 4, Communications document	application/x-msworks		
MS_Works_3_Mac_Draw_Fmt	1396	1327	Microsoft Works for Mac, version 3 and 4, Draw document	application/x-msworks	MSW	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
SAP_VDS_Fmt	1397	1328	SAP 3d Visual Enterprise VDS document		VDS	
ZIPVFS_Fmt	1398	1329	ZIPVFS SQLite compressed read/write database		SQLITE	
Right_Hemisphere_Material_Fmt	1399	1330	Right Hemisphere Material file		RH, RHM	
RH_Thumbnails_Fmt	1400	1331	Right Hemisphere thumbnail collection file		$RH	
Westwood_Studios_Audio_Fmt	1401	1332	Westwood Studios Audio file		AUD	
Shockwave_Stream_Fmt	1402	1333	Shockwave Stream audio-video file		STREAM	
EGG_Video_Fmt	1403	1334	EGG video file		EGG	
IRCAM_Fmt	1404	1335	IRCAM audio file		IRCAM	
Sierra_Audio_Fmt	1405	1336	Sierra Entertainment audio file		SOL	
TiVo_Video_Fmt	1406	1337	TiVo video		TY+	
OptimFROG_Fmt	1407	1338	OptimFROG audio		OFR, OFS	
LPAC_Fmt	1408	1339	Lossless Predictive Audio Compression file		PAC	
RK_Audio_Fmt	1409	1340	RK Audio lossless compressed audio		RKA	
Asylum_Music_Fmt	1410	1341	Asylum Music Format		AMF	
Novastorm_Audio_Fmt	1411	1342	Novastorm Media audio file		SMP	
HHE_Fmt	1412	1343	HHE video		HHE	
Portable_Voice_Fmt	1413	1344	Portable Voice Format audio		PVF	
CNM_Video_Fmt	1414	1345	Arxel CNM audio-video format		CNM	
Phantom_Cine_Fmt	1415	1346	Phantom Cine video file		CINE	
MPEG2_Transport_Stream_Fmt	1416	1347	MPEG-2 Transport Stream video		M2TS	
Audacity_Project_Fmt	1417	1348	Audacity audio project file	application/x-audacity-project	AUP	
Voltage_VSF_Fmt	1418	1349	Micro Focus Voltage VSF encrypted file		VDF	
XLIFF_Fmt	1419	1350	XML Localization Interchange File Format (XLIFF)	application/xliff+xml	XLF	
XBRL_Fmt	1420	1351	Extensible Business Reporting Language (XBRL)		XBRL	
AuditXPressX_Fmt	1421	1352	AuditXPressX file		AXPX	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Box_Note_Fmt	1422	1353	Box Note document		BOXNOTE	
Hikvision_DVR_Fmt	1423	1354	Hikvision DVR video			
Electronic_Arts_TGV_Fmt	1424	1355	Electronic Arts TGV video		TGV	
Electronic_Arts_TGQ_Fmt	1425	1356	Electronic Arts TGQ video		TGQ	
Reaper_Video_Fmt	1426	1357	Reaper Video		FMV	
Lightweight_Video_Fmt	1427	1358	Lightweight Video Format (LVF)		LVF	
Liquid_Audio_Fmt	1428	1359	Liquid Audio		LQT	
Extended_Instrument_Fmt	1429	1360	eXtended Instrument generic audio tracker		XI	
MAML_Fmt	1430	1361	Microsoft Assistance Markup Language		AML	
MS_Chat_Character_Fmt	1431	1362	Microsoft Comic Chat Character		AVB	
MS_Border_Fmt	1432	1363	Microsoft Office Border images		BDR	
MS_Binary_Log_Fmt	1433	1364	Microsoft Binary Log file		BLG	
MS_Reader_eBook_Fmt	1434	1365	Microsoft Reader eBook file		LIT	
MS_Reader_Annotations_Fmt	1435	1366	Microsoft Reader annotation file		EBO	
Amazon_KFX_Aux_Fmt	1436	1367	Amazon KFX eBook auxiliary format (2015)		KFX, AZW	
Amazon_KFX_Ion_Fmt	1437	1368	Amazon KFX eBook Ion format (2015)		KFX, AZW, ION	
MS_DPAPI_Fmt	1438	1369	Microsoft Data Protection API (DPAPI) data			
MS_Streets_Fmt	1439	1370	Microsoft Streets & Trips map		EST	
MS_Fast_Find_Index_Fmt	1440	1371	Microsoft Office Fast Find Index		FFX	
MS_Fresh_Paint_Fmt	1441	1372	Microsoft Fresh Paint image		FPPX	
MS_Mathematics_Fmt	1442	1373	Microsoft Mathematics worksheet		GCW	
MS_Instrument_Definition_Fmt	1443	1374	Microsoft MIDI Instrument Definition File		IDF	
MS_Pocket_Streets_Fmt	1444	1375	Microsoft Pocket Streets map		MPS	
Obfuscated_OpenType_Fmt	1445	1376	Obfuscated OpenType font (ODTTF)	application/vnd.ms-package.obfuscated-opentype	ODTTF	
Pfaff_PCS_Fmt	1446	1377	Pfaff PCS embroidery image		PCS	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Janome_JEF_Fmt	1447	1378	Janome JEF embroidery format		JEF	
Husqvarna_HUS_Fmt	1448	1379	Husqvarna Viking HUS embroidery format		HUS	
Husqvarna_VIP_Fmt	1449	1380	Husqvarna Viking-Pfaff VIP embroidery format		VIP	
Brother_PEC_Fmt	1450	1381	Brother PEC embroidery format		PEC	
Brother_PES_Fmt	1451	1382	Brother PEC embroidery format		PES	
Viking_SHV_Fmt	1452	1383	Viking SHV embroidery format		SHV	
VP3_Fmt	1453	1384	VP3 embroidery format		VP3	
SEW_Fmt	1454	1385	SEW embroidery format		SEW	
Data_Stitch_Tajima_Fmt	1455	1386	Data Stitch Tajima (DST) embroidery image		DST	
Singer_XXX_Fmt	1456	1387	Singer XXX embroidery image		XXX	
Bernina_ART_Fmt	1457	1388	Bernina ART embroidery image		ART	
MS_Prefetch_Fmt	1458	1389	Microsoft Windows Prefetch (uncompressed) file		PF	
MS_Prefetch_Compressed_Fmt	1459	1390	Microsoft Windows Prefetch (compressed) file		PF	
MS_MapPoint_Fmt	1460	1391	Microsoft MapPoint map		PTM	
MS_Live_Meeting_Fmt	1461	1392	Microsoft Office Live Meeting Connection		RTC	
MS_Speech_Definitions_Fmt	1462	1393	Microsoft text-to-speech Speech Definitions File		SDF	
MS_Speech_Data_Fmt	1463	1394	Microsoft text-to-speech Speech Data File		SPD	
MS_SQL_CE_Fmt	1464	1395	Microsoft SQL Server Compact (CE) edition database		SDF	
MS_ICE_Project_Fmt	1465	1396	Microsoft Image Composite Editor (ICE) Project		SPJ	
MS_DVR_Fmt	1466	1397	Microsoft Digital Video Recording (DVR-MS)	video/x-ms-dvr	DVR-MS	
Symbol_Dynamics_EXP_Fmt	1467	1398	Symbol Dynamics EXP document		WXP	stringssr
XNA_Compiled_Fmt	1468	1399	Microsoft XNA Compiled Format		XNB	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Outlook_Shortcut_Fmt	1469	1400	Microsoft Outlook or Exchange folder shortcut		XNK	
ChiWriter_Fmt	1470	1401	ChiWriter document (up to version 3)		CHI	
ChiWriter4_Fmt	1471	1402	ChiWriter document (version 4)		CHI	
Lightning_Strike_Fmt	1472	1403	Lightning Strike image	image/cis-cod	COD	
Blackberry_Executable_Fmt	1473	1404	Blackberry executable		COD	
EndNote_Library_Fmt	1474	1405	EndNote Library (up to version 9)	application/x-endnote-library	ENL	
EndNote_Library_X_Fmt	1475	1406	EndNote Library (version X onwards)		ENL, ENLX	
EndNote_Filter_Fmt	1476	1407	EndNote Filter	application/x-puid-fmt-327	ENF	
EndNote_Style_Fmt	1477	1408	EndNote Style	application/x-endnote-style	ENS	
EndNote_Connection_Fmt	1478	1409	EndNote Connection	application/x-endnote-connect	ENZ	
Camtasia_Recording_Fmt	1479	1410	Camtasia Recording		CAMREC	
Camtasia_Project_Fmt	1480	1411	Camtasia XML Project		CAMPROJ	
TechSmith_Project_Fmt	1481	1412	TechSmith JSON Project		TSCPROJ	
ABIF_Fmt	1482	1413	Applied Biosystems Inc. Format (ABIF)		AB1, FSA	
CIF_Fmt	1483	1414	Crystallographic Information File	chemical/x-cif	CIF	
Sibelius_Fmt	1484	1415	Sibelius musical score		SIB	
Geogebra_Worksheet_Fmt	1485	1416	Geogebra worksheet	application/vnd.geogebra.file	GGB	
Geogebra_Tool_Fmt	1486	1417	Geogebra tool		GGT	
Polynomial_Texture_Map_Fmt	1487	1418	Polynomial Texture Map (PTM)		PTM	
Poly_Tracker_Fmt	1488	1419	Poly Tracker audio		PTM	
PC_Outline_Fmt	1489	1420	PC-Outline document		PCO	
Spline_Font_Database_Fmt	1490	1421	Spline Font Database (SFD) font		SFD	
QuickTime_Image_Fmt	1491	1422	QuickTime (QTIF) image	image/x-quicktime	QTIF, QIF, QTI	
XBin_Image_Fmt	1492	1423	XBin image		XB	
Segmented_Hypergraphics_Fmt	1493	1424	MS Segmented Hypergraphics image		SHG	
LEADTools_CMP_Fmt	1494	1425	LEADTools CMP image		CMP	
WBMP_Fmt	1495	1426	Wireless Bitmap image (WBMP)	image/vnd.wap.wbmp	WBMP	
Blender_Fmt	1496	1427	Blender (v2) CAD file	application/x-blender	BLEND	

Format Name	Number	Category	Description	MIME Type	Extension	Readers
Blender_v1_Fmt	1497	1428	Blender (v1) CAD file	application/x-blender	BLEND	
Scribus_Fmt	1498	1429	Scribus document	application/vnd.scribus	SLA	
LyX_Fmt	1499	1430	LyX document	application/x-lyx	LYX	
NZB_Fmt	1500	1431	NewzBin NZB format	application/x-nzb	NZB	
KWord_Fmt	1501	1432	KOffice KWord document	application/vnd.kde.kword	KWD	
KSpread_Fmt	1502	1433	KOffice KSpread document	application/vnd.kde.kspread	KSP	
KPresenter_Fmt	1503	1434	KOffice KPresenter document	application/vnd.kde.kpresenter	KPR	
KWord_GZ_Fmt	1504	1435	KOffice (up to v1.1) kWord document	application/x-kword	KWD	
KSpread_GZ_Fmt	1505	1436	KOffice (up to v1.1) kSpread document	application/x-kspread	KSP	
KPresenter_GZ_Fmt	1506	1437	KOffice (up to v1.1) kPresenter document	application/x-kpresenter	KPR	
Karbon_Fmt	1507	1438	KOffice Karbon document	application/vnd.kde.karbon	KARBON	
KChart_Fmt	1508	1439	KOffice KChart document	application/vnd.kde.kchart	CHRT	
KPlato_Fmt	1509	1440	KOffice KPlato document	application/x-vnd.kde.kplato	KPLATO	
GIMP_Pattern_Fmt	1510	1441	GIMP Pattern file		PAT	
GIMP_Brush_Fmt	1511	1442	GIMP Brush file		GBR	
GIMP_Animated_Brush_Fmt	1512	1443	GIMP Animated Brush file		GIH	
Git_Pack_Index_Fmt	1513	1444	Git Pack Index format		IDX	
Git_Index_Fmt	1514	1445	Git Index format		INDEX	

[1] MHT, EML, and MBX files might return either format 2, 233, or 395, depending on the text in the file. In general, files that contain fields such as **To**, **From**, **Date**, or **Subject** are considered to be email messages; files that contain fields such as **content-type** and **mime-version** are considered to be MHT files; and files that do not contain any of those fields are considered to be text files.

[2] All CAT file extensions, for example CATDrawing, CATProduct, CATPart, and so on.

[3] This format is returned only if you enable source code identification.

[4] This format is returned only if you enable extended source code identification.

Appendix B: KeyView Document Readers

This section lists the KeyView document readers that are available to filter, export, and view supported file formats.

Key to Document Readers Table

The document readers table includes the following information.

Column	Description
Reader	The name of the reader.
Description	A description of the reader.
Filter	Shows whether KeyView can filter text from the main content of the file.
Export	Shows whether KeyView supports export to HTML, XML, and PDF.
View	Shows whether KeyView provides viewing capability.
Extract	Shows whether KeyView can extract sub-files.
Metadata	Shows whether KeyView can extract metdata (properties such as title, author, and subject).
Charset	Shows whether KeyView can detect and extract the character set. Even though a file format might be able to provide character set information, some documents might not contain character set information. Therefore, the document reader would not be able to determine the character set of the document.
H/F	Shows whether KeyView can extract headers and footers.
Associated File Formats	The file formats that are supported by the reader.

Key to Symbols

Symbol	Description
Y	The feature is supported.

Key to Symbols, continued

Symbol	Description
N	The feature is not supported.
P	Partial metadata is extracted from this format. Some non-standard fields are not extracted.
T	Only text is extracted from this format. Formatting information is not extracted.
M	Only metadata (title, subject, author, and so on) is extracted from this format. Text and formatting information are not extracted.

Document Readers

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
ActiveX components	Microsoft Visio (2013)	N	N	Y[1]	N	Y	N	N	MS_Visio_2013_Fmt
ad1sr	AD1 Evidence file	N	N	Y	Y	N	n/a	N	AD1_Fmt
afsr	ASCII Text	Y	Y	Y	N	N	N	N	ABAP_Fmt, AMPL_Fmt, APL_Fmt, ASCII_Text_Fmt, ASN1_Fmt, ATS_Fmt, Agda_Fmt, Alloy_Fmt, Apex_Fmt, AppleScript_Fmt, Arduino_Fmt, AsciiDoc_Fmt, AspectJ_Fmt, Assembly_Fmt, Awk_Fmt, BlitzMax_Fmt, Bluespec_Fmt, Brainfuck_Fmt, Brightscript_Fmt, CLIPS_Fmt, CMake_Fmt, COBOL_Fmt, CPlusPlus_Fmt, CWeb_Fmt, C_Fmt, CartoCSS_Fmt, Ceylon_Fmt, Chapel_Fmt, Clarion_Fmt, Clean_Fmt, Clojure_Fmt, CoffeeScript_Fmt, Component_Pascal_Fmt, Cool_Fmt, Coq_Fmt, Creole_Fmt, Crystal_

[1] Visio 2013 is supported in Viewing only, with the support of ActiveX components from the Microsoft Visio 2013 Viewer. Image fidelity is supported but other features, such as highlighting, are not.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
									Fmt, Csharp_Fmt, Csound_Document_Fmt, Csound_Fmt, Css_Fmt, Cuda_Fmt, DIGITAL_Command_Language_Fmt, DTrace_Fmt, D_Fmt, Dart_Fmt, Dockerfile_Fmt, ECL_Fmt, E_Fmt, Eiffel_Fmt, Elm_Fmt, Emacs_Lisp_Fmt, EmberScript_Fmt, Erlang_Fmt, Fantom_Fmt, Forth_Fmt, Fortran_Fmt, FreeMarker_Fmt, Frege_Fmt, Fsharp_Fmt, GAMS_Fmt, GAP_Fmt, GDScript_Fmt, GIS_World_File_Fmt, GLSL_Fmt, G_code_Fmt, Game_Maker_Language_Fmt, Gnuplot_Fmt, Go_Fmt, Golo_Fmt, Gosu_Fmt, Gradle_Fmt, GraphQL_Fmt, Graphviz_DOT_Fmt, Groovy_Fmt, HLSL_Fmt, Hack_Fmt, Haml_Fmt, Handlebars_Fmt, Haskell_Fmt, Hy_Fmt, IDL_Fmt, IGOR_Pro_Fmt, Idris_Fmt, Inform_7_Fmt, Ini_Fmt, Ioke_Fmt, Isabelle_Fmt, JSONiq_Fmt, JSX_Fmt, J_Fmt, Jasmin_Fmt, Java_Fmt, Javascript_Fmt, Jolie_Fmt, Julia_Fmt, KiCad_Layout_Fmt, KiCad_Schematic_Fmt, Kotlin_Fmt, LFE_Fmt, LOLCODE_Fmt, Lasso_Fmt, Limbo_Fmt, Lisp_Fmt, LiveScript_Fmt, Lua_Fmt, MAXScript_Fmt, ML_Fmt, MSDOS_Batch_File_Fmt, M_Fmt, Makefile_Fmt, Markdown_Fmt, Mathematica_Fmt, Matlab_Fmt, Max_Code_Fmt, Mercury_Fmt, Modelica_

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
									Fmt, Modula_2_Fmt, Monkey_Fmt, Moocode_Fmt, NL_Fmt, NSIS_Fmt, NetLogo_Fmt, NewLisp_Fmt, Nginx_Fmt, Nix_Fmt, Nu_Fmt, OCaml_Fmt, ObjC_Fmt, ObjCpp_Fmt, ObjJ_Fmt, OpenCL_Fmt, OpenEdge_ABL_Fmt, OpenSCAD_Fmt, Ox_Fmt, Oxygene_Fmt, Oz_Fmt, PAWN_Fmt, PHP_Fmt, PLSQL_Fmt, PLpgSQL_Fmt, Pan_Fmt, Parrot_Assembly_Fmt, Pascal_Fmt, Perl_Fmt, PicoLisp_Fmt, Pike_Fmt, Pony_Fmt, Powershell_Fmt, Processing_Fmt, Prolog_Fmt, Puppet_Fmt, PureBasic_Fmt, Python_Fmt, QMake_Fmt, RAML_Fmt, RDoc_Fmt, REXX_Fmt, R_Fmt, Racket_Fmt, Ragel_Fmt, Rascal_Fmt, Rebol_Fmt, Red_Fmt, RenPy_Fmt, RenderScript_Fmt, Ring_Fmt, RobotFramework_Fmt, Ruby_Fmt, Rust_Fmt, SAS_Fmt, SGML_Fmt, SPARQL_Fmt, SQLPL_Fmt, SQL_Fmt, SaltStack_Fmt, Scala_Fmt, Scheme_Fmt, Scilab_Fmt, Scribe_Fmt, Shell_Fmt, Smalltalk_Fmt, Squirrel_Fmt, Stan_Fmt, Stata_Fmt, Stylus_Fmt, SuperCollider_Fmt, Swift_Fmt, SystemVerilog_Fmt, TSV_Fmt, TSV_Fmt, TXL_Fmt, Tcl_Fmt, Tex_Fmt, Turing_Fmt, Turtle_Fmt, TypeScript_Fmt, UrWeb_Fmt, Verilog_Fmt, Vim_script_Fmt, Visual_Basic_Fmt, WebAssembly_Fmt, WebIDL_Fmt,

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
									Wiki_Fmt, X10_Fmt, XQuery_Fmt, Xojo_Fmt, Xtend_Fmt, YAML_Fmt, YANG_Fmt, Zephir_Fmt, eC_Fmt, reStructuredText_Fmt, xBase_Fmt
aiffsr	Audio Interchange File Format	M	N	N	N	Y	N	N	AIFF_Fmt
asfsr	Advanced Systems Format (1.2)	N	N	N	N	Y	N	N	ASF_Fmt, WMA_Fmt, WMV_Fmt
assr	Applix Spreadsheets (4.2, 4.3, 4.4)	Y	Y	Y	N	N	Y	N	Applix_Spreadsheets_Fmt
awsr	Applix Words (3.11, 4, 4.1, 4.2, 4.3, 4.4)	Y	Y	Y	N	N	Y	Y	Applix_Words_Fmt
axsr	Applix Asterix	Y	T	T	N	N	N	N	Applix_Alis_Fmt
b1sr	B1	N	N	Y	Y	N	n/a	N	B1_Fmt
bkfsr	Microsoft Backup File	N	N	Y	Y	N	n/a	N	BKF_Fmt
bmpsr	Windows Bitmap Image	M	M	N	N	Y	N	N	BMP_Fmt
bzip2sr	Bzip2 Compressed File	N	N	Y	Y	N	n/a	N	BZIP2_Fmt
cabsr	Microsoft Cabinet File (1.3)	N	N	Y	Y	N	n/a	N	CAB_Fmt

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
cdsr	Convergent Technologies DEF Comm. Format	Y	T	T	N	N	N	N	CT_DEF_Fmt
cebsr[1]	Founder Chinese E-paper Basic (3.2.1)	Y	N	N	N	N	N	N	Founder_CEB_Fmt
chmsr	Microsoft Compiled HTML Help (3)	N	N	Y	Y	N	n/a	N	CHM_Fmt
csvsr	CSV (Comma Separated Values)	Y	Y	Y	N	N	N	N	CSV_Fmt
dbfsr	dBase Database (III+, IV)	Y	Y	Y	N	N	N	N	dBase_Fmt
dbxsr	Microsoft Outlook Express DBX Message Database (5.0, 6.0)	N	N	Y	Y	Y	Y	N	MS_OEDBX_Fmt
dcasr	IBM DCA/RFT (Revisable Form Text) (SC23-0758-1)	Y	Y	Y	N	N	Y	N	DCA_RFT_Fmt
dcmsr	Digital Imaging &	M	N	N	N	Y	N	N	Dicom_Fmt

[1] This reader is only supported on Windows 32-bit platforms.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	Communications in Medicine (DICOM)								
difsr	Data Interchange Format	Y	Y	Y	N	N	N	N	DIF_SpreadSheet_Fmt
dmgsr	Mac Disk Copy Disk Image	N	N	Y	Y	N	n/a	N	DMG_Fmt
dw4sr	DisplayWrite (4)	Y	Y	Y	N	N	Y	N	IBM_Display_Write_Fmt
dxlsr	IBM Domino Data in XML format[1]	N	N	Y	Y	Y	N	N	Lotus_Domino_DXL_Fmt
emlsr[2]	Text Mail (MIME) / Microsoft Outlook Express (Windows 6, MacIntosh 5)	Y	T	T	Y	Y	Y	N	SMTP_Fmt
emxsr	Legato EMailXtender Archives	N	N	Y	Y	N	n/a	N	EMX_Fmt
encase2sr	Expert Witness Compression Format (EnCase) (7)	N	N	Y	Y	N	n/a	N	EnCase_Fmt

[1] Supports non-encrypted embedded files only.

[2] This reader supports both clear signed and encrypted S/MIME. KeyView supports S/MIME for PST, EML, MBX, and MSG files.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
encasesr	Expert Witness Compression Format (EnCase) (6)	N	N	Y	Y	N	n/a	N	EnCase_Fmt
entsr	Microsoft Entourage Database (2004)	N	N	Y	Y	Y	Y	N	ENT_Fmt
epubsr	Open Publication Structure eBook (2.0, 3.0)	Y	Y	Y	N	Y	Y	N	Epub_Fmt
exesr	MSDOS/Windows Executable	N	N	Y	N	N	n/a	N	MS_Executable_Fmt
foliosr	Folio Flat File (3.1)	Y	Y	Y	N	Y	Y	Y	Folio_Flat_Fmt
gdsiisr	GDSII data format	Y	T	T	N	N	N	N	GDSII_Fmt
gifsr	GIF (87, 89)	M	M	N	N	Y	N	N	GIF_87a_Fmt, GIF_89a_Fmt
gwfssr	GroupWise FileSurf email	N	N	Y	Y	Y	N	N	GWFS_Email_Fmt
hl7sr	Health level7 message (2.0)	Y	Y	Y	N	Y	Y	N	Hl7_Fmt
htmsr	HTML/XHTML (3, 4)	Y	Y	Y	N	Y[1]	Y	N	HTML_Fmt, Netscape_Bookmark_File_Fmt

[1] HTML only supports partial metadata extraction

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
hwposr	Haansoft Hangul HWP (2002, 2005, 2007, 2010)	Y	Y	Y	Y	Y	Y	N	HWP_Fmt
hwpsr	Haansoft Hangul HWP (97)	Y	Y	Y	N	Y	Y	N	HWP_Fmt
ichatsr	Apple iChat Log (1, AV 2, AV 2.1, AV 3)	Y	Y	Y	N	N	N	N	Apple_iChat_Fmt
icssr	Microsoft Outlook iCalendar (1.0, 2.0)	N	N	Y	Y	Y	Y	N	ICS_Fmt
isosr	ISO-9660 CD Disc Image	N	N	Y	Y	N	n/a	N	ISO_Fmt
iwss13sr[1]	Apple iWork Numbers ('13, '16, '18, iCloud 2018)	Y	T	T	N	N	Y	N	IWSS13_Fmt
iwsssr	Apple iWork Numbers ('08, '09)	Y	Y	Y	N	Y	Y	N	IWSS_Fmt
iwwp13sr[2]	Apple iWork Pages ('13, '16, '18, iCloud 2018)	Y	T	T	N	N	N	N	IWWP13_Fmt

[1] This reader is available only on Windows (32-bit and 64-bit), Linux (32-bit and 64-bit), and Solaris x86-64.

[2] This reader is available only on Windows (32-bit and 64-bit), Linux (32-bit and 64-bit), and Solaris x86-64.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
iwwpsr	Apple iWork Pages ('08, '09)	Y	Y	Y	N	Y	Y	N	IWWP_Fmt
jp2000sr	JPEG (2000)	M	M	N	N	Y	N	N	ISO_JPEG2000_JP2_Fmt, ISO_JPEG2000_JPM_Fmt, ISO_JPEG2000_JPX_Fmt, JPEG_2000_JP2_File_Fmt, JPEG_2000_PGX_Fmt, Motion_JPEG_2000_Fmt
jpgsr	JPEG Interchange Format (JFIF)	M	M	N	N	Y	N	N	JPEG_File_Interchange_Fmt
jtdsr	JustSystems Ichitaro (8 to 2013, 2018)	Y	Y	Y	N	P	N	Y	ICHITARO_Compr_Fmt, ICHITARO_Fmt
kpagrdr	Applix Presents/Graphics (4.0, 4.2, 4.3, 4.4)	Y	Y	Y	N	N	N	N	Applix_Graphics_Fmt
kpanirdr	Windows Animated Cursor	N	Y	Y	N	N	N	N	Windows_Animated_Cursor_Fmt
kpbmprdr	Windows Bitmap Image	N	Y	Y	N	N	N	N	BMP_Fmt
kpCATrdr	CATIA formats (5)	Y	N	N	N	Y	N	N	CATIA_Fmt
kpcdrrdr	CorelDRAW[1] (through 9.0, 10, 11, 12, X3)	N	Y	Y	N	N	N	N	Corel_Draw_Fmt

[1] CDR/CDR with TIFF header.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
kpcgmrdr[1]	Computer Graphics Metafile	Y	Y	Y	N	N	N	N	CGM_Binary_Fmt, CGM_Character_Fmt, CGM_ClearText_Fmt
kpchtrdr	Microsoft Excel (2-7) and Lotus 1-2-3 Charts (2-5)	N	Y	Y	N	N	N	N	
kpdcxrdr	DCX Fax System	N	Y	Y	N	N	N	N	DCX_Fmt
kpDWGrdr[2]	Autodesk AutoCAD DWG Drawing (R13 onwards)	Y	Y	Y	N	Y	Y	N	AutoDesk_DWG_Fmt
kpDXFrdr[3]	Autodesk AutoCAD DXF Drawing (R13 onwards)	Y	Y	Y	N	Y	Y	N	AutoCAD_DXF_Binary_Fmt, AutoCAD_DXF_Text_Fmt
kpemfrdr	Enhanced Metafile	Y	Y	Y	N	Y	N	N	Enhanced_Metafile_Fmt
kpepsrdr	Encapsulated PostScript (raster) (TIFF header)	N	Y	Y	N	N	N	N	EPSF_Fmt, Preview_EPSF_Fmt
kpGFLrdr	Omni Graffle	Y	N	N	N	Y	Y	N	Omni_Graffle_XML_Fmt

[1] Files with non-partitioned data are supported.

[2] The kpODArdr reader can filter, export, and view all versions but is supported only on Windows, Linux, and macOS. The kpDWGrdr reader is used on AIX, FreeBSD, Solaris, and SPARC platforms, but does not support graphics for versions after 2004 or text for versions after 2013.

[3] The kpODArdr reader can filter, export, and view all versions but is supported only on Windows, Linux, and macOS. The kpDXFrdr reader is used on AIX, FreeBSD, Solaris, and SPARC platforms, but does not support graphics for versions after 2004.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
kpgifrdr	GIF (87, 89)	N	Y	Y	N	N	N	N	GIF_87a_Fmt, GIF_89a_Fmt
kpicordr	Windows Icon Cursor	N	Y	Y	N	N	N	N	Windows_Icon_Fmt
kpIWPG13rdr[1]	Apple iWork Keynote ('13, '16, '18, iCloud 2018)	Y	T	N	N	N	N	N	IWPG13_Fmt
kpIWPGrdr	Apple iWork Keynote (2, 3, '08, '09)	Y	Y	Y	N	Y	Y	N	IWPG13_Fmt, IWPG_Fmt
kpJBIG2rdr	JBIG2	N	Y	Y	N	N	N	N	JBIG2_Fmt
kpjp2000rdr	JPEG (2000)	N	Y	Y	N	N	N	N	ISO_JPEG2000_JP2_Fmt, ISO_JPEG2000_JPM_Fmt, ISO_JPEG2000_JPX_Fmt, JPEG_2000_JP2_File_Fmt, JPEG_2000_PGX_Fmt, Motion_JPEG_2000_Fmt
kpjpgrdr	JPEG Interchange Format (JFIF)	N	Y	Y	N	N	N	N	JPEG_File_Interchange_Fmt
kpmacrdr	MacPaint	N	Y	Y	N	N	N	N	MacPaint_Fmt
kpmsordr	Microsoft Office Drawing	N	Y	Y	N	N	N	N	MS_Office_Drawing_Fmt
kpODArdr	ODA	Y	Y	Y	N	Y	Y	N	AutoCAD_DXF_Binary_Fmt, AutoCAD_DXF_Text_Fmt, AutoDesk_DWG_Fmt

[1] This reader is available only on Windows (32-bit and 64-bit), Linux (32-bit and 64-bit), and Solaris x86-64.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
kpodfrdr	OASIS Open Document Format (1, 2[1])	Y	Y	Y	Y[2]	Y	Y	N	ODF_Drawing_Fmt, ODF_Drawing_Template_Fmt, ODF_Presentation_Fmt, ODF_Presentation_Template_Fmt, SO_Drawing_XML_Fmt, SO_Presentation_XML_Fmt
kpONErdr	Microsoft OneNote (2007, 2010, 2013, 2016)	Y	Y	Y	Y	N	Y	N	OneNote_Fmt
kpp40rdr	Microsoft PowerPoint (98)	Y	Y	Y	N	P[3]	N	N	PowerPoint_Win_Fmt
kpp95rdr	Microsoft PowerPoint Windows (95)	Y	Y	Y	N	P	Y	N	PowerPoint_95_Fmt
kpp97rdr	Microsoft PowerPoint (97-2004)	Y	Y	Y	N	P	Y	Y[4]	PowerPoint_2000_Fmt, PowerPoint_97_Fmt
kppctrdr	Macintosh Raster / QuickDraw (2)	N	Y	Y	N	N	N	N	Mac_PICT_Fmt
kppcxrdr	PC PaintBrush (3)	N	Y	Y	N	N	N	N	PC_Paintbrush_Fmt

[1] Generated by OpenOffice Impress 2.0, StarOffice 8 Impress, and IBM Lotus Symphony Presentation 3.0.

[2] Supported using the olesr embedded objects reader.

[3] Microsoft PowerPoint Windows only

[4] Microsoft PowerPoint Windows only

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
kppdf2rdr[1]	Adobe PDF (1.1 to 1.7, 2.0)	N	N	Y	N	N	N	N	PDF_Fmt
kppdfrdr	Adobe PDF (1.1 to 1.7, 2.0)	N	Y	Y	N	N	N	N	PDF_Fmt
kppicrdr	Lotus PIC	Y	Y	Y	N	N	N	N	Lotus_PIC_Fmt
kppngrdr	Portable Network Graphics	N	Y	Y	N	N	N	N	APNG_Fmt, PNG_Fmt
kpppxrdr	Microsoft PowerPoint Windows XML (2007 onwards)	Y	Y	Y	Y	Y	Y	Y	MS_PPT_2007_Fmt, MS_PPT_Macro_2007_Fmt
kpprerdr	Lotus Freelance Graphics 2 (2)	Y	Y	Y	N	N	N	N	Freelance_OS2_Fmt, Freelance_Win_Fmt
kpprzrdr	Lotus Freelance Graphics (96, 97, 98, R9, 9.8)	Y	Y	Y	N	N	N	N	Freelance_96_Fmt, Freelance_97_Fmt, Freelance_DOS_Fmt
kpsddrdr	StarOffice Impress (3, 4, 5)	Y	T	N	N	N	N	N	SO_Presentation_Fmt
kpsdwrdr	Lotus AMIDraw Graphics	N	Y	Y	N	N	N	N	Ami_Pro_Draw_Fmt, SO_Text_Fmt
kpsgirdr	SGI RGB Image	N	Y	Y	N	N	N	N	SGI_Image_Fmt

[1] kppdf2rdr is an alternate graphic-based reader that produces high-fidelity output but does not support other features such as highlighting or text searching.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
kpshwrdr	Corel Presentations (6, 7, 8, 9, 10, 11, 12, X3)	Y	Y	Y	N	N	N	N	Corel_Presentations_Fmt
kpsunrdr	Sun Raster Image	N	Y	Y	N	N	N	N	Sun_Raster_Fmt
kpTGArdr	Truevision Targa (2)	N	Y	Y	N	N	N	N	Targa_Fmt
kptifrdr	TIFF Tagged Image File (through 6.0[1])	N	Y	Y	N	N	N	N	TIFF_Fmt
kpUGrdr	Unigraphics (UG) NX	Y	N	N	N	N	N	N	Unigraphics_NX_Fmt
kpVSD2rdr	Microsoft Visio (4, 5, 2000, 2002, 2003, 2007, 2010[2])	Y	Y	Y	N	Y	Y	N	MS_Visio_Fmt
kpVSDXrdr	Microsoft Visio (2013)	Y	Y	Y	Y	Y	Y	N	MS_Visio_2013_Fmt, MS_Visio_2013_Macro_Fmt, MS_Visio_2013_Stencil_Fmt, MS_Visio_2013_Stencil_Macro_Fmt, MS_Visio_2013_Template_Fmt,

[1] The following compression types are supported: no compression, CCITT Group 3 1-Dimensional Modified Huffman, CCITT Group 3 T4 1-Dimensional, CCITT Group 4 T6, LZW, JPEG (only Gray, RGB and CMYK color space are supported), and PackBits.

[2] Viewing and Export use the graphic reader, kpVSD2rdr for Microsoft Visio 2003, 2007, and 2010, and vsdsr for all earlier versions. Image fidelity in Viewing and Export is therefore only supported for versions 2003 and above. Filter uses the graphic reader kpVSD2rdr for Microsoft Visio 2003, 2007, and 2010, and vsdsr for all earlier versions.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
									MS_Visio_2013_Template_Macro_Fmt
kpwg2rdr	WordPerfect Graphics 2 (2, 7)	N	Y	Y	N	N	N	N	WordPerfect_Graphics_Fmt
kpwmfrdr	Windows Metafile (3)	Y[1]	Y	Y	N	N	N	N	Windows_Metafile_Fmt, Windows_Metafile_NoHdr_Fmt
kpwpgrdr	WordPerfect Graphics 1 (1)	N	Y	Y	N	N	N	N	WordPerfect_Graphics_Fmt
kpXFDLrdr	Extensible Forms Description Language	Y	Y	Y	N	Y	Y	N	XFDL_Fmt
kvgz	GZIP archive (2)	N	N	Y	N	N	n/a	N	GZ_Compress_Fmt
kvgzsr	GZIP archive (2)	N	N	N	Y	N	n/a	N	GZ_Compress_Fmt
kvhqxsr	BinHex	N	N	Y	Y	N	n/a	N	BinHex_Fmt
kvzee	UNIX Compress	N	N	Y	N	N	n/a	N	Compress_Fmt
kvzeesr	UNIX Compress	N	N	N	Y	N	n/a	N	Compress_Fmt
l123sr	Lotus 1-2-3 (96, 97, R9, 9.8)	Y	Y	Y	N	P	Y	N	Lotus_123_97_Fmt, Lotus_123_Format_Fmt, Lotus_123_R9_Fmt

[1] Windows Metafiles can contain both raster images (KeyView file class 4) and vector graphics (KeyView file class 5). Filtering is supported only for vector graphics (class 5).

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
lasr	Lotus AMI Pro and Write Plus (2, 3)	Y	Y	Y	N	P[1]	Y[2]	Y	Ami_Pro_Fmt, Ami_Pro_StyleSheet_Fmt
lwpsr	Lotus Word Pro and SmartMaster (96, 97, R9)	Y	Y	Y	N	P[3]	N	Y[4]	Lotus_Word_Pro_96_Fmt, Lotus_Word_Pro_97_Fmt
lzhsr	Microsoft LZH Compressed Folder	N	N	N	Y	N	n/a	N	LZH_Fmt
macbinsr	MacBinary	N	N	Y	Y	N	n/a	N	MacBinary_Fmt
mbsr	Microsoft Word Macintosh (4, 5, 6, 98)	Y	Y	Y	N	Y	N	Y	MS_Word_Mac_4_Fmt, MS_Word_Mac_Fmt
mbxsr[5]	Text Mail (MIME), Microsoft Outlook Express (Windows 6, MacIntosh 5),	Y[7]	N	T	Y	Y	Y	N	MIME_Fmt

[1] Lotus AMI Pro only

[2] Lotus AMI Pro only

[3] Lotus Word Pro only

[4] Lotus Word Pro only

[5] This reader supports both clear signed and encrypted S/MIME. KeyView supports S/MIME for PST, EML, MBX, and MSG files.

[7] Text Mail only

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	Mailbox[1] (Thunderbird 1.0, Eudora 6.2)								
MCI	Microsoft Media Control Interface	N	N	Y	N	N	N	N	AIFF_Fmt, AU_Audio_Fmt, ISO_QuickTime_Fmt, MIDI_Audio_Fmt, MPEG_Audio_Fmt, MS_Video_Fmt, MS_WAVE_Audio_Fmt, Mobile_QuickTime_Fmt, QuickTime_Fmt
mdbsr	Microsoft Access (95 onwards)	Y	T	T	N	N	Y[2]	N	MS_Access_2000_Fmt, MS_Access_2007_Fmt, MS_Access_95_Fmt, MS_Access_97_Fmt, MS_Access_Fmt
mhtsr	MIME HTML (MHTML)	Y	Y	Y	N	Y	Y	N	MHT_Fmt
mifsr	Adobe FrameMaker Interchange Format (5, 5.5, 6, 7)	Y	Y	Y	N	N	Y	N	Maker_Interchange_Fmt
misr	Microsoft Word Windows (1.0, 2.0)	Y	Y	Y	N	N	N	Y	MS_Word_Win_Fmt
mp3sr	MPEG-1 Audio layer3 (ID3 v1 and v2)	M	M	Y	N	Y	N	N	MPEG_Audio_Fmt

[1] KeyView supports MBX files created by Eudora Email and Mozilla Thunderbird. MBX files created by other common mail applications are typically filtered, converted, and displayed.

[2] Charset is not supported for Microsoft Access 95 or 97.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
mpeg4sr	MPEG video	M	N	N	N	Y	N	N	Adobe_Flash_Audio_Book_Fmt, Adobe_Flash_Audio_Fmt, Adobe_Flash_Protected_Video_Fmt, Adobe_Flash_Video_Fmt, Audible_Audiobook_Fmt, ISO_3GPP2_Fmt, ISO_3GPP_Fmt, ISO_IEC_MPEG_4_Fmt, KDDI_Video_Fmt, MPEG4_AVC_Fmt, MPEG4_M4A_Fmt, MPEG4_M4B_Fmt, MPEG4_M4P_Fmt, MPEG4_M4V_Fmt, MPEG4_Sony_PSP_Fmt, MPEG_21_Fmt, NTT_MPEG4_Fmt, Nero_MPEG4_Audio_Fmt, QuickTime_Fmt, Sony_XAVC_Fmt
mppsr	Microsoft Project (2000 onwards)	Y	Y	Y	Y	Y	Y	N	MS_Project_2000_Fmt, MS_Project_2007_Fmt, MS_Project_41_Fmt, MS_Project_4_Fmt, MS_Project_98_Fmt
msgsr[1]	Microsoft Outlook (97 onwards), Documentum EMCMF	Y[2]	T[3]	Y[4]	Y	Y	Y[5]	N	EMCMF_Fmt, MS_Outlook_Fmt
mspubsr	Microsoft Publisher (98 to 2016)	Y	T	T	Y	Y	Y	N	MS_Publisher_98_Fmt, MS_Publisher_Fmt

[1] This reader supports both clear signed and encrypted S/MIME. KeyView supports S/MIME for PST, EML, MBX, and MSG files.

[2] Except Documentum EMCMF

[3] Except Documentum EMCMF

[4] For Outlook this is Text only

[5] Returns "Unicode" character set for Outlook version 2003 and up, and "Unknown" character set for previous versions.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
msw6sr	Microsoft Works Word Processor for Windows (6, 2000)	Y	Y	Y	N	N	N	Y	MS_Works_Win_WP_Fmt
mswsr	Microsoft Works Word Processor for Windows (1, 2, 3, 4)	Y	Y	Y	N	N	N	Y	MS_Works_Win_WP_Fmt
multiarcsr[1]	Compressed formats	N	N	Y[2]	Y	N	n/a	N	ARJ_Fmt, RAR5_Fmt, XZ_Fmt
mw6sr	Microsoft Word for Windows (6, 7, 8, 95)	Y	Y	Y	N	Y	Y	Y	MS_Word_95_Fmt
mw8sr	Microsoft Word (97-2004)	Y	Y	Y	Y[3]	Y	Y	Y[4]	MS_Word_2000_Fmt, MS_Word_97_Fmt
mwsr	Microsoft Word PC (4-6) and Windows Write (1-3)	Y	Y	Y	N	N	Y[5]	Y[6]	MS_Windows_Write_Fmt, MS_Word_PC_Driver_Fmt, MS_Word_PC_Fmt, MS_Word_PC_Glossary_Fmt, MS_Word_PC_Misc_Fmt, MS_Word_PC_StyleSheet_Fmt

[1] 7zip is supported with the multiarcsr reader on some platforms for Extract.

[2] 7-zip and SUN PEX archives only

[3] Supported using the embedded objects reader olesr.

[4] Microsoft Word for Windows only

[5] Microsoft Windows Write only

[6] Microsoft Word PC only

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
mwssr	Microsoft Works Spreadsheet (2, 3, 4)	Y	Y	Y	N	N	Y	N	MS_Works_DOS_SS_Fmt, MS_Works_Mac_SS_Fmt, MS_Works_Win_SS_Fmt
mwxsr	Microsoft Word XML (2007 onwards)	Y	Y	Y	Y	Y	Y	Y	MS_Word_2007_Flat_XML_Fmt, MS_Word_2007_Fmt, MS_Word_Macro_2007_Fmt
nnsr	NBI OASys Net Archive	Y	T	T	N	N	N	N	NBI_Net_Archive_Fmt
nsfsr	IBM Lotus Notes database (4, 5, 6.0, 6.5, 7.0, 8.0)	N	N	Y	Y	Y	N	N	Lotus_Notes_NSF_Fmt
oa2sr	Fujitsu Oasys (7)	Y	Y	Y	N	P	N	N	Oasys_Fmt
odfsssr	OASIS Open Document Format (1, 2[1])	Y	Y	Y	Y[2]	Y	Y	N	ODF_Spreadsheet_Fmt, ODF_Spreadsheet_Template_Fmt
odfwpsr	OASIS Open Document Format (1, 2[3])	Y	Y	Y	Y[4]	Y	Y	Y	ODF_Text_Fmt, ODF_Text_Template_Fmt, SO_Text_XML_Fmt
olesr	Windows Scrap File	N	N	N	Y	N	n/a	N	OLE_Fmt, Scrap_Fmt, Windows_Installer_Fmt

[1] Generated by OpenOffice Calc 2.0, StarOffice 8 Calc, and IBM Lotus Symphony Spreadsheet 3.0.

[2] Supported using the embedded objects reader olesr.

[3] Generated by OpenOffice Writer 2.0, StarOffice 8 Writer, and IBM Lotus Symphony Documents 3.0.

[4] Supported using the embedded objects reader olesr.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
olmsr	Microsoft Outlook for Macintosh (2011)	N	N	Y	Y	N	Y	N	MS_OutlookOLM_Fmt
onealtsr	Microsoft OneNote Alternative Packaging Format (2007 onwards)	Y	T	T	Y	N	N	N	OneNote_Alternate_Fmt
onmsr	Legato Extender	N	N	Y	Y	Y	N	N	Legato_Extender_ONM_Fmt
oo3sr	Omni Outliner (v3, OPML, OOutline)	Y	Y	Y	N	N	Y	N	OO3_Fmt, OOUTLINE_Fmt, OPML_Fmt
pbixsr	Microsoft Power BI Desktop (1.11)	Y	T	T	N	N	Y	N	MS_Power_BI_Fmt
pdf2sr	Adobe PDF (1.1 to 1.7, 2.0)	N	Y	N	N	N	N	N	PDF_Fmt
pdfsr	Adobe PDF (1.1 to 1.7, 2.0)	Y	Y	N	Y[1]	Y	Y	N	PDF_Fmt, Portfolio_PDF_Fmt
pffsr[2]	Microsoft Outlook Offline Storage File (97 onwards)	N	N	Y	Y	Y	Y	N	MS_OutlookOST_Fmt

[1] Includes support for extraction of subfiles from PDF Portfolio documents.

[2] The reader pffsr is available only on Windows and Linux.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
pfilesr	Rights Management Services (RMS)-protected format	Y[1]	T[2]	T[3]	N	Y	N	N	RMS_Protected_Fmt
pngsr	Portable Network Graphics	M	M	N	N	Y	N	N	PNG_Fmt
psdsr	Adobe Photoshop	N	N	N	N	Y[4]	N	N	PSD_Fmt
pstnsr	Microsoft Outlook Personal Folder[5] (97 onwards)	N	N	Y	Y	Y	Y	N	MS_OutlookPST_Fmt
pstsr[6]	Microsoft Outlook Personal Folder[7] (97 onwards)	N	N	Y	Y	Y	N	N	MS_OutlookPST_Fmt

[1] KeyView filters only the internal redirection text. The underlying document text is not accessible without the decryption key.

[2] KeyView filters only the internal redirection text. The underlying document text is not accessible without the decryption key.

[3] KeyView filters only the internal redirection text. The underlying document text is not accessible without the decryption key.

[4] Only XMP metadata is extracted for this format.

[5] KeyView provides several readers capable of processing PST files. The pstsr reader uses the Microsoft Messaging Application Programming Interface (MAPI), works only on Windows, and requires that you have Microsoft Outlook installed. The pstxsr reader is available for Windows (32-bit and 64-bit) and Linux (64-bit only) and does not require Microsoft Outlook. The pstnsr reader is an alternative reader that does not require Microsoft Outlook, for all platforms not supported by pstxsr.

[6] This reader supports both clear signed and encrypted S/MIME. KeyView supports S/MIME for PST, EML, MBX, and MSG files.

[7] KeyView provides several readers capable of processing PST files. The pstsr reader uses the Microsoft Messaging Application Programming Interface (MAPI), works only on Windows, and requires that you have Microsoft Outlook installed. The pstxsr reader is available for Windows (32-bit and 64-bit) and Linux (64-bit only) and does not require Microsoft Outlook. The pstnsr reader is an alternative reader that does not require Microsoft Outlook, for all platforms not supported by pstxsr.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
pstxsr	Microsoft Outlook Personal Folder[1] (97 onwards)	N	N	Y	Y	Y	Y	N	MS_OutlookPST_Fmt
pwsr	PRIMEWORD	Y	T	T	N	N	N	N	PRIMEWORD_Fmt
qpssr	Corel Quattro Pro (5, 6, 7, 8)	Y	Y	Y	N	P	Y	N	Quattro_Pro_Win_Fmt
qpwsr	Corel Quattro Pro (X4)	Y	N	Y	N	P	Y	N	QPW_Fmt
rarsr	RAR archive (2.0 through 3.5)	N	N	N	Y	N	n/a	N	RAR_Fmt
riffsr	Microsoft Wave Sound	M	N	N	N	Y	N	N	MS_WAVE_Audio_Fmt
rtfsr	Rich Text Format (1 through 1.7)	Y	Y	Y	N	P	Y	Y	MS_Pocket_Word_Fmt, MS_RTF_Fmt
skypesr	Skype Log (3)	Y	Y	Y	N	N	N	N	Skype_Fmt
sosr	OpenOffice, LibreOffice(1-5), StarOffice (6-9)	Y	T	T	N	Y	Y	N	SO_Spreadsheet_XML_Fmt
starcsr	StarOffice Calc (3, 4, 5)	Y	T	T	N	N	N	N	SO_Spreadsheet_Fmt

[1] KeyView provides several readers capable of processing PST files. The pstsr reader uses the Microsoft Messaging Application Programming Interface (MAPI), works only on Windows, and requires that you have Microsoft Outlook installed. The pstxsr reader is available for Windows (32-bit and 64-bit) and Linux (64-bit only) and does not require Microsoft Outlook. The pstnsr reader is an alternative reader that does not require Microsoft Outlook, for all platforms not supported by pstxsr.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
starwsr	StarOffice Writer (3, 4, 5)	Y	T	T	N	N	N	N	SO_Text_Fmt
stringssr	Generic 'strings' reader	Y	T	T	N	N	N	N	BeagleWorks_Word_Fmt, CEOwrite_Fmt, CPT_Comm_Fmt, CWK_Fmt, DG_CDS_Fmt, DSA101_Fmt, Data_Point_VistaWord_Fmt, Enable_WP_Fmt, GreatWorks_Word_Fmt, HP_Word_PC_Fmt, IBM_DCF_Script_Fmt, IBM_Writing_Assistant_Fmt, Lotus_Notes_CDF_Fmt, Lyrix_Fmt, MASS_11_Fmt, MS_Works_DOS_WP_Fmt, MS_Works_Mac_WP_Fmt, MacWrite_Fmt, MacWrite_II_Fmt, Multimate_Adv_Fmt, Multimate_Adv_Fnote_Fmt, Multimate_Adv_II_Fmt, Multimate_Adv_II_Fnote_Fmt, Multimate_Fmt, Multimate_Fnote_Fmt, Navy_DIF_Fmt, ODA_Q1_11_Fmt, ODA_Q1_12_Fmt, Office_Writer_Fmt, Psion_TextEd_Fmt, Psion_Word_3_Fmt, Psion_Word_Fmt, Q_A_DOS_Fmt, Q_A_Win_Fmt, Quadratron_Q_One_v1_Fmt, Quadratron_Q_One_v2_Fmt, Quickword_Fmt, SAMNA_Word_IV_Fmt, Symbol_Dynamics_EXP_Fmt, Targon_Word_Fmt, Uniplex_WP_Fmt, Volkswriter_Fmt, WANG_WITA_Fmt, WANG_WPS_Comm_Fmt, WPS_PLUS_Fmt, WordERA_Fmt, WordMARC_Fmt, WordPerfect_Fmt, WordStar_2000_Fmt, WordStar_Fmt, WordStar_for_Windows_Fmt, Word_

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
									Connection_Fmt, WriteNow_Fmt, Xerox_860_Comm_Fmt, Xerox_Writer_Fmt
swfsr	Macromedia Flash (through 8.0)	Y	Y	Y	N	N	Y[1]	N	Macromedia_Flash_Fmt
swsr	Informix SmartWare II Word Processor	Y	T	T	N	N	N	N	SmartWare_II_WP_Fmt
tarsr	TAR Tape Archive	N	N	Y	Y	N	n/a	N	TAR_Fmt
tifsr	TIFF Tagged Image File (through 6.0[2])	M	M	N	N	Y	N	N	TIFF_Fmt
tnefsr	Transport Neutral Encapsulation Format	N	N	Y	Y	Y	Y	N	TNEF_Fmt
unihtmsr	Unicode HTML	Y	Y	Y	N	Y	Y	N	Unicode_HTML_Fmt
unisr	Unicode Text (3, 4)	Y	Y	Y	N	N	Y	N	Unicode_Fmt
unzip	PKZIP/Zip Compression	N	N	Y[3]	Y	N	n/a	N	Executable_JAR_Fmt, KMZ_Fmt, ODF_Formula_Fmt, ODF_Formula_Template_Fmt, PKZIP_Fmt, Tableau_Packaged_

[1] The character set cannot be determined for versions 5.x and lower.

[2] The following compression types are supported: no compression, CCITT Group 3 1-Dimensional Modified Huffman, CCITT Group 3 T4 1-Dimensional, CCITT Group 4 T6, LZW, JPEG (only Gray, RGB and CMYK color space are supported), and PackBits.

[3] PKZIP, WinZip, and Java Archive only

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
									Data_Source_Fmt, Tableau_Packaged_Workbook_Fmt
uudsr	UU-Encoding (all versions)	N	N	Y	Y	N	n/a	N	UUEncoded_Fmt
vcfsr	Microsoft Outlook vCard Contact (2.1, 3.0, 4.0)	Y	Y	T	N	Y	N	N	VCF_Fmt
vsdsr	Microsoft Visio (4, 5, 2000, 2002, 2003, 2007, 2010[1])	Y	Y	Y	Y[2]	Y	Y	N	MS_Visio_Fmt
wkssr	Lotus 1-2-3 (2, 3, 4, 5)	Y	Y	Y	N	N	Y	N	Lotus_123_Worksheet_Fmt
wosr	Corel WordPerfect Windows (5, 5.1)	Y	Y	Y	N	P	Y	Y	WordPerfect_5_Fmt
wp6sr	Corel WordPerfect (6 onwards)	Y	Y	Y	N	P	Y	N	WordPerfect_6_Fmt
wpmsr	Corel WordPerfect Macintosh (1.02, 2, 2.1, 2.2, 3, 3.1)	Y	Y	Y	N	N	Y	N	WordPerfect_Mac_Fmt
xlsbsr	Microsoft Excel	Y	Y	Y	N	Y	N	N	MS_Excel_Binary_2007_Fmt

[1] Viewing and Export use the graphic reader, kpVSD2rdr for Microsoft Visio 2003, 2007, and 2010, and vsdsr for all earlier versions. Image fidelity in Viewing and Export is therefore only supported for versions 2003 and above. Filter uses the graphic reader kpVSD2rdr for Microsoft Visio 2003, 2007, and 2010, and vsdsr for all earlier versions.
[2] Extraction of embedded OLE objects is supported for Filter on Windows platforms only.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	Binary Format (2007 onwards)								
xlssr	Microsoft Excel (2.2 to 2004)	Y	Y	Y	Y[1]	Y	Y	Y[2]	Excel_2000_Fmt, Excel_95_Fmt, Excel_97_Fmt, Excel_Chart_Fmt, Excel_Fmt, Excel_Macro_Fmt
xlsxsr	Microsoft Excel Windows XML (2007 onwards)	Y	Y	Y	Y	Y	Y	Y	MS_Excel_2007_Fmt, MS_Excel_Macro_2007_Fmt
xmlsr	XML	Y	T	T	N	Y	Y	N	AMF_Fmt, Adobe_XML_Data_Package_Fmt, Atom_Syndication_Fmt, CDXML_Fmt, Chemical_Markup_Language_Fmt, Collada_DAE_Fmt, ESzigno_Fmt, FictionBook_Fmt, Grasshopper_GHX_Fmt, JNLP_Fmt, JavaView_JVX_Fmt, KML_Fmt, MARC_XML_Fmt, METS_Fmt, MODS_Fmt, MS_Excel_XML_Fmt, MS_Management_Pack_MPX_Fmt, MS_Visio_XML_Fmt, MS_Word_XML_Fmt, MXML_Fmt, Metalink_Fmt, Mozilla_XUL_Fmt, MusicXML_Fmt, Open_Diagnostic_Data_Exchange_Fmt, Open_eBook_Fmt, PDF_XML_Forms_Data_Fmt, PLS_Fmt, RDF_XML_Fmt, RSS_Fmt, Really_Simple_Discovery_Fmt, SBML_Fmt, SMIL_Fmt, SPARQL_Results_Fmt, SRGS_Fmt, SRU_Fmt,

[1] Supported using the embedded objects reader olesr.

[2] Microsoft Excel for Windows only

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
									SSML_Fmt, SVG_Fmt, SyncML_Fmt, TEI_Fmt, Tableau_Data_Source_Fmt, Tableau_Map_Source_Fmt, Tableau_Preferences_Fmt, Tableau_Workbook_Fmt, VTK_XML_Fmt, VoiceXML_Fmt, WML_Fmt, Windows_Audio_Playlist_Fmt, XAML_Browser_Application_Fmt, XDF_Fmt, XML_Fmt, XML_Shareable_Playlist_Fmt, XSLT_Fmt, YIN_Fmt
xpssr	Microsoft XML Paper Specification	Y	T	T	N	N	N	N	MS_XPS_Fmt
xywsr	XyWrite / Nota Bene (4.12)	Y	Y	Y	N	N	N	N	XyWrite_Fmt
yimsr[1]	Yahoo! Instant Messenger	Y	Y	Y	N	N	N	N	YIM_Fmt
z7zsr	7-Zip archive (4.57)	N	N	Y	Y	N	n/a	N	Z7Z_Fmt

[1] To successfully use this reader, you must set the KV_YAHOO_ID environment variable to the Yahoo user ID. You can optionally set the KV_OTHER_YAHOO_ID environment variable to the other Yahoo user ID. If you do not set it, "Other" is used by default. If you enter incorrect values for the environment variables, erroneous data is generated.

Appendix C: KeyView Classes

The following table lists KeyView file classes. The numbers are reported in the `DocumentClass` field in documents processed by CFS. Consult the table to determine the class of the file that was imported.

File Classes

Attribute Number	Description
0	No file class
01	Word processor
02	Spreadsheet
03	Database
04	Raster image
05	Vector graphic
06	Presentation
07	Executable
08	Encapsulation
09	Sound
10	Desktop publishing
11	Outline/planning
12	Miscellaneous
13	Mixed format
14	Font
15	Time scheduling
16	Communications
17	Object module
18	Library module
19	Fax
20	Movie

File Classes, continued

Attribute Number	Description
21	Animation
22	Source Code
23	Computer-Aided Design
24	BI and analysis tools
25	Scientific data
26	Geographic Info System

Appendix D: Document Fields

This appendix describes the standard fields that Connectors and CFS add to documents before the documents are indexed into IDOL Server.

- Document Fields
- AUTN_IDENTIFIER

Document Fields

The following fields are added to a document by connectors:

Field	Description
`AUTN_IDENTIFIER`	An identifier that allows a connector to extract the document from the repository again, for example during the collect or view actions. For more information about the identifier, see AUTN_IDENTIFIER, on page 212.
`DocTrackingId`	An identifier used for document tracking functionality.
`DREREFERENCE`	A reference for the document. This is the standard IDOL reference field, which is used for deduplication.
`source_connector_run_id`	(Added only when `IngestSourceConnectorFields=TRUE`). The asynchronous action token of the `fetch` action that ingested the document.
`source_connector_server_id`	(Added only when `IngestSourceConnectorFields=TRUE`). A token that identifies the instance of the connector that retrieved the document (different installations of the same connector populate this field with different IDs). You can retrieve the UID of a connector through `action=GetVersion`.

The following fields are added to a document during ingestion:

Field	Description
`DocumentAttributes`	KeyView document attributes.
`DocumentClass`	The KeyView document class. Consult the KeyView Classes table to determine the file class that was imported.

Field	Description
`DocumentSize`	The size of the document.
`DocumentType`	A number that represents the program that created the file format. Consult the KeyView Supported Formats table to determine the file type that was imported.
`DRECHILDCOUNT`	The number of sub-files that the document contains.
`DREDBNAME`	The name of the IDOL database that the document must be indexed to.
`DREFILENAME`	The file name of the original document.
`DREORIGINALNAME`	The original file name passed to CFS. This is the full path for extracted sub-files.
`DREROOTFAMILYREFERENCE`	The parent document for the family of documents.
`DREROOTFAMILYREFERENCE_ID`	A unique hash for the family of documents.
`FAMILYSORT`	A field used to track families (that is, containers) of documents. It contains a hash unique to the family, with indices appended that describe the depth and number of attachments.
`ImportErrorDescription`	If an error occurs when a document is processed, a description of the error is written to this field.
`ImportMagicExtension`	The file extension of the detected document type.
`ImportOriginalEncoding`	The detected encoding used by the document.
`ImportVersion`	Internal version number.
`InfoFlag`	A KeyView Flag that describes the file type (External, Embedded and so on). 0 = default 1 = This sub file needs further extraction 2 = This sub file is protected 4 = This sub file is an external file 8 = This sub file is a mail item attachment 16 = This sub file is SMIME protected
`KeyviewVersion`	The version of KeyView that CFS was released with.
`UUID`	A unique identifier for the document.
`VersionNumber`	The version of CFS that was used to import the document.

AUTN_IDENTIFIER

An *Identifier* is a base-64 encoded string that identifies the source of a document in IDOL Server. When you use a connector to index documents into IDOL Server, an identifier is added to every document, in the `AUTN_IDENTIFIER` document field.

A connector can use the identifier to extract the original file from the repository. An application might need to extract the original file when presenting the results of a query. The application can request the file by sending a `collect` or `view` action to the connector.

The exact content of the `AUTN_IDENTIFIER` field depends on the connector that retrieved the document, but contains information such as:

- The document reference. The document reference identifies an item in a repository. For the files retrieved from the same repository, a reference is unique. For files retrieved by a File System Connector, the document reference is the path to the file. For e-mail messages retrieved by an Exchange Connector, the document reference includes the name of the message store and folder that contains the message.
- Additional information used to find the document in the repository. Though the document reference identifies a file in the repository, it might not provide sufficient information to retrieve it efficiently. The identifier can include additional information to assist the connector locate the document.
- The name of the fetch task that was used to retrieve the document. When a connector needs to retrieve a file, it can use the same settings by finding the fetch task in its configuration file.

An example identifier appears below:

```
<id section="MyTask1" reference="http://myserver:4567/doc/_vxswdfguhjknbio_
earycqzt_">
  <param name="SERVICEURL" value="http://myserver:4567/service"/>
  <param name="DOCID">_vxswdfguhjknbio_earycqzt_</param>
</id>
```

Sub File Indexes

Documents in IDOL Server can represent sub-files. In these documents, the `AUTN_IDENTIFIER` field contains the identifier of the container file.

To retrieve a sub-file from a repository, a connector must retrieve the container file and send it to KeyView so that the sub-file can be extracted. So that KeyView can extract the correct sub-file, the identifier must include a sub-file index.

When CFS indexes documents into IDOL Server, sub-file indexes are automatically written to the `SubFileIndexCSV` document field. For example:

```
SubFileIndexCSV="1"
```

> **NOTE:** Your connector must be configured with `EnableExtraction=true`. The connector's

> `KeyviewDirectory` parameter must also be set.

The sub-file index in this example (`1`) indicates that the document represents the second file in the container (the sub-files are indexed from 0).

Container files can contain other container files (for example an e-mail message file could contain ZIP file attachments, containing further sub-files). In this case, the sub-file index might include more than one level:

```
SubFileIndexCSV="2,6"
```

A sub-file index of `2,6` indicates that the document represents the seventh file in the third container, in the original container file.

When an action is sent to a connector to retrieve sub-files, the sub-file index must be appended to the identifier of the container. For example:

```
PGlkIHM9Ik15VGFzazEiIHI9Imh0dHA6Ly9teXNlcnZlcjo0NTY3L2RvYy9fdnhzd
2RmZ3VoamtuYmlvX2VhcnljcXp0XyI+PHAgbj0iU0VSVklDRVVSTCIgdj0iaHR0cD
ovL215c2VydmVyOjQ1Njcvc2VydmljZSIvPjxwIG49IkRPQ0lEIiB2PSJfdnhzd2R
mZ3VoamtuYmlvX2VhcnljcXp0XyIvPjwvaWQ+|2.6
```

> **NOTE:** Where sub-file indexes have multiple levels (for example `SubFileIndexCSV="2,6"`, the comma must be replaced by a period).

Append Sub File Indexes to the Document Identifier

You can configure CFS to automatically append sub-file indexes to document identifiers, before the documents are indexed into IDOL Server.

To do this, use the lua script `identifiers.lua`, which is included with CFS in the `scripts` folder. The script is also included below:

```
function handler( document )
   identifier = document:getFieldValue( "AUTN_IDENTIFIER" )

   if identifier then
      indices = document:getFieldValue( "SubFileIndexCSV" )

      if indices then
         indices = string.gsub(indices, ",", ".")
         document:setFieldValue("AUTN_IDENTIFIER", identifier .. "|" .. indices)
      end
   end

   return true
end
```

You must run the script after KeyView has extracted sub-files, so run the script using a *Post* Import task. For example:

```
[ImportTasks]
Post0=Lua:scripts/identifiers.lua
```

Appendix E: Debug Your Lua Scripts

The CFS installation includes files that you can use to debug your Lua scripts with an IDE such as the ZeroBrane Studio Lua IDE. You can use the IDOL and CFS Lua functions through the IDE, with code-completion.

To debug Lua scripts in the ZeroBrane Studio IDE

1. Install ZeroBrane Studio Lua IDE.
2. Copy the files from the `lua_ide_files/ZeroBraneStudio` directory in your CFS installation to the ZeroBrane Studio installation directory, merging them with the files that are already there:

Source (CFS Installation)	Destination
`lua_ide_files/ZeroBraneStudio/api/lua/*`	*`ZeroBraneInstall`*`/api/lua/`
`lua_ide_files/ZeroBraneStudio/bin/*`	*`ZeroBraneInstall`*`/bin/`
`lua_ide_files/ZeroBraneStudio/interpreters/*`	*`ZeroBraneInstall`*`/interpreters/`

3. Open the ZeroBrane Studio IDE.
4. On the **Project** menu, click **Lua Interpreter** > **Lua 5.3 64bit**.
5. On the **Project** menu, click **Project Directory** > **Choose**, and then select a directory to use as the project directory.

> **TIP:** Any relative paths in your Lua scripts are relative to the project directory that you choose.

> **NOTE:** The project directory must include:
>
> - `autn_lua.dll`
> - `autn_lua_cfs.dll`
> - A CFS configuration file. Some of the Lua functions that you can use in your scripts write to log files or use KeyView and the configuration file specifies the paths of these files.
>
> The libraries are available in the `lua` directory of your CFS installation. You can use that folder as your project directory.

An example CFS configuration that is suitable for debugging Lua scripts appears below.

```
[License]
LicenseServerHost=10.0.0.1
LicenseServerACIPort=20000
```

```
[Logging]
LogLevel=NORMAL
0=ApplicationLogStream
1=ImportLogStream

[ApplicationLogStream]
LogTypeCSVs=application
LogFile=application.log

[ImportLogStream]
LogTypeCSVs=import
LogFile=import.log

[ImportService]
KeyviewDirectory=../filters
ExtractDirectory=temp
ThreadCount=3
ImportInheritFieldsCSV=AUTN_GROUP,AUTN_
IDENTIFIER,DREDBNAME,AUTONOMYMETADATA,SECURITYTYPE
```

> **CAUTION:** If you use your CFS configuration file, the Lua scripts that you are debugging can write to the CFS log files, or even index documents into your IDOL Content component.

6. Create a new Lua script file in the project directory. Start your script with the following lines, to provide the path of the configuration file and include the IDOL and CFS Lua functions:

```
-- Configuration file provides paths to log files, KeyView
require_config = "my_config.cfg"
require("autn_lua_cfs").as_global()
require("autn_lua").as_global()
```

7. You can now begin writing your script, using all of the standard IDOL and CFS Lua functions. CFS pre- and post- tasks call a function named `handler` that is passed a document object, so you might want to construct a document and then call the function. For example:

```
-- Configuration file provides paths to log files, KeyView
require_config = "my_config.cfg"
require("autn_lua_cfs").as_global()
require("autn_lua").as_global()



function handler(document)
    print(document:to_xml())
    return true
end
```

```
-- Parse a document from an IDX file to be used as input
local document = parse_document_idx("input.idx", true)

-- Call the handler, storing the return value (expected to be Boolean)
local result = handler(document)

-- Print the result
print("handler returned:", result)
```

8. Run and debug the script as necessary using the ZeroBrane Studio IDE. For more information about using the ZeroBrane Studio IDE, refer to the ZeroBrane documentation.

Glossary

A

ACI (Autonomy Content Infrastructure)
A technology layer that automates operations on unstructured information for cross-enterprise applications. ACI enables an automated and compatible business-to-business, peer-to-peer infrastructure. The ACI allows enterprise applications to understand and process content that exists in unstructured formats, such as email, Web pages, Microsoft Office documents, and IBM Notes.

ACI Server
A server component that runs on the Autonomy Content Infrastructure (ACI).

ACL (access control list)
An ACL is metadata associated with a document that defines which users and groups are permitted to access the document.

action
A request sent to an ACI server.

active directory
A domain controller for the Microsoft Windows operating system, which uses LDAP to authenticate users and computers on a network.

C

Category component
The IDOL Server component that manages categorization and clustering.

Community component
The IDOL Server component that manages users and communities.

connector
An IDOL component (for example File System Connector) that retrieves information from a local or remote repository (for example, a file system, database, or Web site).

Connector Framework Server (CFS)
Connector Framework Server processes the information that is retrieved by connectors. Connector Framework Server uses KeyView to extract document content and metadata from over 1,000 different file types. When the information has been processed, it is sent to an IDOL Server or Distributed Index Handler (DIH).

Content component
The IDOL Server component that manages the data index and performs most of the search and retrieval operations from the index.

D

DAH (Distributed Action Handler)
DAH distributes actions to multiple copies of IDOL Server or a component. It allows you to use failover, load balancing, or distributed content.

DIH (Distributed Index Handler)
DIH allows you to efficiently split and index extremely large quantities of data into multiple copies of IDOL Server or the Content component. DIH allows you to create a scalable solution that delivers high performance and high availability. It provides a flexible way to batch, route, and categorize the indexing of internal and external content into IDOL Server.

I

IDOL

The Intelligent Data Operating Layer (IDOL) Server, which integrates unstructured, semi-structured and structured information from multiple repositories through an understanding of the content. It delivers a real-time environment in which operations across applications and content are automated.

IDOL Proxy component

An IDOL Server component that accepts incoming actions and distributes them to the appropriate subcomponent. IDOL Proxy also performs some maintenance operations to make sure that the subcomponents are running, and to start and stop them when necessary.

Intellectual Asset Protection System (IAS)

An integrated security solution to protect your data. At the front end, authentication checks that users are allowed to access the system that contains the result data. At the back end, entitlement checking and authentication combine to ensure that query results contain only documents that the user is allowed to see, from repositories that the user has permission to access. For more information, refer to the IDOL Document Security Administration Guide.

K

KeyView

The IDOL component that extracts data, including text, metadata, and subfiles from over 1,000 different file types. KeyView can also convert documents to HTML format for viewing in a Web browser.

L

LDAP

Lightweight Directory Access Protocol. Applications can use LDAP to retrieve information from a server. LDAP is used for directory services (such as corporate email and telephone directories) and user authentication. See also: active directory, primary domain controller.

License Server

License Server enables you to license and run multiple IDOL solutions. You must have a License Server on a machine with a known, static IP address.

O

OmniGroupServer (OGS)

A server that manages access permissions for your users. It communicates with your repositories and IDOL Server to apply access permissions to documents.

P

primary domain controller

A server computer in a Microsoft Windows domain that controls various computer resources. See also: active directory, LDAP.

V

View

An IDOL component that converts files in a repository to HTML formats for viewing in a Web browser.

W

Wildcard

A character that stands in for any character or group of characters in a query.

X

XML

Extensible Markup Language. XML is a language that defines the different attributes of document content in a format that can be read by humans and machines. In IDOL Server, you can index documents in XML format. IDOL Server also returns action responses in XML format.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Administration Guide (Micro Focus Connector Framework Server 12.7)

Add your feedback to the email and click **Send**.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to swpdl.idoldocsfeedback@microfocus.com.

We appreciate your feedback!