



Hyland Document Filters v25.4

Implementation Guide

November 2025

Table of contents

1. Getting started	10
1.1 Getting started with Document Filters	10
1.2 Getting Started with .NET	14
1.3 Getting Started with C	18
1.4 Getting Started with C++	26
1.5 Getting Stated with COM	35
1.6 Getting Stated with Java	36
1.7 Getting Stated with Python	42
1.8 About accessibility info extraction	46
1.9 About conversion profiles	47
1.10 About custom streams and extended streams	49
1.11 About File Type Identification	53
1.12 About Fonts	56
1.13 About Multi-Part Archives	61
1.14 About Multithreading	62
1.15 About Optical Character Recognition (OCR)	63
1.16 Content Enrichment	73
1.17 Output Formats	79
2. Tutorials	109
2.1 Tutorials	109
2.2 Opening Files	113
2.2.1 How do I open a document from disk?	113
2.2.2 How do I open a document from memory?	115
2.3 Converting Files	117
2.3.1 How do I convert a document to a PDF file?	117
2.3.2 How do I convert a document to Classic HTML?	119
2.3.3 How do I convert a document to JSON in Hi-Def mode?	121
2.3.4 How do I convert a document to Markdown in Hi-Def mode?	123

2.3.5	How do I convert a document to Markdown in Text mode?	125
2.3.6	How do I convert a document to paginated HiDef HTML?	127
2.3.7	How do I convert a document to PNG images?	128
2.3.8	How do I convert a document to Structured XML?	130
2.3.9	How do I convert a PDF to Markdown with Table Detection?	131
2.3.10	How do I convert HTML to Hi-Def?	134
2.4	Extracting Files	138
2.4.1	How do I extract metadata from a document?	138
2.4.2	How do I extract sub-documents from documents and archives?	140
2.4.3	How do I extract text and metadata from a document?	142
2.4.4	How do I handle password-protected files?	144
2.5	Other	146
2.5.1	How do I capture log messages?	146
2.5.2	How do I compare documents?	148
2.5.3	How do I create a barcode?	150
2.5.4	How do I integrate with a custom OCR engine?	151
2.5.5	How do I localize metadata?	156
3.	Reference	158
3.1	Overview	158
3.2	C reference	159
3.2.1	C reference	159
3.2.2	Functions	175
3.2.3	Structures	405
3.2.4	Constants	468
3.3	Object reference	553
3.3.1	Overview	553
3.3.2	Annotation	559
3.3.3	AnnotationAction	577
3.3.4	AnnotationActionGoTo	580
3.3.5	AnnotationActionGoToRemote	585

3.3.6	AnnotationActionNamed	587
3.3.7	AnnotationActionUri	589
3.3.8	AnnotationAppearanceStream	591
3.3.9	AnnotationAppearanceStreams	595
3.3.10	AnnotationAztec	599
3.3.11	AnnotationBarcode	600
3.3.12	AnnotationBorderStyle	609
3.3.13	AnnotationCode128	614
3.3.14	AnnotationCode39	615
3.3.15	AnnotationDataMatrix	616
3.3.16	AnnotationDefaultAppearance	617
3.3.17	AnnotationEllipse	621
3.3.18	AnnotationFreeText	625
3.3.19	AnnotationGS1_128	629
3.3.20	AnnotationHighlight	633
3.3.21	AnnotationInk	634
3.3.22	AnnotationLine	637
3.3.23	AnnotationLineEndings	642
3.3.24	AnnotationLink	647
3.3.25	AnnotationNamedDestination	652
3.3.26	AnnotationNote	655
3.3.27	AnnotationNoteList	660
3.3.28	AnnotationPDF417	666
3.3.29	AnnotationPoint	667
3.3.30	AnnotationPointList	670
3.3.31	AnnotationPointListList	676
3.3.32	AnnotationPolygon	682
3.3.33	AnnotationPolyline	686
3.3.34	AnnotationPopup	690
3.3.35	AnnotationQrCode	693

3.3.36	AnnotationRect	694
3.3.37	AnnotationRectangle	699
3.3.38	AnnotationSquiggly	703
3.3.39	AnnotationStamp	704
3.3.40	AnnotationStickyNote	707
3.3.41	AnnotationStrikeout	710
3.3.42	AnnotationTextMarkup	711
3.3.43	AnnotationUnderline	713
3.3.44	Bookmark	714
3.3.45	Canvas	733
3.3.46	CompareDocumentSettings	778
3.3.47	CompareDocumentSource	782
3.3.48	CompareResultDifference	784
3.3.49	CompareResultDifferenceDetail	790
3.3.50	CompareResults	794
3.3.51	CompareSettings	797
3.3.52	DocumentFilters	800
3.3.53	Extractor	824
3.3.54	FormElement	926
3.3.55	FormElementOption	947
3.3.56	FormKeyValue	951
3.3.57	Hyperlink	955
3.3.58	IGRFormat	973
3.3.59	IGROption	980
3.3.60	Page	989
3.3.61	PageElement	1032
3.3.62	RenderPageProperties	1042
3.3.63	SubFile	1051
3.3.64	Word	1070

3.4	Structured XML	1084
3.4.1	Overview	1084
3.4.2	<doc> element	1085
3.4.3	<head> element	1086
3.4.4	<meta> element	1087
3.4.5	<page> element	1088
3.4.6	<header>/<footer> elements	1091
3.4.7	<section> element	1093
3.4.8	<column> element	1094
3.4.9	<p> element	1096
3.4.10	<line> element	1099
3.4.11	<run> element	1100
3.4.12	<word> element	1103
3.4.13	<float> element	1104
3.4.14	 element	1106
3.4.15	<table> element	1107
3.4.16	<tr> element	1109
3.4.17	<td> element	1111
4.	Tech Info	1113
4.1	Supported platforms	1113
4.2	Platform Requirements	1114
4.2.1	Windows	1114
4.2.2	Linux	1114
4.2.3	Alpine Linux	1115
4.2.4	FreeBSD	1116
4.2.5	Solaris	1116
4.2.6	macOS	1118
4.2.7	IBM AIX	1118
4.3	Platform Differences	1120

4.4 Supported File Formats	1121
4.4.1 Archive	1121
4.4.2 CAD	1123
4.4.3 Database	1124
4.4.4 Email and Messaging	1125
4.4.5 Multimedia	1127
4.4.6 Other	1129
4.4.7 Presentation	1131
4.4.8 Raster Image	1132
4.4.9 Spreadsheets	1136
4.4.10 Text and Markup	1138
4.4.11 Vector Image	1140
4.4.12 Word Processor and Office	1141
4.5 Document format codes	1146
4.6 Processing Options	1163
4.7 Environment Variables	1204
4.8 Third-Party Libraries	1206
4.9 Known Issues	1268
5. Releases	1270
5.1 Releases	1270
5.2 Document Filters 25.4.0 (Nov 2025)	1273
5.3 Document Filters 25.3.0 (Aug 2025)	1275
5.4 Document Filters 25.2.0 (May 2025)	1278
5.5 Document Filters 25.1.0 (Feb 2025)	1281
5.6 Document Filters 24.4.0 (Nov 2024)	1283
5.7 Document Filters 24.3.0 (Aug 2024)	1285
5.8 Document Filters 24.2.0 (May 2024)	1288
5.9 Document Filters 24.1.0 (Feb 2024)	1291
5.10 Document Filters 23.3.0 (Sep 2023)	1294
5.11 Document Filters 23.2.0 (May 2023)	1298

5.12	Document Filters 23.1.0 (Feb 2023)	1300
5.13	Document Filters 22.4.0 (Nov 2022)	1303
5.14	Document Filters 22.3.0 (Aug 2022)	1306
5.15	Document Filters 22.2.0 (May 2022)	1309
5.16	Document Filters 22.1.0 (Feb 2022)	1312
5.17	Document Filters 21.11.0 (Nov 2021)	1314
5.18	Document Filters 21.8.0 (Aug 2021)	1316
5.19	Document Filters 21.5.0 (May 2021)	1318
5.20	Document Filters 21.2.0 (Feb 2021)	1321
5.21	Document Filters 11.4.20 (Nov 2020)	1323
5.22	Document Filters 11.4.19 (Jul 2020)	1325
5.23	Document Filters 11.4.18 (May 2020)	1328
5.24	Document Filters 11.4.17 (Feb 2020)	1330
5.25	Document Filters 11.4.16 (December 2019)	1332
5.26	Document Filters 11.4.15 (September 2019)	1334
5.27	Document Filters 11.4.14 (June 2019)	1336
5.28	Document Filters 11.4.13 (March 2019)	1338
5.29	Document Filters 11.4 Build 3054 (November 2018)	1340
5.30	Document Filters 11.4 Build 2990 (September 2018)	1342
5.31	Document Filters 11.4 Build 2934 (July 2018)	1343
5.32	Document Filters 11.4 Build 2878 (May 2018)	1345
5.33	Document Filters 11.4 Build 2822 (March 2018)	1346
5.34	Document Filters 11.4 Build 2766 (January 2018)	1348
5.35	Document Filters 11.4 Build 2710 (November 2017)	1349
5.36	Document Filters 11.4 Build 2647 (September 2017)	1352
5.37	Document Filters 11.4 Build 2600 (August 2017)	1354
5.38	Document Filters 11.4 Build 2581 (July 2017)	1355
5.39	Document Filters 11.4 Build 2543 (June 2017)	1356
5.40	Document Filters 11.4 Build 2452 (March 2017)	1357
5.41	Document Filters 11.4 Build 2480 (April 2017)	1360

5.42	Document Filters 11.3 Build 2400 (January 2017)	1362
5.43	Document Filters 11.3 Build 2356 (December 2016)	1364
5.44	Document Filters 11.3 Build 2322 (November 2016)	1366
5.45	Document Filters 11.3 Build 2228 (August 2016)	1369
5.46	Document Filters 11.3 Build 2172 (June 2016)	1372
5.47	Document Filters 11.3 Build 2116 (April 2016)	1376
5.48	Document Filters 11.3 Build 2074 (March 2016)	1379
5.49	Document Filters 11.3 Build 2040 (January 2016)	1382
5.50	Document Filters 11.3 Build 1998 (December 2015)	1385
5.51	Document Filters 11.2 Build 1962 (November 2015)	1387
5.52	Document Filters 11.2 Build 1914 (September 2015)	1390
5.53	Document Filters 11.2 Build 1884 (September 2015)	1392
5.54	Document Filters 11.2 Build 1858 (August 2015)	1396
5.55	Document Filters 11.2 Build 1808 (June 2015)	1398
5.56	Document Filters 11.2 Build 1766 (May 2015)	1401
5.57	Document Filters 11.2 Build 1732 (March 2015)	1404
5.58	Document Filters 11.2 Build 1666 (January 2015)	1408
5.59	Document Filters 11.2 (November 2014)	1412
5.60	Document Filters 11.1 Build 1546 (October 2014)	1419
5.61	Document Filters 11.1 Build 1464 (July 2014)	1424
5.62	Document Filters 11.1 Build 1394 (April 2014)	1428
5.63	Document Filters 11.1 Build 1333 (March 2014)	1431
5.64	Document Filters 11.1 Build 1262 (December 2013)	1434
5.65	Document Filters 11.1 (October 2013)	1437
5.66	Document Filters 11.0 (July 2013)	1441

1. Getting started

1.1 Getting started with Document Filters

- Document Filters is a comprehensive SDK designed for software developers looking to integrate robust file identification, data extraction, document transformation, and format conversion features into their applications.
- This SDK is available as Dynamic Link Libraries (DLLs) for Windows and Shared Objects (SOs) for UNIX-based systems. With Document Filters, developers can:
- Identify a wide range of file types.
 - Extract text and metadata from hundreds of document formats.
 - Retrieve sub-documents and attachments from various formats, including MS Office documents, ZIP files, RARs, 7-Zips, ISOs, CABs, PSTs, and OSTs.
 - Convert popular document formats into High-Definition outputs that preserve styles, layouts, and images, with support for formats such as TIFF, HTML, PDF, and Structured XML.
 - Utilize Canvas and Drawing functions for document markup, permanent annotations, and redaction.

For sample code and header definitions in C++, C#, HTML, Java, and Python, please visit [GitHub - Hyland Document Filters](#).

1.1.1 In This Section

Getting Started with .NET	Hyland Document Filters provides robust document processing capabilities that can be easily integrated into your .NET applications. Follow these instructions to set up your environment.
Getting Started with C	Hyland Document Filters allows you to integrate powerful document processing capabilities into your C applications. Below are instructions on how to set up your application depending on your build system, such as CMake, Visual Studio, or Make.
Getting Started with C++	Hyland Document Filters allows you to integrate powerful document processing capabilities into your C++ applications. Below are instructions on how to set up your application depending on your build system, such as CMake, Visual Studio, or Make.
Getting Started with COM	Hyland Document Filters offers powerful document processing capabilities that can be seamlessly integrated into your Windows applications using COM. However, using COM involves extra setup steps, so first-level bindings (e.g., .NET, Java, or C++) should be chosen when available for easier integration.

Getting Started with Java	Hyland Document Filters provides robust document processing capabilities that can be easily integrated into your Java applications. Follow these instructions to set up your environment.
Getting Started with Python	Hyland Document Filters provides robust document processing capabilities that can be easily integrated into your Python applications. Follow these instructions to set up your environment.
About Accessibility Info Extraction	Document Filters supports the extraction of accessibility information from MS Office documents, aiding in the development of accessible products. The extracted "alt text" labels include images, drawings, objects, word art, smart art, charts, icons, shapes, and text boxes across Word (DOC and DOCX), PowerPoint (PPT and PPTX), Excel (XLS and XLSX), and Visio (VSDX) files. The extracted alt text is surfaced in various output formats -- HTML5, Classic HTML, XML, and PDF -- when the appropriate options for creating accessible output are provided.
About Conversion Profiles	The Hyland Document Filters allow you to customize conversion settings through a configuration file named <code>ISYS11df.ini</code> . This file should be placed in the executable folder. If present, the settings specified in this file will be applied during processing, unless explicitly overridden by the application in the <code>Open</code> or <code>Canvas</code> methods.
About Custom Streams and Extended Streams	Hyland Document Filters supports customizable streams, allowing you to read from storage systems not natively recognized by Document Filters. For example, you may want to read files directly from a database or an FTP site. Additionally, you can utilize a specialized type of custom stream known as an Extended Stream. Extended Streams help Document Filters retrieve additional information about your stream when needed, such as handling requests for specific parts of a multi-part archive.
About File Type Identification	File type identification is the process of determining a file's format based on its content rather than its extension. This is crucial for accurate processing, security scanning, and compatibility with different software applications. Hyland Document Filters identifies file types by analyzing a file's byte stream, starting with the first 2048 bytes and reading additional data if necessary. It applies a combination of signature-based detection, heuristics, and container recognition to determine the format.
About Fonts	To effectively render a document into HD format, Document Filters must have access to the appropriate fonts to accurately represent each character on the page. Fonts not only provide the visual representation of characters but also contain size and measurement information necessary for proper content positioning.
About Multi-Part Archives	

	Multi-part archive files, such as certain ZIP and RAR formats, are comprised of two or more files packaged together. Document Filters supports the processing of these multi-part archives, allowing users to easily manage complex file structures.
About Multithreading	Document Filters may be run in a multithreaded application with minimal effort. Following a basic set of rules is essential to avoid issues such as data corruption or race conditions.
About Optical Character Recognition (OCR)	Document Filters provides comprehensive support for Optical Character Recognition (OCR), enabling users to extract text from a variety of document formats. It includes built-in support for the Tesseract OCR engine, which is a widely-used open-source OCR solution. Additionally, Document Filters allows for the use of other versions of Tesseract, giving you the flexibility to choose the version that best fits your needs.

Content Enrichment

Overview	The Content Enrichment section introduces key features that enhance the processing of documents by identifying and extracting valuable structural elements such as tables, headers, and footers. These capabilities allow users to transform raw document data into a more organized and meaningful format, facilitating improved data extraction and analysis. By leveraging these features, you can enrich your documents with critical insights and streamline workflows, making it easier to work with complex content.
Table Detection	Table Detection is a powerful feature of the Hyland Document Filters that enables automatic extraction and processing of tables from various document types. This functionality is particularly valuable when working with file formats that inherently store structured tabular data, such as Microsoft Office documents (Word, Excel, PowerPoint) and other similar productivity applications.

Output Formats

Overview	Hyland Document Filters offers a variety of output modes, including text, images, PDF, and XML. This section delves into some of these options in more detail.
JSON Output	Hyland Document Filters provides robust options for generating JSON output from various document formats. By configuring the output settings, you can customize the generated JSON to suit your specific needs, from data structures to metadata inclusion.

Markdown Output	Hyland Document Filters provides powerful options for generating Markdown output from various document formats. By configuring the output settings, you can customize the generated Markdown to suit your specific needs, from table styles to metadata inclusion.
PDF Output	Hyland Document Filters offers robust features for generating PDF output from a variety of document formats. By adjusting the output settings, you can tailor the resulting PDF to meet your specific requirements, including layout preferences, security features, and metadata inclusion.

1.2 Getting Started with .NET

Hyland Document Filters provides robust document processing capabilities that can be easily integrated into your .NET applications. Follow these instructions to set up your environment.

1.2.1 Installing the Bindings

The Document Filters .NET bindings are available on NuGet under the package name `Hyland.DocumentFilters`, which provides the easiest and recommended way to integrate Document Filters into your .NET applications.

If needed, you can also access the source code for the bindings in the `bindings/dotnet` directory of the [Document Filters GitHub repository](#). This allows for modifications and manual builds when customization is required.

Note

Earlier versions utilized `Perceptive.DocumentFilters` as a pre-compiled project, which remains included in the package. To upgrade a project to use `Hyland.DocumentFilters`, simply update the project reference and change the namespace accordingly.

Note

DocumentFilters DLLs are designed for a specific architecture. When your application targets AnyCPU, the architecture can be determined at runtime, so both 32-bit and 64-bit versions of the DocFilters DLLs must be present.

If you're using the NuGet package and targeting .NET Core, the selection of the appropriate DLLs is automated. However, if you are targeting .NET Framework, you must ensure that the DLLs are discoverable according to the [Default Probing rules for Unmanaged \(native\) libraries](#).

Using Published Bindings

To use the official Document Filters .NET bindings, you can add the `Hyland.DocumentFilters` package from NuGet to your project using Visual Studio or the command line.

Via the Command Line:

Run the following command in your terminal:

```
dotnet add package Hyland.DocumentFilters
```

Via Visual Studio:

Follow these steps to add the bindings via Visual Studio:

1. Open your project in Visual Studio.
2. In the Solution Explorer, right-click on your project and select **Manage NuGet Packages**.
3. In the **NuGet Package Manager** window, select the Browse tab.
4. In the search box, type `Hyland.DocumentFilters`.
5. Once found, click on the `Hyland.DocumentFilters` package and select the desired version.
6. Click Install to add the package to your project.
7. After installation, the package will be listed under your project's Dependencies in Solution Explorer, and you can start using the Document Filters API.

This ensures that you always have the latest stable version of the bindings, making integration and updates straightforward.

Building the Bindings

If you need to modify the .NET bindings or build them manually, follow these steps:

1. Clone the [Document Filters GitHub repository](#).
2. Navigate to the `bindings/dotnet` directory.
3. Open the solution file in Visual Studio or another preferred IDE.
4. Make any necessary changes to the code.
5. Build the project to generate the custom .NET bindings for your application.

Once built, you can reference the custom bindings in your .NET project manually, or package them into a local NuGet package for reuse.

1.2.2 Initializing and calling Document Filters

C#

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
```

The preceding code loads the `DocumentFilters` package into global scope, then creates a new `Api` singleton and `Initialize`s it with a license key. Replace "YOUR_LICENSE_KEY_HERE" with your actual license key.

The second argument controls where DocFilters should look for resources, such as configuration files and fonts. The `.` tells it to look in the same directory as the DocFilter's shared libraries.

Note: ISYSdf11.dll must be either in same folder as the currently executing Assembly or found by [Default Probing rules for Unmanaged \(native\) libraries](#).

1.2.3 Extracting Text

C#

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.GetExtractor("filename.doc");
7 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta)
8
9 while (!doc.EndOfStream)
10 {
11     var text = doc.GetText(4096);
12     Console.Out.WriteLine(text);
13 }
```

The preceding code loads the file `filename.doc` into an extractor `doc`. By using a scoped `using` block, the extractor will be closed when `doc` goes out of scope.

It then opens the extractor with `BodyAndMeta` indicating that we want to extract both the text (body) and the metadata of the document.

Finally, it loops over calling `GetText` until the extractor reports `EndOfStream`.

1.2.4 Converting a Document

C#

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.GetExtractor("filename.doc");
7 using var canvas = api.MakeOutputCanvas("output.pdf", Hyland.DocumentFilters.CanvasType.PDF);
8
9 doc.Open(Hyland.DocumentFilters.OpenType.FormatImage)
10
11 canvas.RenderPages(doc);
```

The preceding code loads the file `filename.doc` into an extractor `doc`. By using a scoped `using` block, the extractor will be closed when `doc` goes out of scope.

It also creates a new `canvas` object of type `CanvasType.PDF` and stores it as `canvas`. The canvas will also be automatically closed when it goes out of scope.

It then opens the extractor with `FormatImage` indicating that we want to convert the file to an image based output. This triggers the file to be paginated.

Finally, it calls `RenderPages` to output every page from `doc` into the `canvas`. You could also manually iterate over the pages and call `RenderPages`.

**Did you know?**

You can render more than one document to a canvas. If you want to stitch multiple files together, simply load each document into its own `Extractor`, then call `RenderPage/s` onto a single canvas.

1.3 Getting Started with C

Hyland Document Filters allows you to integrate powerful document processing capabilities into your C applications. Below are instructions on how to set up your application depending on your build system, such as CMake, Visual Studio, or Make.

1.3.1 Clone and Include the Document Filters Repository

The [Document Filters GitHub repository](#) contains the necessary header files and libraries. If you are not using CMake's `FetchContent`, clone the repository manually into your project:

```
git clone https://github.com/Hyland/DocumentFilters.git
```

The main header file you need to work with is `DocumentFilters.h`, located in the `bindings/c/include` directory. If you're using CMake's `FetchContent`, manually cloning the repository is not required (details below).

1.3.2 Integrating with CMake

If you're using CMake, you can either manually clone the repository or use the `FetchContent` module to automatically fetch it without cloning. Here's how:

Using FetchContent (No Cloning Required)

CMake's `FetchContent` allows you to fetch the repository directly without manually cloning it. Add the following to your `CMakeLists.txt`:

CMakeLists.txt

```
include(FetchContent)

FetchContent_Declare(
  HylandDocumentFiltersC
  GIT_REPOSITORY https://github.com/Hyland/DocumentFilters.git
  GIT_TAG <desired-tag-or-branch>
  SOURCE_SUBDIR bindings/c
)

FetchContent_MakeAvailable(HylandDocumentFiltersC)
```

Once included, link the `DocumentFilters::C` library to your target:

CMakeLists.txt

```
target_link_libraries(your_target PRIVATE DocumentFilters::C)
```

Manual Cloning with CMake

If you prefer to manually clone the repository, first clone the repo into your project and then link it manually:

```
bash

git clone https://github.com/Hyland/DocumentFilters.git
```

After cloning, add the include directory and the library manually in your `CMakeLists.txt` :

```
CMakeLists.txt

add_subdirectory(path/to/DocumentFilters/bindings/c)
```

1.3.3 Integrating with Visual Studio (vcxproj-based)

To use Hyland Document Filters in a Visual Studio project, follow these steps:

Include the Header Files

- Go to **Project > Properties > Configuration Properties > C/C++ > General > Additional Include Directories**.
- Add the path to the `bindings/c/include` directory:

```
c:\path\to\DocumentFilters\bindings\c\include
```

Link the ISYS11df Library

- Go to **Project > Properties > Configuration Properties > Linker > General > Additional Library Directories**, and add the directory containing the `ISYS11df.lib` library:

```
c:\path\to\DocumentFilters\bindings\c\libs\x64
```

- In **Linker > Input > Additional Dependencies**, add:

```
ISYS11df.lib
```

1. Copy Required DLLs (if needed):
2. Ensure that the `ISYS11df.dll` file is accessible at runtime by copying it to your project output directory or ensuring it's available in your system's PATH.

1.3.4 Integrating with Makefiles projects

If you're using a manual build system, such as `Make` with a `Makefile`, you will need to manually specify the include paths and link the required libraries.

Include Header Files and Link Libraries in Makefile

Update your `Makefile` to include the necessary paths for headers and link the `ISYS11df` library. Here's an example setup:

```
Makefile

CC = gcc
CFLAGS = -Ipath/to/DocumentFilters/bindings/c/include
LDFLAGS = -Lpath/to/DocumentFiltersBinaries -lISYS11df

all: my_app

my_app: main.o
    $(CC) -o my_app main.o $(LDFLAGS)

main.o: main.c
    $(CC) $(CFLAGS) -c main.c

clean:
    rm -f *.o my_app
```

Compile the Application

After setting up the Makefile, you can build your application by running:

```
bash

make
```

With this approach, you'll need to manually download the Document Filters Release Binaries and specify their location as a link library directory, referred to above as `path/to/DocumentFiltersBinaries`.

1.3.5 Initializing and calling Document Filters

After setting up the necessary includes and linking the required libraries, the next step is to initialize the Document Filters library. This initialization is essential, as it prepares the library to process documents effectively. Below is a sample code snippet demonstrating how to perform this initialization in your C application.

C

```
1 #include <stdio.h>
2 #include "DocumentFilters.h"
3
4 int main(int argc, char** argv)
5 {
6     Instance_Status_Block isb;
7     Error_Control_Block ecb;
8     SHORT inst;
9
10    strncpy_s(isb.Licensee_ID1, "YOUR_LICENSE_KEY_HERE", sizeof(isb.Licensee_ID1) - 1);
11    Init_Instance(0, ".", &isb, &inst, &ecb);
12    if (ecb.Msg[0])
13        return 1;
14
15    return 1;
16 }
```

In the code above, we initialize the global Document Filters library, which is required only once per application. Replace "YOUR_LICENSE_KEY_HERE" with your actual license key.

The second argument in the `Init_Instance` function specifies where the Document Filters library should search for resources, such as configuration files and fonts. By passing `.` as the argument, you instruct the library to look in the same directory as the shared libraries of Document Filters. This ensures that the library can access all necessary resources for proper functioning.

1.3.6 Extracting Text

Once the Document Filters library is initialized, you can start extracting text from documents. The following code snippet illustrates how to load a document and retrieve its contents using the Document Filters API. This example focuses on extracting text from a Word document (`.doc` file) and converting it to a more accessible format.

C

```

1  #include <stdio.h>
2  #include <string.h>
3  #include "DocumentFilters.h"
4
5  const char LICENSE_KEY[] = "YOUR_LICENSE_KEY_HERE";
6  const char SOURCE_FILE[] = "filename.doc";
7
8  int main(int argc, char** argv)
9  {
10     Instance_Status_Block isb = { 0 };
11     Error_Control_Block ecb = { 0 };
12     SHORT inst = 0;
13
14     strncpy(isb.Licensee_ID1, LICENSE_KEY, sizeof(isb.Licensee_ID1) - 1);
15     Init_Instance(0, ".", &isb, &inst, &ecb);
16     if (ecb.Msg[0])
17     {
18         printf("%s\n", ecb.Msg);
19         return 1;
20     }
21
22     IGR_UCS2 source_file[255] = { 0 };
23     IGR_UCS2 canvas_file[255] = { 0 };
24     IGR_UCS2 body_text[1024] = { 0 };
25     IGR_LONG caps = 0, filetype = 0, handle = 0, body_size = sizeof(body_text) / sizeof(body_text[0]) - 1;
26     char ascii_body_text[2048] = { 0 };
27
28     UTF8_to_Widechar_Ex(SOURCE_FILE, sizeof(SOURCE_FILE) - 1, source_file, sizeof(source_file) /
sizeof(source_file[0]));
29
30     if (IGR_Open_File_Ex(source_file, IGR_BODY_AND_META, NULL, &caps, &filetype, &handle, &ecb) != IGR_OK)
31         goto bad;
32
33     while (IGR_Get_Text(handle, body_text, &body_size, &ecb) == IGR_OK)
34     {
35         // convert the buffer from UTF-16 into UTF-8 for printing
36         Widechar_to_UTF8_Ex(body_text, body_size, ascii_body_text, sizeof(ascii_body_text)-1);
37
38         // replace DF markers
39         for (size_t i = 0; i < sizeof(ascii_body_text) && ascii_body_text[i]; ++i)
40         {
41             if (ascii_body_text[i] == IGR_CHAR_PARA_BREAK)
42                 ascii_body_text[i] = '\n';
43         }
44         printf("%s", ascii_body_text);
45         body_size = sizeof(body_text) / sizeof(body_text[0]) - 1;
46     }
47
48 bad:
49     if (ecb.Msg[0])
50         printf("%s\n", ecb.Msg);
51     if (handle)
52         IGR_Close_File(handle, &ecb);
53 }

```

In the code snippet above, we begin by defining the license key and the source file name. The program initializes the Document Filters library and opens the specified document (`filename.doc`) using the `IGR_Open_File_Ex` function. This function provides a handle to the document, which is necessary for subsequent operations.

The Document Filters API uses UCS2/UTF-16 encoding for strings, so the example includes conversion functions to handle the text appropriately. The program enters a loop, calling `IGR_Get_Text` to retrieve text from the document until an error occurs. Each retrieved chunk of text is converted from UTF-16 to UTF-8 for easier printing.

If any error messages are encountered during the process, they are displayed to the user. Finally, the document handle is closed with `IGR_Close_File` to release any resources associated with it, ensuring proper cleanup.

1.3.7 Converting a Document

After successfully initializing the Document Filters library and extracting text, you can proceed to convert documents into different formats, such as PDF. The following code snippet demonstrates how to load a Word document (`.doc` file) and convert it into a PDF using the Document Filters API.

C

```

1  #include <stdio.h>
2  #include <string.h>
3  #include "DocumentFilters.h"
4
5  const char LICENSE_KEY[] = "YOUR_LICENSE_KEY_HERE";
6  const char SOURCE_FILE[] = "filename.doc";
7  const char CANVAS_FILE[] = "output.pdf";
8
9  int main(int argc, char** argv)
10 {
11     Instance_Status_Block isb = { 0 };
12     Error_Control_Block ecb = { 0 };
13     SHORT inst = 0;
14
15     strncpy(isb.Licensee_ID1, LICENSE_KEY, sizeof(isb.Licensee_ID1) - 1);
16     Init_Instance(0, ".", &isb, &inst, &ecb);
17     if (ecb.Msg[0])
18     {
19         printf("%s\n", ecb.Msg);
20         return 1;
21     }
22
23     IGR_UCS2 source_file[255] = { 0 };
24     IGR_UCS2 canvas_file[255] = { 0 };
25     IGR_UCS2 body_text[1024] = { 0 };
26     IGR_LONG caps = 0, filetype = 0, doc_handle = 0, page_count = 0;
27     HCANVAS canvas_handle = 0;
28     HPAGE page_handle = 0;
29
30     UTF8_to_Widechar_Ex(SOURCE_FILE, sizeof(SOURCE_FILE) - 1, source_file, sizeof(source_file) /
sizeof(source_file[0]));
31     UTF8_to_Widechar_Ex(CANVAS_FILE, sizeof(CANVAS_FILE) - 1, canvas_file, sizeof(canvas_file) /
sizeof(canvas_file[0]));
32
33     if (IGR_Open_File_Ex(source_file, IGR_FORMAT_IMAGE, NULL, &caps, &filetype, &doc_handle, &ecb) !=
IGR_OK)
34         goto bad;
35
36     if (IGR_Make_Output_Canvas(IGR_DEVICE_PDF, canvas_file, NULL, &canvas_handle, &ecb) != IGR_OK)
37         goto bad;
38
39     if (IGR_Get_Page_Count(doc_handle, &page_count, &ecb) != IGR_OK)
40         goto bad;
41
42     for (IGR_LONG page_index = 0; page_index < page_count; ++page_index)
43     {
44         if (IGR_Open_Page(doc_handle, page_index, &page_handle, &ecb) == IGR_OK)
45         {
46             IGR_Render_Page(page_handle, canvas_handle, &ecb);
47             IGR_Close_Page(page_handle, &ecb);
48         }
49     }
50
51 bad:
52     if (ecb.Msg[0])
53         printf("%s\n", ecb.Msg);
54     if (canvas_handle)
55         IGR_Close_Canvas(canvas_handle, &ecb);

```



```
56 |     if (doc_handle)
57 |         IGR_Close_File(doc_handle, &ecb);
58 | }
```

In this example, the program initializes the Document Filters library with a specified license key and opens the document (`filename.doc`) using the `IGR_Open_File_Ex` function. The file is opened in `IGR_FORMAT_IMAGE`, indicating that we want to convert it into an image-based output format, which will paginate the document accordingly.

Note: The Document Filters API uses UCS2/UTF-16 encoding for strings, so all input and output strings are converted into UTF-8.

The code then creates a new PDF canvas by calling `IGR_Make_Output_Canvas` with `IGR_DEVICE_PDF` , preparing for the output file.

Next, the program retrieves the total page count of the document with `IGR_Get_Page_Count` and enters a loop to process each page. For every page, it calls `IGR_Open_Page` to obtain a handle to the page, which is then rendered to the PDF canvas using `IGR_Render_Page` .

In case of any errors during the conversion process, appropriate messages are printed. Finally, the program ensures proper cleanup by closing the canvas and the document handle with `IGR_Close_Canvas` and `IGR_Close_File` , respectively. This structured approach ensures that resources are managed efficiently while converting documents.

1.3.8 Troubleshooting

If you encounter the error message:

```
libISYS11df.so: cannot open shared object file: No such file or directory
```

this indicates that the application is unable to find the Document Filters Shared Libraries. The standard `dlopen` rules apply for locating these libraries.

To resolve this issue, you can add the directory containing the libraries to the `LD_LIBRARY_PATH` environment variable. This adjustment will help the application locate the necessary shared libraries and eliminate the error.

1.4 Getting Started with C++

Hyland Document Filters allows you to integrate powerful document processing capabilities into your C++ applications. Below are instructions on how to set up your application depending on your build system, such as CMake, Visual Studio, or Make.

1.4.1 Clone and Include the Document Filters Repository

The [Document Filters GitHub repository](#) contains the necessary header files and libraries. If you are not using CMake's `FetchContent`, clone the repository manually into your project:

```
git clone https://github.com/Hyland/DocumentFilters.git
```

The primary header file required for your work is `DocumentFiltersObjects.h`, which can be found in the `bindings/cpp{VERSION}/include` directory. When using CMake's `FetchContent`, there is no need to manually clone the repository (details are provided below).

Note

Using the C++ bindings is not mandatory; your application can opt to interact directly with the C API instead. Follow the instructions on [Getting Started with C](#)

1.4.2 Choosing a C++ binding version

The Document Filters library offers bindings for C++17, located in the `bindings/cpp17` directory, as well as deprecated bindings for C++11 found in the `bindings/cpp11` directory. The C++17 bindings deliver comprehensive Object API functionality, encompassing annotations and document comparison, while the C++11 bindings offer limited support. Although the C++11 bindings are labeled as deprecated, they should only be utilized when a modern C++ compiler is unavailable.

Memory Management:

- **C++11 bindings:** Requires manual memory management with raw pointers.
- **C++17 bindings:** Automatic object lifetime management, eliminating manual `delete` calls. Objects are internally reference counted, and can be safely stored on the stack or heap without the caller needing to track their life-time.
- **Migration strategy:** REplace instances of manual memory management, including explicit calls to `delete`, or wrapping objects in `smartptrs`. Instead, objects should be placed on the stack, and allow them to be self-managed.

String Handling:

- **C++11 bindings:** Provided a mix of ansi string and wide string.

- **C++17 bindings:** Returned strings are normalized to `std::wstring`, with utility functions for converting between UTF-8 variants. Function parameters can be provided in either utf-8 or wide strings.
- **Migration strategy:** Use the provided string conversion function to convert between string types if your application does not work natively with `std::wstring`.

Object API Coverage:

- **C++11 bindings:** Limited coverage, restricted to text extraction, subfile handling, and rendering. Advanced features such as annotations and document comparison are not supported directly in the C++11 bindings and must be achieved via the C API.
- **C++17 bindings:** Provides full coverage of the Object API, including advanced features like annotations and document comparison, with feature parity across .NET, Python, and Java.
- **Migration strategy:** No required changes unless leveraging new features.

Stream Handling:

- **C++11 bindings:** Streams are handled via raw pointers and manual read/write operations.
- **C++17 bindings:** Streams are supported using modern C++ constructs, including `std::istream` and `std::ostream`, allowing for seamless integration with standard C++ I/O streams.
- **Migration strategy:** Replace raw pointer-based stream handling with `std::istream` and `std::ostream`-based implementations. This will ensure more efficient and safer stream operations.

Collection Iteration:

- **C++11 bindings:** Collections like lists or arrays are often iterated manually, typically using raw loops or index-based access.
- **C++17 bindings:** The bindings take advantage of range-based loops and standard C++ iterator patterns, improving readability and reducing potential off-by-one errors.
- **Migration strategy:** Migrate to range-based for-loops or use iterators for collections. This will make the code more concise and easier to maintain.

Callbacks:

- **C++11 bindings:** Callbacks are implemented using function pointers, requiring manual management and less flexibility when passing stateful functions.
- **C++17 bindings:** Callbacks can now be implemented using `std::function`, allowing the use of lambdas and stateful objects to handle callbacks more elegantly.
- **Migration strategy:** Replace function pointers with `std::function` where callbacks are used. This allows greater flexibility, enabling the use of lambdas or member functions with bound state.

Lambda Improvements:

- **C++11 bindings:** Basic lambda support exists, but capturing more complex objects or state (like move-only types) requires more manual management.
- **C++17 bindings:** C++17 lambdas support move-only types and generalized lambda captures, which allow for more flexible handling of objects within closures.
- **Migration strategy:** Refactor complex lambdas, especially those that handle state, to take advantage of C++17's generalized captures. This reduces boilerplate and makes callbacks and functional programming patterns easier to manage.

Filesystem Support:

- **C++11 bindings:** File handling may rely on older APIs or require third-party libraries to handle filesystem operations like directory iteration and path manipulations.
- **C++17 bindings:** C++17 brings native support for filesystem operations via the `std::filesystem` library, which includes functions for path manipulation, file I/O, and directory traversal.
- **Migration strategy:** Replace any custom or third-party filesystem libraries with `std::filesystem`. This simplifies file and directory handling with a standardized and efficient API.

1.4.3 Integrating with CMake

If you're using CMake, you can either manually clone the repository or use the `FetchContent` module to automatically fetch it without cloning. Here's how:

Using FetchContent

CMake's `FetchContent` allows you to fetch the repository directly without manually cloning it. Add the following to your `CMakeLists.txt`:

```
CMakeLists.txt

include(FetchContent)

FetchContent_Declare(
  HylandDocumentFiltersCpp17
  GIT_REPOSITORY https://github.com/Hyland/DocumentFilters.git
  GIT_TAG <desired-tag-or-branch>
  SOURCE_SUBDIR bindings/cpp17
)

FetchContent_MakeAvailable(HylandDocumentFiltersCpp17)
```

Once included, link the `DocumentFilters::Cpp17` library to your target:

CMakeLists.txt

```
target_link_libraries(your_target PRIVATE DocumentFilters::Cpp17)
```

Manual Cloning with CMake

If you prefer to manually clone the repository, first clone the repo into your project and then link it manually:

bash

```
git clone https://github.com/Hyland/DocumentFilters.git
```

After cloning, add the include directory and the library manually in your `CMakeLists.txt` :

CMakeLists.txt

```
add_subdirectory(path/to/DocumentFilters/bindings/cpp17)
```

1.4.4 Integrating with Visual Studio projects (vcxproj-based)

To use Hyland Document Filters in a Visual Studio project, follow these steps:

Include the DocumentFiltersCpp17 project in your Solution

- Right-click on your Solution in **Solution Explorer**, then select **Add > Existing Project**.
- Navigate to the `bindings/cpp17` directory and select the `DocumentFiltersCpp17.vcxproj` file.
- In your project, right-click on the project name, then select **Add > Reference*. Ensure that `DocumentFiltersCpp17` is checked.

Include the Header Files

- Go to **Project > Properties > Configuration Properties > C/C++ > General > Additional Include Directories**.
- Add the path to the `bindings/cpp17/include` and `bindings/c/include` directories, as follows:

```
c:\path\to\DocumentFilters\bindings\cpp17\include;c:\  
\path\to\DocumentFilters\bindings\c\include;
```

 Note

You need to include both the `Cpp` and `C` include directories in your project.

Link the ISYS11df Library

- No additional steps are required to manually link the `ISYS11df` library, as it is already a dependency of the `DocumentFiltersCpp17` project.
- However, ensure that the `ISYS11df.dll` file is accessible at runtime. You can either copy it to your project output directory or make sure it is available in your system's `PATH`.

1.4.5 Integrating with Makefiles projects

If you're using a manual build system, such as `Make` with a `Makefile`, you will need to manually specify the include paths and link the required libraries.

Include Header Files, build object api and Link Libraries in Makefile

You can modify your `Makefile` to compile the source files and link the required libraries as specified in the CMake script. Below is an example `Makefile` setup:

Makefile

```
CXX = g++
CXXFLAGS = -std=c++17 -Ipath/to/DocumentFilters/bindings/c/include -Ipath/to/DocumentFilters/bindings/cpp17/
include
LDFLAGS = -Lpath/to/DocumentFiltersBinaries -lISYS11df
SOURCES = src/DocumentFiltersObjects.cpp src/DocFiltersAnnotations.cpp src/DocFiltersBookmark.cpp \
src/DocFiltersCanvas.cpp src/DocFiltersCommon.cpp src/DocFiltersCompareDocumentSettings.cpp \
src/DocFiltersCompareDocumentSource.cpp src/DocFiltersCompareResultDifference.cpp \
src/DocFiltersCompareResultDifferenceDetail.cpp src/DocFiltersCompareResults.cpp \
src/DocFiltersCompareSettings.cpp src/DocFiltersDateTime.cpp src/DocFiltersExtractor.cpp \
src/DocFiltersFormat.cpp src/DocFiltersFormElement.cpp src/DocFiltersHyperlink.cpp \
src/DocFiltersOption.cpp src/DocFiltersPage.cpp src/DocFiltersPageElement.cpp \
src/DocFiltersPagePixels.cpp src/DocFiltersRenderPageProperties.cpp src/DocFiltersStreams.cpp \
src/DocFiltersStrings.cpp src/DocFiltersSubFile.cpp src/DocFiltersWord.cpp
HEADERS = include/DocumentFiltersObjects.h

all: my_cpp_app

# Rule to build the DocumentFilters library
libDocumentFilters.a: $(SOURCES) $(HEADERS)
    $(CXX) $(CXXFLAGS) -c $(SOURCES)
    ar rcs libDocumentFilters.a *.o

# Link the library to the main application
```

```
my_cpp_app: main.o libDocumentFilters.a
$(CXX) -o my_cpp_app main.o libDocumentFilters.a $(LDFLAGS)

main.o: main.cpp
$(CXX) $(CXXFLAGS) -c main.cpp

clean:
rm -f *.o my_cpp_app libDocumentFilters.a
```

Compile the Application

After setting up the Makefile, you can build your application by running:

```
bash

make
```

With this approach, you'll need to manually download the Document Filters Release Binaries and specify their location as a link library directory, referred to above as path/to/DocumentFiltersBinaries.

1.4.6 Initializing and calling Document Filters

After setting up the necessary includes and linking the required libraries, the next step is to initialize the Document Filters library. This initialization is essential, as it prepares the library to process documents effectively. Below is a sample code snippet demonstrating how to perform this initialization in your Cpp application.

C++

```
#include <iostream>
#include <memory>
#include <DocumentFiltersObjects.h>

static const char* LICENSE_KEY = "YOUR_LICENSE_KEY_HERE";

int main() {
    try {
        Hyland::DocFilters::Api api;
        api.Initialize(LICENSE_KEY, ".");
    } catch (const std::exception& error) {
        std::cout << "Initialization error: " << error.what() << std::endl;
        return 1;
    }

    std::cout << "Document Filters initialized successfully!" << std::endl;
    return 0;
}
```

Explanation:

- **License Key:** The LICENSE_KEY is provided as part of the initialization process and is required to unlock the full functionality of the Document Filters library. Replace "YOUR_LICENSE_KEY_HERE" with your actual license key.
- **Resource Directory:** The second parameter to Initialize, ".", specifies the directory where Document Filters should look for resources such as configuration files and fonts. In this case, "." means it will search the current working directory where the shared libraries reside. You can adjust this path based on your setup.
- **Error Handling:** The try-catch block captures any exceptions thrown during initialization and outputs the error message. This ensures you can handle initialization failures gracefully.

The DocumentFilters objects are now initialized and ready to be used for various document processing tasks, such as extraction, rendering, or comparison.

1.4.7 Extracting Text

Once the Document Filters library is initialized, you can begin extracting text from documents. The following C++ code snippet demonstrates how to load a document and extract its contents using the Document Filters API. This example focuses on extracting text from a Word document (.doc file).

C++

```
#include <iostream>
#include <memory>
#include <DocumentFiltersObjects.h>

static const char* LICENSE_KEY = "YOUR_LICENSE_KEY_HERE";

int main() {
    try {
        Hyland::DocFilters::Api api;
        api.Initialize(LICENSE_KEY, ".");

        // Load the document for text extraction
        auto doc = api.GetExtractor(L"filename.doc");

        // Open the document for both body text and metadata extraction
        doc.Open(IGR_BODY_AND_META);

        // Loop through the document until EOF and extract text
        while (!doc->getEOF()) {
            auto text = doc.GetText();
            std::wcout << text << std::endl;
        }
    } catch (const std::exception& error) {
        std::cout << "Error: " << error.what() << std::endl;
        return 1;
    }
}
```



```
    return 0;
}
```

Explanation:

- **Document Loading:** The document is loaded using `api.GetExtractor`, which returns an object for extracting content. In this case, `"filename.doc"` is the input Word document. Update the file path accordingly.
- **Opening the Document:** The `doc.Open(IGR_BODY_AND_META)` method is called to open the document for text extraction. The `IGR_BODY_AND_META` flag specifies that both the document body and metadata should be extracted.
- **Text Extraction:** The code enters a loop, repeatedly calling `doc.GetText()` until the `getEOF()` method indicates the end of the document. The extracted text is output using `std::wcout` to handle wide characters.
- **Automatic Resource Management:** Since the document object is automatically managed, it will be closed when it goes out of scope, ensuring proper cleanup without manual intervention.

This approach efficiently extracts text from documents while managing resources automatically.

1.4.8 Converting a Document

After initializing the Document Filters library, you can convert documents into different formats, such as PDF. The following C++ code snippet demonstrates how to load a Word document (`.doc` file) and convert it into a PDF using the Document Filters API.

C++

```
#include <iostream>
#include <memory>
#include <DocumentFiltersObjects.h>

static const char* LICENSE_KEY = "YOUR_LICENSE_KEY_HERE";

int main() {
    try {
        Hyland::DocFilters::Api api;
        api.Initialize(LICENSE_KEY, ".");

        // Load the document for conversion
        auto doc = api.GetExtractor(L"filename.doc");

        // Create an output canvas for PDF rendering
        auto canvas = api.MakeOutputCanvas(L"result.pdf", IGR_DEVICE_IMAGE_PDF, "");

        // Open the document in image format for rendering
        doc.Open(IGR_FORMAT_IMAGE);

        // Loop through the document pages and render each one to the PDF canvas
        for (auto&& page : doc.pages()) {
            canvas.RenderPage(page);
        }
    }
}
```

```
    } catch (const std::exception& error) {  
        std::cout << "Error: " << error.what() << std::endl;  
        return 1;  
    }  
  
    return 0;  
}
```

Explanation:

- **Document Loading:** The document is loaded using `api.GetExtractor` with the path `"filename.doc"`. Replace this with your actual file path.
- **Canvas Creation:** A PDF canvas is created with the `MakeOutputCanvas` function, specifying the output file as `"result.pdf"`. The `IGR_DEVICE_IMAGE_PDF` flag is used to indicate that the output format is PDF.
- **Opening the Document:** The document is opened using `doc.Open(IGR_FORMAT_IMAGE)`, which prepares the document for rendering as an image-based format (used for PDF conversion).
- **Page Rendering:** The code iterates through each page of the document using a range-based loop. For each page, the `canvas.RenderPage(page)` method is called to render it onto the PDF canvas.
- **Automatic Resource Management:** Both the `doc` and `canvas` objects are automatically managed and will be closed when they go out of scope, ensuring proper cleanup.

This approach efficiently converts documents to PDF format while handling resources automatically.

1.5 Getting Stated with COM

Hyland Document Filters offers powerful document processing capabilities that can be seamlessly integrated into your Windows applications using COM. However, using COM involves extra setup steps, so first-level bindings (e.g., .NET, Java, or C++) should be chosen when available for easier integration.

1.5.1 Clone and Include the Document Filters Repository

The [Document Filters GitHub repository](#) contains the necessary files and libraries.

1.5.2 Additional Setup for COM

The `ISYS11df.dll` must be registered with Windows using the `regsvr32` command. This step is required for COM to function correctly.

1.5.3 Initializing and calling Document Filters

- Register and import the type library information from **ISYS11df.dll**.
- Create a "Hyland.DocumentFilters.11" object.
- Call `DocumentFilters.Initialize`.
- Call `DocumentFilters.GetExtractor` for one or more documents.
- Call `Extractor.GetText` to retrieve a document's text.
- Call `Extractor.Close` when finished with each document.

1.6 Getting Stated with Java

Hyland Document Filters provides robust document processing capabilities that can be easily integrated into your Java applications. Follow these instructions to set up your environment.

1.6.1 Clone and Include the Document Filters Repository

The [Document Filters GitHub repository](#) contains the necessary files and libraries.

1.6.2 Installing the Bindings

The Java bindings JAR file, `ISYS11df.jar`, can be found in the `bindings/java/lib` directory of the [Document Filters GitHub repository](#). While the same JAR file can be used across all platforms, you will need to obtain the appropriate native binaries for each platform you wish to support. The native binaries are included in the release ZIP files for each platform.

Note

The `run.sh` and `run.cmd` scripts included with the Java samples in the [Document Filters GitHub repository](#) automatically handle downloading the release binaries for the current platform.

Integrating with Maven

1. **Add the JAR and native binaries:** Since `ISYS11df.jar` is not hosted on Maven Central, you'll need to manually include the JAR file and native binaries.
2. **Add the dependencies:** Copy the `ISYS11df.jar` into your project directory (e.g., `libs` folder).
3. **Update your `pom.xml`:**

pom.xml

```
<dependencies>
  <!-- Add Document Filters JAR as a system-scoped dependency -->
  <dependency>
    <groupId>com.perceptive</groupId>
    <artifactId>documentfilters</artifactId>
    <version>11.0</version>
    <scope>system</scope>
    <systemPath>${project.basedir}/libs/ISYS11df.jar</systemPath>
  </dependency>
</dependencies>
```

```
<build>
  <plugins>
    <!-- Ensure native libraries are accessible by setting up system
properties -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-antrun-plugin</artifactId>
      <version>1.8</version>
      <executions>
        <execution>
          <phase>process-resources</phase>
          <configuration>
            <tasks>
              <copy file="path/to/native/binaries/ISYS11df.dll" todir="$
{project.build.directory}/native/" />
            </tasks>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

4. **Configure Native Libraries:** You may need to set the library path in your code or Maven build script using the `java.library.path` system property.

Integrating with Gradle

1. **Add the JAR and native binaries:** Copy the `ISYS11df.jar` to your project's `libs` directory.
2. **Update your build.gradle:**

build.gradle

```
groovy
Copy code
dependencies {
    // Add Document Filters JAR as a compile-time dependency
    implementation files('libs/ISYS11df.jar')
}

task copyNativeLibs(type: Copy) {
    from 'path/to/native/binaries'
    into "$buildDir/nativeLibs"
```

```
}

// Ensure native libraries are available
run {
    dependsOn copyNativeLibs
    systemProperty 'java.library.path', "$buildDir/nativeLibs"
}
```

3. **Configure Native Libraries:** Like Maven, you can set the `java.library.path` in your Gradle run configuration.

Integrating with Ant

1. **Add the JAR and native binaries:** Place the `ISYS11df.jar` and native binaries in your project folder (e.g., `lib` and `native` folders).
2. **Update `build.xml`:**

build.xml

```
<project name="DocumentFiltersProject" basedir="." default="run">

  <path id="classpath">
    <pathelement location="lib/ISYS11df.jar"/>
  </path>

  <target name="run">
    <java classname="com.perceptive.App" fork="true">
      <classpath refid="classpath"/>
      <jvmarg value="-Djava.library.path=./native"/>
    </java>
  </target>

</project>
```

3. **Configure Native Libraries:** Set the `java.library.path` using the `jvmarg` to point to the directory containing the native binaries.

1.6.3 Initializing and calling Document Filters

Once the package is installed, you can begin using it in your application.

Java

App.java

```
import com.perceptive.documentfilters.*;

public class App {

    private static final String LICENSE_KEY = "YOUR_LICENSE_KEY_HERE";

    public static void main(String[] args) {
        try {
            DocumentFilters api = new DocumentFilters();
            api.Initialize(LICENSE_KEY, ".");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Explanation:

- The code imports the DocumentFilters package.
- A new DocumentFilters instance is created and initialized using a license key. Replace "YOUR_LICENSE_KEY_HERE" with your actual license key.
- The second parameter "." specifies the directory for configuration files and resources, such as fonts.

Note: ISYS11df.(dll/so/dylib) will be loaded by a call to System.loadLibrary("ISYS11df"). For more details, refer to [System.LoadLibrary](#).

1.6.4 Extracting Text

Once the Document Filters library is initialized, you can begin extracting text from documents. The following Java code snippet demonstrates how to load a document and extract its contents using the Document Filters API. This example focuses on extracting text from a Word document (.doc file).

Java

App.java

```
import com.perceptive.documentfilters.*;

public class App {

    private static final String LICENSE_KEY = "YOUR_LICENSE_KEY_HERE";

    public static void main(String[] args) {
        try {
            DocumentFilters api = new DocumentFilters();
            api.Initialize(LICENSE_KEY, ".");

            try (Extractor doc = api.GetExtractor("filename.doc")) {
```

```

        doc.Open(isis_docfilters.IGR_BODY_AND_META);

        while (!doc.getEOF()) {
            String text = doc.GetText(4096);
            System.out.println(text);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

Explanation:

- The code initializes `DocumentFilters` and loads a document `filename.doc` into an `Extractor` instance.
- It uses `IGR_BODY_AND_META` to extract both the document body and metadata.
- The `GetText` method reads the document's content in chunks of 4096 characters, looping until the `EOF` (End of File) is reached.

1.6.5 Converting a Document

After initializing the Document Filters library, you can convert documents into different formats, such as PDF. The following Java code snippet demonstrates how to load a Word document (`.doc` file) and convert it into a PDF using the Document Filters API.

Java

App.java

```

import com.perceptive.documentfilters.*;

public class App {

    private static final String LICENSE_KEY = "YOUR_LICENSE_KEY_HERE";

    public static void main(String[] args) {
        try {
            DocumentFilters api = new DocumentFilters();
            api.Initialize(LICENSE_KEY, ".");

            try (Extractor doc = api.GetExtractor("filename.doc");
                Canvas canvas = api.MakeOutputCanvas("output.pdf", isys_docfilters.IGR_DEVICE_IMAGE_PDF,
                "")) {

                doc.Open(isis_docfilters.IGR_FORMAT_IMAGE);

                for (int pageIndex = 0, pageCount = doc.GetPageCount(); pageIndex < pageCount; ++pageIndex) {
                    try (Page page = doc.GetPage(pageIndex)) {
                        canvas.RenderPage(page);
                    }
                }
            }
        }
    }
}

```



```
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Explanation:

- This code converts `filename.doc` to a PDF by rendering each page into a `Canvas`.
- The extractor is opened with `IGR_FORMAT_IMAGE`, which sets the document for image-based output, triggering pagination.
- Each page of the document is rendered using `RenderPage`, looping through all available pages until complete.

1.7 Getting Stated with Python

Hyland Document Filters provides robust document processing capabilities that can be easily integrated into your Python applications. Follow these instructions to set up your environment.

1.7.1 Clone and Include the Document Filters Repository

The [Document Filters GitHub repository](#) contains the necessary files and libraries.

Note

Before version 22.2, Python bindings for Document Filters were only available through shared object packages, supporting Python versions up to 3.3. Version 22.2 introduced ctypes-based Python bindings, which are compatible with Python versions 3.7 and later, while still supporting the legacy shared object bindings. Starting from version 24.4, the shared object bindings have been deprecated and removed.

1.7.2 Installing the Bindings

The Python bindings are provided as source in the `bindings/python` directory of the Document Filters GitHub [Document Filters GitHub repository](#).

The easiest way to work with them is to install them as a system package, however you can also include them within your application by copying the directory.

You can install the Python bindings in three ways:

1. Install Directly from the Directory

Navigate to the `bindings/python` directory and run the following command:

```
pip install .
```

2. Install from GitHub

You can also install the bindings directly from GitHub using the following command:

```
pip install git+https://github.com/Hyland/DocumentFilters.git@<desired-tag-or-branch>#subdirectory=bindings/python
```

Replace `<desired-tag-or-branch>` with the specific tag or branch you want to use.

3. Add to requirements.txt

If you're using a `requirements.txt` file for your project, you can add the following line to include the Python bindings:

```
DocumentFilters @ git+https://github.com/Hyland/DocumentFilters.git@<desired-tag-or-branch>#subdirectory=bindings/python
```

Again, replace `<desired-tag-or-branch>` with the specific tag or branch you want to use.

After updating the `requirements.txt` file, run the following command to install the dependencies:

```
pip install -r requirements.txt
```

1.7.3 Initializing and calling Document Filters

Once the package is installed, you can begin using it in your application.

Python

```
1 from DocumentFilters import *
2
3 df = Api()
4 df.Initialize("YOUR_LICENSE_KEY_HERE", ".")
```

The code above imports the `DocumentFilters` package into the global scope, then creates an instance of the `Api` class and initializes it with a license key. Replace `"YOUR_LICENSE_KEY_HERE"` with your actual license key.

The second argument specifies the directory where Document Filters should search for resources like configuration files and fonts. The `.` indicates that it should look in the same directory as the Document Filters shared libraries.

1.7.4 Extracting Text

Once the Document Filters library is initialized, you can begin extracting text from documents. The following Python code snippet demonstrates how to load a document and extract its contents using the Document Filters API. This example focuses on extracting text from a Word document (.doc file).

Python

```
1 from DocumentFilters import *
2
3 df = Api()
4 df.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with df.GetExtractor("filename.doc") as doc:
7     doc.Open(IGR_BODY_AND_META)
8
9     while not doc.EOF:
10         text = doc.GetText(4096, True)
11         print(text)
```

In this code snippet, the file `filename.doc` is loaded into an extractor named `doc`. By using a `with` statement, the extractor is automatically closed when it goes out of scope.

The extractor is opened with the `IGR_BODY_AND_META` option, which allows for the extraction of both the document's body text and its metadata.

The loop then repeatedly calls `GetText` until the extractor indicates that it has reached the end of the file (EOF).

1.7.5 Converting a Document

After initializing the Document Filters library, you can convert documents into different formats, such as PDF. The following Python code snippet demonstrates how to load a Word document (`.doc` file) and convert it into a PDF using the Document Filters API.

Python

```
1 from DocumentFilters import *
2
3 df = Api()
4 df.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with df.GetExtractor("filename.doc") as doc:
7     with df.MakeOutputCanvas("output.pdf", IGR_DEVICE_IMAGE_PDF) as canvas:
8         doc.Open(IGR_FORMAT_IMAGE)
9
10    canvas.RenderPages(doc)
```

This code snippet loads `filename.doc` into an extractor named `doc`. Again, using a `with` block ensures that the extractor is closed automatically when it goes out of scope.

Additionally, a new `canvas` object of type `IGR_DEVICE_IMAGE_PDF` is created, which will also be closed automatically when it is no longer needed.

The extractor is opened with the `IGR_FORMAT_IMAGE` option, indicating that the file should be converted into an image-based output, triggering pagination.

Finally, `RenderPages` is called to render each page from `doc` onto the `canvas`. Alternatively, you could iterate over the pages manually and call `RenderPages` for each one.

Did you know?

You can render more than one document to a canvas. If you want to stitch multiple files together, simply load each document into its own `Extractor`, then call `RenderPage/s` onto a single canvas.

1.7.6 Troubleshooting

```
libISYS11df.so: cannot open shared object file: No such file or directory
```

If you see an error similar to above, it means that `Python` was unable to locate the Document Filters Shared Libraries. Standard `dlopen` rules are used to locate the libraries.

This can often be worked around by adding the path containing the libraries to the `LD_LIBRARY_PATH` environment variable.

1.8 About accessibility info extraction

Document Filters supports the extraction of accessibility information from MS Office documents, aiding in the development of accessible products. The extracted "alt text" labels include images, drawings, objects, word art, smart art, charts, icons, shapes, and text boxes across Word (DOC and DOCX), PowerPoint (PPT and PPTX), Excel (XLS and XLSX), and Visio (VSDX) files. The extracted alt text is surfaced in various output formats -- HTML5, Classic HTML, XML, and PDF -- when the appropriate options for creating accessible output are provided.

1.8.1 Output Format Specifics

HTML5 / Classic HTML

To include alt text in HTML5 or Classic HTML output, pass the `HDHTML_ACCESSIBILITY=ON` option as a parameter of the [IGR_Open_File_Ex](#) or [Extractor::Open](#) method.

The alt text is rendered as follows:

- `alt` attribute on `<img>` tags
- `aria-label` attribute on other tags (e.g. `<div>` or `<span>`)

XML

For XML output, no additional options are needed to include alt text. The alt text is represented as `alt-text` attributes within the XML.

PDF

In PDF output, alt text is included when the `PDF_VERSION=PDFA1A` or `PDF_TAGGED=ON` option is specified as a parameter of the [IGR_Open_File_Ex](#) or [Extractor::Open](#) method. The alt text is embedded in the PDF with an `/Alt` entry in the property list for a tag.

1.8.2 Using Page Elements Style API

Alt text is also available as the "altText" style on page elements. You can retrieve this using [IGR_Get_Page_Element_Style](#) to get the "altText" style on a page element acquired with [IGR_Enum_Page_Elements](#), or by using the [PageElement](#) object in the Object API.

1.9 About conversion profiles

The Hyland Document Filters allow you to customize conversion settings through a configuration file named `ISYS11df.ini`. This file should be placed in the executable folder. If present, the settings specified in this file will be applied during processing, unless explicitly overridden by the application in the `Open` or `Canvas` methods.

Conversion profiles enable you to define specific options based on three criteria:

1. **Input File Type:** Different options can be configured for varying document types, such as bitmaps versus textual documents.
2. **Conversion Mode:** You can choose among different conversion modes—text, classic HTML, or HiDef (HD).
3. **Output Format:** The output format can be customized based on your requirements. This flexibility is particularly useful when working with diverse input files, allowing you to tailor the conversion process to suit the characteristics of each document.

1.9.1 Configuring a Conversion Class

A conversion class is a way to map specific file types to a named class. This classification allows you to define rules governing how that group of file types should be processed.

Defining Class Rules

A class rule specifies the options that will be applied to a given file class. These rules are defined in sections within the `ISYS11df.ini` file, following this syntax:

```
[Class:{CLASS_NAME}[:{OUTPUT}]]
```

- **Class:** This part is fixed and indicates that the section defines a class.
- **{CLASS_NAME}:** This is the name of the file type or class as defined in the Classes section.
- **:{OUTPUT} (optional):** This optional segment specifies the output type. If omitted, the settings will apply to all output modes.

Examples:

- `[Class:IMAGE:TEXT]` applies to image file types specifically for text-mode output.
- `[Class:SPREADSHEET]` applies to spreadsheet file types and affects all output modes. Within each section, you can define options using the name=value format. These values serve as default options and can be overridden by the user as needed.

Within each section, you can define options using the `name=value` format. These values serve as default options and can be overridden by the user as needed.

1.10 About custom streams and extended streams

Hyland Document Filters supports customizable streams, allowing you to read from storage systems not natively recognized by Document Filters. For example, you may want to read files directly from a database or an FTP site. Additionally, you can utilize a specialized type of custom stream known as an Extended Stream. Extended Streams help Document Filters retrieve additional information about your stream when needed, such as handling requests for specific parts of a multi-part archive.

1.10.1 Custom Streams in C

In C and C++, you can customize streams by providing an `IGR_Stream` struct populated with pointers to essential functions, including `Read`, `Seek`, and `Write`. For more details, see [IGR_Stream Data Type](#).

If your implementation requires handling callbacks — such as reading multi-part archives from a custom stream — you will need to provide an Extended Stream. An Extended Stream functions similarly to a standard stream but includes a user-defined callback to handle situations when more information is needed or when specific actions must be performed. For further information, see `IGR_Extend_Stream`.

Alternatively, you can create custom streams using [IGR_Make_Stream_From_Functions](#), where you provide callback functions along with the necessary context.

1.10.2 Custom Streams in C++

The C++17 bindings provide a convenient way to create custom streams by inheriting from the `Hyland::DocFilters::Stream` class and implementing the following methods:

- `seek` (required) - Seeks to the specified `offset` in the file.
- `read` (required) - Reads `size` bytes into the buffer.
- `write` (optional) - Supports writing data.

All custom streams **must** be seekable; forward-only read streams are not supported.

Alternatively, if you are working with C++ `std::istream` or `std::iostream`, you can pass these directly to stream functions or create `IGR_Stream` wrappers using the `Stream::bridge` functions.

C++ code can also utilize the C approach for creating streams.

1.10.3 Custom Streams in .NET

In C#, you can create custom streams by extending the `IGRStream` class and implementing the following methods:

- `Seek` (required) - Seeks to the specified `offset` in the file.

- `Read` (required) - Reads `size` bytes into the buffer.
- `Write` (optional) - Supports writing data.
- `GetStreamPart` (optional) - Supports multi-part handling.

C#

```
class IGRStream
{
    public virtual uint Read(uint Size, IGRStream_Data Dest);
    public virtual uint Seek(long Offset, int Origin);
    public virtual uint Write(byte[] bytes, uint size);
    public virtual IGRStream GetStreamPart(string partName, string partFullName, int partIndex);
}
```

If `GetStreamPart` method is not needed, you can also extend `System.IO.Stream`.

Alternatively, if you are working with C# `System.IO.Stream`, you can pass these streams directly to the relevant stream functions.

1.10.4 Custom Streams in Java

In Java, you can create custom streams by extending the `CIGRStream` class and implementing the following methods:

- `Seek` (required): Seeks to the specified offset in the file.
- `Read` (required): Reads `size` bytes into the buffer.
- `Write` (optional): Supports writing data.
- `GetStreamPart` (optional): Supports multi-part handling.

Java

```
class IGRStream
{
    public long Read(long Size, IGRStream_Data Dest);
    public long Seek(long Offset, int Origin);
    public long Write(byte[] bytes);
    public IGRStream GetStreamPart(String partName, String partFullName, int partIndex);
}
```

1.10.5 Custom Streams in Python

In Python, you can create custom streams by extending the `IGRStream` class and implementing the following methods:

- `Seek` (required): Seeks to the specified offset in the file.
- `Read` (required): Reads `size` bytes into the buffer.

- `Write` (optional): Supports writing data.
- `GetStreamPart` (optional): Supports multi-part handling.

Python

```
class IGRStream:
    def Read(self, size, igr_stream_data_dest):
        pass # Implementation goes here

    def Seek(self, offset, origin):
        pass # Implementation goes here

    def Write(self, bytes):
        pass # Implementation goes here

    def GetStreamPart(self, partName, partFullName, partIndex):
        pass # Implementation goes here
```

If the `GetStreamPart` method is not required, you can also extend `io.IOBase`.

1.10.6 Implementation Details

When implementing a custom stream, the following behaviors are expected:

Read

You should read the specified number of bytes, indicated by `size`, into the provided `buffer`. The method should return the actual number of bytes read.

Seek

The `Seek` method moves the file pointer to a new position based on the specified `offset` and `origin`, returning the new absolute position within the file. The `origin` parameter can take the following values:

- `0`: Seek from the beginning of the file.
- `1`: Seek from the current position in the file.
- `2`: Seek from the end of the file.

1.10.7 Write

In the `Write` method, you should write the specified number of bytes from the provided `buffer` to the underlying resource. The method should return the actual number of bytes written.

1.10.8 GetStreamPart

The `GetStreamPart` method is called to provide a stream for the specified file, allowing Document Filters to access the second and subsequent parts of multi-part archives. The parameters are as follows:

- `partName` : The filename of the requested file (if known), without any path information.
- `partFullName` : The complete filename and path of the requested file (if known).
- `partIndex` : The guaranteed part number of the requested file.

You should create and return a new instance of a stream object corresponding to the given `partIndex`.

1.11 About File Type Identification

File type identification is the process of determining a file's format based on its content rather than its extension. This is crucial for accurate processing, security scanning, and compatibility with different software applications. Hyland Document Filters identifies file types by analyzing a file's byte stream, starting with the first 2048 bytes and reading additional data if necessary. It applies a combination of signature-based detection, heuristics, and container recognition to determine the format.

The identification process involves several steps:

- **Profiling the byte stream** – Counting common character and byte groupings, such as brackets, dots, printable characters, punctuation, and whitespace.
- **Detecting container formats** – Recognizing common file structures like ZIP archives and Microsoft Compound File Binary (CFB).
- **Matching known signatures** – Comparing the file's initial bytes against a predefined set of fixed patterns.
- **Analyzing embedded formats** – Identifying file types that are built on top of ZIP or CFB.
- **Applying heuristic detection** – Scanning for common patterns or opcodes and assigning a probability score to determine the most likely format.

In some cases, the first 2048 bytes do not contain enough information to accurately determine the file type. When this happens, Document Filters reads additional data as needed to improve identification.

1.11.1 API

The C API provides functions [IGR_Get_File_Type](#) and [IGR_Get_Stream_Type](#) for file type identification.

The Object APIs enable file type identification through the [Extractor::FileType](#) property and the [Extractor::getFileType](#) method.

The returned file type ID will be one of those described in [File Format IDs](#).

1.11.2 File Type Identification on Limited Data

Hyland Document Filters can identify a file type even from truncated content, reducing the need to provide the entire file.

Using truncated file content for identification offers several key advantages:

- **Reduced bandwidth and data usage** – Especially beneficial for large files, as only a small portion needs to be transferred.
- **Faster processing** – Limits data transfer, allowing quicker identification and reducing wait times.

- **Enhanced efficiency** – Minimizes resource consumption by processing less data without sacrificing accuracy.

In most cases, Document Filters can identify the file type when provided the first 2048 bytes.

If the file type cannot be determined with the first 2048 bytes, Document Filters will identify it as one of three types:

- 267 - Microsoft Compound Binary File (Corrupted) [`cfb_corrupt`]
- 268 - ZIP Archive (Corrupted) [`zip_archive_corrupt`]
- 0 - Unknown [`unknown`]

For `cfb_corrupt` and `zip_archive_corrupt`, Document Filters recognizes the file as a CFB or ZIP archive but determines it is either corrupt or incomplete. More bytes are needed for precise identification.

If Document Filters classifies a file as `unknown`, it means the file type could not be determined from the first 2048 bytes, and no additional format details are available. However, providing the full file may allow Document Filters to correctly identify its type. Some known formats that often return `unknown` when detecting on limited data but can be correctly identified with the complete file include:

- 93 – ISO Disk Image [`iso`]
- 243 – Kodak Photo CD [`pcd_image`]

Example Code

The following example demonstrates how to provide the first 2048 bytes of the file, or more bytes if needed, to Document Filters for file type identification.

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 string filename = "filename.doc";
8 int fileTypeId = 0;
9
10 byte[] buffer = new byte[2048];
11
12 using var fileStream = System.IO.File.OpenRead(filename);
13
14 int bytesRead = fileStream.Read(buffer, 0, buffer.Length);
15
16 if (bytesRead < buffer.Length)
17 {
18     Array.Resize(ref buffer, bytesRead);
19 }
20
21 using var truncatedDoc = api.GetExtractor(buffer);
22
23 fileTypeId = truncatedDoc.FileType;
24
25 const int FILE_TYPE_CFB_CORRUPT = 267;
26 const int FILE_TYPE_ZIP_ARCHIVE_CORRUPT = 268;
27
28 if (fileTypeId == FILE_TYPE_CFB_CORRUPT || fileTypeId == FILE_TYPE_ZIP_ARCHIVE_CORRUPT)
29 {
30     var fileBytes = System.IO.File.ReadAllBytes(filename);
31     using var doc = api.GetExtractor(fileBytes);
32     fileTypeId = doc.FileType;
33 }
34
35 Console.WriteLine("File Type ID: " + fileTypeId);
```

1.12 About Fonts

To effectively render a document into HD format, Document Filters must have access to the appropriate fonts to accurately represent each character on the page. Fonts not only provide the visual representation of characters but also contain size and measurement information necessary for proper content positioning.

The highest quality renditions are produced when the fonts used to create the document are available on the machine performing the conversion. However, this is not always feasible, and substitutions may be necessary. Additionally, fonts do not encompass all Unicode characters (144,697 as of Unicode 14), yet it is expected that text will render correctly, even if the selected font cannot display certain characters.

Document Filters employs various techniques to render documents effectively, independent of format and international writing systems.

1.12.1 Font Substitution

Font Substitution refers to the ability to replace a requested font that is unavailable with one that is available. Document Filters attempts to resolve a font in the following order:

- **Requested font** - as stored in the original document.
- **Font aliases** - registered in Font Mappings (see below).
- **Format dependent mappings** - some file types store extra information to aid in finding substitutes.
- **Unicode Fallback Font**.

1.12.2 Font Fallback

Font Fallback is the mechanism that allows rendering of a character or glyph, even if it is not present in the requested fonts. Very few fonts support a wide range of alphabets, and we do not expect users to know precisely which fonts support which characters. For instance, if a user selects the Arial font to enter text in English, Hebrew, and then Thai, Arial supports Latin and Hebrew characters but lacks Thai glyphs. If Arial cannot render Thai, a fallback font is automatically identified, saving the user from manual intervention. Document Filters resolves a glyph in the following order:

- The requested font or its substitute as determined above.
- The visually closest font that claims support for the requested glyph.
- The Unicode Fallback Font.
- The "missing character" symbol.

The Google Noto Fonts project is particularly valuable for rendering a broad variety of Unicode characters. While Document Filters does not ship with the full set of Noto fonts due to size constraints, if you are deploying your application in low-font environments, consider copying the complete set.

- <https://github.com/googlefonts/noto-fonts>
- <https://github.com/googlefonts/noto-cjk>

1.12.3 Configuring fonts

Document Filters uses an INI file called `fonts.ini` for managing font-specific configurations. This file is loaded once when the library is first initialized, allowing you to specify font settings that will affect document rendering.

The `fonts.ini` will be search for in the following locations

- `$ISYS_FONTS/fonts.ini`
- `$ISYS_INIT_PATH/fonts.ini` (i.e. the path used in the call to `Init_Instance`)
- `$ISYS_INIT_PATH/fonts/fonts.ini`
- `$ISYS_DLL_PATH/fonts.ini` (i.e. the path where the DLL/shared libraries are loaded)
- `$ISYS_DLL_PATH/fonts/fonts.ini`

This order of searching allows for flexibility in configuring fonts, ensuring that users can place their font configurations in the most convenient location.

Example: An example of the contents of a `fonts.ini` file might include font paths or aliases for specific fonts to ensure they are loaded correctly during document rendering.

By appropriately configuring the `fonts.ini` file, you can enhance the rendering quality and ensure that the correct fonts are used for displaying text in your documents.

1.12.4 Finding or Adding Fonts

Document Filters looks for system fonts by scanning common font directories on the host system for OpenType Fonts (OTF, TTF, and TTC). The following directories are scanned by default:

- `$USERPROFILE\AppData\Local\Microsoft\Windows\Fonts`
- `$SystemRoot\Fonts`
- `/usr/share/fonts/truetype`
- `/usr/share/fonts/opentype`

Additionally, you can supply fonts alongside your application without needing to install them into a system location. Document Filters will also look for fonts in a directory called `fonts` that exists in the same directory as the Document Filters' DLLs or in the same directory as your application. If these are the same location, it will only be scanned once:

- `$ISYS_INIT_PATH/fonts`
- `$ISYS_DLL_PATH/fonts`

You can add new fonts in a few ways:

- Install the font using the operating system's preferred method.
- Copy the font into the application's `fonts` directory.
- Add the font directory to the `[FontLocations]` section of the `fonts.ini` file.

fonts.ini

```
[FontLocations]

# Locations to scan for fonts (additional paths may be added here)
$ISYS_INIT_PATH/fonts
$ISYS_DLL_PATH/fonts
$SystemRoot\Fonts/**
$USERPROFILE\AppData\Local\Microsoft\Windows\Fonts
/usr/share/fonts/truetype/**
/usr/share/fonts/opentype/**
/opt/path-to-more-fonts/**
```

The directories specified will be scanned in order. If a font exists in more than one location, the first one found will be used. Names starting with `$` are considered environment variables and will be substituted at load time.

To recursively load a directory, append `**` to the end of the path.

By effectively managing font locations, you can enhance the rendering quality of documents processed with Document Filters.

1.12.5 Font Mappings and Aliases

Document Filters will attempt to render with an exact match for any referenced font. If the font does not exist on the system, it may fall back to a different font or be aliased to another font.

Font mappings can be added to or edited by modifying the `[FontMappings]` section of the `fonts.ini` file.

fonts.ini

```
[FontMappings]

# Canonical font types (each in order of preference)
$ISYS_FONTS_SANS_SERIF=Arial;Liberation Sans;Droid Sans
$ISYS_FONTS_SERIF=Times New Roman;Liberation Serif;Droid Serif
$ISYS_FONTS_MONO=Courier New;Liberation Mono;Droid Mono
```

```
$ISYS_FONTS_SANS_UNICODE=Arial Unicode;Arial Unicode MS;Droid Sans Fallback;Unifont
$ISYS_FONTS_SYMBOL=Symbol;OpenSymbol

# Font substitutions (if the exact font is not available on the system)
Arial;Helvetica=$ISYS_FONTS_SANS_SERIF
Times;Times New Roman=$ISYS_FONTS_SERIF
Courier;CourierPS;Courier New=$ISYS_FONTS_MONO
Arial Unicode;Arial Unicode MS=$ISYS_FONTS_SANS_UNICODE
Symbol=$ISYS_FONTS_SYMBOL
```

To add a new font mapping, use the following format:

```
OriginalFontName[;SecondaryFontName]=FontAlias1[;FontAlias2;FontAlias3;FontAliasN]
```

Document Filters will create an alias from OriginalFontName to the first font alias found.

There is a special alias installed called \$ISYS_FONTS_SANS_UNICODE; this font is used to render Unicode text if the selected font does not support Unicode characters.

Example Use Cases

- **User Scenario:** If a document references "Comic Sans," but it's not available on the system, you can add a mapping to substitute it with a similar font like "Arial" to ensure proper rendering.
- **Unicode Handling:** For documents containing mixed language text, the \$ISYS_FONTS_SANS_UNICODE alias ensures that characters from different scripts are rendered correctly even if the primary font does not support them.

By utilizing font mappings and aliases, users can enhance the flexibility of document rendering in Document Filters, accommodating various fonts and character sets effectively.

1.12.6 Character Mapping

Character mapping is a critical feature of Document Filters that allows for the rendering of characters when the original font is not available. This is particularly important for symbol-based fonts, such as WingDings and Symbol. When a full Unicode font is available and is listed as `ISYS_FONTS_SANS_UNICODE` in the `fonts.ini` file, fallback and substitution will occur automatically, ensuring that characters render correctly even in the absence of the specified font.

Automatic mapping is provided for WingDings and WingDings 2, and to a lesser extent, WebDings. This feature helps maintain the integrity of the document's visual presentation when original fonts are missing.

To build a character mapping table, modify the `fonts.ini` file to add a `[CharMappings]` section, with one line per mapping in the form of:

fonts.ini

```
FontName:CodePoint=UnicodeCodePoint
```

Example

For instance, you can map specific character codes from the Symbol font to their corresponding Unicode values as follows:

```
fonts.ini

[CharMappings]
Symbol:183=8226
Symbol:167=9827
```

Use Cases

- **Document Compatibility:** If a document created with the Symbol font references special characters (e.g., bullet points or symbols) that are not available on the system, character mapping allows these symbols to render using their Unicode equivalents.
- **User Convenience:** Users can add custom mappings for any special characters needed in their documents, ensuring consistent rendering across different environments and font installations.

Limitations

While character mapping is a powerful tool, it is important to note that not all fonts or symbols may be adequately represented. Users should ensure that mappings are correct and that necessary Unicode fonts are available for full compatibility.

1.12.7 Diagnostics

Document Filters provides a diagnostics feature that displays information about loaded fonts, their locations, and the font fallback and aliasing processes that occur during document rendering. This functionality is essential for troubleshooting font-related issues and ensuring that documents render correctly across different environments.

To enable diagnostics, set the environment variable `ISYS_FONTS_DIAG` to a numeric value:

- `1` - Prints a list of all loaded fonts, including their name, style, and filename.
- `2` - Prints the same font loading information as above, along with details about font aliasing and fallbacks.

The information is printed to Standard Out, allowing users to easily review the font loading and mapping processes.

1.13 About Multi-Part Archives

Multi-part archive files, such as certain ZIP and RAR formats, are comprised of two or more files packaged together. Document Filters supports the processing of these multi-part archives, allowing users to easily manage complex file structures.

To process a multi-part archive, the API user needs only to submit the first file to Document Filters. The system will then handle the extraction of child files as if it were an ordinary single-part archive. When reading multi-part archives from an ordinary file system on disk, the second and subsequent files are processed automatically without additional effort from the API user.

1.13.1 Limitations in Non-File System Contexts

If the later parts of the archive are not directly stored on a file system, they do not process automatically. Document Filters cannot assume the location of these files. For instance, a multi-part ZIP file attached to an email or nested within another archive is treated as individual files and requires special handling.

1.13.2 Processing Multi-Part Archives with Extended Streams

To process multi-part archives that are not directly on a file system, you need to create an Extended Stream. This stream assists Document Filters in locating the later parts of the archive. If the API user attempts to process secondary parts directly, those parts will be identified, but no further processing will occur.

When using an Extended Stream, Document Filters invokes the nominated callback function whenever a later part is required. In this scenario, the callback function receives parameters where:

- **actionID** = IGR_ACTION_GET_STREAM_PART
- **actionData** = a struct of type [IGR_T_ACTION_GET_STREAM_PART](#).

For more information, see [IGR_T_ACTION_GET_STREAM_PART Data Type](#)

1.14 About Multithreading

Document Filters may be run in a multithreaded application with minimal effort. Following a basic set of rules is essential to avoid issues such as data corruption or race conditions.

- **A document/extractor MUST be processed in a single thread only.** This ensures that state and data integrity are maintained throughout the processing.
- **A document can move between threads, but the caller is responsible for ensuring that only a single thread is calling an API for any given handle.** This means that care must be taken to synchronize access when transferring documents between threads.
- **These rules also apply to sub-files**, unless the subfile is saved to a file or stream first and then loaded as a new document. In such cases, the new document handle can be safely processed in a different thread.

Example: If you are processing a main document in Thread A and want to analyze its subfile in Thread B, ensure that all API calls for the subfile are managed by Thread B after it is saved and loaded as a new document.

By adhering to these guidelines, you can effectively implement multithreading in your applications while leveraging the full capabilities of the Hyland Document Filters.

1.15 About Optical Character Recognition (OCR)

Document Filters provides comprehensive support for Optical Character Recognition (OCR), enabling users to extract text from a variety of document formats. It includes built-in support for the Tesseract OCR engine, which is a widely-used open-source OCR solution. Additionally, Document Filters allows for the use of other versions of Tesseract, giving you the flexibility to choose the version that best fits your needs.

Moreover, Document Filters is designed to integrate with external OCR engines, providing even greater versatility in processing documents. This means you can configure Document Filters to work with any OCR solution that meets your requirements, whether it's a different version of Tesseract or a completely separate OCR engine.

You can enable the functionality by passing the `OCR=ON` option as a parameter of the `IGR_Open_File_Ex` or `Extractor::Open` method. This option is available for both text-mode and high-definition outputs.

It is important to note that invoking the `OCR=ON` option will have no effect on formats that are not supported.



Supported Graphic Types

BMP, BRK, CALS, CGM, CUR, DCX, EMF, EPS, GEM, GIF, ICO, IFF, IMNET, JEDICS, JPG, JPK, JXR, MACPAINT, MDI, MSPAINT, NCR, PBM, PCX, PICT, PNG, PSP, SCANNED PDFS, SVM, TGA, TIFF, WBMP, WEBP, WMF, WPG, XBM, XPM, XWD.

When processing a PDF file, only pages that do not contain a text layer will be processed by the OCR engine.

For optimal results, ensure that the input images have sufficient quality, with a recommended resolution of at least 300 dpi. Document Filters includes features to automatically detect images that may be too low in resolution for effective OCR, helping you avoid potential issues during processing.

To enhance the effectiveness of the OCR process, Document Filters skips any image with a width of less than 1,000 pixels. You can adjust this threshold by specifying `OCR_MIN_WIDTH` in the document open flags.

1.15.1 Using the built-in version of Tesseract

Document Filters integrates with the [Tesseract OCR engine](#) 3.02.02 to provide OCR as an optional processing step to extract text from document image formats.

Prerequisites

To utilize the built-in version of Tesseract, ensure that the base set of training data is available. This data is provided in the `ISYSreadersocr.dat` file, included in the `assets.zip` archive available on GitHub with each release. The `ISYSreadersocr.dat` file contains the English and OSD (Orientation and Script Detection) training sets.

The `ISYSreadersocr.dat` file must be present in the directory specified in the call to either `Initialize` or `Init_Instance`. The simplest approach is to place the file in the same directory as the `ISYS11df.[dll|so|dylib]` file and pass `.` as the directory argument in the initialization call.

Language Support

To perform OCR on documents in other languages, you can manually install the necessary dictionaries from the Tesseract OCR project site. Ensure that you download the language files that match the version of Tesseract in use (3.02.02). To install a dictionary file, place the `.traineddata` file in your application directory.

To specify the language for OCR processing, set the `OCR_LANGUAGE=xyz`, where `xyz` is the 3-digit language code. For example:

- English = `eng`
- French = `fra`
- German = `deu`

Why Tesseract 3.02.02

The embedded version of Tesseract, 3.02.02, is selected for compatibility across all platforms supported by Document Filters. This specific release ensures optimal support and consistent output. However, you can also configure Document Filters to utilize a system-installed version of Tesseract if preferred.

Disclaimer

Document Filters employs an unaltered version of the Tesseract OCR engine source code and does not guarantee the accuracy of the recognized text or the performance of the engine. While this integration aims to streamline your experience, Tesseract OCR remains a third-party engine and is not a native component of Document Filters. Users are encouraged to integrate any OCR engine that meets their specific needs.

Example Usage in C#

With the `ISYSreadersocr.dat` in place, you can enable OCR in your code by passing the `OCR=ON` option when opening a document.

C#

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.OpenExtractor("image.tiff", OpenMode.Paginated, options: "OCR=ON");
7
8 while (!doc.EndOfStream)
9 {
10     var text = doc.GetText(4096);
11     Console.Out.WriteLine(text);
12 }
```

1.15.2 Using a different version of Tesseract

As mentioned earlier, the version of Tesseract bundled with Document Filters is 3.02. You have the flexibility to set up Document Filters to utilize a locally installed version of Tesseract on the host system. When opting for the host system's Tesseract, ensure it aligns with your application or environment's architecture (e.g., 32-bit, 64-bit) and is supplied as a Shared Library or DLL.

The installation process for Tesseract depends on your operating system. Refer to the [Tesseract Installation Documentation](#) for details.

To setup Document Filters to support different versions of Tesseract, you must create configure it with the details of the alternate OCR engine. This can be done through either INI file or Environment Variables.

Before you start, you will need to know the name and path of the shared library. This can be found by running `find /usr -iname "*tesseract*.so"`. For example, Ubuntu 20.04 reports as `/usr/lib/x86_64-linux-gnu/libtesseract.so`.

You can either replace the default OCR engine with the system version, or add a new "named" OCR engine, which can then be specified with the `OCR_ENGINE={name}` setting. The approach to registering the engine is identical, and is keyed off of the name. Naming your engine `TESSERACT` will replace the default engine.

**Disclaimer**

Document Filters includes a convenient pass-through integration with the open-source Tesseract OCR engine as an ancillary service. While we provide this integration to streamline your experience, Tesseract OCR remains an unaltered, third-party engine and is not a native component of Document Filters. Document Filters makes no claims or warranties regarding the accuracy of the recognized text or the performance of Tesseract OCR, or any other OCR engine. Document Filters users are welcome to integrate any OCR engine of their choice to suit their specific needs.

Registering via INI file

Registering an OCR engine via the `ISYS11df.ini` file is the simplest way to permanently set up the engine for the host system, and it's easy to manage across environments. Multiple OCR engines can be registered, each in its own unique section prefixed with `OCR:` (e.g., `[OCR:tesseract5]`).

`ISYS11df.ini`

```
[OCR:tesseract5]
enabled=true
tesseract_lib=libtesseract.so.5
tesseract_dll=tesseract53.dll
tesseract_properties=textord_min_blobs_in_row=4 textord_spline_minblobs=8
```

Each value within the section is described below:

Name	Description
enabled	Enables or disables the OCR engine. When set to <code>true</code> , the engine can be used by the application.
tesseract_lib	Specifies the name of the Tesseract shared library for Linux/macOS. This can be a fully qualified path or relative, relying on standard shared library discovery rules.
tesseract_dll	Specifies the name of the Tesseract DLL for Windows. This can be a fully qualified path or relative, relying on standard DLL discovery rules.
tesseract_properties	A space-separated list of options to pass to Tesseract during initialization. These options can control Tesseract behavior, such as text layout analysis. For example, <code>textord_min_blobs_in_row</code> and <code>textord_spline_minblobs</code> adjust text line detection parameters.

Registering via Environment Variables

Registering an OCR engine using environment variables is particularly convenient when working with containerized environments or testing different configurations without modifying configuration files. This method allows you to quickly switch between different OCR engines and configurations.

This method is ideal for scenarios such as containerized deployments where modifying configuration files is not practical, or for quickly testing different OCR engines without needing to restart or reconfigure the application.

The environment variable names must follow the pattern `DOCFILTERS_OCR_{NAME}__{PROPERTY}`, where `{NAME}` is the OCR engine's name, and `{PROPERTY}` corresponds to a setting from the INI file. Both `{NAME}` and `{PROPERTY}` must be uppercase, and a double underscore (`__`) separates them.

Name	Description
<code>DOCFILTERS_OCR_{NAME}__ENABLED</code>	Enables or disables the OCR engine. When set to <code>true</code> , the engine can be used by the application.
<code>DOCFILTERS_OCR_{NAME}__TESSERACT_LIB</code>	Specifies the name of the Tesseract shared library for Linux/macOS. This can be a fully qualified path or relative, relying on standard shared library discovery rules.
<code>DOCFILTERS_OCR_{NAME}__TESSERACT_DLL</code>	Specifies the name of the Tesseract DLL for Windows. This can be a fully qualified path or relative, relying on standard DLL discovery rules.
<code>DOCFILTERS_OCR_{NAME}__TESSERACT_PROPERTIES</code>	A space-separated list of options to pass to Tesseract during initialization. These options can control Tesseract behavior, such as text layout analysis. For example, <code>textord_min_blobs_in_row</code> and <code>textord_spline_minblobs</code> adjust text line detection parameters.

PRECEDENCE OF ENVIRONMENT VARIABLES

Environment variables override any values stored in the INI file. If both an environment variable and a setting in the INI file define the same property, the environment variable takes precedence and will be applied at runtime.

EXAMPLE CONFIGURATION

To register a new OCR engine called `tesseract5` using environment variables, the following variables would be set:

```
export DOCFILTERS_OCR_tesseract5__ENABLED=true
export DOCFILTERS_OCR_tesseract5__TESSERACT_LIB="libtesseract.so.5"
export DOCFILTERS_OCR_tesseract5__TESSERACT_DLL="tesseract53.dll"
export DOCFILTERS_OCR_tesseract5__TESSERACT_PROPERTIES="textord_min_blobs_in_row=4 textord_spline_minblobs=8"
```

USING ENVIRONMENT VARIABLES IN CONTAINERS

When working with containers, you can inject these environment variables during container startup. For example, in Docker:

```
docker run -e DOCFILTERS_OCR_TESS5__ENABLED=true \
-e DOCFILTERS_OCR_TESS5__TESSERACT_LIB=/usr/local/lib/libtesseract.so.5 \
your-container-image
```

Or in Kubernetes, you can define these in a ConfigMap:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: ocr-config
data:
  DOCFILTERS_OCR_TESS5__ENABLED: "true"
  DOCFILTERS_OCR_TESS5__TESSERACT_LIB: "/usr/local/lib/libtesseract.so.5"
```

Example Usage in C#

After configuring the OCR engine, you can enable OCR in your code by passing the OCR=ON option when opening a document.

C#

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.OpenExtractor("image.tiff", OpenMode.Paginated, options:
7   "OCR=ON;OCR_ENGINE=tesseract5");
8
9 while (!doc.EndOfStream)
10 {
11     var text = doc.GetText(4096);
12     Console.Out.WriteLine(text);
13 }
```

1.15.3 Using a Different OCR Engine via Callback

Document Filters can integrate with alternative OCR engines by invoking a callback into your application. This allows you to tightly integrate an OCR engine written in the same language as your application.

To provide a custom OCR engine, implement a callback for the `OcrImage` hook, or handle the `IGR_Open_Ex` callback for the `IGR_OPEN_CALLBACK_ACTION_OCR_IMAGE` action. This mechanism is available in C, C++17, Java, .NET, and Python.

Next, enable OCR support by passing `OCR=on` when opening the file, and set `OCR_ENGINE=CUSTOM` to trigger your callback.

Your application will then be called whenever an image needs to be OCR'd.

At that point, your application is responsible for calling `AddText` for each word, along with its bounding box and any optional style information. If your OCR engine supports layout detection, you can also call `StartBlock` and

`EndBlock` to mark the start and end of paragraphs, columns, or other layout regions. If it does not, Document Filters' built-in layout detection will be used instead.

1.15.4 Using a different OCR engine via External Process

Document Filters can integrate with alternative OCR engines by invoking an external process. To ensure proper interaction with the OCR engine, the engine must meet the following requirements:

- Read **PNG files** from disk.
- Write **HOCR** output to disk.

Configuration for External OCR Engines

Registering an external OCR engine uses a similar configuration format as the Tesseract engine but includes additional options specific to the external process:

Name	Description
exe	The executable name for the OCR engine on Windows.
proc	The executable name for the OCR engine on all other platforms.
args	Command-line arguments to pass to the external process.
output_ext	The optional file extension appended to the output filename.

COMMAND-LINE ARGUMENT VARIABLES

The `args` value can utilize specific variables that will be expanded at runtime:

Variable	Description
<code>\${inputFile}</code>	The filename of the temporary file that contains the image to be processed.
<code>\${outputFile}</code>	The filename of a temporary file where the output is to be written.
<code>\${language}</code>	The OCR language as passed from the user.

EXAMPLE CONFIGURATION

To configure an OCR engine that utilizes the Tesseract command-line interface instead of a shared library, you might set up your configuration like this:

```
ISYS11df.ini

[OCR:tesseract_cli]
enabled=true
```

```
exe=tesseract.exe
proc=/usr/bin/tesseract
args=-c "tessedit_create_hocr=1" -c "hocr_font_info=1" -l "${language}" "${inputFile}" "${outputFile}"
output_ext=.hocr
```

In this example:

- The options `-c "tessedit_create_hocr=1"` and `-c "hocr_font_info=1"` enable the HOCR output format.
- The Tesseract CLI automatically appends a `.hocr` suffix to the output filename, which is why we specify `output_ext=.hocr` in the configuration.

ACTIVATING THE EXTERNAL OCR ENGINE

To utilize this external engine within your application, you can set the options when opening a document:

```
"OCR=on;OCR_ENGINE=tesseract_cli"
```

TROUBLESHOOTING EXTERNAL OCR ENGINES

If you encounter issues when using a different OCR engine, consider the following steps:

1. **Verify Executable Paths:** Ensure that the paths for `exe` and `proc` are correct and that the executable is accessible from the environment where Document Filters runs.
2. **Check Arguments:** Review the arguments passed in the `args` configuration to confirm they are valid and properly formatted. Pay attention to quotes and escape characters if needed.
3. **Output File Permissions:** Ensure that the process has the necessary permissions to write to the output file location.
4. **Logging and Error Messages:** Enable detailed logging in your application to capture any error messages returned by the external OCR engine. This can provide insights into what may be going wrong.
5. **Testing the External Process:** Manually test running the OCR engine from the command line with the same arguments to ensure it functions correctly outside of the Document Filters context.
6. **Review HOCR Output:** After processing, inspect the generated HOCR output for correctness and completeness. If issues persist, consider adjusting the command-line options based on Tesseract's documentation.

1.15.5 Inline OCR of Embedded Images

In addition to performing OCR on full-page raster inputs (for example, scanned TIFF pages or image-only PDF pages), Document Filters can now extract text from images that are embedded inline within supported document formats. This capability helps retain meaningful textual content that appears inside screen captures, diagrams, slide artwork, pasted photos, and other inline graphics.

Enable this behavior by supplying the options:

```
OCR=ON;OCR_INLINE_IMAGES=ON
```

You may combine this with any other OCR options (such as `OCR_MIN_WIDTH` or `OCR_REORIENT_PAGES`). When enabled, each qualifying embedded image is passed through the OCR pipeline and any recognized text is merged into the output flow (e.g., Markdown, JSON, HTML overlay) without altering the visual appearance of the original rendered image content.

Supported Formats

Inline image OCR is currently supported for: email and web archive formats (HTML, MSG, EML, MHT, ICS/VCAL, OLK15) and office productivity formats (Apple Keynote, Apple Numbers, Apple Pages, HWPX, OpenOffice, RTF, Microsoft Word, WordPerfect, Microsoft Visio, Microsoft PowerPoint).

Related Options

Option	Description
<code>OCR_INLINE_IMAGES</code>	Enables OCR on embedded (inline) images inside supported document pages. Default: <code>OFF</code> . Requires <code>OCR=ON</code> .
<code>OCR_MIN_WIDTH</code>	Minimum pixel width for an image to be considered for OCR. Smaller images are skipped (default threshold documented above).
<code>OCR_REORIENT_PAGES</code>	Attempts to detect / correct rotation prior to recognition (applies to inline images as well).

Notes:

- If `OCR` is not enabled, `OCR_INLINE_IMAGES` is ignored.
- Images are only pre-filtered by width (`OCR_MIN_WIDTH`). If an image meets the width threshold but contains no readable text (e.g., a low-contrast logo, photograph, gradient, or decorative icon), the OCR engine may simply return no words; this is expected and not an additional skip rule.
- Caching strategies within the implementation prevent re-OCR of identical embedded images encountered multiple times in a single document session.

Example (C#) – Extracting Markdown With Inline Image OCR

```
using Hyland.DocumentFilters;  
  
var api = new Api();  
api.Initialize("YOUR_LICENSE_KEY", ".");
```

```
using var doc = api.OpenExtractor("example.docx", OpenMode.Paginated,
    options: "OCR=ON;OCR_INLINE_IMAGES=ON;OCR_MIN_WIDTH=800");

while (!doc.EndOfStream)
{
    var md = doc.GetText(8192); // Markdown output
    System.Console.WriteLine(md);
}
```

Example (Command Options)

```
OCR=ON;OCR_INLINE_IMAGES=ON;OCR_MIN_WIDTH=800
```

When to Use Inline Image OCR

Use `OCR_INLINE_IMAGES=ON` when downstream workflows (search, summarization, semantic enrichment, AI classification) benefit from textual content that is visually embedded but not otherwise present in the document's natural text layer. Leaving it `OFF` avoids extra processing when inline graphics are unlikely to contain meaningful text.

Performance & Tuning

Keep runtime in check by:

- Enabling `OCR_INLINE_IMAGES` only where inline image text adds value.
- Raising or lowering `OCR_MIN_WIDTH` to balance coverage vs. cost.
- Using `OCR_CACHE` for repeated / duplicate documents.

1.16 Content Enrichment

1.16.1 Content Enrichment

The Content Enrichment section introduces key features that enhance the processing of documents by identifying and extracting valuable structural elements such as tables, headers, and footers. These capabilities allow users to transform raw document data into a more organized and meaningful format, facilitating improved data extraction and analysis. By leveraging these features, you can enrich your documents with critical insights and streamline workflows, making it easier to work with complex content.

In This Section

Table Detection	Table Detection is a powerful feature of the Hyland Document Filters that enables automatic extraction and processing of tables from various document types. This functionality is particularly valuable when working with file formats that inherently store structured tabular data, such as Microsoft Office documents (Word, Excel, PowerPoint) and other similar productivity applications.

1.16.2 Table Detection

Table Detection is a powerful feature of the Hyland Document Filters that enables automatic extraction and processing of tables from various document types. This functionality is particularly valuable when working with file formats that inherently store structured tabular data, such as Microsoft Office documents (Word, Excel, PowerPoint) and other similar productivity applications.

- **Automatic Extraction:** In supported file types, Document Filters can automatically identify and extract table structures, preserving the integrity of the data and ensuring that it is presented in a meaningful format. This automatic extraction allows users to work with the data efficiently, facilitating tasks such as data analysis, reporting, and content manipulation.
- **Processing Tables in Flattened Formats:** Table detection becomes even more crucial when dealing with formats like PDF. In these formats, the original document structure may be lost, and the content is often flattened into a series of drawing instructions. This makes it challenging to identify tables directly. However, Document Filters employs advanced detection algorithms to reconstruct tables from the available data, ensuring that users can access and utilize table information even when it is not explicitly formatted.

Benefits

By incorporating Table Detection into your document processing workflows, you can achieve the following benefits:

- **Enhanced Data Accessibility:** Extracted tables can be easily accessed and manipulated, improving overall data usability.
- **Improved Data Integrity:** Automatically detected tables help maintain the original context and relationships of the data.
- **Streamlined Workflows:** By automating the table extraction process, users can save time and reduce manual effort in preparing data for analysis or reporting.

Enabling Table Detection for PDFs

To utilize this functionality, you must enable table detection when opening a PDF document. Once enabled, Document Filters will automatically detect tables during processing, ensuring that they are accurately represented in the output. This feature is particularly beneficial when generating Markdown or JSON output formats.

The calling sequence:

1. Open the PDF file with the `PDF_TABLE_DETECTION=on` option enabled.
2. Create a Markdown or other canvas type.
3. Render the pages to the canvas.
4. Verify the output.

Explore our tutorials on creating Markdown output with table detection:

- [How do I convert a PDF to Markdown with Table Detection?](#)

PDF Table Detection Options

To customize the table detection process in PDF files, several options are available. Each option allows you to fine-tune how tables are detected and processed, ensuring optimal results based on the specific layout and structure of your PDF documents. Below are the available options:

Option Name	Description	Default Value	Type
<code>PDF_TABLE_DETECTION</code>	Enable or disable table detection.	OFF	bool
<code>PDF_TABLE_EDGE_MIN_LENGTH</code>	Set the value of the minimum edge length considered when detecting table edges in PDF files.	11	number
<code>PDF_TABLE_HORZ_STRATEGY</code>	Set the value of horizontal table detection strategy for detecting tables in PDF files.	LINES	string:enum

Option Name	Description	Default Value	Type
PDF_TABLE_INTERSECTION_TOLERANCE	Set the tolerance value to determine edge combination into cells.	3	number
PDF_TABLE_INTERSECTION_X_TOLERANCE	Set the value of edge intersection tolerance along the x-axis.	3	number
PDF_TABLE_INTERSECTION_Y_TOLERANCE	Set the value of y-axis tolerance for edge intersection when detecting tables in PDF files.	3	number
PDF_TABLE_JOIN_TOLERANCE	Set the value of tolerance for joining line segments when detecting tables in PDF files.	3	number
PDF_TABLE_JOIN_X_TOLERANCE	Set the x-axis tolerance for joining line segments.	3	number
PDF_TABLE_JOIN_Y_TOLERANCE		3	number

Option Name	Description	Default Value	Type
	Set the y-axis tolerance for joining line segments.		
PDF_TABLE_MIN_CELL_HEIGHT	Set the minimum cell height for table detection in PDF files.	8	number
PDF_TABLE_MIN_CELL_WIDTH	Set the minimum cell width for table detection in PDF files.	8	number
PDF_TABLE_MIN_COLS	Set the minimum number of columns required for table detection in PDF files.	2	number
PDF_TABLE_MIN_ROWS	Set the minimum number of rows required for table detection in PDF files.	1	number
PDF_TABLE_SNAP_TOLERANCE	Set the line tolerance to control parallel line snapping for	3	number

Option Name	Description	Default Value	Type
	detecting tables in PDF files.		
PDF_TABLE_SNAP_X_TOLERANCE	Set the x-axis tolerance for snapping parallel lines when detecting tables in PDF files.	3	number
PDF_TABLE_SNAP_Y_TOLERANCE	Set the y-axis tolerance for snapping parallel lines when detecting tables in PDF files.	3	number
PDF_TABLE_VERT_STRATEGY	Set the value of vertical table detection strategy for detecting tables in PDF files.	LINES	string:enum

1.17 Output Formats

1.17.1 Output Overview

Hyland Document Filters offers a variety of output modes, including text, images, PDF, and XML. This section delves into some of these options in more detail.

In This Section

JSON Output	Hyland Document Filters provides robust options for generating JSON output from various document formats. By configuring the output settings, you can customize the generated JSON to suit your specific needs, from data structures to metadata inclusion.
Markdown Output	Hyland Document Filters provides powerful options for generating Markdown output from various document formats. By configuring the output settings, you can customize the generated Markdown to suit your specific needs, from table styles to metadata inclusion.
PDF Output	Hyland Document Filters offers robust features for generating PDF output from a variety of document formats. By adjusting the output settings, you can tailor the resulting PDF to meet your specific requirements, including layout preferences, security features, and metadata inclusion.

1.17.2 About JSON Output

Hyland Document Filters provides robust options for generating JSON output from various document formats. By configuring the output settings, you can customize the generated JSON to suit your specific needs, from data structures to metadata inclusion.

Hyland Document Filters supports three JSON schemas: Full, Simplified, and MDAST. The Full schema contains the complete object hierarchy, representing the original document's DOM. The MDAST schema is a JSON representation of the Markdown output. The Simplified schema flattens the structure, making it ideal for use with AI applications.

What is JSON

JSON (JavaScript Object Notation) is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate. It is widely used for APIs, configuration files, and data storage due to its simplicity and interoperability with various programming languages.

JSON's structured format, with its key-value pairs and arrays, makes it more than just a data representation tool. Its flexibility and ease of use make it an ideal format for various advanced applications in AI, analytics, and data processing systems. Here are some key use cases for JSON beyond traditional data interchange:

1. JSON in AI and Machine Learning Systems

2. **Training Data Generation:** JSON, with its structured format, is useful for generating training data for AI models that require labeled datasets. Its ability to represent complex data structures helps in organizing data hierarchically, making it easier to identify features and relationships.

- **Example:** A machine learning model trained to classify documents can easily parse JSON to focus on relevant fields and values.

3. **Natural Language Processing (NLP) Tasks:** JSON documents can be preprocessed by NLP engines for various tasks, such as sentiment analysis, named entity recognition, or topic modeling. Its clear structure allows AI models to focus on content while ignoring irrelevant information.

4. **AI-Driven Data Transformation:** JSON is used in systems where AI transforms data from one format to another, such as converting datasets into different structures or summarizing large datasets. JSON's format facilitates easier processing and manipulation of data.

5. JSON in Analytics and Reporting Systems

6. **Structured Data Aggregation:** In analytics and reporting systems, JSON is often used as a lightweight way to store structured data, allowing analysts to include explanations alongside visualizations. JSON's text format enables easy integration with reporting tools that generate automated reports.

7. **Data-Driven Documentation:** JSON is ideal for automatically generating documentation that evolves as data changes. When combined with automated tools, JSON can be used to create real-time updated reports, financial documents, or data dashboards where content is generated based on underlying data models.

8. JSON in Data Processing Systems

9. **Data Comparison and Diffing:** JSON's text-based format is ideal for data comparison (diffing) systems, which need to detect changes between different data versions. Since JSON is plain text, it can be diffed more easily than complex formats like XML or proprietary formats. Systems can highlight changes in data structure or content, facilitating version control.

10. **Content Cleaning and Normalization:** In data processing pipelines, JSON can be used as an intermediate format for content cleaning and normalization. JSON simplifies the structure of datasets, allowing processing systems to focus on cleaning up data, such as removing duplicates or formatting inconsistencies before further analysis.

11. **Automated Documentation Generation:** JSON serves as a common format in automated documentation generation systems where structured content like API responses, configurations, and data specifications are programmatically converted into documentation. Systems can dynamically generate JSON based on data inputs, enabling easy updates.

Creating JSON Output

Generating JSON output with Hyland Document Filters is a powerful feature that allows you to create structured data representations from various source formats. Please note that JSON creation is supported exclusively in Hi-Def mode, which ensures that the document's structure and semantics are preserved accurately during the conversion process.

To create JSON output in Hi-Def mode, follow these steps:

1. **Select Your Source Document:** Choose the document you wish to convert to JSON. Ensure it is a supported format for Hi-Def mode.
2. **Configure Output Settings:** Set your desired output options, including:
 3. Security settings (e.g., permissions for accessing the data)
 4. Metadata to include (e.g., title, author, keywords)
5. **Create JSON Canvas:** Ensure that Hi-Def mode is selected in your conversion settings to define the schema for the JSON output. This schema will be used to accurately represent the data structure of the source document.
6. **Render Data to Canvas:** Initiate the rendering process, where the system will convert each relevant element of the document to the JSON format, ensuring fidelity in structure and content.
7. **Review the Output:** Once the JSON is created, review the data to ensure that the structure, metadata, and overall representation meet your expectations.

By following these steps, you can effectively create professional-grade JSON documents that retain the integrity of your original content, making them suitable for sharing, processing, and integration with other systems.

Explore our tutorials on creating JSON output to enhance your workflow:

- [How do I convert a document to a JSON file?](#)

JSON Schema and Formatting

This section outlines the options available for configuring the JSON schema and formatting in Hyland Document Filters. It includes details on how to select the output schema for Markdown generation and whether to enable formatted output for improved readability.

JSON SCHEMA OPTIONS

JSON_OUTPUT_SCHEMA

`JSON_OUTPUT_SCHEMA` determines the schema to be used when generating JSON output.

- **FULL:** Represents the complete document object model.
- **PIPELINE:** Represents a simplified, flatter version of the document object model.

- **MDAST**: Provides a JSON representation of Markdown output.

JSON_FORMAT_OUTPUT

`JSON_FORMAT_OUTPUT` specifies whether the JSON output should be formatted with newlines and indentation. While this increases the file size, it enhances human readability.

- **true**: Output will include newlines and indentation.
- **false**: Output will not include newlines and indentation.

MDAST SCHEMA AND LOCATION INFORMATION

The **MDAST** schema provides a JSON representation of the Markdown output and is ideal for rendering Markdown documents with structure that can be easily parsed. A key feature of the MDAST schema is the inclusion of location information when the `JSON_INCLUDE_BOUNDS=ON` or `MARKDOWN_INCLUDE_LOCATIONS=ON` option is enabled.

Location Metadata

When `JSON_OUTPUT_SCHEMA` is set to **MDAST**, and `JSON_INCLUDE_BOUNDS` or `MARKDOWN_INCLUDE_LOCATIONS` is enabled (set to `ON`), the output includes **location metadata** for block elements (e.g. "type" of "heading", "paragraph", "list", "listitem", "table", "tableRow", "tableCell") and images. This metadata is helpful for mapping content back to its physical position in the document, such as in rendered PDF or other visual formats. The location metadata can include:

- **File Name**: Optionally included if `JSON_INCLUDE_DOC_METADATA_PER_ELEMENT` is enabled.
- **Page Number**: The page on which the element appears.
- **Bounding Box**: The position of the element in the form of `[left, top, right, bottom]` coordinates, indicating its location on the page.

Example Structure with Location Information

```
{
  "type": "heading",
  "location": {
    "filename": "example_document.pdf",
    "page": 1,
    "box": [72, 120, 468, 150]
  },
  "children": [
    {
      "type": "text",
      "value": "Heading 1"
    }
  ],
  "depth": 1
}
```

Content Inclusion

The Content Inclusion options allow users to control various elements in the generated JSON output. These options specify whether bookmarks, fields, headers, footers, images, links, and metadata are included, providing flexibility in how the final document is structured. Additionally, users can choose the format for including metadata, enhancing compatibility with different systems. By customizing these settings, users can create JSON documents that meet their specific needs and preferences.

CONTENT INCLUSION OPTIONS

JSON_INCLUDE_BOUNDS

This option controls the inclusion of element bounds information in the generated JSON. The default value is:

- Default value: `ON` , which includes bounds information in the output.

JSON_INCLUDE_BOOKMARKS

This option controls the inclusion of bookmarks in the generated JSON. The default value is:

- Default value: `ON` , which includes bookmarks in the output.

JSON_INCLUDE_DOC_METADATA_PER_ELEMENT

This option enables or disables the inclusion of document-level metadata for each element in the generated JSON. The default value is:

- Default value: `OFF` , which does not include metadata for each element.

JSON_INCLUDE_ELEMENT_ID

This option controls whether an element ID is included in the generated JSON output. The default value is:

- Default value: `OFF` , which does not include element IDs.

JSON_INCLUDE_FIELDS

This option determines whether fields are included in the generated JSON. The default value is:

- Default value: `ON` , which includes fields in the output.

JSON_INCLUDE_FOOTERS

This option specifies whether page footers are included in the JSON output. The available values for this option are:

- Default value: `OFF` , which does not include footers.
- `ON` : Includes footers for all pages.
- `FIRST` : Includes the footer from the first page only.

JSON_INCLUDE_HEADERS

This option specifies whether page headers are included in the JSON output. The available values for this option are:

- Default value: `OFF` , which does not include headers.
- `ON` : Includes headers for all pages.
- `FIRST` : Includes the header from the first page only.

JSON_INCLUDE_IMAGE_DATA

This option controls the inclusion of image data in the generated JSON. The default value is:

- Default value: `ON` , which includes image data in the output.

JSON_INCLUDE_IMAGES

This option controls the inclusion of images in the generated JSON. The default value is:

- Default value: `ON` , which includes images in the output.

JSON_INCLUDE_LINKS

This option determines whether hyperlinks are included in the generated JSON. The default value is:

- Default value: `ON` , which includes links in the output.

JSON_INCLUDE_METADATA

This option specifies whether document metadata is included in the JSON output. The default value is:

- Default value: `OFF` , which does not include metadata.

JSON_INCLUDE_STYLES

This option determines whether element style information is included in the generated JSON. The default value is:

- Default value: `ON` , which includes style information in the output.

JSON_INCLUDE_WHITESPACE

This option enables or disables the inclusion of whitespace words in the generated JSON. The default value is:

- Default value: `OFF` , which does not include whitespace words.

JSON_INCLUDE_WORDS

This option determines whether word-level information is included in the generated JSON. The default value is:

- Default value: `OFF` , which does not include word-level information.

JSON_INCLUDE_FORMATTING

This option determines whether text formatting (e.g., bold, italic) is included in the JSON output. The default value is:

- Default value: `ON` , which includes text formatting in the output.

Content Cleaning

The Content Cleaning feature allows users to improve the readability and machine-friendliness of generated content, making it ideal for downstream processing in AI/ML systems. This feature offers multiple configurable cleaning options, including removing non-ASCII characters and normalizing quotes.

CONTENT CLEANING OPTION**JSON_CLEAN_CONTENT**

`JSON_CLEAN_CONTENT` specifies an unordered list of cleaning functions to apply to each element's text when generating simplified JSON output with option `JSON_OUTPUT_SCHEMA=PIPELINE` .

The unordered list of cleaning functions is separated by commas and is case insensitive.

The default value is an empty string, which applies no cleaning functions.

CLEANING FUNCTIONS

- **clean_non_ascii_chars**: Removes non-ASCII characters, leaving only standard ASCII characters.
- **normalize_quotes**: Standardizes a variety of Unicode single and double quotes by replacing them with ASCII single and double quotes.
- **normalize_dashes**: Standardizes a variety of Unicode dashes by replacing them with a standard ASCII hyphen.

The cleaning functions are applied in the following fixed order:

1. `normalize_quotes`
2. `normalize_dashes`
3. `clean_non_ascii_chars`

EXAMPLES

- Option value `normalize_quotes` applies the **normalize_quotes** function.
- Option value `clean_non_ascii_chars,normalize_quotes` applies the **normalize_quotes** and **clean_non_ascii_chars** functions.

1.17.3 About Markdown Output

Hyland Document Filters provides powerful options for generating Markdown output from various document formats. By configuring the output settings, you can customize the generated Markdown to suit your specific needs, from table styles to metadata inclusion.

What is Markdown

Markdown is a lightweight markup language designed for creating formatted text using a simple and easy-to-read syntax. It's widely used for documentation, web content, and various other writing tasks due to its readability in plain text form and ease of conversion to HTML or other formats. Markdown is designed to be a writing format that can be converted to rich text but also be readable in its raw form.

Markdown's simplicity and flexibility make it more than just a tool for humans to write documentation. Its lightweight structure, combined with the ability to preserve document semantics, makes it an ideal format for various advanced

applications in AI, analytics, and text processing systems. Here are some key use cases for Markdown beyond traditional human-readable documentation:

1. Markdown in AI and Machine Learning Systems

2. **Training Data Generation:** Markdown, with its minimal formatting and clear text structure, is useful for generating training data for AI models that require labeled text. Since Markdown retains the document hierarchy, such as headers, lists, and tables, without the complexities of full HTML or rich-text formats, it simplifies the process of identifying sections, entities, and relationships within a document.

- **Example:** A machine learning model trained to summarize documents or extract specific sections (e.g., headings, key points) can easily parse Markdown to focus on relevant text elements like headers (#), lists (-), and code blocks.

3. **Natural Language Processing (NLP) Tasks:** Markdown documents can be preprocessed by NLP engines for a variety of tasks, such as sentiment analysis, named entity recognition, or topic modeling. The human-readable nature of Markdown, along with its hierarchical structure, allows AI models to focus on the content while ignoring unnecessary formatting tags.

4. **AI-Driven Content Transformation:** Markdown is used in systems where AI transforms content from one format to another, such as converting technical documentation into different languages, summarizing long reports, or even converting written instructions into voice commands. Markdown's structured format makes it easier for AI systems to process the content and maintain a logical flow.

5. Markdown in Analytics and Reporting Systems

6. **Simplified Data Aggregation:** In analytics and reporting systems, Markdown is often used as a lightweight way to store structured narrative data, allowing analysts to include text explanations alongside visualizations and data outputs. Markdown's plain text format allows for easy integration with other reporting tools or systems that generate automated reports.

7. **Data-Driven Documentation:** Markdown is ideal for automatically generating data-driven documentation that evolves as data changes. When combined with automated tools, Markdown can be used to create real-time updated reports, financial documents, or data analytics dashboards where the content is generated and formatted based on underlying data models.

8. Markdown in Text Processing Systems

9. **Document Comparison and Diffing:** Markdown's text-based format is ideal for document comparison (diffing) systems, which need to detect changes between different versions of a document. Since Markdown is plain text, it can be diffed more easily than complex formats like PDF or Word. Systems can show changes in a document's content, structure, or metadata, allowing for more granular version control.

10. **Content Cleaning and Normalization:** In text processing pipelines, Markdown can be used as an intermediate format for content cleaning and normalization. Markdown simplifies the structure of documents, allowing text processing systems to focus on cleaning up content such as extra whitespace, special characters, and formatting inconsistencies before further processing.

11. **Automated Documentation Generation:** Markdown serves as a common format in automated documentation generation systems where structured content like code comments, API specifications, and technical descriptions

are programmatically converted into documentation. Systems can dynamically generate Markdown based on source code or data inputs, allowing for easy updates.

Creating Markdown Output

Hyland Document Filters enables users to generate high-quality Markdown output from a variety of document formats with flexibility and ease. There are two modes for Markdown creation: **Hi-Def** and **Text-Mode**.

In **Hi-Def mode**, you create a Markdown canvas and then render the pages into it. This mode captures more visual information while maintaining the document structure, making it ideal for documents where layout and formatting are essential, as it preserves elements like tables and images more effectively.

In contrast, **Text-Mode** focuses on the structural elements of the document. To use this mode, you open the document with the `IGR_FORMAT_MARKDOWN` and call `GetText` to extract the content. This method is particularly useful when the Markdown generated from paginated output is suboptimal, such as in the case of spreadsheets, ensuring a more reliable representation of the underlying content without complex formatting.

By configuring the output settings, you can customize the generated Markdown to suit your specific needs, from table styles to metadata inclusion.

Explore our tutorials on creating Markdown output to enhance your workflow:

- [How do I convert a document to Markdown in Hi-Def mode?](#)
- [How do I convert a document to Markdown in Text mode?](#)

Table Styling

In Markdown, tables can be represented in two main ways: **simple** and **complex**. When processing documents, Hyland Document Filters automatically determines the complexity of each table and maps it to the appropriate type. Simple tables are used when the content allows for a minimal representation, while complex tables are used for more advanced structures.

You can control how these tables are rendered using the options described below, allowing you to balance clarity, functionality, and ease of processing based on your specific needs.

SIMPLE TABLES

Simple tables are the most straightforward type of tables in Markdown. They are typically used for small, uniform tables where the layout is minimal, and the content is easy to represent without any advanced formatting. Simple tables generally consist of headers and rows of data separated by pipes (`|`), and columns are aligned using dashes (`-`) for headers.

Characteristics of Simple Tables:

- **Limited to basic formatting:** Simple tables only support basic table structure and are rendered with pipes and dashes to separate columns and rows.
- **No complex cells:** There is no support for merged cells (colspan/rowspan) or nested elements like lists, images, or multiple lines of text.
- **Fast rendering:** Simple tables are lightweight and easy for Markdown parsers to process, making them ideal for smaller data sets or where table styling isn't the main focus.
- **Uniform column widths:** Columns are automatically spaced based on the longest cell in each column.

Example of Simple Table:

	Name		Age		City	
	-----		---		-----	
	Alice		30		New York	
	Bob		25		San Francisco	
	Charlie		35		Chicago	

COMPLEX TABLES

Complex tables are used when more advanced table layouts are required. These tables allow for more intricate formatting, including features like nested block elements (e.g., lists, paragraphs or images within table cells), merged cells, or varying column widths. Complex tables often rely on the flexibility of HTML or other Markdown extensions to represent their structure.

Characteristics of Complex Tables:

- **Advanced formatting:** Complex tables can include merged cells (colspan/rowspan), nested content, multi-line text, and even embedded images or code blocks.
- **HTML support:** In some Markdown flavors (e.g., GitHub Flavored Markdown), complex tables may require or allow embedding HTML to achieve the desired level of complexity.
- **Variable column widths:** Complex tables can have columns with varying widths to better display irregular content or nested elements.
- **Rendering complexity:** Since complex tables may rely on raw HTML, their rendering depends on the Markdown processor being used. Some renderers may strip or sanitize HTML, affecting table appearance. Additionally, HTML-based tables may require more processing and impact document readability in plain-text Markdown views.

TABLE FORMAT

Document Filters provides flexible options for rendering tables in Markdown. Depending on the complexity of a table, it may be represented using different styles to balance readability and compatibility with various Markdown flavors.

Tables can be rendered in **pipe-style**, **grid-style**, **HTML**, or **paragraph**, with additional hybrid options for enhanced formatting. The examples below illustrate the different table rendering styles available.

1. PIPE Mode (Default for simple tables):

A standard Markdown table using pipes (|) to separate columns. This format is widely supported and easy to read.

```
| Name      | Age | Occupation      |
|-----|-----|-----|
| Alice     | 29  | Engineer        |
| Bob       | 34  | Data Scientist  |
```

2. GRID Mode:

A more structured table format using +, -, and | to define cell borders. This style improves readability, especially for fixed-width text editors.

```
+-----+-----+-----+
| Name      | Age | Occupation      |
+-----+-----+-----+
| Alice     | 29  | Engineer        |
+-----+-----+-----+
| Bob       | 34  | Data Scientist  |
+-----+-----+-----+
```

3. HTML Mode

Renders the table as raw HTML, providing maximum flexibility for styling and layout customization. Recommended for environments that fully support HTML rendering.

```
<table>
  <tr><th>Name</th><th>Age</th><th>Occupation</th></tr>
  <tr><td>Alice</td><td>29</td><td>Engineer</td></tr>
  <tr><td>Bob</td><td>34</td><td>Data Scientist</td></tr>
</table>
```

4. PIPE_WITH_HTML Mode:

A hybrid format that uses pipe-style tables for structure while allowing inline HTML for enhanced formatting, such as bold text or custom styling.

```
| Name      | Age | Occupation      |
|-----|-----|-----|
| Alice     | 29  | <b>Engineer</b>   |
| Bob       | 34  | <b>Data Scientist</b> |
```

5. PARAGRAPH Mode:

Tables are converted into a simple paragraph-style format, listing each cell's content sequentially. This mode is useful for cases where table structures need to be flattened for easier text processing.

```
Name
Age
Occupation
Alice
29
**Engineer**
Bob
34
**Data Scientist**
```

TABLE OPTIONS

MARKDOWN_COMPLEX_TABLE_STYLE

This option controls the rendering style for **complex tables** in Markdown, which might include multi-line text, merged cells, or other advanced layouts. The available values for this option are:

- `PIPE` : A basic pipe (`|`) style table.
- `GRID` : A grid-style table using `+`, `-`, and `|` to create cell borders.
- `HTML` : Renders the table as HTML for full flexibility.
- `PIPE_WITH_HTML` : A combination of pipe-style for basic table structures and HTML for more complex features like merged cells.
- `PARAGRAPHS` : Tables are converted into a simple paragraph-style format, listing each cell's content sequentially.

MARKDOWN_SIMPLE_TABLE_STYLE

This option controls the rendering style for **simple tables**, typically smaller tables without complex formatting. The values for this option are the same as for complex tables, and they include:

- `PIPE` : A simple pipe-style table.
- `GRID` : A grid-style table with cell borders.
- `HTML` : Uses HTML to render the table.

- `PIPE_WITH_HTML` : A mix of pipe-style with HTML for added formatting flexibility.
- `PARAGRAPHS` : Tables are converted into a simple paragraph-style format, listing each cell's content sequentially.

`MARKDOWN_TABLE_PADDING`

Enables or disables padding in pipe-style tables. Defaults to `ON`.

Markdown Formatting

Markdown formatting options allow users to customize the output to suit different needs and preferences. This includes controlling the flavor of Markdown to align with specific platform requirements, defining header styles for better organization, and setting preferred line lengths for optimal readability. By adjusting these settings, users can create Markdown documents that are not only syntactically correct but also visually appealing and easier to navigate.

FORMATTING OPTIONS

`MARKDOWN_FLAVOR`

This option controls the flavor of Markdown used in the output. Different flavors can introduce variations in syntax or features that are supported by specific platforms. The available values for this option are:

- `GFM` : GitHub-Flavored Markdown, which supports features like task lists, tables, and autolinks.
- `GPT` : GPT Markdown, optimized for AI-related markdown outputs.

`MARKDOWN_HEADERS_STYLE`

This option specifies the style of headers in the generated Markdown, defining how header levels are formatted. The available values for this option are:

- `ATX` : Uses hash (`#`) symbols to denote headers.
- `SETEXT` : Uses `=` for level 1 headers and `-` for level 2 headers.

`MARKDOWN_PREFERRED_LINE_LENGTH`

This option sets the preferred line length for wrapping Markdown content. It controls how lines are broken to improve readability in the rendered Markdown. The default value is: `80`.

Content Inclusion

The Content Inclusion options allow users to control various elements in the generated Markdown output. These options specify whether bookmarks, fields, headers, footers, images, links, and metadata are included, providing flexibility in how the final document is structured. Additionally, users can choose the format for including metadata, enhancing compatibility with different systems. By customizing these settings, users can create Markdown documents that meet their specific needs and preferences.

CONTENT INCLUSION OPTIONS**MARKDOWN_INCLUDE_BOOKMARKS**

This option controls the inclusion of bookmarks in the generated Markdown. The default value is: `ON`.

MARKDOWN_INCLUDE_FIELDS

This option determines whether fields are included in the generated Markdown. The default value is: `ON`.

MARKDOWN_INCLUDE_FORMS

The option determines whether form elements are include in the generated Markdown. The default value is: `ON`.

MARKDOWN_INCLUDE_FOOTERS

This option specifies whether page footers are included in the Markdown output. The available values for this option are:

- Default value: `OFF`, which does not include footers.
- `ON`: Includes footers for all pages.
- `FIRST`: Includes the footer from the first page only.

MARKDOWN_INCLUDE_HEADERS

This option specifies whether page headers are included in the Markdown output. The available values for this option are:

- Default value: `OFF`, which does not include headers.
- `ON`: Includes headers for all pages.
- `FIRST`: Includes the header from the first page only.

MARKDOWN_INCLUDE_IMAGES

This option controls the inclusion of images in the generated Markdown. The default value is: `ON`.

MARKDOWN_INCLUDE_LINKS

This option determines whether hyperlinks are included in the generated Markdown. The default value is: `ON`.

MARKDOWN_INCLUDE_METADATA

This option specifies whether document metadata is included in the Markdown output. The default value is: `OFF`.

MARKDOWN_METADATA_FORMAT

This option specifies the format for including metadata in the Markdown output. The available values for this option are:

- Default value: `COMMENTS`, which includes metadata as comments in the output.
- `YAML`: Uses YAML format for metadata.

- **TOML** : Uses TOML format for metadata.
- **JSON** : Uses JSON format for metadata.

Note: `MARKDOWN_INCLUDE_METADATA` must be enabled for metadata to be output.

Example Outputs:

1. **COMMENTS** (default)

```
<!--  
Title: Example Document  
Author: John Doe  
Date: 2024-10-16  
-->
```

2. **YAML**

```
---  
title: Example Document  
author: John Doe  
date: 2024-10-16  
---
```

3. **TOML**

```
+++  
title = "Example Document"  
author = "John Doe"  
date = "2024-10-16"  
+++
```

4. **JSON**

```
{  
  "title": "Example Document",  
  "author": "John Doe",  
  "date": "2024-10-16"  
}  
---
```

MARKDOWN_INCLUDE_FORMATTING

This option determines whether text formatting (e.g., bold, italic) is included in the Markdown output. The default value is: `ON`.

MARKDOWN_INCLUDE_LOCATIONS

This option controls the inclusion of location information for extracted content. When enabled, location metadata is inserted into the Markdown output as HTML comments, providing context for machine learning and large language model systems. The default value is: `OFF`.

Location comments are emitted before the following Markdown elements:

- Headings (`#` , `##` , etc.)
- Paragraphs (non-empty or image-only, if image output is enabled)
- Lists
- Tables

The location comment follows this format:

```
<!-- LOC: N, (l,t,r,b) -->
```

Where: * N is the page number. * (l,t,r,b) are the bounding box coordinates: left, top, right, bottom.

If the page changes and no bounding box is available (e.g., due to continued flow), the comment is:

```
<!-- LOC: N -->
```

Example

```
<!-- LOC: 1, (10,10,110,110) -->
# Heading 1

<!-- LOC: 1, (10,20,110,120) -->
Content of Heading 1

<!-- LOC: 1, (10,30,110,130) -->
## Heading 2

<!-- LOC: 1, (10,40,110,140) -->
Content of Heading 2
```

Page Structure

The Page Structure options provide control over the layout and visual separation of content in the generated Markdown. Users can choose to insert horizontal rules between pages for clearer delineation or add new lines after

block-level HTML elements to enhance readability. Customizing these options allows for better organization and presentation of the document's structure, catering to different formatting preferences.

PAGE STRUCTURE OPTIONS

MARKDOWN_HR_BETWEEN_PAGES

This option controls the insertion of a horizontal rule (`<hr>`) between pages in the generated Markdown. The default value is: `OFF` .

MARKDOWN_LF_AFTER_BLOCK

This option specifies whether new lines are added after each block-level HTML element in the generated Markdown. The default value is: `OFF` .

Content Cleaning

The Content Cleaning feature allows users to improve the readability and machine-friendliness of generated content, making it ideal for downstream processing in AI/ML systems. This feature offers multiple configurable cleaning options, including removing non-ASCII characters and normalizing quotes.

CONTENT CLEANING OPTION

MARKDOWN_CLEAN_CONTENT

`MARKDOWN_CLEAN_CONTENT` specifies an unordered list of cleaning functions to apply to the text of each block, paragraph, or element when generating Markdown output.

The unordered list of cleaning functions is separated by commas and is case insensitive.

The default value is an empty string, which applies no cleaning functions.

CLEANING FUNCTIONS

- **clean_non_ascii_chars**: Removes non-ASCII characters, leaving only standard ASCII characters.
- **normalize_quotes**: Standardizes a variety of Unicode single and double quotes by replacing them with ASCII single and double quotes.
- **normalize_dashes**: Standardizes a variety of Unicode dashes by replacing them with a standard ASCII hyphen.

The cleaning functions are applied in the following fixed order:

1. `normalize_quotes`
2. `normalize_dashes`
3. `clean_non_ascii_chars`

EXAMPLES

- Option value `normalize_quotes` applies the **normalize_quotes** function.

- Option value `clean_non_ascii_chars, normalize_quotes` applies the **normalize_quotes** and **clean_non_ascii_chars** functions.

Format-Specific Mappings

Markdown provides a relatively flat document structure. While some Markdown flavors support complex nesting, the general approach is to maintain a simplified representation of the document.

Hyland Document Filters processes documents with this principle in mind, flattening and simplifying structures where possible while preserving the document's meaning.

HTML

HTML has evolved significantly since its introduction in 1993, with numerous iterations shaping its capabilities. However, when HTML was adopted for email formatting, many email clients imposed limitations on HTML support. To achieve the desired visual layout, email authors often relied on `<table>` elements—not for tabular data, but as a means of structuring content.

When converting HTML to Markdown, Document Filters applies additional logic to distinguish between true data tables and layout tables, ensuring that only meaningful tabular data is preserved in the output.

Table Flattening Rules

To prevent unnecessary complexity in Markdown output, Document Filters flattens tables under the following conditions:

- If a table contains another table, the outer table is flattened into paragraphs.
- If a table explicitly sets cellpadding and cellspacing to zero, it is treated as a layout table and flattened into paragraphs.
- If a table has fewer than two rows or two columns, it is considered non-tabular content and flattened into paragraphs.

Form Elements

HTML forms contain various interactive elements such as text fields, checkboxes, radio buttons, and dropdowns. When converting HTML to Markdown, Document Filters processes form elements in a way that preserves relevant content while simplifying the structure. Since Markdown does not natively support form elements, the following rules apply:

- **Text Inputs and Textareas:** Converted to labeled text representations, with stored value.

```
**Name:** Name
```

```
**Comments:**  
Comments
```

- **Checkboxes:** Represented as Markdown checklists when appropriate.

```
[x] Option 1  
[ ] Option 2
```

- **Radio Buttons:** Represented as Markdown checklists when appropriate.

```
(o) Option 1  
( ) Option 2
```

- **Dropdowns (Select Elements):** Flattened into a list showing available options.

```
**Select a category:**  
- Option A  
- Option B (selected)  
- Option C
```

1.17.4 About PDF Output

Hyland Document Filters offers robust features for generating PDF output from a variety of document formats. By adjusting the output settings, you can tailor the resulting PDF to meet your specific requirements, including layout preferences, security features, and metadata inclusion.

What is PDF

PDF, or Portable Document Format, is a versatile file format developed by Adobe Systems for presenting documents, including text formatting and images, in a manner independent of application software, hardware, and operating systems. PDFs are widely used for documents like reports, manuals, and forms due to their consistent appearance across different platforms.

The PDF format preserves the integrity of a document's layout, ensuring that it appears the same to every viewer, regardless of the device or software used to access it. This makes PDFs an ideal choice for sharing important information, as they can include hyperlinks, forms, annotations, and other interactive elements while retaining high-quality graphics and text.

Beyond its traditional use as a means for sharing documents, PDF has become a fundamental format in various advanced applications, including digital signatures, document management systems, and archiving solutions. Here are some key use cases for PDF beyond simple document distribution:

1. PDF in Document Management Systems

2. **Archiving and Compliance:** PDFs are often used in document management systems for archiving important records, ensuring compliance with regulatory standards while maintaining accessibility and integrity over time.

- **Example:** Organizations may store contracts and legal documents in PDF format to meet compliance requirements while preserving their original layout and formatting.

3. **Collaboration and Review:** PDFs allow multiple users to annotate and review documents, facilitating collaborative workflows. Users can highlight text, add comments, and suggest changes without altering the original content.

4. **Integration with Workflow Systems:** PDF output can be integrated into automated workflows, enabling seamless document routing and approval processes within organizations.

5. PDF in Digital Publishing

6. **E-Books and Online Publications:** The PDF format is widely used in digital publishing for e-books and online content distribution, allowing for rich formatting, images, and interactivity.

7. **Print-Ready Documents:** PDFs are often the preferred format for print-ready documents, ensuring that the final product matches the designer's intent and specifications.

8. PDF in Data Presentation

9. **Reports and Dashboards:** PDF output is ideal for generating professional reports and dashboards, allowing users to present data alongside rich visual elements while maintaining the document's visual integrity.

10. **Interactive Forms:** PDF forms can include interactive fields for user input, making them a practical choice for applications that require data collection or customer feedback.

Creating PDF Output

Generating PDF output with Hyland Document Filters is a powerful feature that allows you to create high-quality documents in Portable Document Format (PDF) from various source formats. Please note that PDF creation is supported exclusively in Hi-Def mode, which ensures that the document's structure and formatting are preserved accurately during the conversion process.

To create a PDF output in Hi-Def mode, follow these steps:

1. **Select Your Source Document:** Choose the document you wish to convert to PDF. Ensure it is a supported format for Hi-Def mode.
2. **Configure Output Settings:** Set your desired output options, including:
 3. Security settings (e.g., password protection, restrictions)
 4. Metadata to include (e.g., title, author, keywords)
5. **Create PDF Canvas:** Ensure that Hi-Def mode is selected in your conversion settings to create a canvas for the PDF output. This canvas will be used to accurately represent the layout and structure of the source document.
6. **Render Pages to Canvas:** Initiate the rendering process, where the system will convert each page of the document to the PDF canvas, ensuring fidelity in layout and formatting.
7. **Review the Output:** Once the PDF is created, review the document to ensure that the formatting, images, and overall layout meet your expectations.

By following these steps, you can effectively create professional-grade PDF documents that retain the integrity of your original content, making them suitable for sharing, printing, and archiving.

Explore our tutorials on creating PDF output to enhance your workflow:

- [How do I convert a document to a PDF file?](#)

PDF Options

PDF options allow users to customize the behavior and characteristics of PDF output. These options include settings for annotations, bookmarks, compression, encryption, image quality, and other properties, ensuring that generated PDFs meet specific requirements.

PDF_ENCRYPT PDF_SET_OWNER_PASSWORD PDF_SET_USER_PASSWORD PDF_ENCRYPT_PERMISSIONS

PDF_VERSION

PDF (Portable Document Format) is a widely used file format developed by Adobe Systems for document exchange. The PDF specification has evolved over the years, with various versions introducing new features, enhancements, and compliance standards to meet the changing needs of users and industries. Understanding the different PDF versions is crucial for ensuring compatibility and leveraging the latest capabilities, especially when creating, editing, or sharing PDF documents.

PDF_VERSION

This option allows users to specify the compliance level of the generated PDF files. Different versions of PDF support various features, such as transparency, layers, and advanced security options. It is essential to choose the correct

version to ensure that the document displays correctly across different PDF readers and platforms. Note that certain features may not be available in all versions, so users should carefully consider their requirements when selecting a version.

• **Values:**

- `1.7` - **Default** - The standard version that supports advanced features like layers and transparency.
- `A1` - Compliance with PDF/A-1, a format for long-term archiving of electronic documents.
- `A1A` - A conformance level of PDF/A-1 ensuring accessibility and long-term preservation.
- `A1B` - A conformance level of PDF/A-1 ensuring that the visual appearance of the document is preserved.
- `A2` - Compliance with PDF/A-2, allowing the use of JPEG2000 compression and transparency.
- `A2U` - A conformance level of PDF/A-2 ensuring accessibility and compliance with Unicode.
- `PDF1.7` - Similar to `1.7`, supports all features available in that version.
- `PDFA` - General indication of compliance with PDF/A standards.
- `PDFA1` - Compliance with PDF/A-1 standards.
- `PDFA1A` - Ensures accessibility and long-term preservation in compliance with PDF/A-1 standards.
- `PDFA1B` - Ensures visual fidelity and long-term preservation.
- `PDFA2` - Compliance with PDF/A-2 standards, which includes new features from PDF 1.6 and 1.7.
- `PDFA2U` - Ensures accessibility and compliance with Unicode, compliant with PDF/A-2 standards.

PDF_TAGGED

The `PDF_TAGGED` option allows users to create tagged PDF documents, which are essential for accessibility and assistive technologies. Tagged PDFs provide a logical structure to the content, allowing screen readers and other assistive devices to interpret and present the information correctly. This is particularly important for users with disabilities who rely on these technologies to access document content.

• **Values:**

- `true` - Generates a tagged PDF that includes a logical structure and supports accessibility features.
- `false` - Generates a standard PDF without any tagging for accessibility.

Note: this value will be auto-enabled when generating accessible PDF-A files.

PDF ENCRYPTION OPTIONS

PDF encryption options enhance the security of PDF documents by restricting access and controlling permissions. These options ensure that sensitive information is protected and that only authorized users can perform certain actions on the document.

PDF_ENCRYPT

The `PDF_ENCRYPT` option enables encryption for the generated PDF document. This ensures that the content is protected from unauthorized access and manipulation.

• Values:

- `true` - Enables encryption for the output PDF.
- `false` - **Default* - Disables encryption, allowing unrestricted access.

PDF_SET_OWNER_PASSWORD

The `PDF_SET_OWNER_PASSWORD` option allows users to set an owner password for the PDF document. This password controls permissions such as printing, copying, and editing the document.

PDF_SET_USER_PASSWORD

The `PDF_SET_USER_PASSWORD` option lets users set a user password for opening the PDF document. This password is required to access the content of the PDF.

PDF_ENCRYPT_PERMISSIONS

The `PDF_ENCRYPT_PERMISSIONS` option what permissions should be set when encrypting the document. The permissions are provided as a comma separated list of the following:

- **accessibility** : Allows users with disabilities to access the document's content. This permission ensures that assistive technologies can read and interpret the document.
- **annotate_and_form** : Grants permission to add comments and annotations to the document and to fill out interactive form fields. This is important for collaboration and data entry in PDFs.
- **assemble** : Allows the user to rearrange, insert, or delete pages within the document. This permission is essential for users who need to manage document structure without altering content.
- **encrypt_metadata** : Specifies whether to encrypt the metadata of the document, such as the title, author, and subject. When enabled, it helps protect sensitive information embedded in the document properties.
- **extract** : Permits the extraction of text and images from the document. This permission is useful for users who need to repurpose content from the PDF.
- **form_filling** : Allows users to fill out and save forms within the document. This is crucial for documents that serve as applications, surveys, or contracts requiring user input.
- **modify_other** : Grants permission to modify the document in ways other than those explicitly defined (e.g., adding watermarks or modifying existing content). This permission is generally broader and can vary based on context.
- **printing_full** : Allows the document to be printed at any resolution, including high-quality prints. This permission is typically necessary for documents that need to be distributed in physical form.
- **printing** : Permits printing of the document, but may restrict the resolution (e.g., low-resolution printing). This permission is essential for controlling how the document is reproduced in printed format.

ADDITIONAL PDF OPTIONS

In addition to versioning and tagging, there are several other options available for customizing the behavior and features of PDF documents. These options enhance the functionality, quality, and accessibility of the generated PDFs, catering to specific user needs.

PDF_PRESERVE_PAGE_SIZE

The `PDF_PRESERVE_PAGE_SIZE` option determines whether the original page size of the source document is preserved in the output PDF. This is particularly useful for maintaining the intended layout and formatting when converting PDF to PDF.

- `true` - Preserves the original page size of the input document.
- `false` - Allows the page size to be adjusted according to the PDF settings.

PDF_PRESERVE_ORIGINAL

The `PDF_PRESERVE_ORIGINAL` option allows users to keep the original content of the document intact during the conversion process. This is important for retaining the fidelity of the source document while still generating a new PDF.

- `true` - Preserves the original content without alterations.
- `false` - Allows modifications to the content as per the conversion settings.

PDF_A_JPEG_QUALITY

The `PDF_A_JPEG_QUALITY` option specifies the quality level for JPEG images when creating PDF/A compliant documents. This is critical for balancing file size and image fidelity in archived documents.

- **Range:** 0-100 (where 0 is the lowest quality and 100 is the highest)
- **Default value:** 75
- **Description:** A higher value results in better image quality but larger file sizes, while a lower value reduces image quality and file size.

PDF_BOOKMARKS

The `PDF_BOOKMARKS` option enables the inclusion of bookmarks in the generated PDF, providing a navigational aid for users. Bookmarks allow quick access to specific sections or pages within the document.

- `true` - Includes bookmarks in the output PDF.
- `false` - Excludes bookmarks from the PDF.

PDF_LINEARIZE

The `PDF_LINEARIZE` option allows for the linearization of PDF files, optimizing them for efficient online viewing. Linearized PDFs can be loaded and displayed incrementally, which improves the user experience, especially for large documents.

- `true` - Creates a linearized PDF for faster access over the web.
- `false` - Generates a standard PDF without linearization.

2. Tutorials

2.1 Tutorials

This section provides a comprehensive collection of tutorials designed to help users effectively utilize the Hyland Document Filters SDK. Each tutorial offers step-by-step guidance, covering various aspects of document processing, including document conversion, data extraction, and customization options. Whether you are a beginner or an experienced developer, these tutorials will enhance your understanding and facilitate your integration of Document Filters into your applications. Follow along to discover best practices and leverage the full potential of our powerful document processing capabilities.

Opening Files

How do I open a document from disk?	This sample demonstrates how to use the Hyland Document Filters SDK to extract text from a document. It provides a high-level workflow for initializing the Document Filters API, opening a document, and reading its text content in chunks until the end of the document is reached.
How do I open a document from memory?	This sample demonstrates how to use the Hyland Document Filters SDK to extract text from a document loaded into memory. It provides a high-level workflow for initializing the Document Filters API, opening a document, and reading its text content in chunks until the end of the document is reached.

Converting Files

How do I convert a document to a PDF file?	This sample demonstrates how to use the Hyland Document Filters SDK to convert a document into PDF format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering its pages into a structured PDF output.
How do I convert a document to Classic HTML?	This sample demonstrates how to use the Hyland Document Filters SDK to convert a document into classic HTML format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and extracting its content and images for web presentation.
How do I convert a document to JSON in Hi-Def mode?	This sample demonstrates how to use the Hyland Document Filters SDK to convert a document into HiDef JSON format. It provides a high-level workflow for initializing the

	Document Filters API, opening a document, and rendering its content into a structured JSON output suitable for web presentation.
How do I convert a document to Markdown in Hi-Def mode?	This sample demonstrates how to use the Hyland Document Filters SDK to convert a document into HiDef Markdown format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering its content into a structured Markdown output suitable for web presentation.
How do I convert a document to Markdown in Text mode?	This sample demonstrates how to use the Hyland Document Filters SDK to convert a document into Text-mode Markdown format. It outlines a high-level workflow for initializing the Document Filters API, opening a document in Text mode, and rendering its content into structured Markdown output suitable for web presentation.
How do I convert a document to paginated HiDef HTML?	This sample demonstrates how to use the Hyland Document Filters SDK to convert a document into HiDef HTML format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering its content into a structured HTML output suitable for web presentation.
How do I convert a document to PNG images?	This sample demonstrates how to use the Hyland Document Filters SDK to convert a document into PNG format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering each page into individual PNG images.
How do I convert a document to Structured XML?	This sample demonstrates how to use the Hyland Document Filters SDK to convert a document into XML format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering its pages into a structured XML output.
How do I convert a PDF to Markdown with Table Detection?	This sample demonstrates how to use the Hyland Document Filters SDK to convert a PDF document into HiDef Markdown format with table detection enabled. It provides a high-level workflow for initializing the Document Filters API, opening a PDF document, and rendering its content into a structured Markdown output suitable for web presentation.
How do I convert HTML to Hi-Def?	This sample demonstrates how to use the Hyland Document Filters SDK to convert an HTML document into Hi-Def format. It outlines a high-level workflow for initializing the Document Filters API, opening an HTML file, and handling embedded images. The tutorial also covers different methods for resolving images, including external URLs, local disk references, and in-memory stream processing.

Extracting Files

How do I extract metadata from a document?	This sample demonstrates how to use the Hyland Document Filters SDK to extract metadata from a document. It provides a high-level workflow for initializing the Document Filters API, opening a document, and retrieving its metadata.
How do I extract sub-documents from documents and archives?	This sample demonstrates how to use the Hyland Document Filters SDK to extract metadata from subfiles within a document archive. It provides a high-level workflow for initializing the Document Filters API, opening a document, and iterating through its subfiles to read their associated metadata.
How do I extract text and metadata from a document?	This sample demonstrates how to use the Hyland Document Filters SDK to extract text from a document. It provides a high-level workflow for initializing the Document Filters API, opening a document, and reading its text content in chunks until the end of the document is reached.
How do I process password-protected files?	Document Filters provides a flexible callback-based approach to unlock password-protected files, whether they are documents or nested within containers. This includes scenarios where each item may have different passwords.

Other

How do I capture log messages?	Hyland Document Filters provides a callback-based logging mechanism that allows applications to dynamically control log levels and capture detailed diagnostic information. This system gives developers full visibility into internal operations across modules while maintaining control over verbosity and performance.
How do I compare documents?	Comparing documents presents a multifaceted challenge due to the dynamic nature of content. Documents can undergo various changes such as additions, deletions, modifications, reformatting, and even movement across pages. These alterations not only impact the visual appearance but also the underlying structure and semantics of the content.
How do I create a barcode?	This sample demonstrates how to use the Hyland Document Filters SDK to generate a QR code and save it as a PNG image. It provides a high-level workflow for initializing the Document Filters API, creating an output canvas, and annotating it with a QR code.
How do I integrate with another OCR engine	Hyland Document Filters provides built-in OCR integration with Tesseract, but you can also use a custom OCR engine such as EasyOCR, PaddleOCR, or any AI-powered service by registering an OCR callback. This allows you to incorporate OCR into the document processing pipeline while leveraging Document Filters' layout analysis and flexible output formats like Markdown and JSON.

How do I localize metadata?	This sample demonstrates how to use the Hyland Document Filters SDK to localize an Outlook message and generate a PDF document. It provides a high-level workflow for initializing the Document Filters API, localizing message fields, opening the document, and rendering its content into a PDF format.

2.2 Opening Files

2.2.1 How do I open a document from disk?

In this Article

This sample demonstrates how to use the Hyland Document Filters SDK to extract text from a document. It provides a high-level workflow for initializing the Document Filters API, opening a document, and reading its text content in chunks until the end of the document is reached.

What you will learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Extraction:** Understand how to open a document for text extraction by using the `GetExtractor` method, specifying the document and the mode (`BodyAndMeta`) for reading the body and metadata.
3. **Text Streaming:** Discover how to read the document's text content in chunks using the `GetText()` method, which allows for efficient extraction of large documents by processing portions of the text in a loop.
4. **Resource Management:** Learn about proper resource management in .NET by using using statements to ensure that document and API objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, handling document extraction, and iterating over the document's content efficiently.

Installing Document Filters

Document Filters has several NuGet packages available. For most scenarios, however, you typically only need `Hyland.DocumentFilters`.

You can install it using the following command:

```
dotnet add package Hyland.DocumentFilters
```

For full setup instructions see [Getting Started with .NET](#)

Extracting Text

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc"); // (2) Create an instance of the Extractor for a file
8
9 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta); // (3) Open the document for reading in text-mode
10
11 while (!doc.EndOfStream) // (4) Read the document in 4KB chunks and write it to the console
12 {
13     var text = doc.GetText(4096);
14     Console.Out.WriteLine(text);
15 }
16
17 doc.Close();
```

[See our C# samples on GitHub](#)

2.2.2 How do I open a document from memory?

In this Article

This sample demonstrates how to use the Hyland Document Filters SDK to extract text from a document loaded into memory. It provides a high-level workflow for initializing the Document Filters API, opening a document, and reading its text content in chunks until the end of the document is reached.

What you will learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Extraction:** Understand how to open a document for text extraction by using the `GetExtractor` method, specifying the document and the mode (`BodyAndMeta`) for reading the body and metadata.
3. **Text Streaming:** Discover how to read the document's text content in chunks using the `GetText()` method, which allows for efficient extraction of large documents by processing portions of the text in a loop.
4. **Resource Management:** Learn about proper resource management in .NET by using using statements to ensure that document and API objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, handling document extraction, and iterating over the document's content efficiently.

Installing Document Filters

Document Filters has several NuGet packages available. For most scenarios, however, you typically only need `Hyland.DocumentFilters`.

You can install it using the following command:

```
dotnet add package Hyland.DocumentFilters
```

For full setup instructions see [Getting Started with .NET](#)

Extracting Text

```
1 using Hyland.DocumentFilters;  
2 using System;  
3  
4 var api = new Hyland.DocumentFilters.Api();  
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");  
6  
7 var fileBytes = System.IO.File.ReadAllBytes("filename.doc");  
8  
9 using var doc = api.GetExtractor(fileBytes);  
10 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);  
11  
12 while (!doc.EndOfStream)  
13 {  
14     var text = doc.GetText(4096);  
15     Console.Out.WriteLine(text);  
16 }
```

[See our C# samples on GitHub](#)

2.3 Converting Files

2.3.1 How do I convert a document to a PDF file?

In this Article

This sample demonstrates how to use the **Hyland Document Filters** SDK to convert a document into PDF format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering its pages into a structured PDF output.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Opening:** Understand how to open a document for conversion to PDF using the `OpenExtractor` method, specifying the appropriate open mode (e.g., `Paginated`) to ensure proper rendering of pages.
3. **Output Canvas Creation:** Discover how to create an output canvas for the PDF file using the `MakeOutputCanvas` method, specifying the desired output file name and canvas type (e.g., `PDF`) for generating the PDF document.
4. **Page Rendering:** Learn how to render all pages of the document into the output canvas, efficiently transforming the document's content into a PDF format.
5. **Resource Management:** Gain insight into proper resource management in .NET by using `using` statements to ensure that document and canvas objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, converting documents to PDF format, and efficiently rendering document content for PDF output.

See [PDF Output](#) for more details.

Converting to PDF

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
7 using var canvas = api.MakeOutputCanvas("filename.pdf", CanvasType.PDF);
8
9 canvas.RenderPages(doc);
```

[See our C# samples on GitHub](#)

2.3.2 How do I convert a document to Classic HTML?

In this Article

This sample demonstrates how to use the **Hyland Document Filters** SDK to convert a document into classic HTML format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and extracting its content and images for web presentation.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Conversion:** Understand how to open a document for conversion to classic HTML using the `GetExtractor` method, specifying the appropriate open mode (`ClassicHTML`) and type (`BodyAndMeta`) to access both the document body and metadata.
3. **Content Extraction:** Discover how to extract the document's body text in chunks, efficiently assembling it for HTML output.
4. **Image Handling:** Learn how to iterate through the images in the document and save them to a specified directory, ensuring that all visual elements are preserved in the conversion process.
5. **Resource Management:** Gain insight into proper resource management in .NET by using `using` statements to ensure that document and image objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, extracting metadata from subfiles, and efficiently accessing important information from document archives.

Converting to Classic HTML

```
1 using Hyland.DocumentFilters;
2 using System.Text;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.ClassicHTML, OpenType.BodyAndMeta);
9
10 var destDir = ".";
11
12 // save the body
13 var htmlBody = new StringBuilder();
14 while (!doc.EndOfStream)
15     htmlBody.Append(doc.GetText(4096));
16
17 // save the images
18 foreach (var image in doc.Images)
19 {
20     using (image)
21     {
22         image.CopyTo(System.IO.Path.Combine(destDir, image.getName()));
23     }
24 }
```

[See our C# samples on GitHub](#)

2.3.3 How do I convert a document to JSON in Hi-Def mode?

In this Article

This sample demonstrates how to use the **Hyland Document Filters** SDK to convert a document into HiDef JSON format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering its content into a structured JSON output suitable for web presentation.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Opening:** Understand how to open a document for conversion to HiDef JSON using the `OpenExtractor` method, specifying the appropriate open mode (e.g., `Paginated`) to ensure proper rendering.
3. **Output Canvas Creation:** Discover how to create an output canvas using the `MakeOutputCanvas` method, specifying the desired output file name and canvas type (e.g., `JSON`) for generating the JSON structure.
4. **Content Rendering:** Learn how to render the document's pages into the output canvas, efficiently transforming the document's content into a JSON format.
5. **Resource Management:** Gain insight into proper resource management in .NET by using `using` statements to ensure that document and canvas objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, converting documents to HiDef JSON, and efficiently rendering document content for web applications.

See [JSON Output](#) for more details.

Converting to HiDef JSON

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 var canvasOptions = "JSON_OUTPUT_SCHEMA=MDAST;";
7 canvasOptions += "JSON_FORMAT_OUTPUT=ON;";
8
9 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
10 using var canvas = api.MakeOutputCanvas("output.json", CanvasType.JSON, canvasOptions);
11
12 canvas.RenderPages(doc);
```

[See our C# samples on GitHub](#)

2.3.4 How do I convert a document to Markdown in Hi-Def mode?

In this Article

This sample demonstrates how to use the **Hyland Document Filters** SDK to convert a document into HiDef Markdown format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering its content into a structured Markdown output suitable for web presentation.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Opening:** Understand how to open a document for conversion to HiDef Markdown using the `OpenExtractor` method, specifying the appropriate open mode (e.g., `Paginated`) to ensure proper rendering.
3. **Output Canvas Creation:** Discover how to create an output canvas using the `MakeOutputCanvas` method, specifying the desired output file name and canvas type (e.g., `MARKDOWN`) for generating the Markdown structure.
4. **Content Rendering:** Learn how to render the document's pages into the output canvas, efficiently transforming the document's content into a Markdown format.
5. **Resource Management:** Gain insight into proper resource management in .NET by using `using` statements to ensure that document and canvas objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, converting documents to HiDef Markdown, and efficiently rendering document content for web applications.

Converting to HiDef Markdown

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 var canvasOptions = "MARKDOWN_SIMPLE_TABLE_STYLE=GRID;";
7 canvasOptions += "MARKDOWN_COMPLEX_TABLE_STYLE=HTML;";
8 canvasOptions += "MARKDOWN_INCLUDE_FOOTERS=OFF;";
9 canvasOptions += "MARKDOWN_INCLUDE_HEADERS=OFF;";
10 canvasOptions += "MARKDOWN_INCLUDE_FIELDS=OFF;";
11
12 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
13 using var canvas = api.MakeOutputCanvas("output.md", CanvasType.MARKDOWN, canvasOptions);
14
15 canvas.RenderPages(doc);
```

[See our C# samples on GitHub](#)

2.3.5 How do I convert a document to Markdown in Text mode?

In this Article

This sample demonstrates how to use the **Hyland Document Filters** SDK to convert a document into Text-mode Markdown format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering its content into structured Markdown output suitable for web presentation.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Opening:** Understand how to create an instance of the Extractor for a file and open it in Text mode using the `Open` method with the appropriate options.
3. **Markdown Options Configuration:** Discover how to configure Markdown options to customize the output style for tables and other elements in the Markdown file.
4. **Content Reading:** Learn how to read the document's content in chunks and output it, transforming the document's content into Markdown format.
5. **Resource Management:** Gain insight into proper resource management in .NET by using `using` statements to ensure that document objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, converting documents to Markdown in Text mode, and efficiently rendering document content for web applications.

Converting to Markdown in Text mode

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc"); // (2) Create an instance of the Extractor for a file
8
9 var markdownOptions = "MARKDOWN_SIMPLE_TABLE_STYLE=GRID;";
10 markdownOptions += "MARKDOWN_COMPLEX_TABLE_STYLE=HTML;";
11 markdownOptions += "MARKDOWN_INCLUDE_FOOTERS=OFF;";
12 markdownOptions += "MARKDOWN_INCLUDE_HEADERS=OFF;";
13 markdownOptions += "MARKDOWN_INCLUDE_FIELDS=OFF;";
14
15 doc.Open(ISYS11dfConstants.IGR_FORMAT_MARKDOWN, markdownOptions); // (3) Open the document for reading in
    text-mode
16
17 while (!doc.EndOfStream) // (4) Read the document in 4KB chunks and write it to the console
18 {
19     var text = doc.GetText(4096);
20     Console.Out.WriteLine(text);
21 }
22
23 doc.Close();
```

[See our C# samples on GitHub](#)

2.3.6 How do I convert a document to paginated HiDef HTML?

In this Article

This sample demonstrates how to use the Hyland Document Filters SDK to convert a document into HiDef HTML format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering its content into a structured HTML output suitable for web presentation.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Opening:** Understand how to open a document for conversion to HiDef HTML using the `OpenExtractor` method, specifying the appropriate open mode (e.g., `Paginated`) to ensure proper rendering.
3. **Output Canvas Creation:** Discover how to create an output canvas using the `MakeOutputCanvas` method, specifying the desired output file name and canvas type (e.g., `HTML`) for generating the HTML structure.
4. **Content Rendering:** Learn how to render the document's pages into the output canvas, efficiently transforming the document's content into a web-friendly format.
5. **Resource Management:** Gain insight into proper resource management in .NET by using `using` statements to ensure that document and canvas objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, converting documents to HiDef HTML, and efficiently rendering document content for web applications.

Converting to HiDef HTML5

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
7 using var canvas = api.MakeOutputCanvas("filename.html", CanvasType.HTML);
8
9 canvas.RenderPages(doc);
```

[See our C# samples on GitHub](#)

2.3.7 How do I convert a document to PNG images?

In this Article

This sample demonstrates how to use the **Hyland Document Filters** SDK to convert a document into PNG format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering each page into individual PNG images.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Opening:** Understand how to open a document for conversion to PNG using the `OpenExtractor` method, specifying the appropriate open mode (e.g., `Paginated`) to ensure correct page access.
3. **Page Iteration:** Discover how to iterate through the pages of the document, accessing each page for rendering.
4. **Output Canvas Creation:** Learn how to create an output canvas for each page using the `MakeOutputCanvas` method, specifying the desired output file name and canvas type (e.g., `PNG`) for generating the image files.
5. **Page Rendering:** Gain insight into rendering each page into the output canvas, efficiently creating PNG images from the document's content.
6. **Resource Management:** Understand proper resource management in .NET by using `using` statements to ensure that document and canvas objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, converting documents to PNG format, and efficiently rendering each page for image output.

Converting to PNG Images

```
1 using System.Linq;
2 using Hyland.DocumentFilters;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
8
9 foreach (var pageIter in doc.Pages.Select((page, index) => new { page, index }))
10 {
11     using (pageIter.page)
12     {
13         using var canvas = api.MakeOutputCanvas($"page-{pageIter.index+1}.png", CanvasType.PNG);
14         canvas.RenderPage(pageIter.page);
15     }
16 }
```


[See our C# samples on GitHub](#)

2.3.8 How do I convert a document to Structured XML?

In this Article

This sample demonstrates how to use the **Hyland Document Filters** SDK to convert a document into XML format. It provides a high-level workflow for initializing the Document Filters API, opening a document, and rendering its pages into a structured XML output.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Opening:** Understand how to open a document for conversion to XML using the `OpenExtractor` method, specifying the appropriate open mode (e.g., `Paginated`) to ensure proper rendering of pages.
3. **Output Canvas Creation:** Discover how to create an output canvas for the XML file using the `MakeOutputCanvas` method, specifying the desired output file name and canvas type (e.g., `XML`) for generating the XML document.
4. **Page Rendering:** Learn how to render all pages of the document into the output canvas, efficiently transforming the document's content into an XML format.
5. **Resource Management:** Gain insight into proper resource management in .NET by using `using` statements to ensure that document and canvas objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, converting documents to XML format, and efficiently rendering document content for XML output.

Converting to XML

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
7 using var canvas = api.MakeOutputCanvas("filename.xml", CanvasType.XML);
8
9 canvas.RenderPages(doc);
```

[See our C# samples on GitHub](#)

2.3.9 How do I convert a PDF to Markdown with Table Detection?

In this Article

This sample illustrates how to utilize the **Hyland Document Filters** SDK to convert a PDF document into HiDef Markdown format with enabled table detection. It provides a straightforward workflow for initializing the Document Filters API, opening the PDF document, and rendering its content into a well-structured Markdown output.

What You Will Learn:

1. **API Initialization:** Understand how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Opening:** Learn how to open a PDF document for conversion to HiDef Markdown using the `OpenExtractor` method, ensuring the use of the correct open mode (e.g., `Paginated`) and enabling table detection with specific options.
3. **Output Canvas Creation:** Discover how to create an output canvas using the `MakeOutputCanvas` method, specifying the desired output file name and canvas type (e.g., `MARKDOWN`) to generate the Markdown structure.
4. **Content Rendering:** Learn how to render the document's pages into the output canvas, efficiently transforming the document's content, including detected tables, into Markdown format.
5. **Resource Management:** Gain insights into effective resource management in .NET by utilizing `using` statements to ensure that document and canvas objects are properly disposed of after use.

By following this sample, you will become familiar with the basics of setting up the Document Filters API, converting PDF documents to HiDef Markdown with table detection, and efficiently rendering document content for web applications.

See [Table Detection](#) for more details.

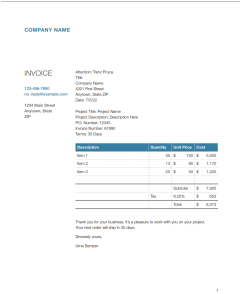
Converting a PDF to Markdown with Table Detection

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 var documentOptions = "PDF_TABLE_DETECTION=ON;";
7
8 var canvasOptions = "MARKDOWN_SIMPLE_TABLE_STYLE=GRID;";
9 canvasOptions += "MARKDOWN_COMPLEX_TABLE_STYLE=HTML;";
10 canvasOptions += "MARKDOWN_INCLUDE_FOOTERS=OFF;";
11 canvasOptions += "MARKDOWN_INCLUDE_HEADERS=OFF;";
12 canvasOptions += "MARKDOWN_INCLUDE_FIELDS=OFF;";
13
14 using var doc = api.OpenExtractor("invoice.pdf", OpenMode.Paginated, options: documentOptions);
15 using var canvas = api.MakeOutputCanvas("output.md", CanvasType.MARKDOWN, canvasOptions);
16
17 canvas.RenderPages(doc);
```

[See our C# samples on GitHub](#)

Reviewing the Output

Let's examine the following PDF invoice file.



With PDF_TABLE_DETECTION

With `PDF_TABLE_DETECTION` enabled, the content is accurately represented in table format:

Description	**Quantity**	**Unit Price**	**Cost**
Item 1	55	100	5,500
Item 2	13	90	1,170
Item 3	25	50	1,250
	Subtotal	7,920	
	Tax	8.25%	653
	Total	8,573	

Thank you for your business. It's a pleasure to work with you on your project.
Your next order will ship in 30 days.

Without PDF_TABLE_DETECTION

When `PDF_TABLE_DETECTION` is not enabled, the line items in the table are treated as standard text, resulting in the following markdown output:

```
Item 1
55    100    5,500
Item 2
13    90     1,170
Item 3
25    50     1,250
Subtotal 7,920
Tax 8.25%   653
Total 8,573
```

Thank you for your business. It's a pleasure to work with you on your project.
Your next order will ship in 30 days.



Note

Table detection is not an exact science; it involves interpreting the original intent from the available information in the file. As a result, it may occasionally misidentify content as a table or fail to detect existing tables.

2.3.10 How do I convert HTML to Hi-Def?

In this Article

This sample demonstrates how to use the **Hyland Document Filters** SDK to convert an HTML document into Hi-Def format. It provides a high-level workflow for initializing the Document Filters API, opening an HTML document, handling images, and rendering structured output.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Opening:** Understand how to create an instance of the Extractor for an HTML file and open it in Hi-Def mode.
3. **Handling Images in HTML:** Learn how to resolve and process images referenced in the HTML document, whether they are local files or external URLs.
4. **Resource Management:** Gain insight into proper resource management to ensure document objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, converting HTML to Hi-Def, and effectively managing embedded resources.

Handling Images in HTML

When processing HTML files, referenced images may need to be resolved and included in the output. The Document Filters SDK provides multiple ways to handle images depending on their source:

- **External Images (HTTP/HTTPS URLs):**

If an `<img>` tag references an image using a full URL (e.g., `http://example.com/image.png`), the image can be automatically downloaded by enabling the `HTTP_EXTERNAL_IMAGES=on` option. This will fetch the image, even from the internet.

- **Approving Image Downloads:**

The `HTTP_EXTERNAL_IMAGES` option can be used with the `IGR_OPEN_CALLBACK_ACTION_APPROVE_EXTERNAL_RESOURCE` callback. This allows the application to approve or deny external image downloads, enabling control over which images are retrieved.

- **Local Disk Images:**

If the HTML file is opened using a filename (not a stream or byte array), referenced images can be resolved from the local disk. However, the images must be in the same directory or a subdirectory of the HTML file.

- **Processing Images from Streams:**

When opening an HTML file from a stream or byte array, the SDK provides the

`IGR_OPEN_CALLBACK_ACTION_GET_RESOURCE_STREAM` callback. This allows the application to provide image data dynamically without writing files to disk. This method is ideal for in-memory processing workflows.

Converting HTML to Hi-Def

This example demonstrates loading a HTML-based document type, loading any referenced external image.

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 // Open the document with the HTTP_EXTERNAL_IMAGES option, which allows
7 // external images to be download and displayed
8 using var doc = api.OpenExtractor("filename.html", OpenMode.Paginated,
9     options: "HTTP_EXTERNAL_IMAGES=on");
10
11 // Create a PDF canvas and render the pages
12 using var canvas = api.MakeOutputCanvas("filename.pdf", CanvasType.PDF);
13
14 canvas.RenderPages(doc);
```

[See our C# samples on GitHub](#)

Converting HTML to Hi-Def with Download Approval

This example demonstrates loading a HTML-based document type, loading approved referenced external image only.

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 var cb = new Func<OpenCallback, int>(info =>
8 {
9     if (info.Action == ISYS11dfConstants.IGR_OPEN_CALLBACK_ACTION_APPROVE_EXTERNAL_RESOURCE)
10     {
11         if (info.ApproveExternalResource.url.Contains("http://images.com"))
12             return ISYS11dfConstants.IGR_OK;
13         else
14             return ISYS11dfConstants.IGR_CANCELLED;
15     }
16     return 0;
17 });
18
19 string html = @"
20 <html>
21     <body>
22         <img src='http://images.com/foo.png' />
23         <img src='http://no-images.com/bar.png' />
24     </body>
25 </html>";
26 byte[] htmlBytes = System.Text.Encoding.UTF8.GetBytes(html);
27
28 // Create an extractor for the HTML from a byte array.
29 using var doc = api.GetExtractor(htmlBytes);
30
31 // Open the document with the HTTP_EXTERNAL_IMAGES option and a callback, which
32 // allows external images to be download and displayed once approved
33 doc.Open(OpenMode.Paginated, options: "HTTP_EXTERNAL_IMAGES=on", callback: cb);
34
35 // Create a PDF canvas and render the pages
36 using var canvas = api.MakeOutputCanvas("filename.pdf", CanvasType.PDF);
37 canvas.RenderPages(doc);
```

[See our C# samples on GitHub](#)

Converting HTML to Hi-Def with Resource Streams

This example demonstrates loading a HTML-based document type, loading images from app-provided resource streams.

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 string html = @"
8 <html>
9     <body>
10         <img src='foo.png' />
11         <img src='bar.png' />
12     </body>
13 </html>";
14 byte[] htmlBytes = System.Text.Encoding.UTF8.GetBytes(html);
15
16 // Create an extractor for the HTML from a byte array.
17 using var doc = api.GetExtractor(htmlBytes);
18
19 // Set the callback to fetch the external images
20 doc.GetResourceCallback = (url) =>
21 {
22     if (url == "foo.png" || url == "bar.png")
23         return System.IO.File.OpenRead("my-files/" + url);
24     else
25         return null;
26 };
27
28 // Open the document in hi-def mode. The resource stream callback will
29 // be called for each image.
30 doc.Open(OpenMode.Paginated, options: "HTTP_EXTERNAL_IMAGES=on");
31
32 // Create a PDF canvas and render the pages
33 using var canvas = api.MakeOutputCanvas("filename.pdf", CanvasType.PDF);
34 canvas.RenderPages(doc);
```

[See our C# samples on GitHub](#)

2.4 Extracting Files

2.4.1 How do I extract metadata from a document?

In this Article

This sample demonstrates how to use the Hyland Document Filters SDK to extract metadata from a document. It provides a high-level workflow for initializing the Document Filters API, opening a document, and retrieving its metadata.

What you will learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Extraction:** Understand how to open a document for text extraction by using the `GetExtractor` method, specifying the document and the mode (`BodyAndMeta`) for reading the body and metadata.
3. **Reading Metadata:** Discover how to retrieve specific metadata fields, such as title, author, creation date, and other relevant information, by utilizing appropriate methods provided by the SDK.
4. **Resource Management:** Learn about proper resource management in .NET by using using statements to ensure that document and API objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, handling document extraction, and iterating over the document's content efficiently.

Installing Document Filters

Document Filters has several NuGet packages available. For most scenarios, however, you typically only need `Hyland.DocumentFilters`.

You can install it using the following command:

```
dotnet add package Hyland.DocumentFilters
```

For full setup instructions see [Getting Started with .NET](#)

Extracting Metadata

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(Hyland.DocumentFilters.OpenType.MetaOnly);
9
10 while (!doc.EndOfStream)
11 {
12     var text = doc.GetText(4096);
13     Console.Out.WriteLine(text);
14 }
15
16 doc.Close();
```

[See our C# samples on GitHub](#)

2.4.2 How do I extract sub-documents from documents and archives?

In this Article

This sample demonstrates how to use the **Hyland Document Filters SDK** to extract metadata from subfiles within a document archive. It provides a high-level workflow for initializing the Document Filters API, opening a document, and iterating through its subfiles to read their associated metadata.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Extraction:** Understand how to open a document for metadata extraction using the `GetExtractor` method, specifying the document archive (e.g., a ZIP file) and the mode (`BodyAndMeta`) for accessing both the body and metadata.
3. **Subfile Metadata Access:** Discover how to iterate through the subfiles within the document archive, accessing key metadata fields such as name, ID, date, and size.
4. **Resource Management:** Learn about proper resource management in .NET by using `using` statements to ensure that subfile objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, extracting metadata from subfiles, and efficiently accessing important information from document archives.

Extracting SubFiles

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.zip");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 foreach (var subfile in doc.SubFiles)
11 {
12     using (subfile)
13     {
14         // act on subfile
15         Console.Out.WriteLine("Name: " + subfile.Name);
16         Console.Out.WriteLine("ID: " + subfile.ID);
17         Console.Out.WriteLine("Date: " + subfile.FileDate);
18         Console.Out.WriteLine("Size: " + subfile.FileSize);
19     }
20 }
```

[See our C# samples on GitHub](#)

2.4.3 How do I extract text and metadata from a document?

In this Article

This sample demonstrates how to use the Hyland Document Filters SDK to extract text from a document. It provides a high-level workflow for initializing the Document Filters API, opening a document, and reading its text content in chunks until the end of the document is reached.

What you will learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Extraction:** Understand how to open a document for text extraction by using the `GetExtractor` method, specifying the document and the mode (`BodyAndMeta`) for reading the body and metadata.
3. **Text Streaming:** Discover how to read the document's text content in chunks using the `GetText()` method, which allows for efficient extraction of large documents by processing portions of the text in a loop.
4. **Resource Management:** Learn about proper resource management in .NET by using using statements to ensure that document and API objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, handling document extraction, and iterating over the document's content efficiently.

Installing Document Filters

Document Filters has several NuGet packages available. For most scenarios, however, you typically only need `Hyland.DocumentFilters`.

You can install it using the following command:

```
dotnet add package Hyland.DocumentFilters
```

For full setup instructions see [Getting Started with .NET](#)

Extracting Text

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc"); // (2) Create an instance of the Extractor for a file
8
9 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta); // (3) Open the document for reading in text-mode
10
11 while (!doc.EndOfStream) // (4) Read the document in 4KB chunks and write it to the console
12 {
13     var text = doc.GetText(4096);
14     Console.Out.WriteLine(text);
15 }
16
17 doc.Close();
```

[See our C# samples on GitHub](#)

2.4.4 How do I handle password-protected files?

In this Article

Dealing with password-protected files is a common requirement in content extraction and processing workflows. Files may require passwords to access their content or even to enumerate their internal structure if they're embedded within containers such as ZIP, RAR, or compound documents.

Hyland Document Filters supports a flexible, callback-based mechanism that enables your application to provide passwords on demand. This approach allows for secure and customizable handling of different protection scenarios, including files with multiple layers of password protection.

TYPES OF PASSWORD-PROTECTED FILES

There are three primary categories of password protection:

1. **Encrypted Documents:** Individual files, such as PDFs or Microsoft Word documents, may require a password to open and view their content.
2. **Protected Containers:** Archive formats such as ZIP, RAR, and some compound document types may require a password simply to list their contents.
3. **Item-Level Protection in Containers:** Some containers allow each embedded item to be independently password-protected. These items may have different passwords or none at all.

CALLBACK-BASED PASSWORD HANDLING

To accommodate these scenarios, Document Filters uses a **password callback** mechanism. When a password is required, your application is notified with an identifier for the item requiring the password. Your callback can respond with the correct password—or cycle through known passwords—based on the item's ID and context.

- **Empty ID:** Indicates the top-level document or container.
- **SubFile.Id:** Identifies a specific embedded item within a container.

This mechanism allows precise control over which password is attempted and when, making it ideal for processing nested or variably protected content.

IMPORTANT CONSIDERATIONS

- Document Filters does **not** track which passwords have been tried—**you must manage retries** in your callback logic.
- If you repeatedly supply the same incorrect password for a given item, the API will not re-invoke your callback for that same password again.

This ensures efficient operation and prevents redundant work, especially when working with large archives or complex document hierarchies.

What You Will Learn:

1. **API Initialization:** How to initialize the Document Filters API and register a password callback function.
2. **Password Callback Implementation:** How to implement the password callback function to handle top-level documents, protected containers, and item-level passwords.
3. **Context Awareness:** How to distinguish between top-level and embedded items using the provided identifier.
4. **Retry Management:** Best practices for managing and re-attempting passwords only when necessary.
5. **Secure Handling:** How to safely manage passwords in memory and avoid reuse of failed credentials.

By the end of this tutorial, you'll understand how to integrate password handling into your document extraction workflow, supporting secure and scalable processing of protected content.

Handling Password-Protected Files

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 var licenseKey = Environment.GetEnvironmentVariable("DOCFILTERS_LICENSE_KEY") ?? "YOUR_LICENSE_KEY_HERE";
6 api.Initialize(licenseKey, ".");
7
8 using var container = api.GetExtractor("sample.7z"); // (2) Create an instance of the extractor
9 container.PasswordCallback = (string id) => "password1234"; // (3) Set the password callback
10 container.Open(OpenMode.Text);
11
12 foreach (var item in container.SubFiles)
13 {
14     Console.WriteLine($"{item.FileDate:yyyy-MM-dd HH:mm} {item.FileSize,10} {item.Name}");
15 }
```

[See our C# samples on GitHub](#)

2.5 Other

2.5.1 How do I capture log messages?

In this Article

Capturing diagnostic and operational information during document processing is essential for debugging, auditing, and optimizing application performance. Document Filters provides a powerful and flexible way to capture logs through a callback-based logging interface.

With this system, your application can dynamically specify the desired level of verbosity per internal module and receive structured log messages during execution. This mechanism gives you fine-grained control over what messages are emitted and how they're handled.

TWO-PART LOGGING CALLBACK MECHANISM

The logging system is composed of **two callbacks** that work together:

- 1. **LogLevelCallback:** This callback lets your application define the log level to use for each internal module of the Document Filters library.
- 2. **LogMessageCallback:** This callback receives actual log messages emitted by the library, filtered according to the levels set in the `LogLevelCallback`.

LOG LEVELS

You can return one of the following values from `IGR_LOG_LEVEL_TYPE` in your `LogLevelCallback`:

Constant	Value	Description
<code>IGR_LOG_LEVEL_NONE</code>	0	No messages will be logged.
<code>IGR_LOG_LEVEL_CRITICAL</code>	1	Only critical failures are logged.
<code>IGR_LOG_LEVEL_ERROR</code>	2	Errors and critical issues are logged.
<code>IGR_LOG_LEVEL_WARNING</code>	3	Warnings, errors, and critical issues.
<code>IGR_LOG_LEVEL_INFORMATION</code>	4	Informational messages and above.
<code>IGR_LOG_LEVEL_DEBUG</code>	5	Debug-level detail and higher.
<code>IGR_LOG_LEVEL_TRACE</code>	6	Maximum verbosity; all logs are captured.

Log messages are only delivered to your `LogMessageCallback` if their severity is less than or equal to the level returned by the corresponding `LogLevelCallback`.

USE CASES

- **Debugging:** Enable `TRACE` or `DEBUG` level logs to diagnose subtle issues during development or integration.
- **Monitoring:** Capture `INFORMATION` and `WARNING` logs in production to monitor the health of your document processing workflow.
- **Auditing:** Retain `ERROR` and `CRITICAL` logs for audit trails or incident response.

What You Will Learn:

1. **API Initialization:** How to initialize the Document Filters API and register logging callbacks.
2. **Module-Aware Logging:** How to define per-module logging behavior using the `LogLevelCallback`.
3. **Capturing Messages:** How to implement a `LogMessageCallback` that receives and records structured logs.
4. **Performance Tips:** Understand the performance implications of higher log levels and when to adjust verbosity.

By working through this tutorial, you'll be able to integrate rich logging support into your application, aiding in diagnostics, auditing, and system observability.

Capturing Log Messages

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 var licenseKey = Environment.GetEnvironmentVariable("DOCFILTERS_LICENSE_KEY") ?? "YOUR_LICENSE_KEY_HERE";
6 api.Initialize(licenseKey, ".");
7
8 using var container = api.GetExtractor("sample.pst"); // (2) Create an instance of the extractor
9
10 // (3) Set the logging callbacks
11 container.LogLevelCallback = (string module) => ISYS11dfConstants.IGR_LOG_LEVEL_TRACE;
12 container.LogMessageCallback = (int level, string module, string message) => Console.WriteLine($"{module}
13 {message}");
14 container.Open(OpenMode.Text);
15
16 foreach (var item in container.SubFiles)
17 {
18     Console.WriteLine($"{item.FileDate:yyyy-MM-dd HH:mm} {item.FileSize,10} {item.Name}");
19 }
```

[See our C# samples on GitHub](#)

2.5.2 How do I compare documents?

In this Article

Comparing documents presents a multifaceted challenge due to the dynamic nature of content. Documents can undergo various changes such as additions, deletions, modifications, reformatting, and even movement across pages. These alterations not only impact the visual appearance but also the underlying structure and semantics of the content.

One of the critical challenges arises from elements like headers, footers, fields (common in applications like Microsoft Word), and Table of Contents. These elements often contain metadata or navigational information that may change between document versions. For instance, page numbers in headers or footers may differ, yet these changes are typically not relevant to the content itself and should be excluded from the comparison.

Handling these intricacies is crucial for accurate document comparison, ensuring that only meaningful content differences are identified while disregarding irrelevant variations caused by formatting or metadata alterations.

DOCUMENT-LEVEL, PAGE-LEVEL, AND ELEMENT-LEVEL DIFFING

The API empowers developers with the flexibility to perform comparisons at different granularities, including document-level, page-level, and element-level diffing. This granularity ensures that comparisons can be tailored to specific requirements, whether it's assessing entire document revisions, individual page modifications, or detailed analysis of specific elements such as paragraphs, tables, or images.

EFFICIENT HANDLING OF LARGE DOCUMENTS

Moreover, the Document Filters API excels in handling large documents efficiently. Its ability to load content on demand minimizes memory footprint and enhances performance, particularly when dealing with extensive documents. This capability ensures smooth and responsive processing, even when dealing with substantial volumes of content, thereby enhancing productivity and user experience.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Document Opening:** Understand how to open a document for conversion to XML using the `OpenExtractor` method, specifying the appropriate open mode (e.g., `Paginated`) to ensure proper rendering of pages.
3. **Output Canvas Creation:** Discover how to create an output canvas for the XML file using the `MakeOutputCanvas` method, specifying the desired output file name and canvas type (e.g., `XML`) for generating the XML document.
4. **Page Rendering:** Learn how to render all pages of the document into the output canvas, efficiently transforming the document's content into an XML format.
5. **Resource Management:** Gain insight into proper resource management in .NET by using `using` statements to ensure that document and canvas objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, converting documents to XML format, and efficiently rendering document content for XML output.

Comparing Documents

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using (var doc1 = api.OpenExtractor("original.docx", OpenMode.Paginated))
7 using (var doc2 = api.OpenExtractor("revision.docx", OpenMode.Paginated))
8 using (var compare = doc1.Compare(doc2))
9 {
10     while (compare.MoveNext())
11     {
12         var diff = compare.Current;
13         // work with diff...
14     }
15 }
```

[See our C# samples on GitHub](#)

2.5.3 How do I create a barcode?

In this Article

This sample demonstrates how to use the **Hyland Document Filters** SDK to generate a QR code and save it as a PNG image. It provides a high-level workflow for initializing the Document Filters API, creating an output canvas, and annotating it with a QR code.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable barcode generation.
2. **Output Canvas Creation:** Understand how to create an output canvas for the PNG file using the `MakeOutputCanvas` method, specifying the desired output file name and canvas type (e.g., `PNG`).
3. **Page Preparation:** Discover how to prepare a blank page on the output canvas with specified dimensions, ensuring there is space for the QR code.
4. **QR Code Annotation:** Learn how to annotate the canvas with a QR code using the `Annotate` method, including setting properties such as the caption, text, and positioning of the QR code rectangle.
5. **Resource Management:** Gain insight into proper resource management in .NET by using `using` statements to ensure that the canvas object is properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, generating QR codes, and rendering them to an image file.

Creating a Barcode

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var canvas = api.MakeOutputCanvas("barcode.png", CanvasType.PNG);
7 canvas.BlankPage(400, 400);
8 canvas.Annotate(new Hyland.DocumentFilters.Annotations.QrCode {
9     Caption = "Hello World",
10    Text = "http://www.hyland.com",
11    Rect = System.Drawing.Rectangle.FromLTRB(20, 20, 380, 380)
12 });
```

[See our C# samples on GitHub](#)

2.5.4 How do I integrate with a custom OCR engine?

In this Article

Hyland Document Filters includes built-in support for Tesseract OCR. However, if you're working with AI-enhanced OCR models like EasyOCR, PaddleOCR, or a proprietary solution, you can register a custom OCR callback to process rasterized image regions yourself.

This integration gives you the flexibility to choose an OCR engine that meets your accuracy, language support, or AI requirements—while still benefiting from Document Filters' powerful layout analysis and downstream output options, including structured formats like Markdown and JSON.

Document Filters takes care of:

- Identifying image regions that require OCR
- Segmenting the page into blocks, paragraphs, and lines
- Invoking your OCR callback with image data
- Merging the OCR results back into the document's logical structure

WHEN TO USE A CUSTOM OCR CALLBACK

Use a custom OCR callback when:

- You want to leverage a machine learning-based OCR engine
- You require specialized language models or handwriting recognition
- You need tighter control over OCR accuracy or performance
- You're integrating with an existing OCR microservice or distributed system

BENEFITS OF USING THE DOCUMENT FILTERS PIPELINE

Even when using your own OCR engine, routing through the Document Filters pipeline provides:

- Unified processing of raster and vector documents
- Consistent layout analysis to group OCR output logically
- Multi-format output (e.g., JSON, XML, Markdown)
- Minimal effort to integrate OCR into a larger document processing pipeline

What You Will Learn:

1. **Register an OCR Callback:** Learn how to implement and register a custom OCR callback with the API.
2. **Understand the OCR Request:** Discover how the API passes image data, bounding boxes, and hints to your OCR function.
3. **Return Recognized Text:** Learn how to format and return recognized text back to Document Filters.
4. **Render Output:** See how the recognized text is integrated into the document output, with layout preserved.
5. **Best Practices:** Explore tips for performance, parallelism, and integration with ML-based services.

By following this tutorial, you'll be able to plug in any OCR engine and produce structured text output via Document Filters.

Integrating a Custom OCR Engine

This sample demonstrates how to use the `TesseractOCR` .NET package to perform OCR on each page image.

To get started, install the `TesseractOCR` package in your project:


```
dotnet add MyProject.csproj package TesseractOCR
```

```

1 using Hyland.DocumentFilters;
2 using System;
3 using System.Linq;
4
5 var ocrCallback = (OpenCallbackActionOcrImage image) =>
6 {
7     var tessdata = Environment.GetEnvironmentVariable("TESSDATA") ?? "./tessdata";
8
9     Console.WriteLine($"OCR'ing image {image.SourcePageIndex}. Tessdata path: {tessdata}");
10
11     var testImagePath = "test_image.png";
12     image.SaveImage(testImagePath, "image/png");
13     using var ocrEngine = new TesseractOCR.Engine(tessdata, "eng");
14     using var img = TesseractOCR.Pix.Image.LoadFromFile(testImagePath);
15     img.XRes = 300;
16     img.YRes = 300;
17     using var ocrPage = ocrEngine.Process(img);
18
19     var words = ocrPage.Layout
20         .SelectMany(block => block.Paragraphs)
21         .SelectMany(paragraph => paragraph.TextLines)
22         .SelectMany(textLine => textLine.Words);
23
24     foreach (var word in words)
25     {
26         if (word.BoundingBox == null)
27             continue;
28
29         System.Drawing.Rectangle rect = System.Drawing.Rectangle.FromLTRB(
30             word.BoundingBox.Value.X1,
31             word.BoundingBox.Value.Y1,
32             word.BoundingBox.Value.X2,
33             word.BoundingBox.Value.Y2);
34         image.AddText(word.Text, rect);
35     }
36
37     return true;
38 };
39
40 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
41 var licenseKey = Environment.GetEnvironmentVariable("DOCFILTERS_LICENSE_KEY") ?? "YOUR_LICENSE_KEY_HERE";
42 api.Initialize(licenseKey, ".");
43
44 using var doc = api.GetExtractor("sample.tif"); // (2) Create an instance of the Extractor for a file
45 doc.OcrImageCallback = ocrCallback;
46 doc.Open(ISYS11dfConstants.IGR_FORMAT_IMAGE, "OCR=on;OCR_ENGINE=CUSTOM"); // (3) Open the document for
    reading in text-mode
47
48 foreach (var page in doc.Pages.Select((value, index) => new { Index = index, Page = value })) // (4)
    Iterate through the pages of the document with index
49 {
50     using (page.Page)
51     {
52         // This is the page object for the current page
53         // You can use this to get information about the page, such as its text length
54         Console.WriteLine($"Page Index: {page.Index + 1}, Text: {page.Page.Text}");
55     }
56 }

```

[See our C# samples on GitHub](#)

2.5.5 How do I localize metadata?

In this Article

This sample demonstrates how to use the **Hyland Document Filters** SDK to localize an Outlook message and generate a PDF document. It provides a high-level workflow for initializing the Document Filters API, localizing message fields, opening the document, and rendering its content into a PDF format.

What You Will Learn:

1. **API Initialization:** Learn how to initialize the Hyland Document Filters API with a valid license key to enable document processing.
2. **Message Localization:** Understand how to localize specific fields of an Outlook message, such as the subject, recipient, and sender, using the `Localize` property to set translations.
3. **Document Opening:** Discover how to open the Outlook message for rendering by specifying the appropriate format (e.g., `IGR_FORMAT_IMAGE`).
4. **Output Canvas Creation:** Learn how to create an output canvas for the PDF file using the `MakeOutputCanvas` method, specifying the desired output file name and canvas type (e.g., `PDF`).
5. **Content Rendering:** Gain insight into rendering the localized message pages into the output canvas, transforming the document's content into a PDF format.
6. **Resource Management:** Understand proper resource management in .NET by using `using` statements to ensure that document and canvas objects are properly disposed of after use.

By working through this sample, you will become familiar with the basics of setting up the Document Filters API, localizing Outlook messages, and efficiently generating PDF documents.

Localizing Metadata

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using (var doc = api.GetExtractor("outlook.msg"))
7 {
8     doc.Localize["Subject"] = "Objet";
9     doc.Localize["To"] = "À";
10    doc.Localize["From"] = "De";
11    doc.Localize["Bcc"] = "Cci";
12    doc.Localize["Sent"] = "Envoyé";
13    doc.Open(OpenMode.Paginated);
14
15    using (var canvas = api.MakeOutputCanvas("output.pdf", CanvasType.PDF))
16    {
17        canvas.RenderPages(doc);
18    }
19 }
```

[See our C# samples on GitHub](#)

3. Reference

3.1 Overview

C reference	The “C” API is implemented as a DLL or Shared Library, depending on the platform. These functions are designed for procedural languages and are callable from C and other languages, such as Delphi and Visual Basic.
Object reference	List of available objects in Document Filters.
Structured XML	Structured XML is a hierarchical Document Object Model (DOM) that represents the paginated view of a document. Most nodes have two distinct sections:

3.2 C reference

3.2.1 C reference

The “C” API is implemented as a DLL or Shared Library, depending on the platform. These functions are designed for procedural languages and are callable from C and other languages, such as Delphi and Visual Basic.

Functions

Init_Instance	Init_Instance initializes the Document Filters engine and authenticates the license. Init_Instance must always be the first call made by any application to the Document Filters library.
Close_Instance	Close_Instance advises the Document Filters engine that the program is finished.
IGR_Calculate_MD5	IGR_Calculate_MD5 will calculate the MD5 hash of an input stream for unique document identification.
IGR_Calculate_SHA1	IGR_Calculate_SHA1 will calculate the SHA1 hash of an input stream for unique document identification.
IGR_Canvas_Annotate_JSON	IGR_Canvas_Annotate_JSON draws the given annotation onto the current page. The annotation is delivered as a string of JSON text. The schema for Annotations can be found in the Document Filters installation directory in “Annotation-Schema.json”.
IGR_Canvas_Arc	Draws an arc on the image along the perimeter of the ellipse bounded by the specified rectangle, with the current pen.
IGR_Canvas_Blank_Page	Inserts a blank page of the specified size into the Canvas. This can be useful when constructing documents, where you need to insert constructed pages created with the drawing or markup APIs.
IGR_Canvas_Bookmarks_Append	Append a single bookmark to the canvas. Bookmarks are added one at a time and the level parameter in the IGR_Bookmark determines the hierarchy. Bookmarks in a hierarchy must be added in order and a bookmark’s level

	cannot be more than one more level deep than the previous bookmark.
IGR_Canvas_Bookmarks_Clear	Remove existing bookmarks from the canvas.
IGR_Canvas_Chord	Draws a closed figure represented by the intersection of a line and an ellipse, with the current pen. The ellipse is bisected by a line that runs between X3,Y3 and X4,Y4.
IGR_Canvas_DrawImage	Renders the image specified by the ImageData parameter on the canvas at the given location given by the X and Y coordinates.
IGR_Canvas_DrawScaleImage	Renders the image specified by the ImageData parameter on the canvas at the given location given by the X and Y coordinates, scaling the output to width and height.
IGR_Canvas_Ellipse	Draws an ellipse defined by a bounding rectangle on the canvas, outlined with the current pen and filled with the current brush.
IGR_Canvas_LineTo	Draws a line on the canvas from the current pen position to the point specified by X and Y, and sets the pen position to (X, Y).
IGR_Canvas_MeasureText	Returns the width and height in pixels, of a string if rendered with the current font.
IGR_Canvas_MoveTo	Changes the current drawing position to the point (X,Y).
IGR_Canvas_Pie	Draws a pie-shaped section of the ellipse bounded by the rectangle (X1, Y1) and (X2, Y2) on the canvas.
IGR_Canvas_Rect	Draws a rectangle using the current brush and pen of the canvas to fill and draw the border.
IGR_Canvas_Reset	IGR_Canvas_Reset restores the canvas object back to the default set of options, including opacity, rotation, pens, and brushes.
IGR_Canvas_Rotation	IGR_Canvas_Rotation sets the rotation to be applied for subsequent drawing methods.
IGR_Canvas_RoundRect	Draws a rectangle with rounded corners, outlined with the current pen and filled with the current brush, on the canvas.

IGR_Canvas_SetBrush	Updates the current brush on the canvas with the given color and style; brushes are used when drawing rectangles, shapes and text.
IGR_Canvas_SetFont	Specifies the font to use when drawing text to the canvas. All subsequent calls to TextOut and MeasureText will use this font.
IGR_Canvas_SetOpacity	Set the opacity/transparency for future drawing routines.
IGR_Canvas_SetPen	Updates the canvas pen on the canvas with the specific color, width and style.
IGR_Canvas_TextOut	Writes a string on the canvas, starting at X and Y, and then updates the pen position to the end of the string. The text is written with the current font, and filled with the current brush.
IGR_Canvas_TextRect	Writes a string inside a clipping rectangle, using the current brush and font.
IGR_Close_Canvas	IGR_Close_Canvas releases the resources associated with the canvas handle. It must be called for every canvas opened by IGR_Make_Output_Canvas or IGR_Make_Output_Canvas_On , and must be called before closing the document with IGR_Close_File .
IGR_Close_File	IGR_Close_File releases the resources associated with the file handle. It must be called for every document opened by IGR_Open_File .
IGR_Close_Page	IGR_Close_Page releases the resources associated with the page handle. It must be called for every page opened by IGR_Open_Page , and must be called before closing the document with IGR_Close_File .
IGR_Convert_File	IGR_Convert_File converts the specified document into a plain text or HTML file, without the need to call IGR_Open_File and IGR_Get_Text .
IGR_Enum_Bookmarks	Enumerate all bookmarks, up-to <code>recurse_depth</code> , to an application-defined callback function.
IGR_Enum_Page_Elements	IGR_Enum_Page_Elements recursively walks the page elements of <code>source</code> calling the user-supplied <code>callback</code> .

IGR_Extend_Stream	IGR_Extend_Stream allows the C / C++ API user to create a custom stream that accepts callbacks from Document Filters. The callbacks allow the passing of Additional information about the stream.
IGR_Extract_Image	IGR_Extract_Image extracts an image to disk from a document opened with HTML or Image conversion in affect.
IGR_Extract_Image_Stream	IGR_Extract_Image_Stream extracts an image to a stream from a document, given the ID of the image.
IGR_Extract_Page_Image	<code>IGR_Extract_Page_Image</code> extracts the embedded image referenced by <code>id</code> to the file specified by <code>destination</code> .
IGR_Extract_Page_Image_Stream	<code>IGR_Extract_Page_Image_Stream</code> extracts the embedded image referenced by <code>id</code> as an IGRStream.
IGR_Extract_Subfile	IGR_Extract_Subfile extracts a sub-document to disk from a compound document, given the ID of the sub-document.
IGR_Extract_Subfile_Stream	IGR_Extract_Subfile_Stream extracts a sub-document to a stream from a compound document, given the ID of the sub-document.
IGR_Free_Page_Pixels	The IGR_Free_Page_Pixels method is used to dispose of page pixels extracted with IGR_Get_Page_Pixels .
IGR_Get_Bookmark_First_Child	IGR_Get_Bookmark_First_Child returns the first child bookmark of the provided bookmark, or NULL if the current item has no children.
IGR_Get_Bookmark_First_Child	IGR_Get_Bookmark_First_Child returns the first child bookmark of the provided bookmark, or NULL if the current item has no children.
IGR_Get_Bookmark_Next_Sibling	IGR_Get_Bookmark_Next_Sibling returns the peer/sibling of the provided bookmark, or NULL if the current item is the last in it's chain.
IGR_Get_Bookmark_Root	IGR_Get_Bookmark_Root provides access to the top-level root node of the bookmark hierarchy.
IGR_Get_File_Type	IGR_Get_File_Type gets the type and the capabilities of a given document.

IGR_Get_Format_Attribute	IGR_Get_Format_Attribute returns information about the supported file type.
IGR_Get_Handle_File_Type	IGR_Get_Handle_File_Type gets the type and the capabilities of an open document.
IGR_Get_Image_Entry	IGR_Get_Image_Entry enumerates the set of images, when HTML or Image conversion is in affect.
IGR_Get_Images_Enumerator	IGR_Get_Images_Enumerator returns a new enumerator for the set of images, when HTML conversion is in effect as set by the Open Document Flags, contained in a previously opened document
IGR_Get_Memory_Status	IGR_Get_Memory_Status is a Windows-only memory diagnostic.
IGR_Get_Option_Attribute	IGR_Get_Option_Attribute returns information about each published option value.
IGR_Get_Page_Annotation_Count	IGR_Get_Page_Annotation_Count returns the number of annotations that are contained on the page. For Office documents, comments are extracted and included in the annotation count. For comments spanning multiple pages, the OFFICE_COMMENT_PAGE open document option determines if the first, last or all pages containing the comments should return an annotation.
IGR_Get_Page_Annotation_Long	IGR_Get_Page_Annotation_Long copies an integer value from a given property of an existing annotation into the user supplied buffer.
IGR_Get_Page_Annotation_Str	IGR_Get_Page_Annotation_Str copies a string value from a given property of an existing annotation into the user supplied buffer.
IGR_Get_Page_Annotations	IGR_Get_Page_Annotations copies references of page annotations into the user supplied array. The caller can iterate over all the page annotations by incrementing the Index parameter. For Office documents, comments will be extracted and returned in the page annotations array. For comments spanning multiple pages, the OFFICE_COMMENT_PAGE

	open document option determines if the first, last or all pages containing the comments should return an annotation.
IGR_Get_Page_Attribute	IGR_Get_Page_Attribute returns style or properties of an open page; see <page> under Structured XML for a full list of options.
IGR_Get_Page_Count	IGR_Get_Page_Count returns the number of pages generated for an open document. This method only works on functions opened with IGR_FORMAT_IMAGE.
IGR_Get_Page_Dimensions	IGR_Get_Page_Dimensions returns the size of the given page in pixels.
IGR_Get_Page_Element_First_Child	IGR_Get_Page_Element_First_Child fills <code>dest</code> with the first child of the <code>parent</code> element.
IGR_Get_Page_Element_Next_Sibling	IGR_Get_Page_Element_Next_Sibling fills <code>dest</code> with the next sibling node of <code>element</code> element.
IGR_Get_Page_Element_Root	IGR_Get_Page_Element_Root fills {dest} with the root element of the page.
IGR_Get_Page_Element_Style	IGR_Get_Page_Element_Style returns the named style value from the page element.
IGR_Get_Page_Element_Styles	IGR_Get_Page_Element_Styles calls the <code>callback</code> with each style for the given <code>source</code> element.
IGR_Get_Page_Element_Text	IGR_Get_Page_Element_Text returns any text associated with the page element.
IGR_Get_Page_Elements	IGR_Get_Page_Elements fills the <code>dest</code> buffer with <code>dest_count</code> page elements from the parent object, starting from the <code>first</code> index. If parent is NULL, the root element is copied into <code>dest</code> .
IGR_Get_Page_Form_Element_Count	IGR_Get_Page_Form_Element_Count returns the number of form elements that are contained on the page.
IGR_Get_Page_Form_Element_Option_Str	IGR_Get_Page_Form_Element_Option_Str copies string information from an existing form element's option into the user supplied buffer.

IGR_Get_Page_Form_Element_Str	IGR_Get_Page_Form_Element_Str copies string information from an existing form element into the user supplied buffer.
IGR_Get_Page_Form_Elements	IGR_Get_Page_Form_Elements copies references of page form elements into the user supplied array. The caller can iterate over all the page form elements by incrementing the Index parameter.
IGR_Get_Page_Handle_File_Type	IGR_Get_Page_Handle_File_Type gets the type and the capabilities of an open page.
IGR_Get_Page_Hyperlink_Count	IGR_Get_Page_Hyperlink_Count returns the number of hyperlinks that are contained on the page.
IGR_Get_Page_Hyperlink_Str	IGR_Get_Page_Hyperlink_Str copies string information from an existing hyperlink into the user supplied buffer.
IGR_Get_Page_Hyperlinks	IGR_Get_Page_Hyperlinks copies references of page hyperlinks into the user supplied array. The caller can iterate over all the page hyperlinks by incrementing the Index parameter.
IGR_Get_Page_Image_Entry	IGR_Get_Page_Image_Entry enumerates the set of images on an open page.
IGR_Get_Page_Images_Enumerator	<code>IGR_Get_Page_Images_Enumerator</code> creates a new enumerator for page images. Unlike IGR_Get_Page_Image_Entry , it facilitates the creation of multiple enumerators, enabling the reprocessing of images within an open page. Remember to free a subfile enumerator by calling <code>IGR_Subfiles_Close</code> once it is no longer needed.
IGR_Get_Page_Pixels	The IGR_Get_Page_Pixels method provides low-level access to a rectangular portion of the page and provides a temporary buffer that you can use to read the pixel data in a specified format.
IGR_Get_Page_Text	IGR_Get_Page_Text extracts the text of a previously opened page of a document.
IGR_Get_Page_Word_Count	IGR_Get_Page_Word_Count returns the number of words of the given page.

IGR_Get_Page_Words	IGR_Get_Page_Words copies references of page words into the user supplied array. The caller can iterate over all the page words by incrementing the Index parameter.
IGR_Get_Stream_Type	IGR_Get_Stream_Type gets the type and the capabilities of a given stream object.
IGR_Get_Subfile_Entry	IGR_Get_Subfile_Entry enumerates the sub-documents contained in a previously opened compound document, such as message documents (MSG) or archive documents (ZIP).
IGR_Get_Subfile_Entry_Ex	IGR_Get_Subfile_Entry_Ex enumerates the sub-documents contained in a previously opened compound document, such as message documents (MSG) or archive documents (ZIP).
IGR_Get_Subfiles_Enumerator	<code>IGR_Get_Subfiles_Enumerator</code> creates a new enumerator for subfiles. Unlike IGR_Get_Subfile_Entry , it facilitates the creation of multiple enumerators, enabling the reprocessing of subfiles within an open document. Remember to free a subfile enumerator by calling IGR_Subfiles_Close once it is no longer needed.
IGR_Get_Text	IGR_Get_Text extracts the text of previously opened document.
IGR_Make_Output_Canvas	IGR_Make_Output_Canvas creates a new canvas that is used for rendering page content. The output data will be written to the file specified in Filename. To write to memory or stream, see IGR_Make_Output_Canvas_On.
IGR_Make_Output_Canvas_On	IGR_Make_Output_Canvas_On creates a new canvas that is used for rendering page content, the output data will be written to the stream specified. The Stream must be a caller created IGR_Writable_Stream derivative.
IGR_Make_Stream_From_File	IGR_Make_Stream_From_File creates a stream based on a file for use with the document stream functions.
IGR_Make_Stream_From_Functions	<code>IGR_Make_Stream_From_Functions</code> creates a new IGR stream based off the context and functions. This is functionally identical to constructing the IGR_Stream structure by hand; this function can be used by languages that do not have precise control over memory layout.

IGR_Make_Stream_From_Memory	IGR_Make_Stream_From_Memory creates a stream based on a memory buffer for use with the document stream functions.
IGR_Multiplex	The IGR_Multiplex function is used to supply extensible functionality to the Document Filters API. It is used as a general purpose extension mechanism to avoid disturbing the published Document Filters API.
IGR_Open_DDB	Opens a document from an OS based, device-dependant bitmap (Windows-Only).
IGR_Open_DIB	IGR_Open_DIB opens a document for content extraction or enumeration of sub-documents from an existing in-memory image.
IGR_Open_Ex	IGR_Open_Ex opens a document for text extraction or enumeration of sub-documents, and takes an IGR_OPEN_CALLBACK parameter that allows for interaction with the API user during API calls.
IGR_Open_File	IGR_Open_File opens a document for content extraction or enumeration of sub-documents.
IGR_Open_File_Ex	IGR_Open_File_Ex opens a document for content extraction or enumeration of sub-documents and controls the output format, including converting the source document to HTML.
IGR_Open_Page	IGR_Open_Page gives access to page specific content for documents opened using the IGR_FORMAT_IMAGE flag, including page words, images, and structured XML.
IGR_Open_Stream	IGR_Open_Stream opens a document from a stream object for content extraction or enumeration of contained sub-documents.
IGR_Open_Stream_Ex	IGR_Open_Stream_Ex opens a document from a stream object for content extraction or enumeration of contained sub-documents and controls the output format.
IGR_Redact_Page_Text	IGR_Redact_Page_Text removes the words and blacks out the location for the specified range from the page.
IGR_Render_Page	IGR_Render_Page draws the page content into the specified output canvas.

IGR_Render_Page_Ex	IGR_Render_Page_Ex draws the page content into the specified output canvas
IGR_Subfiles_Close	IGR_Subfiles_Close releases the resources associated with an enumeration generated by IGR_Get_Subfiles_Enumerator or a similar function. It is imperative to free enumerators before closing the owning document.
IGR_Subfiles_Next	IGR_Subfiles_Next iterates through the sub-documents within a subfile enumeration obtained from IGR_Get_Subfiles_Enumerator or a similar function.
IGR_Subfiles_Next_Ex	IGR_Subfiles_Next_Ex iterates through the sub-documents within a subfile enumeration obtained from IGR_Get_Subfiles_Enumerator or a similar function.
IGR_Subfiles_Reset	IGR_Subfiles_Reset resets an enumeration to its initial state, positioning it at the beginning. Subsequently, invoke IGR_Subfiles_Next to commence reprocessing the document. Notably, invoking IGR_Subfiles_Reset is unnecessary for the initial iteration.
IGR_Text_Compare_Close	Closes the text comparison enumerator and frees associated resources.
IGR_Text_Compare_Difference_Dispose	Disposes of a difference result returned by the text comparison enumerator.
IGR_Text_Compare_Documents	Compares two documents and returns an enumerator for iterating through the differences.
IGR_Text_Compare_Elements	Compares elements within pages of documents and returns an enumerator for iterating through the differences.
IGR_Text_Compare_Next	Retrieves the next difference from the text comparison enumerator.
IGR_Text_Compare_Pages	Compares two pages within documents and returns an enumerator for iterating through the differences.
IGR_Text_Compare_Reset	Resets the text comparison enumerator to its initial state.
UTF8_to_Widechar	UTF8_to_Widechar converts a UTF-8 string into a UCS2 string.

UTF8_to_Widechar_Ex	UTF8_to_Widechar converts a UTF-8 string into a UCS2 string.
Widechar_to_UTF8	Widechar_to_UTF8 converts a UCS2 string into UTF-8.
Widechar_to_UTF8_Ex	Widechar_to_UTF8 converts a UCS2 string into UTF-8

Structures

Error_Control_Block data type	Used to return error messages from the C API.
IGR_Annotation data type	The IGR_Annotation structure is used by the IGR_Get_Page_Annotations , IGR_Get_Page_Annotation_L , IGR_Get_Page_Annotation_Str methods to return information as annotation type, position, and dimensions for annotations given page.
IGR_Bookmark data type	The IGR_Bookmark structure is used by the IGR_Get_Bookmarks_Root method to return information about hierarchical navigation of a document.
IGR_CALLBACK data type	The IGR_CALLBACK data type represents the function signature of a callback from Document Filters.
IGR_Compare_Documents_Callback_Context data type	Context structure for the callback functions used in document comparison.
IGR_Compare_Documents_Difference data type	Structure representing a difference between compared documents.
IGR_Compare_Documents_Difference_Item data type	Structure representing a difference item in a compared document.
IGR_FPoint data type	Structure representing a size with signed floating-point width and height.
IGR_FRect data type	Structure representing a rectangle with floating-point coordinates. The coordinates are defined by the left, top, right, and bottom.
IGR_Hyperlink data type	The IGR_Hyperlink structure is used by the IGR_Get_Page_Hyperlinks method to return information such as position and destination for hyperlinks on a given page.
IGR_OPEN_CALLBACK data type	The IGR_OPEN_CALLBACK data type represents the function signature of a callback passed to IGR_Open_Ex .

IGR_Open_Callback_Action_Approve_External_Resource data type	The IGR_Open_Callback_Action_Approve_External_Resource structure is the payload argument for IGR_OPEN_CALLBACK when the action argument is IGR_OPEN_CALLBACK_ACTION_APPROVE_EXTERNAL_RESOURCE.
IGR_Open_Callback_Action_Get_Resource_Stream data type	The IGR_Open_Callback_Action_Get_Resource_Stream structure is the payload argument for IGR_OPEN_CALLBACK when the action argument is IGR_OPEN_CALLBACK_ACTION_GET_RESOURCE_STREAM.
IGR_Open_Callback_Action_Heartbeat data type	The IGR_Open_Callback_Action_Heartbeat structure is the payload argument for IGR_OPEN_CALLBACK when the action argument is IGR_OPEN_CALLBACK_ACTION_HEARTBEAT.
IGR_Open_Callback_Action_Localize data type	The IGR_Open_Callback_Action_Localize structure is the payload argument for IGR_OPEN_CALLBACK when the action argument is IGR_OPEN_CALLBACK_ACTION_LOCALIZE.
IGR_Open_Callback_Action_Log_Level data type	The IGR_Open_Callback_Action_Log_Level structure is the payload argument for IGR_OPEN_CALLBACK when the action argument is IGR_OPEN_CALLBACK_ACTION_LOG_LEVEL.
IGR_Open_Callback_Action_Log_Message data type	The IGR_Open_Callback_Action_Log_Message structure is the payload argument for IGR_OPEN_CALLBACK when the action argument is IGR_OPEN_CALLBACK_ACTION_LOG_MESSAGE.
IGR_Open_Callback_Action_OCR_Image data type	Structure that is payload argument for IGR_OPEN_CALLBACK when action argument is IGR_OPEN_CALLBACK_ACTION_OCR_IMAGE.
IGR_Open_Callback_Action_OCR_Image_Style_Info data type	Structure that is payload argument when calling AddText from IGR_OPEN_CALLBACK for IGR_OPEN_CALLBACK_ACTION_OCR_IMAGE.
IGR_Open_Callback_Action_Password data type	The IGR_Open_Callback_Action_Password structure is the payload argument for IGR_OPEN_CALLBACK when the action argument is IGR_OPEN_CALLBACK_ACTION_PASSWORD.
IGR_Open_DIB_Info data type	Structure representing information about an open Device Independent Bitmap (DIB). It includes details such as size, width, height, stride, pixel format, pixel data, palette information, and palette count.

IGR_Page_Element data type	IGR_Open_DIB_Info is used by IGR_Get_Page_Elements and related functions to enumerate the elements on a page.
IGR_Page_Form_Element data type	The IGR_Page_Form_Element structure is used by the IGR_Get_Page_Form_Elements method to return information such as position, dimensions, and text about the form items on a page.
IGR_Page_Pixels data type	The IGR_Page_Pixels structure is used by IGR_Get_Page_Pixels to retrieve the pixel data for a given page.
IGR_Page_Word data type	The IGR_Page_Word structure is used by the IGR_Get_Page_Words method to return information such as position, dimensions, and text about the words on a given page.
IGR_QuadPoint data type	Structure representing the points for a rectangle, in clockwise order.
IGR_Rect data type	The IGR_Rect structure is used to specify the position of a rectangle.
IGR_Render_Page_Form_Values data type	The IGR_Render_Page_Form_Values structure is used with IGR_Render_Page_Properties to specify new values for form items.
IGR_Render_Page_Properties data type	The IGR_Render_Page_Properties structure is used by the IGR_Render_Page_Ex method and provides additional information for rendering a page to a canvas.
IGR_Render_Page_Redactions data type	Structure representing a redaction area for rendering on a page. A redaction area is defined by its position (x, y) and size (width and height), along with the color to use for rendering and additional reserved data.
IGR_SRect data type	Structure representing a rectangle with signed integer coordinates. The coordinates are defined by the left, top, right, and bottom values.
IGR_Size data type	The IGR_Size structure is used to specify the size of a rectangle.
IGR_Stream data type	The IGR_Stream data type allows for the creation of custom streams.
IGR_Subfile_Info data type	Used to return subfile information from the IGR_Get_Subfile_Entry_Ex C API.
IGR_T_ACTION_GET_STREAM_PART data type	

	When a second or later part of a multi-part archive is required of processing an Extended Stream, a callback will request the structure to be populated with a new stream instance.
IGR_Text_Compare_Document_Source data type	Structure representing a source document for text comparison.
IGR_Text_Compare_Settings data type	Settings structure for text comparison operations.
IGR_Writable_Stream data type	Similar to the IGR_Stream , the IGR_Writable_Stream data type allows for the creation of custom output streams. It's important to note, all methods must be implemented, not just the Write.
Instance_Status_Block data type	The Instance_Status_Block data type is a control block returned by Init_Instance with information about the installed Document Filters engine.

Constants

IGR_RETURN_CODE	Indicates the success or failure of a call.
IGR_ANNOTATION_FLAGS_TYPE	Contains the annotation flags.
IGR_ANNOTATION_TYPE	Contains the annotation type.
IGR_BRUSH_STYLE	Contains the brush types available when calling IGR_Canvas_SetBrush or Canvas.SetBrush .
IGR_CANVAS_TYPE	Canvas Types specify the type of output to create when rendering a page. They are passed to the IGR_Make_Output_Canvas and DocumentFilters::MakeOutputCanvas functions.
IGR_CAPABILITY	Document Capabilities are a bitmask representing the operations that may be performed on a document.
IGR_CHAR_CODE	These character codes are output by Document Filters when using the

	IGR_Get_Text function and Extractor.GetText.
IGR_COMPARE_DOCUMENTS_COMPARE_TYPE	Contains the list of document comparison types.
IGR_COMPARE_DOCUMENTS_DIFFERENCE_SOURCE_TYPE	Contains the list of comparison difference source types.
IGR_COMPARE_DOCUMENTS_DIFFERENCE_TYPE	Contains the list of comparison difference types.
IGR_COMPARE_DOCUMENTS_FLAGS_TYPE	Contains the list of comparison flags.
IGR_FILETYPE_CATEGORY	The File Type Category provides extra classification of a file type. It is returned by calls to IGR_Get_Format_Attribute .
IGR_FONT_STYLE	These font styles are used with the IGR_Canvas_SetFont method.
IGR_GET_PAGE_PIXELS_FLAGS_TYPE	Contains the flags available when calling IGR_Page_Get_Pixels .
IGR_LOG_LEVEL_TYPE	IGR_LOG_LEVEL_TYPE is used by IGR_Open_Callback_Action_Log_Level when determining the log-level for a module.
IGR_OPEN_BITMAP_FLAGS_TYPE	IGR_OPEN_BITMAP_FLAGS_TYPE is used by IGR_Open_DIB_Info and IGR_Open_DIB when creating a Document Filters document from an existing in-memory image.
IGR_OPEN_BITMAP_PIXEL_TYPE	IGR_OPEN_BITMAP_PIXEL_TYPE is used by IGR_Open_DIB_Info and IGR_Open_DIB when creating a Document Filters document from an existing in-memory image; or IGR_Get_Page_Pixels when accessing pixels of a page.
IGR_OPEN_CALLBACK_ACTION	

	IGR_OPEN_CALLBACK_ACTION is used by IGR_OPEN_CALLBACK and IGR_Open_Ex when opening a document for text extraction or enumeration of sub-documents.
IGR_OPEN_FLAGS	Open Document Flags specify the type of content that is to be retrieved from the document. The flags are a bitwise combination of the values below.
IGR_OPEN_FROM	IGR_OPEN_FROM is used by IGR_Open_Ex when opening a document for text extraction or enumeration of sub-documents.
IGR_PAGE_ELEMENT_TYPE	Contains the list of form element types.
IGR_PAGE_FORM_ELEMENT_FLAG_TYPE	Contains the form element type flags.
IGR_PAGE_FORM_ELEMENT_GET_TYPE	Contains the options when requesting form element data.
IGR_PAGE_FORM_ELEMENT_TYPE	Contains the form element type flags.
IGR_PEN_STYLE	Contains the pen types available when calling <code>IGR_Canvas_SetPen</code> or <code>Canvas.SetPen</code>.
IGR_STRING_ID	Contains the list of strings that can be localized.
IGR_SUBFILE_INFO_FLAG_TYPE	Contains the subfile flag information.
IGR_TEXT_STYLE	Contains the text style types available when calling <code>IGR_Canvas_DrawText</code> or <code>Canvas.DrawText</code>.

3.2.2 Functions

Overview

Init_Instance	Init_Instance initializes the Document Filters engine and authenticates the license. Init_Instance must always be the first call made by any application to the Document Filters library.
Close_Instance	Close_Instance advises the Document Filters engine that the program is finished.
IGR_Calculate_MD5	IGR_Calculate_MD5 will calculate the MD5 hash of an input stream for unique document identification.
IGR_Calculate_SHA1	IGR_Calculate_SHA1 will calculate the SHA1 hash of an input stream for unique document identification.
IGR_Canvas_Annotate_JSON	IGR_Canvas_Annotate_JSON draws the given annotation onto the current page. The annotation is delivered as a string of JSON text. The schema for Annotations can be found in the Document Filters installation directory in “Annotation-Schema.json”.
IGR_Canvas_Arc	Draws an arc on the image along the perimeter of the ellipse bounded by the specified rectangle, with the current pen.
IGR_Canvas_Blank_Page	Inserts a blank page of the specified size into the Canvas. This can be useful when constructing documents, where you need to insert constructed pages created with the drawing or markup APIs.
IGR_Canvas_Bookmarks_Append	Append a single bookmark to the canvas. Bookmarks are added one at a time and the level parameter in the IGR_Bookmark determines the hierarchy. Bookmarks in a hierarchy must be added in order and a bookmark’s level cannot be more than one more level deep than the previous bookmark.
IGR_Canvas_Bookmarks_Clear	Remove existing bookmarks from the canvas.
IGR_Canvas_Chord	

	Draws a closed figure represented by the intersection of a line and an ellipse, with the current pen. The ellipse is bisected by a line that runs between X3,Y3 and X4,Y4.
IGR_Canvas_DrawImage	Renders the image specified by the ImageData parameter on the canvas at the given location given by the X and Y coordinates.
IGR_Canvas_DrawScaleImage	Renders the image specified by the ImageData parameter on the canvas at the given location given by the X and Y coordinates, scaling the output to width and height.
IGR_Canvas_Ellipse	Draws an ellipse defined by a bounding rectangle on the canvas, outlined with the current pen and filled with the current brush.
IGR_Canvas_LineTo	Draws a line on the canvas from the current pen position to the point specified by X and Y, and sets the pen position to (X, Y).
IGR_Canvas_MeasureText	Returns the width and height in pixels, of a string if rendered with the current font.
IGR_Canvas_MoveTo	Changes the current drawing position to the point (X,Y).
IGR_Canvas_Pie	Draws a pie-shaped section of the ellipse bounded by the rectangle (X1, Y1) and (X2, Y2) on the canvas.
IGR_Canvas_Rect	Draws a rectangle using the current brush and pen of the canvas to fill and draw the border.
IGR_Canvas_Reset	IGR_Canvas_Reset restores the canvas object back to the default set of options, including opacity, rotation, pens, and brushes.
IGR_Canvas_Rotation	IGR_Canvas_Rotation sets the rotation to be applied for subsequent drawing methods.
IGR_Canvas_RoundRect	Draws a rectangle with rounded corners, outlined with the current pen and filled with the current brush, on the canvas.
IGR_Canvas_SetBrush	Updates the current brush on the canvas with the given color and style; brushes are used when drawing rectangles, shapes and text.

IGR_Canvas_SetFont	Specifies the font to use when drawing text to the canvas. All subsequent calls to <code>TextOut</code> and <code>MeasureText</code> will use this font.
IGR_Canvas_SetOpacity	Set the opacity/transparency for future drawing routines.
IGR_Canvas_SetPen	Updates the canvas pen on the canvas with the specific color, width and style.
IGR_Canvas_TextOut	Writes a string on the canvas, starting at X and Y, and then updates the pen position to the end of the string. The text is written with the current font, and filled with the current brush.
IGR_Canvas_TextRect	Writes a string inside a clipping rectangle, using the current brush and font.
IGR_Close_Canvas	<code>IGR_Close_Canvas</code> releases the resources associated with the canvas handle. It must be called for every canvas opened by IGR_Make_Output_Canvas or IGR_Make_Output_Canvas_On , and must be called before closing the document with IGR_Close_File .
IGR_Close_File	<code>IGR_Close_File</code> releases the resources associated with the file handle. It must be called for every document opened by IGR_Open_File .
IGR_Close_Page	<code>IGR_Close_Page</code> releases the resources associated with the page handle. It must be called for every page opened by IGR_Open_Page , and must be called before closing the document with IGR_Close_File .
IGR_Convert_File	<code>IGR_Convert_File</code> converts the specified document into a plain text or HTML file, without the need to call IGR_Open_File and IGR_Get_Text .
IGR_Enum_Bookmarks	Enumerate all bookmarks, up-to <code>recurse_depth</code> , to an application-defined callback function.
IGR_Enum_Page_Elements	<code>IGR_Enum_Page_Elements</code> recursively walks the page elements of <code>source</code> calling the user-supplied <code>callback</code> .

IGR_Extend_Stream	IGR_Extend_Stream allows the C / C++ API user to create a custom stream that accepts callbacks from Document Filters. The callbacks allow the passing of Additional information about the stream.
IGR_Extract_Image	IGR_Extract_Image extracts an image to disk from a document opened with HTML or Image conversion in affect.
IGR_Extract_Image_Stream	IGR_Extract_Image_Stream extracts an image to a stream from a document, given the ID of the image.
IGR_Extract_Page_Image	<code>IGR_Extract_Page_Image</code> extracts the embedded image referenced by <code>id</code> to the file specified by <code>destination</code> .
IGR_Extract_Page_Image_Stream	<code>IGR_Extract_Page_Image_Stream</code> extracts the embedded image referenced by <code>id</code> as an IGRStream.
IGR_Extract_Subfile	IGR_Extract_Subfile extracts a sub-document to disk from a compound document, given the ID of the sub-document.
IGR_Extract_Subfile_Stream	IGR_Extract_Subfile_Stream extracts a sub-document to a stream from a compound document, given the ID of the sub-document.
IGR_Free_Page_Pixels	The IGR_Free_Page_Pixels method is used to dispose of page pixels extracted with IGR_Get_Page_Pixels .
IGR_Get_Bookmark_First_Child	IGR_Get_Bookmark_First_Child returns the first child bookmark of the provided bookmark, or NULL if the current item has no children.
IGR_Get_Bookmark_First_Child	IGR_Get_Bookmark_First_Child returns the first child bookmark of the provided bookmark, or NULL if the current item has no children.
IGR_Get_Bookmark_Next_Sibling	IGR_Get_Bookmark_Next_Sibling returns the peer/sibling of the provided bookmark, or NULL if the current item is the last in it's chain.
IGR_Get_Bookmark_Root	IGR_Get_Bookmark_Root provides access to the top-level root node of the bookmark hierarchy.
IGR_Get_File_Type	IGR_Get_File_Type gets the type and the capabilities of a given document.

IGR_Get_Format_Attribute	IGR_Get_Format_Attribute returns information about the supported file type.
IGR_Get_Handle_File_Type	IGR_Get_Handle_File_Type gets the type and the capabilities of an open document.
IGR_Get_Image_Entry	IGR_Get_Image_Entry enumerates the set of images, when HTML or Image conversion is in affect.
IGR_Get_Images_Enumerator	IGR_Get_Images_Enumerator returns a new enumerator for the set of images, when HTML conversion is in effect as set by the Open Document Flags, contained in a previously opened document
IGR_Get_Memory_Status	IGR_Get_Memory_Status is a Windows-only memory diagnostic.
IGR_Get_Option_Attribute	IGR_Get_Option_Attribute returns information about each published option value.
IGR_Get_Page_Annotation_Count	IGR_Get_Page_Annotation_Count returns the number of annotations that are contained on the page. For Office documents, comments are extracted and included in the annotation count. For comments spanning multiple pages, the OFFICE_COMMENT_PAGE open document option determines if the first, last or all pages containing the comments should return an annotation.
IGR_Get_Page_Annotation_Long	IGR_Get_Page_Annotation_Long copies an integer value from a given property of an existing annotation into the user supplied buffer.
IGR_Get_Page_Annotation_Str	IGR_Get_Page_Annotation_Str copies a string value from a given property of an existing annotation into the user supplied buffer.
IGR_Get_Page_Annotations	IGR_Get_Page_Annotations copies references of page annotations into the user supplied array. The caller can iterate over all the page annotations by incrementing the Index parameter. For Office documents, comments will be extracted and returned in the page annotations array. For comments spanning multiple pages, the OFFICE_COMMENT_PAGE

	open document option determines if the first, last or all pages containing the comments should return an annotation.
IGR_Get_Page_Attribute	IGR_Get_Page_Attribute returns style or properties of an open page; see <page> under Structured XML for a full list of options.
IGR_Get_Page_Count	IGR_Get_Page_Count returns the number of pages generated for an open document. This method only works on functions opened with IGR_FORMAT_IMAGE.
IGR_Get_Page_Dimensions	IGR_Get_Page_Dimensions returns the size of the given page in pixels.
IGR_Get_Page_Element_First_Child	IGR_Get_Page_Element_First_Child fills <code>dest</code> with the first child of the <code>parent</code> element.
IGR_Get_Page_Element_Next_Sibling	IGR_Get_Page_Element_Next_Sibling fills <code>dest</code> with the next sibling node of <code>element</code> element.
IGR_Get_Page_Element_Root	IGR_Get_Page_Element_Root fills {dest} with the root element of the page.
IGR_Get_Page_Element_Style	IGR_Get_Page_Element_Style returns the named style value from the page element.
IGR_Get_Page_Element_Styles	IGR_Get_Page_Element_Styles calls the <code>callback</code> with each style for the given <code>source</code> element.
IGR_Get_Page_Element_Text	IGR_Get_Page_Element_Text returns any text associated with the page element.
IGR_Get_Page_Elements	IGR_Get_Page_Elements fills the <code>dest</code> buffer with <code>dest_count</code> page elements from the parent object, starting from the <code>first</code> index. If parent is NULL, the root element is copied into <code>dest</code> .
IGR_Get_Page_Form_Element_Count	IGR_Get_Page_Form_Element_Count returns the number of form elements that are contained on the page.
IGR_Get_Page_Form_Element_Option_Str	IGR_Get_Page_Form_Element_Option_Str copies string information from an existing form element's option into the user supplied buffer.

IGR_Get_Page_Form_Element_Str	IGR_Get_Page_Form_Element_Str copies string information from an existing form element into the user supplied buffer.
IGR_Get_Page_Form_Elements	IGR_Get_Page_Form_Elements copies references of page form elements into the user supplied array. The caller can iterate over all the page form elements by incrementing the Index parameter.
IGR_Get_Page_Handle_File_Type	IGR_Get_Page_Handle_File_Type gets the type and the capabilities of an open page.
IGR_Get_Page_Hyperlink_Count	IGR_Get_Page_Hyperlink_Count returns the number of hyperlinks that are contained on the page.
IGR_Get_Page_Hyperlink_Str	IGR_Get_Page_Hyperlink_Str copies string information from an existing hyperlink into the user supplied buffer.
IGR_Get_Page_Hyperlinks	IGR_Get_Page_Hyperlinks copies references of page hyperlinks into the user supplied array. The caller can iterate over all the page hyperlinks by incrementing the Index parameter.
IGR_Get_Page_Image_Entry	IGR_Get_Page_Image_Entry enumerates the set of images on an open page.
IGR_Get_Page_Images_Enumerator	<code>IGR_Get_Page_Images_Enumerator</code> creates a new enumerator for page images. Unlike IGR_Get_Page_Image_Entry , it facilitates the creation of multiple enumerators, enabling the reprocessing of images within an open page. Remember to free a subfile enumerator by calling <code>IGR_Subfiles_Close</code> once it is no longer needed.
IGR_Get_Page_Pixels	The IGR_Get_Page_Pixels method provides low-level access to a rectangular portion of the page and provides a temporary buffer that you can use to read the pixel data in a specified format.
IGR_Get_Page_Text	IGR_Get_Page_Text extracts the text of a previously opened page of a document.
IGR_Get_Page_Word_Count	IGR_Get_Page_Word_Count returns the number of words of the given page.

IGR_Get_Page_Words	IGR_Get_Page_Words copies references of page words into the user supplied array. The caller can iterate over all the page words by incrementing the Index parameter.
IGR_Get_Stream_Type	IGR_Get_Stream_Type gets the type and the capabilities of a given stream object.
IGR_Get_Subfile_Entry	IGR_Get_Subfile_Entry enumerates the sub-documents contained in a previously opened compound document, such as message documents (MSG) or archive documents (ZIP).
IGR_Get_Subfile_Entry_Ex	IGR_Get_Subfile_Entry_Ex enumerates the sub-documents contained in a previously opened compound document, such as message documents (MSG) or archive documents (ZIP).
IGR_Get_Subfiles_Enumerator	<code>IGR_Get_Subfiles_Enumerator</code> creates a new enumerator for subfiles. Unlike IGR_Get_Subfile_Entry , it facilitates the creation of multiple enumerators, enabling the reprocessing of subfiles within an open document. Remember to free a subfile enumerator by calling IGR_Subfiles_Close once it is no longer needed.
IGR_Get_Text	IGR_Get_Text extracts the text of previously opened document.
IGR_Make_Output_Canvas	IGR_Make_Output_Canvas creates a new canvas that is used for rendering page content. The output data will be written to the file specified in Filename. To write to memory or stream, see IGR_Make_Output_Canvas_On.
IGR_Make_Output_Canvas_On	IGR_Make_Output_Canvas_On creates a new canvas that is used for rendering page content, the output data will be written to the stream specified. The Stream must be a caller created IGR_Writable_Stream derivative.
IGR_Make_Stream_From_File	IGR_Make_Stream_From_File creates a stream based on a file for use with the document stream functions.
IGR_Make_Stream_From_Functions	<code>IGR_Make_Stream_From_Functions</code> creates a new IGR stream based off the context and functions. This is functionally identical to constructing the IGR_Stream structure by hand; this function can be used by languages that do not have precise control over memory layout.

IGR_Make_Stream_From_Memory	IGR_Make_Stream_From_Memory creates a stream based on a memory buffer for use with the document stream functions.
IGR_Multiplex	The IGR_Multiplex function is used to supply extensible functionality to the Document Filters API. It is used as a general purpose extension mechanism to avoid disturbing the published Document Filters API.
IGR_Open_DDB	Opens a document from an OS based, device-dependant bitmap (Windows-Only).
IGR_Open_DIB	IGR_Open_DIB opens a document for content extraction or enumeration of sub-documents from an existing in-memory image.
IGR_Open_Ex	IGR_Open_Ex opens a document for text extraction or enumeration of sub-documents, and takes an IGR_OPEN_CALLBACK parameter that allows for interaction with the API user during API calls.
IGR_Open_File	IGR_Open_File opens a document for content extraction or enumeration of sub-documents.
IGR_Open_File_Ex	IGR_Open_File_Ex opens a document for content extraction or enumeration of sub-documents and controls the output format, including converting the source document to HTML.
IGR_Open_Page	IGR_Open_Page gives access to page specific content for documents opened using the IGR_FORMAT_IMAGE flag, including page words, images, and structured XML.
IGR_Open_Stream	IGR_Open_Stream opens a document from a stream object for content extraction or enumeration of contained sub-documents.
IGR_Open_Stream_Ex	IGR_Open_Stream_Ex opens a document from a stream object for content extraction or enumeration of contained sub-documents and controls the output format.
IGR_Redact_Page_Text	IGR_Redact_Page_Text removes the words and blacks out the location for the specified range from the page.
IGR_Render_Page	IGR_Render_Page draws the page content into the specified output canvas.

IGR_Render_Page_Ex	IGR_Render_Page_Ex draws the page content into the specified output canvas
IGR_Subfiles_Close	IGR_Subfiles_Close releases the resources associated with an enumeration generated by IGR_Get_Subfiles_Enumerator or a similar function. It is imperative to free enumerators before closing the owning document.
IGR_Subfiles_Next	IGR_Subfiles_Next iterates through the sub-documents within a subfile enumeration obtained from IGR_Get_Subfiles_Enumerator or a similar function.
IGR_Subfiles_Next_Ex	IGR_Subfiles_Next_Ex iterates through the sub-documents within a subfile enumeration obtained from IGR_Get_Subfiles_Enumerator or a similar function.
IGR_Subfiles_Reset	IGR_Subfiles_Reset resets an enumeration to its initial state, positioning it at the beginning. Subsequently, invoke IGR_Subfiles_Next to commence reprocessing the document. Notably, invoking IGR_Subfiles_Reset is unnecessary for the initial iteration.
IGR_Text_Compare_Close	Closes the text comparison enumerator and frees associated resources.
IGR_Text_Compare_Difference_Dispose	Disposes of a difference result returned by the text comparison enumerator.
IGR_Text_Compare_Documents	Compares two documents and returns an enumerator for iterating through the differences.
IGR_Text_Compare_Elements	Compares elements within pages of documents and returns an enumerator for iterating through the differences.
IGR_Text_Compare_Next	Retrieves the next difference from the text comparison enumerator.
IGR_Text_Compare_Pages	Compares two pages within documents and returns an enumerator for iterating through the differences.
IGR_Text_Compare_Reset	Resets the text comparison enumerator to its initial state.
UTF8_to_Widechar	UTF8_to_Widechar converts a UTF-8 string into a UCS2 string.

UTF8_to_Widechar_Ex	UTF8_to_Widechar converts a UTF-8 string into a UCS2 string.
Widechar_to_UTF8	Widechar_to_UTF8 converts a UCS2 string into UTF-8.
Widechar_to_UTF8_Ex	Widechar_to_UTF8 converts a UCS2 string into UTF-8

Init_Instance

Init_Instance initializes the Document Filters engine and authenticates the license. Init_Instance must always be the first call made by any application to the Document Filters library.

PROTOTYPE

C/C++

```
void Init_Instance(  
    IGR_LONG Reserved,  
    const char* InstallPath,  
    Instance_Status_Block* InstanceBlock,  
    IGR_SHORT* InstanceHandle,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Reserved: IGR_LONG

Reserved. Must be 0.

InstallPath: ANSI string

Path to installed executables and data files.

InstanceBlock: Pointer to Instance_Status_Block

Prior to the call: Contains your application License Code.

After the call: Returns your licensee information.

InstanceHandle: Pointer to IGR_SHORT

Returns an instance handle.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

None

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;  
IGR_SHORT DocumentFilters;  
Instance_Status_Block ISB;
```

```
strncpy(ISB.Licensee_ID1, "Your License Key Here", 40);  
Init_Instance(0, "Your Document Filters Executables Path Here", &ISB,  
             &DocumentFilters, &ISYSError);  
  
// Process documents...  
  
Close_Instance(&ISYSError);
```

ADDITIONAL INFORMATION

The application must call [Close_Instance](#) when finished.

SEE ALSO

- [Close_Instance](#)

Close_Instance

Close_Instance advises the Document Filters engine that the program is finished.

PROTOTYPE

C/C++

```
void Close_Instance(  
    Error_Control_Block* ISYSError)
```

PARAMETERS

ISYSError: Pointer to Error_Control_Block

ISYSError

RETURN VALUE

None

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;  
  
Close_Instance(&ISYSError);
```

IGR_Calculate_MD5

IGR_Calculate_MD5 will calculate the MD5 hash of an input stream for unique document identification.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Calculate_MD5(  
    IGR_Stream *Stream,  
    IGR_UCS2* Name,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Stream: IGR_Stream **

An open [IGR_Stream](#).

Name: Unicode string (UCS2)

A buffer to receive the null-terminated MD5 hash (as a Unicode string). Must be allocated by the caller and be able to hold at least 33 UCS2 characters.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
IGR_UCS2 strHexOut[ISYS_SZ_MD5_HEX_LEN];  
Error_Control_Block ISYSError;  
  
IGR_LONG RC = IGR_Calculate_MD5(pStream, strHexOut, &ISYSError);  
  
if (RC == IGR_OK)  
    // strHexOut will now contain a MD5 hash for the stream expressed as  
    // hexadecimal characters
```

IGR_Calculate_SHA1

IGR_Calculate_SHA1 will calculate the SHA1 hash of an input stream for unique document identification.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Calculate_SHA1(  
    IGR_Stream *Stream,  
    IGR_UCS2* Name,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Stream: IGR_Stream **

An open [IGR_Stream](#).

Name: Unicode string (UCS2)

A buffer to receive the null-terminated SHA1 hash (as a Unicode string). Must be allocated by the caller and be able to hold at least 41 UCS2 characters.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
IGR_UCS2 strHexOut[ISYS_SZ_SHA1_HEX_LEN];  
Error_Control_Block ISYSError;  
  
IGR_LONG RC = IGR_Calculate_SHA1(pStream, strHexOut, &ISYSError);  
  
if (RC == IGR_OK)  
    // strHexOut will now contain a SHA1 hash for the stream expressed as  
    // hexadecimal characters
```

ADDITIONAL INFORMATION

The stream must be created by one of the IGR stream creation functions or a custom IGR stream that has been opened.

IGR_Canvas_Annotate_JSON

IGR_Canvas_Annotate_JSON draws the given annotation onto the current page. The annotation is delivered as a string of JSON text. The schema for Annotations can be found in the Document Filters installation directory in “Annotation-Schema.json”.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Annotation_JSON(  
    HCANVAS Canvas,  
    const IGR_UCS2* Json,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Json: Unicode string (UCS2)

The json payload of the annotation to render to output of the canvas. The string must be NULL terminated.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note Annotations are supported for bitmap, PDF and HTML canvas, however are only end-user interactive in PDF output. For other output types, the annotation is flattened/rastered into the output

IGR_Canvas_Arc

Draws an arc on the image along the perimeter of the ellipse bounded by the specified rectangle, with the current pen.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Arc(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    IGR_LONG x2,  
    IGR_LONG y2,  
    IGR_LONG x3,  
    IGR_LONG y3,  
    IGR_LONG x4,  
    IGR_LONG y4,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas : HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

X : IGR_LONG

Left-most coordinate of the bounding box.

Y : IGR_LONG

Top-most coordinate of the bounding box.

X2 : IGR_LONG

Right-most coordinate of the bounding box.

Y2 : IGR_LONG

Bottom-most coordinate of the bounding box.

X3 : IGR_LONG

X coordinate of the start point.

Y3 : IGR_LONG

Y coordinate of the start point.

X4: IGR_LONG

X coordinate of the end point.

Y4: IGR_LONG

Y coordinate of the end point.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Use IGR_Canvas_Arc to draw a curved line with the current Pen onto the canvas. The arc follows the perimeter of the ellipse that is bounded by X, Y, X2 and Y2. The arc will follow the perimeter of the ellipse from the starting point to the ending point.

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetPen](#)

IGR_Canvas_Blank_Page

Inserts a blank page of the specified size into the Canvas. This can be useful when constructing documents, where you need to insert constructed pages created with the drawing or markup APIs.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Blank_Page(  
    HCANVAS canvas,  
    const IGR_UCS2* options,  
    IGR_LONG width,  
    IGR_LONG height,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Options: const IGR_UCS2*

Option string providing any extended processing options.

Width: IGR_LONG

Width of the page to insert, in pixels.

Height: IGR_LONG

Height of the page to insert, in pixels.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: LONG

Returns IGR_OK.

Failure: LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Canvas_Bookmarks_Append

Append a single bookmark to the canvas. Bookmarks are added one at a time and the level parameter in the [IGR_Bookmark](#) determines the hierarchy. Bookmarks in a hierarchy must be added in order and a bookmark's level cannot be more than one more level deep than the previous bookmark.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Bookmarks_Append (  
    HCANVAS canvas,  
    const struct IGR_Bookmark* bookmark,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Bookmark: Pointer to IGR_Bookmark

Pointer to bookmark data to add.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: LONG

Returns IGR_OK.

Failure: LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Canvas_Bookmarks_Clear

Remove existing bookmarks from the canvas.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Bookmarks_Clear (  
    HCANVAS canvas,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: LONG

Returns IGR_OK.

Failure: LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Canvas_Chord

Draws a closed figure represented by the intersection of a line and an ellipse, with the current pen. The ellipse is bisected by a line that runs between X3,Y3 and X4,Y4.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Chord(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    IGR_LONG x2,  
    IGR_LONG y2,  
    IGR_LONG x3,  
    IGR_LONG y3,  
    IGR_LONG x4,  
    IGR_LONG y4,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

X: IGR_LONG

Left-most coordinate of the bounding box.

Y: IGR_LONG

Top-most coordinate of the bounding box.

X2: IGR_LONG

Right-most coordinate of the bounding box.

Y2: IGR_LONG

Bottom-most coordinate of the bounding box.

X3: IGR_LONG

X coordinate of the start point.

Y3: IGR_LONG

Y coordinate of the start point.

X4: IGR_LONG

X coordinate of the end point.

Y4: IGR_LONG

Y coordinate of the end point.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: LONG

Returns IGR_OK.

Failure: LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Use IGR_Canvas_Chord to a shape that is an arc and a line that joins the endpoints of the arc. The chord consists of a portion of an ellipse that is bounded by X1, Y1, X2 and Y2. The ellipse is bisected by a line that runs between X3,Y3 and X4,Y4.

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetPen](#)
- [IGR_Canvas_SetBrush](#)

IGR_Canvas_DrawImage

Renders the image specified by the ImageData parameter on the canvas at the given location given by the X and Y coordinates.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_DrawImage(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    void* imagedata,  
    size_t imagesize,  
    const IGR_UCS2* mimetype,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

X: IGR_LONG

The X coordinate where the image is to be drawn.

Y: IGR_LONG

The Y coordinate where the image is to be drawn.

ImageData: Pointer to Bytes

Pointer to the image data loaded into memory.

ImageSize: IGR_LONG

The size of the buffer pointed to by ImageData.

MimeType: Pointer to Unicode String

A Unicode string indicating the MIME type of ImageData. Accepted values are:

image/jpg; image/jpeg; image/bmp; image/png

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

IGR_Canvas_DrawScaleImage

Renders the image specified by the ImageData parameter on the canvas at the given location given by the X and Y coordinates, scaling the output to width and height.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_DrawScaleImage(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    IGR_LONG width,  
    IGR_LONG height,  
    void* imagedata,  
    size_t imagesize,  
    const IGR_UCS2* mimetype,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

X: IGR_LONG

The X coordinate where the image is to be drawn.

Y: IGR_LONG

The Y coordinate where the image is to be drawn.

Width: IGR_LONG

The width that the image should be drawn.

Height: IGR_LONG

The height that the image should be drawn.

ImageData: Pointer to Bytes

Pointer to the image data loaded into memory.

ImageSize: IGR_LONG

The size of the buffer pointed to by ImageData.

MimeType: Pointer to Unicode String

A Unicode string indicating the MIME type of ImageData. Accepted values are:

```
image/jpg or image/jpeg  
image/bmp  
image/png
```

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

IGR_Canvas_Ellipse

Draws an ellipse defined by a bounding rectangle on the canvas, outlined with the current pen and filled with the current brush.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Ellipse(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    IGR_LONG x2,  
    IGR_LONG y2,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

X: IGR_LONG

Left-most coordinate of the bounding box.

Y: IGR_LONG

Top-most coordinate of the bounding box.

X2: IGR_LONG

Right-most coordinate of the bounding box.

Y2: IGR_LONG

Bottom-most coordinate of the bounding box.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetPen](#)
- [IGR_Canvas_SetBrush](#)

IGR_Canvas_LineTo

Draws a line on the canvas from the current pen position to the point specified by X and Y, and sets the pen position to (X, Y).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_LineTo(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

X: IGR_LONG

The X coordinate of the end point.

Y: IGR_LONG

The Y coordinate of the end point.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Use IGR_Canvas_LineTo to draw a line from the current pen position to the new coordinates. The pen position will be updated to the new coordinates.

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetPen](#)
- [IGR_Canvas_MoveTo](#)

IGR_Canvas_MeasureText

Returns the width and height in pixels, of a string if rendered with the current font.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_MeasureText(  
    HCANVAS canvas,  
    const IGR_UCS2* text,  
    IGR_LONG* width,  
    IGR_LONG* height,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Text: Pointer to Unicode String

Pointer to a NULL terminated Unicode string to be measured.

Width: Pointer to an IGR_LONG

Pointer to an integer that is populated with the calculated width.

Height: Pointer to an IGR_LONG

Pointer to an integer that is populated with the calculated height.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetFont](#)

IGR_Canvas_MoveTo

Changes the current drawing position to the point (X,Y).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_MoveTo(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

X: IGR_LONG

The X coordinate for the new pen position.

Y: IGR_LONG

The Y coordinate for the new pen position.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Use IGR_Canvas_MoveTo to move the current pen position, without drawing onto the canvas.

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_LineTo](#)

IGR_Canvas_Pie

Draws a pie-shaped section of the ellipse bounded by the rectangle (X1, Y1) and (X2, Y2) on the canvas.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Pie(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    IGR_LONG x2,  
    IGR_LONG y2,  
    IGR_LONG x3,  
    IGR_LONG y3,  
    IGR_LONG x4,  
    IGR_LONG y4,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas : HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

X : IGR_LONG

Left-most coordinate of the bounding box.

Y : IGR_LONG

Top-most coordinate of the bounding box.

X2 : IGR_LONG

Right-most coordinate of the bounding box.

Y2 : IGR_LONG

Bottom-most coordinate of the bounding box.

X3 : IGR_LONG

X coordinate of the start point.

Y3 : IGR_LONG

Y coordinate of the start point.

X4: IGR_LONG

X coordinate of the end point.

Y4: IGR_LONG

Y coordinate of the end point.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Use IGR_Canvas_Pie to draw a pie-shaped wedge on the image. The wedge is defined by the ellipse bounded by the rectangle determined by X, Y, X2 and Y2. The section drawn is determined by two lines radiating from the center of the ellipse through X3, Y3 and X4, Y4. The wedge is outlined using Pen, and filled using Brush.

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetPen](#)
- [IGR_Canvas_SetBrush](#)

IGR_Canvas_Rect

Draws a rectangle using the current brush and pen of the canvas to fill and draw the border.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Rect(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    IGR_LONG x2,  
    IGR_LONG y2,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas.

X: IGR_LONG

Left-most coordinate of the bounding box.

Y: IGR_LONG

Top-most coordinate of the bounding box.

X2: IGR_LONG

Right-most coordinate of the bounding box.

Y2: IGR_LONG

Bottom-most coordinate of the bounding box.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetPen](#)
- [IGR_Canvas_SetBrush](#)
- [IGR_Canvas_RoundRect](#)

IGR_Canvas_Reset

IGR_Canvas_Reset restores the canvas object back to the default set of options, including opacity, rotation, pens, and brushes.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Reset(  
    HCANVAS canvas,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

IGR_Canvas_Rotation

IGR_Canvas_Rotation sets the rotation to be applied for subsequent drawing methods.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_Rotation(  
    HCANVAS canvas,  
    IGR_LONG degrees,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Degrees: IGR_LONG

The rotation angle to be applied, in radians.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

IGR_Canvas_RoundRect

Draws a rectangle with rounded corners, outlined with the current pen and filled with the current brush, on the canvas.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_RoundRect(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    IGR_LONG x2,  
    IGR_LONG y2,  
    IGR_LONG radius,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas.

X: IGR_LONG

Left-most coordinate of the bounding box.

Y: IGR_LONG

Top-most coordinate of the bounding box.

X2: IGR_LONG

Right-most coordinate of the bounding box.

Y2: IGR_LONG

Bottom-most coordinate of the bounding box.

Radius: IGR_LONG

The radius to use for the rounded corner.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetPen](#)
- [IGR_Canvas_SetBrush](#)
- [IGR_Canvas_Rect](#)

IGR_Canvas_SetBrush

Updates the current brush on the canvas with the given color and style; brushes are used when drawing rectangles, shapes and text.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_SetBrush(  
    HCANVAS canvas,  
    IGR_LONG color,  
    IGR_LONG style,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Color: IGR_LONG

The color expressed as a 32-bit integer.

Style: IGR_LONG

The brush style used when drawing. See [Brush Styles](#).

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Color is expressed as a 32-bit integer, where the 4 bytes represent Alpha, Red, Green and Blue components.

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetFont](#)

IGR_Canvas_SetFont

Specifies the font to use when drawing text to the canvas. All subsequent calls to TextOut and MeasureText will use this font.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_SetFont(  
    HCANVAS canvas,  
    const IGR_UCS2* fontFamily,  
    IGR_LONG size,  
    IGR_LONG style,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

FontFamily: Pointer to Unicode String

The font family (or typeface) of the font.

Size: IGR_LONG

The size, in points, of the font.

Style: IGR_LONG

A bitmask of style information, can be zero or more of the following:

Character Value	Description
0x01	FONT_STYLE_BOLD
0x02	FONT_STYLE_ITALICS
0x04	FONT_STYLE_UNDERLINE
0x08	FONT_STYLE_STRIKEOUT
0x10	FONT_STYLE_SERIF
0x20	FONT_STYLE_MONO
0x40	FONT_STYLE_RTL

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_TextOut](#)
- [IGR_Canvas_TextRect](#)
- [IGR_Canvas_MeasureText](#)
- [Font Styles](#)

IGR_Canvas_SetOpacity

Set the opacity/transparency for future drawing routines.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_SetOpacity(  
    HCANVAS canvas,  
    BYTE opacity,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Opacity: BYTE

Indicates the opacity value between 0 and 255, where 255 is opaque.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

The opacity is expressed as a number between 0 and 255. 255 indicates that there is no transparency and 0 indicates full transparency.

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

IGR_Canvas_SetPen

Updates the canvas pen on the canvas with the specific color, width and style.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_SetPen(  
    HCANVAS canvas,  
    IGR_LONG color,  
    IGR_LONG width,  
    IGR_LONG style,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Color: IGR_LONG

The color expressed as a 32-bit integer.

Width: IGR_LONG

The width of the pen, expressed in points.

Style: IGR_LONG

The pen style used when drawing. See [Pen Styles](#).

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Color is expressed as a 32-bit integer, where the 4 bytes represent Alpha, Red, Green and Blue components.

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetBrush](#)

IGR_Canvas_TextOut

Writes a string on the canvas, starting at X and Y, and then updates the pen position to the end of the string. The text is written with the current font, and filled with the current brush.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_TextOut(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    const IGR_UCS2* text,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

X: IGR_LONG

Left-most coordinate of the bounding box.

Y: IGR_LONG

Top-most coordinate of the bounding box.

Text: Unicode string (UCS2)

The text to output to the canvas. The string must be NULL terminated.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SEE ALSO

- [IGR_Canvas_SetPen](#)
- [IGR_Canvas_SetBrush](#)

IGR_Canvas_TextRect

Writes a string inside a clipping rectangle, using the current brush and font.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Canvas_TextRect(  
    HCANVAS canvas,  
    IGR_LONG x,  
    IGR_LONG y,  
    IGR_LONG x2,  
    IGR_LONG y2,  
    const IGR_UCS2* text,  
    IGR_LONG flags,  
    Error_Control_Block* error);
```

PARAMETERS

Canvas: HCANVAS

Handle to a canvas.

X: IGR_LONG

Left-most coordinate of the bounding box.

Y: IGR_LONG

Top-most coordinate of the bounding box.

X2: IGR_LONG

Right-most coordinate of the bounding box.

Y2: IGR_LONG

Bottom-most coordinate of the bounding box.

Text: Pointer to Unicode String

The text to output to the canvas. The string must be NULL terminated.

Flags: IGR_LONG

Reserved for future use.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

IGR_Close_Canvas

IGR_Close_Canvas releases the resources associated with the canvas handle. It must be called for every canvas opened by [IGR_Make_Output_Canvas](#) or [IGR_Make_Output_Canvas_On](#), and must be called before closing the document with [IGR_Close_File](#).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Close_Canvas(
    HCANVAS CanvasHandle,
    Error_Control_Block* Error);
```

PARAMETERS

CanvasHandle: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#) or [IGR_Make_Output_Canvas_On](#).

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle, PageCount;
HPAGE PageHandle;

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &ISYSError);
if (RC == IGR_OK)
{
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSError) == IGR_OK)
    {
        for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)
        {
            if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSError) == IGR_OK)
            {
                HCANVAS CanvasHandle;
```

```
        if (IGR_Make_Output_Canvas(IGR_DEVICE_IMAGE_PNG, L"page.png",
                                   &CanvasHandle, &ISYSError) == IGR_OK)
        {
            IGR_Render_Page(PageHandle, CanvasHandle, &ISYSError);
            IGR_Close_Canvas(CanvasHandle, &ISYSError);
        }
        IGR_Close_Page(PageHandle, &ISYSError);
    }
}
IGR_Close_File(DocHandle, &ISYSError);
}
```

SEE ALSO

- [IGR_Make_Output_Canvas](#)
- [IGR_Make_Output_Canvas_On](#)

IGR_Close_File

DESCRIPTION

IGR_Close_File releases the resources associated with the file handle. It must be called for every document opened by [IGR_Open_File](#).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Close_File(  
    IGR_LONG DocHandle,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

DocHandle: IGR_LONG

Handle to a document, opened by a call to [IGR_Open_File](#), [IGR_Open_File_Ex](#), [IGR_Open_Stream](#) or [IGR_Open_Stream_Ex](#).

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;  
IGR_LONG Capabilities, DocType, DocHandle;  
  
IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_BODY_AND_META, &Capabilities, &DocType, &DocHandle,  
&ISYSError);  
if (RC == IGR_OK)  
{  
    // Extract document text or sub-documents...  
    IGR_Close_File(DocHandle, &ISYSError);  
}
```


ADDITIONAL INFORMATION

The stream must be created by one of the IGR stream creation functions or a custom IGR stream that has been opened.

IGR_Close_Page

IGR_Close_Page releases the resources associated with the page handle. It must be called for every page opened by [IGR_Open_Page](#), and must be called before closing the document with [IGR_Close_File](#).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Close_Page(
    HPAGE PageHandle,
    Error_Control_Block* Error);
```

PARAMETERS

PageHandle: HPAGE

Handle to a page opened by [IGR_Open_Page](#).

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;
IGR_LONG Caps, DocType, DocHandle, PageCount;
HPAGE PageHandle;

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Caps, &DocType, &DocHandle, &ISYSError);
if (RC == IGR_OK)
{
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSError) == IGR_OK)
    {
        for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)
        {
            if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSError) == IGR_OK)
            {
                // Process Page Element
                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
}
```

```
        }  
    }  
}  
IGR_Close_File(DocHandle, &ISYSError);  
}
```

IGR_Convert_File

IGR_Convert_File converts the specified document into a plain text or HTML file, without the need to call [IGR_Open_File](#) and [IGR_Get_Text](#).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Convert_File(  
    const IGR_UCS2* FileName,  
    IGR_LONG Flags,  
    const IGR_UCS2* Options,  
    const IGR_UCS2* OutputFilename  
    Error_Control_Block* ISYSError);
```

PARAMETERS

FileName: Unicode string (UCS2)

Path to the document to be converted.

Flags: IGR_LONG

Specifies processing options controlling the output. See [Open Document Flags](#) on page .

Note: IGR_Convert_File supports Text and Classic HTML modes only.

Options: Unicode string (UCS2)

Extended processing options used when converting the document to HTML. The [Open Document Options](#) are expressed as Name=Value with a semicolon delimiter.

OutputFilename: Unicode string (UCS2)

The filename where the output document is to be saved.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```
Error_Control_Block ISYSError;  
IGR_LONG RC;  
  
if (IGR_OK == IGR_Convert_File(_UCS2("TEST.DOC"), IGR_BODY_AND_META | IGR_FORMAT_HTML, _UCS2("IMAGEPATH=C:\\  
Temp"), _UCS2("C:\\Temp\\TEST.HTML"), &ISYSError)) { ... }
```

IGR_Enum_Bookmarks

Enumerate all bookmarks, up-to recurse_depth, to an application-defined callback function.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Enum_Bookmarks(  
    IGR_LONG docHandle,  
    const struct IGR_Bookmark* root,  
    IGR_LONG recurse_depth,  
    IGR_CALLBACK_BOOKMARK callback,  
    void* context,  
    Error_Control_Block* error  
)
```

PARAMETERS

docHandle: IGR_LONG

Handle to a HD document.

root: Pointer to IGR_Bookmark

Optional bookmark value to enumerate, or NULL to enumerate from root.

recurse_depth: IGR_LONG

Maximum depth to scan.

callback: IGR_CALLBACK_BOOKMARK

A pointer to an application-defined callback function.

context: Pointer to void

An application-defined value to be passed to the callback function.

error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Enum_Page_Elements

IGR_Enum_Page_Elements recursively walks the page elements of `source` calling the user-supplied `callback`.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Enum_Page_Elements(  
    IGR_HPAGE page,  
    const struct IGR_Page_Element* source,  
    IGR_ULONG flags,  
    IGR_ULONG max_depth,  
    IGR_PAGE_ELEMENT_CALLBACK callback,  
    void* context,  
    Error_Control_Block* error  
)
```

PARAMETERS

`page`: IGR_HPAGE

Page element containing the page elements to extract.

`source`: Pointer to IGR_Page_Element

The element of the element where the styles should be extracted.

`flags`: IGR_ULONG

Reserved for future use.

`max_depth`: IGR_ULONG

The maximum depth of page elements to walk.

`callback`: IGR_PAGE_ELEMENT_CALLBACK

User supplied callback where the element information is to be sent.

`context`: Pointer to void

User supplied callback context to pass when calling {callback}.

`error`: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

IGR_Extend_Stream

IGR_Extend_Stream allows the C / C++ API user to create a custom stream that accepts callbacks from Document Filters. The callbacks allow the passing of Additional information about the stream.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Extend_Stream(  
    IGR_Stream* Stream,  
    IGR_CALLBACK Callback,  
    void* Context,  
    IGR_Stream** ExtStream,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Stream: Pointer to IGR_Stream

A valid [IGR_Stream](#) instance.

Callback: Pointer to callback function

Pointer to the API user's function to handle callback generated while processing the stream.

Context: void Pointer

API user-supplied context information.

ExtStream: Pointer to IGR_Stream pointer

The extended stream which should be used instead of the original stream. See notes below.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```

IGR_LONG HandleCallback(int actionID, void* actionData, void* context)
{
    MyFileInfoStruct* pFileInfo;
    IGR_T_ACTION_GET_STREAM_PART* pStreamPartInfo;

    // Process the action...
    pFileInfo = (MyFileInfoStruct*)context;
    if (actionID == IGR_ACTION_GET_STREAM_PART)
    {
        pStreamPartInfo = (IGR_T_ACTION_GET_STREAM_PART*) actionData;
        // Open a new stream based on the stream part info...
        // The new stream does not need to be an extended stream.
    }
    return 0; // OK
}

void ProcessFile()
{
    Error_Control_Block ISYSError;
    IGR_Stream *pStream;
    IGR_Stream* pExtendedStream;
    MyFileInfoStruct fileInfo;

    SetFileInfoName(fileInfo, "TEST.RAR");

    if (IGR_Make_Stream_From_File(_UCS2(fileInfo.name), 0, &pStream,
        &ISYSError) == IGR_OK)
    {
        if (IGR_Extend_Stream(pStream, &HandleCallback, &fileInfo, &pExtendedStream,
            &ISYSError))
        {
            // Process the file using pExtendedStream only...
            IGR_Open_Stream(pExtendedStream, ...);
            // ...
            pExtendedStream->Close(pExtendedStream);
        }
    }
}

```

ADDITIONAL INFORMATION

To create and use an Extended Stream, complete the following steps.

- Get or create an instance of an ordinary stream.
- Call `IGR_Extend_Stream`.
- Use the returned Extended Stream instead of the original stream.

Once you have successfully created an Extended Stream, do not use the original stream pointer any further, and do not close or release it. When you are finished with the Extended Stream, call `Close` on the Extended Stream directly and the original stream closes automatically.

SEE ALSO

- [Multi-Part Archives](#)
- [Custom Streams](#)
- [IGR_CALLBACK Data Type](#)
- [IGR_Make_Stream_From_File](#)
- [IGR_Stream](#)

IGR_Extract_Image

IGR_Extract_Image extracts an image to disk from a document opened with HTML or Image conversion in affect.

The image ID is obtained previously by [IGR_Get_Image_Entry](#) from the document.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Extract_Image(
    IGR_LONG DocHandle,
    const IGR_UCS2* ID,
    const IGR_UCS2* Destination,
    Error_Control_Block* ISYSError);
```

PARAMETERS

DocHandle: IGR_LONG

Is a handle to a file, opened by a call to [IGR_Open_File](#).

ID: Unicode string (UCS2)

Unique ID of the sub-document to be extracted, obtained by a call to [IGR_Get_Image_Entry](#).

Destination: Unicode string (UCS2)

Path to a file on disk, where the binary the sub-document will be written.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle;
IGR_UCS2 ID[4096], Name[1024];
```

```
INT64 FileDate, FileSize;

IGR_LONG RC = IGR_Open_File_Ex(_UCS2("TEST.DOC"), IGR_BODY_AND_META | IGR_FORMAT_HTML, _UCS2(""),
&Capabilities, &DocType, &DocHandle, &ISYSSError);
if (RC == IGR_OK)
{
    // Extract document HTML via IGR_Get_Text first, then...
    while (true)
    {
        rc = IGR_Get_Image_Entry(DocHandle, ID, Name, &FileDate, &FileSize, &ISYSSError);
        if (rc != IGR_OK)
        {
            if (rc != IGR_NO_MORE)
                // ReportError(rc);
            break;
        }
        rc = IGR_Extract_Image(DocHandle, ID, ID, &ISYSSError);
        if (rc != IGR_OK)
            // ReportError(rc);
    }
    IGR_Close_File(DocHandle, &ISYSSError);
}
```

SEE ALSO

- [IGR_Get_Image_Entry](#)

IGR_Extract_Image_Stream

IGR_Extract_Image_Stream extracts an image to a stream from a document, given the ID of the image.

The image ID is obtained previously by [IGR_Get_Image_Entry](#) from the document, after being opened by [IGR_Open_File](#) or [IGR_Open_Stream](#) with the HTML conversion Open Document Flags set.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Extract_Image_Stream(  
    IGR_LONG DocHandle,  
    const IGR_UCS2* ID,  
    IGR_Stream **Stream,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

DocHandle: IGR_LONG

Handle to a document, opened by a call to [IGR_Open_File](#), [IGR_Open_File_Ex](#), [IGR_Open_Stream](#) or [IGR_Open_Stream_Ex](#).

ID: Unicode string (UCS2)

Unique ID of the sub-document to be extracted, obtained by a call to [IGR_Get_Image_Entry](#).

Stream: IGR_Stream **

A pointer to a system allocated memory stream will be returned.

It is the caller's responsibility to free the stream object by calling `Stream->Close()`.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```
Error_Control_Block ISYSError;
IGR_Stream *Stream;
IGR_UCS2 ID[4096], Name[1024];
INT64 FileDate, FileSize;

while (true)
{
    IGR_LONG rc = IGR_Get_Image_Entry(DocHandle, ID, Name, &FileDate, &FileSize, &ISYSError);
    if (rc != IGR_OK)
    {
        if (rc != IGR_NO_MORE)
            // ReportError(rc);
        break;
    }
    rc = IGR_Extract_Image_Stream(DocHandle, ID, &Stream, &ISYSError);
    if (rc != IGR_OK)
        // ReportError(rc);
    else
        // DoSomethingWithTheStream(Stream);
    Stream->close();
}
```

SEE ALSO

- [IGR_Get_Image_Entry](#)

IGR_Extract_Page_Image

`IGR_Extract_Page_Image` extracts the embedded image referenced by `id` to the file specified by `destination`.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Extract_Page_Image(  
    HPAGE pageHandle,  
    const IGR_UCS2* id,  
    const IGR_UCS2* destination,  
    Error_Control_Block* error  
)
```

PARAMETERS

`pageHandle`: HPAGE

Handle to a page, opened by a call to [IGR_Open_Page](#).

`id`: Pointer to IGR_UCS2

Unique ID of the sub-document to be extracted, obtained by a call to [IGR_Get_Image_Entry](#).

`destination`: Pointer to IGR_UCS2

Path to a file on disk, where the binary the sub-document will be written.

`error`: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

`Success`: IGR_LONG

Returns IGR_OK.

`Failure`: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

IGR_Extract_Page_Image_Stream

`IGR_Extract_Page_Image_Stream` extracts the embedded image referenced by `id` as an IGRStream.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Extract_Page_Image_Stream(  
    HPAGE pageHandle,  
    const IGR_UCS2* id,  
    IGR_Stream** stream,  
    Error_Control_Block* error  
)
```

PARAMETERS

`pageHandle`: HPAGE

Handle to a page, opened by a call to [IGR_Open_Page](#).

`id`: Pointer to IGR_UCS2

Unique ID of the sub-document to be extracted, obtained by a call to [IGR_Get_Image_Entry](#).

`stream`: Pointer to IGR_Stream

A pointer to a system allocated memory stream will be returned.

It is the caller's responsibility to free the stream object by calling `Stream->Close()`.

`error`: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

`Success`: IGR_LONG

Returns IGR_OK.

`Failure`: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

IGR_Extract_Subfile

IGR_Extract_Subfile extracts a sub-document to disk from a compound document, given the ID of the sub-document.

The sub-document ID is obtained previously by [IGR_Get_Subfile_Entry](#) from the compound document, after being opened by [IGR_Open_File](#).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Extract_Subfile(  
    IGR_LONG DocHandle,  
    const IGR_UCS2* ID,  
    const IGR_UCS2* Destination,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

DocHandle: IGR_LONG

Is a handle to a file, opened by a call to [IGR_Open_File](#).

ID: Unicode string (UCS2)

Unique ID of the sub-document to be extracted, obtained by a call to [IGR_Get_Subfile_Entry](#).

Destination: Unicode string (UCS2)

Path to a file on disk, where the binary the sub-document will be written.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;  
IGR_UCS2 ID[4096], Name[1024];  
INT64 FileDate, FileSize;  
  
while (true)  
{  
    IGR_LONG rc = IGR_Get_Subfile_Entry(DocHandle, ID, Name, &FileDate, &FileSize,  
        &ISYSError);  
    if (rc != IGR_OK)  
    {  
        if (rc != IGR_NO_MORE)  
            // ReportError(rc);  
        break;  
    }  
    rc = IGR_Extract_Subfile(DocHandle, ID, _UCS2("TEMP.DAT"), &ISYSError);  
    if (rc != IGR_OK)  
        // ReportError(rc);  
    else  
        // DoSomethingWithTheFile("TEMP.DAT", ID, Name);  
}
```

SEE ALSO

- [IGR_Get_Subfile_Entry](#)

IGR_Extract_Subfile_Stream

IGR_Extract_Subfile_Stream extracts a sub-document to a stream from a compound document, given the ID of the sub-document.

The sub-document ID is obtained previously by [IGR_Get_Subfile_Entry](#) from the compound document, after being opened by [IGR_Open_File](#) or [IGR_Open_Stream](#).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Extract_Subfile_Stream(  
    IGR_LONG DocHandle,  
    const IGR_UCS2* ID,  
    IGR_Stream **Stream,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

DocHandle: IGR_LONG

Handle to a document, opened by a call to [IGR_Open_File](#), [IGR_Open_File_Ex](#), [IGR_Open_Stream](#) or [IGR_Open_Stream_Ex](#).

ID: Unicode string (UCS2)

Unique ID of the sub-document to be extracted, obtained by a call to [IGR_Get_Subfile_Entry](#).

Stream: Pointer to an IGR_Stream pointer

A pointer to a system allocated memory stream will be returned.

It is the caller's responsibility to free the stream object by calling

Stream->Close()

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E](#) error codes.

SAMPLE CODE**C/C++**

```
Error_Control_Block ISYSError;
IGR_Stream *Stream;
IGR_UCS2 ID[4096], Name[1024];
INT64 FileDate, FileSize;

while (true)
{
    IGR_LONG rc = IGR_Get_Subfile_Entry(DocHandle, ID, Name, &FileDate, &FileSize,
                                        &ISYSError);
    if (rc != IGR_OK)
    {
        if (rc != IGR_NO_MORE)
            // ReportError(rc);
        break;
    }
    rc = IGR_Extract_Subfile_Stream(DocHandle, ID, &Stream, &ISYSError);
    if (rc != IGR_OK)
        // ReportError(rc);
    else
        // DoSomethingWithTheStream(Stream);
    Stream->close();
}
```

SEE ALSO

- [IGR_Get_Subfile_Entry](#)
- [IGR_Stream](#)

IGR_Free_Page_Pixels

The IGR_Free_Page_Pixels method is used to dispose of page pixels extracted with [IGR_Get_Page_Pixels](#).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Free_Page_Pixels(  
    struct IGR_Page_Pixels* pixel_data,  
    Error_Control_Block* error);
```

PARAMETERS

`pixel_data`: Pointer to IGR_Page_Pixels

Pointer to an IGR_Page_Pixels populated by a called to [IGR_Get_Page_Pixels](#).

`ISYSError`: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

`Success`: IGR_LONG

Returns IGR_OK.

`Failure`: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

- [IGR_Get_Page_Pixels](#)
- [IGR_Page_Pixels](#)

IGR_Get_Bookmark_First_Child

IGR_Get_Bookmark_First_Child returns the first child bookmark of the provided bookmark, or NULL if the current item has no children.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Bookmark_First_Child (  
    IGR_LONG DocHandle,  
    const IGR_Bookmark *Source,  
    IGR_Bookmark *Dest,  
    Error_Control_Block* Error);
```

PARAMETERS

DocHandle: IGR_LONG

Handle to a document, opened by a call to [IGR_Open_File](#), [IGR_Open_File_Ex](#), [IGR_Open_Stream](#) or [IGR_Open_Stream_Ex](#).

Source: Pointer to IGR_Bookmark

Pointer to an IGR_Bookmark returned from a previous call to [IGR_Get_Bookmark_Root](#), [IGR_Get_Bookmark_First_Child](#), or [IGR_Get_Bookmark_Next_Sibling](#).

Dest: Pointer to IGR_Bookmark

Pointer to a user-supplied IGR_Bookmark that will be populated enabling supplementary calls to bookmark methods.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++


```
Error_Control_Block Err;
IGR_LONG Capabilities, DocType, DocHandle, PageCount;
HPAGE PageHandle;

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &Err);
if (RC == IGR_OK)
{
    IGR_Bookmark Bookmarks, Child;
    if (IGR_Get_Bookmark_Root(DocHandle, &Bookmarks, &Err) == IGR_OK
        && IGR_Get_Bookmark_First_Child(DocHandle, &Bookmarks, &Child, &Err) == IGR_OK)
    {
        do
        {
            // ...
        } while (IGR_Get_Bookmark_Next_Sibling(DocHandle, &Child, &Child, &Err) == IGR_OK);
    }
    IGR_Close_File(DocHandle, &ISYSError);
}
```

SEE ALSO

- [IGR_Bookmark](#)
- [IGR_Get_Bookmark_Next_Sibling](#)

IGR_Get_Bookmark_Next_Sibling

IGR_Get_Bookmark_Next_Sibling returns the peer/sibling of the provided bookmark, or NULL if the current item is the last in it's chain.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Bookmark_Next_Sibling(  
    IGR_LONG DocHandle,  
    const IGR_Bookmark *Source,  
    IGR_Bookmark *Dest,  
    Error_Control_Block* Error);
```

PARAMETERS

DocHandle: IGR_LONG

Handle to a document, opened by a call to [IGR_Open_File](#), [IGR_Open_File_Ex](#), [IGR_Open_Stream](#) or [IGR_Open_Stream_Ex](#).

Source: Pointer to IGR_Bookmark

Pointer to an IGR_Bookmark returned from a previous call to [IGR_Get_Bookmark_Root](#), [IGR_Get_Bookmark_First_Child](#), or [IGR_Get_Bookmark_Next_Sibling](#).

Dest: Pointer to IGR_Bookmark

Pointer to a user-supplied IGR_Bookmark that will be populated enabling supplementary calls to bookmark methods.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block Err;
IGR_LONG Capabilities, DocType, DocHandle, PageCount;
HPAGE PageHandle;

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &Err);
if (RC == IGR_OK)
{
    IGR_Bookmark Bookmarks, Child;
    if (IGR_Get_Bookmark_Root(DocHandle, &Bookmarks, &Err) == IGR_OK
        && IGR_Get_Bookmark_First_Child(DocHandle, &Bookmarks, &Child, &Err) == IGR_OK)
    {
        do
        {
            // ...
        } while (IGR_Get_Bookmark_Next_Sibling(DocHandle, &Child, &Child, &Err) == IGR_OK);
    }
    IGR_Close_File(DocHandle, &ISYSError);
}
```

SEE ALSO

- [IGR_Bookmark](#)
- [IGR_Get_Bookmark_First_Child](#)

IGR_Get_Bookmark_Root

IGR_Get_Bookmark_Root provides access to the top-level root node of the bookmark hierarchy.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Bookmark_Root(  
    IGR_LONG DocHandle,  
    IGR_Bookmark *Dest,  
    Error_Control_Block* Error);
```

PARAMETERS

DocHandle: IGR_LONG

Handle to a document, opened by a call to [IGR_Open_File](#), [IGR_Open_File_Ex](#), [IGR_Open_Stream](#) or [IGR_Open_Stream_Ex](#).

Dest: Pointer to IGR_Bookmark

Pointer to a user-supplied IGR_Bookmark that will be populated enabling supplementary calls to bookmark methods.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block Err;  
IGR_LONG Capabilities, DocType, DocHandle, PageCount;  
HPAGE PageHandle;  
  
IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &Err);  
if (RC == IGR_OK)  
{  
    IGR_Bookmark Bookmarks, Child;  
    if (IGR_Get_Bookmark_Root(DocHandle, &Bookmarks, &Err) == IGR_OK
```

```
    && IGR_Get_Bookmark_First_Child(DocHandle, &Bookmarks, &Child, &Err) == IGR_OK)
{
    do
    {
        // ...
    } while (IGR_Get_Bookmark_Next_Sibling(DocHandle, &Child, &Child, &Err)==IGR_OK);
}
IGR_Close_File(DocHandle, &ISYSError);
}
```

SEE ALSO

- [IGR_Bookmark](#)
- [IGR_Get_Bookmark_First_Child](#)
- [IGR_Get_Bookmark_Next_Sibling](#)

IGR_Get_File_Type

IGR_Get_File_Type gets the type and the capabilities of a given document.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_File_Type(  
    const IGR_UCS2* FileName,  
    IGR_LONG* Capabilities,  
    IGR_LONG* DocType,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

FileName: Unicode string (UCS2)

Path to the document to be opened.

Capabilities: Pointer to IGR_LONG

Returns the [Document Capabilities](#) as a bit field.

DocType: Pointer to IGR_LONG

Returns the [Document Format Code](#) of the document.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;  
IGR_LONG Capabilities, DocType;  
  
IGR_LONG rc = IGR_Get_File_Type(_UCS2("TEST.TXT"), &Capabilities, &DocType, &ISYSError);
```

ADDITIONAL INFORMATION

If the document has the IGR_FILE_SUPPORTS_TEXT capability, text may be directly extracted from the document by calling [IGR_Get_Text](#) (e.g. a Word document).

If the document has the IGR_FILE_SUPPORTS_SUBFILES capability, then it is a container for other documents and it is valid to enumerate and/or extract its sub-documents.

It is valid for a document to have both capabilities (for example email message documents have their own text and also can have attached documents).

Document Filters also has the ability to identify certain document formats, without being able to extract content. In this situation, the capabilities will be returned as 0. See [Document Format Codes](#) for a list of these formats on page .

The compound documents can include other compound documents, for example an MSG with a ZIP attachment, which contains ZIPs and MSGs. The calling application can navigate as far down as needed.

IGR_Get_Format_Attribute

IGR_Get_Format_Attribute returns information about the supported file type.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Format_Attribute(  
    int formatid,  
    int what,  
    char* buffer,  
    Error_Control_Block* error);;
```

PARAMETERS

formatid: Integer

The format id, as returned by [IGR_Get_File_Type](#).

what: Integer

Indicates the information to return, can be one of:

Value	Description
0	copy the long form of the format name
1	copy the short form of the format name
2	copy the config file form of the format name (as it would appear in an Enterprise Search config file)
3	copy the class of the format
4	indicate if the format is a legacy format
5	default MIME type for file format
6	string representation of the IGR_FILETYPE_CATEGORY

buffer: Pointer to an Ansi string

The buffer must be at least 255 bytes, it will be populated based on the value of 'what'.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

- [IGR_FILETYPE_CATEGORY](#)

IGR_Get_Handle_File_Type

IGR_Get_Handle_File_Type gets the type and the capabilities of an open document.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Handle_File_Type(  
    IGR_HDOC handle,  
    IGR_LONG* capabilities,  
    IGR_LONG* filetype,  
    Error_Control_Block* error  
)
```

PARAMETERS

handle: IGR_HDOC

Handle to an open document.

capabilities: Pointer to IGR_LONG

Returns the Document Capabilities as a bit field.

filetype: Pointer to IGR_LONG

Returns the Document Format Code of the document.

error: Pointer to Error_Control_Block

Contains any error text.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

- [IGR_Get_Page_Handle_File_Type](#)
- [Document Format Codes](#)
- [Document Capability Codes](#)

IGR_Get_Image_Entry

IGR_Get_Image_Entry enumerates the set of images, when HTML or Image conversion is in affect.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Image_Entry(  
    IGR_LONG DocHandle,  
    IGR_UCS2* ID,  
    IGR_UCS2* Name,  
    LONGLONG* FileDate,  
    LONGLONG* FileSize,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

DocHandle: IGR_LONG

Is a handle to a file, opened by a call to [IGR_Open_File](#).

ID: Unicode string (UCS2)

Application allocated memory block of 8192 bytes that will be filled with up to 4096 Unicode characters that specify the unique ID of the next sub-document.

Name: Unicode string (UCS2)

Application allocated memory block of 2048 bytes that will be filled with up to 1024 Unicode characters that specify the name of the sub-document.

FileDate: Pointer to INT64

Returns the date and time of the sub-document in [FILETIME](#) format.

FileSize: Pointer to INT64

Returns the size in bytes of the sub-document.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Success and the end of the document was reached: IGR_LONG

Returns IGR_NO_MORE.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle;
IGR_UCS2 ID[4096], Name[1024];
INT64 FileDate, FileSize;

IGR_LONG RC = IGR_Open_File_Ex(_UCS2("TEST.DOC"), IGR_BODY_AND_META | IGR_FORMAT_HTML, _UCS2(""),
&Capabilities, &DocType, &DocHandle, &ISYSError);
if (RC == IGR_OK)
{
    // Extract document HTML via IGR_Get_Text first, then...
    while (true)
    {
        rc = IGR_Get_Image_Entry(DocHandle, ID, Name, &FileDate, &FileSize, &ISYSError);
        if (rc != IGR_OK)
        {
            if (rc != IGR_NO_MORE)
                // ReportError(rc);
            break;
        }
        rc = IGR_Extract_Image(DocHandle, ID, ID, &ISYSError);
        if (rc != IGR_OK)
            // ReportError(rc);
    }
    IGR_Close_File(DocHandle, &ISYSError);
}
```

ADDITIONAL INFORMATION

The previously opened document must have the IGR_FILE_SUPPORTS_HDHTML capability.

After a successful call to [IGR_Open_File](#) / [Stream](#), each call to [IGR_Get_Image_Entry](#) will retrieve information about the images contained in the document, referenced by DocHandle. To traverse all the images, the application will need to call this method in a loop until IGR_NO_MORE is returned.

Note that the null-terminating character will also be copied to the ID and Name parameters. If the function succeeds, the ID is guaranteed not to be empty and will be unique among all traversed images retrieved from the document. The returned ID can be used in a call to [IGR_Extract_Image](#) to save the binary content of the image to disk.

IGR_Get_Images_Enumerator

IGR_Get_Images_Enumerator returns a new enumerator for the set of images, when HTML conversion is in effect as set by the Open Document Flags, contained in a previously opened document

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Images_Enumerator(  
    IGR_LONG docHandle,  
    HSUBFILES* enumerator,  
    Error_Control_Block* error  
)
```

PARAMETERS

docHandle: IGR_LONG

Handle to a document, opened by a call to IGR_Open_File, IGR_Open_Stream, IGR_Open_File_Ex or IGR_Open_Stream_Ex.

enumerator: Pointer to HSUBFILES

Populated with handle to new enumerator, which can be passed to the enumeration functions.

error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Get_Memory_Status

IGR_Get_Memory_Status is a Windows-only memory diagnostic.

PROTOTYPE

C/C++

```
size_t IGR_Get_Memory_Status(  
    char* buffer,  
    size_t len  
)
```

PARAMETERS

buffer: Pointer to char

Pointer to a buffer that will be populated with process memory details.

len: size_t

The size of the buffer.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Get_Option_Attribute

IGR_Get_Option_Attribute returns information about each published option value.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Option_Attribute(  
    int option_index,  
    int what,  
    char* buffer,  
    Error_Control_Block* error);;
```

PARAMETERS

option_index: Integer

0-based index of the option to fetch.

what: Integer

Indicates the information to return, can be one of:

Value	Description
0	copy the display name of the option
1	copy the English description. Localized versions are not available.
2	copy the default value
3	copy the option flags

buffer: Pointer to an Ansi string

The buffer that will be populated based on the value of 'what'.

bufferSize: Size_t

The size of the provided buffer

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Get_Page_Annotation_Count

IGR_Get_Page_Annotation_Count returns the number of annotations that are contained on the page. For Office documents, comments are extracted and included in the annotation count. For comments spanning multiple pages, the OFFICE_COMMENT_PAGE open document option determines if the first, last or all pages containing the comments should return an annotation.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Annotation_Count (  
    HPAGE Page,  
    IGR_LONG* Count,  
    Error_Control_Block* Error);
```

PARAMETERS

Page: Handle to Page

The handle to a page that was opened using [IGR_Open_Page](#).

Count: Pointer to IGR_LONG

Returns the number of annotations on the page on success.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block Error;  
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;  
  
IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,  
    &Error);
```

```

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)
    {
        for (IGR_LONG i = 0; i < PageCount; i++)
        {
            if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
            {
                IGR_LONG AnnotationCount = 0;
                IGR_Get_Page_Annotation_Count(PageHandle, &AnnotationCount, &Error);

                for (IGR_LONG j = 0; j < AnnotationCount; j++)
                {
                    IGR_LONG read = 1;
                    IGR_Annotation annotation;

                    if (IGR_Get_Page_Annotations(PageHandle, j, &read, &annotation,
                        &Error) == IGR_OK)
                    {
                        // ...
                    }
                }

                IGR_Close_Page(PageHandle, &Error);
            }
        }
    }
}

```

SEE ALSO

- [IGR_Annotation](#)
- [IGR_Get_Page_Annotations](#)
- [IGR_Get_Page_Annotation_Long](#)
- [IGR_Get_Page_Annotation_Str](#)

IGR_Get_Page_Annotation_Long

IGR_Get_Page_Annotation_Long copies an integer value from a given property of an existing annotation into the user supplied buffer.

Path is a JSON string adhering to the Annotation schema that can be found in the Document Filters installation directory in “Annotation-Schema.json”; additionally, adding “.length” to the path of any given property that is an Array will return the number of elements in the Array.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Annotation_Long (  
    const struct IGR_Annotation* Item,  
    const IGR_UCS2* Path,  
    IGR_LONG* Buffer,  
    Error_Control_Block* Error);
```

PARAMETERS

Item: Pointer to IGR_Annotation

Pointer to a annotation returned by a call to [IGR_Get_Page_Annotations](#).

Path: Pointer to IGR_UCS2

JSON string specifying the annotation property to be extracted, following the schema defined in “Annotation-Schema.json” in the Document Filters installation directory; adding “.length” to the path of any given property that is an Array will return the number of elements in the Array.

Buffer: Pointer to IGR_LONG

Returns the integer value of the property specified by Path for the given Item.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```

Error_Control_Block Error;
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,
&Error);

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)
    {
        for (IGR_LONG i = 0; i < PageCount; i++)
        {
            if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
            {
                IGR_LONG AnnotationCount = 0;
                IGR_Get_Page_Annotation_Count(PageHandle, &AnnotationCount, &Error);

                for (IGR_LONG j = 0; j < AnnotationCount; j++)
                {
                    IGR_LONG read = 1;
                    IGR_Annotation annotation;

                    if (IGR_Get_Page_Annotations(PageHandle, j, &read, &annotation,
&Error) == IGR_OK)
                    {
                        IGR_LONG value;

                        IGR_Get_Page_Annotation_Long(&annotation,
                            _UCS2("rect.left"),
                            &value,
                            &Error);
                    }
                }

                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
}

```

SEE ALSO

- [IGR_Annotation](#)
- [IGR_Get_Page_Annotations](#)
- [IGR_Get_Page_Annotation_Count](#)
- [IGR_Get_Page_Annotation_Str](#)

IGR_Get_Page_Annotation_Str

IGR_Get_Page_Annotation_Str copies a string value from a given property of an existing annotation into the user supplied buffer.

Path is a JSON string adhering to the Annotation schema that can be found in the Document Filters installation directory in “Annotation-Schema.json”; additionally, adding “.length” to the path of any given property that is an Array will return the number of elements in the Array.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Annotation_Str (  
    const struct IGR_Annotation* Item,  
    const IGR_UCS2* Path,  
    IGR_LONG* CharCount,  
    IGR_UCS2* Buffer,  
    Error_Control_Block* Error);
```

PARAMETERS

Item: Pointer to IGR_Annotation

Pointer to a annotation returned by a call to [IGR_Get_Page_Annotations](#).

Path: Pointer to IGR_UCS2

JSON string specifying the annotation property to be extracted, following the schema defined in “Annotation-Schema.json” in the Document Filters installation directory; adding “.length” to the path of any given property that is an Array will return the number of elements in the Array.

CharCount: Pointer to IGR_LONG

Prior to the call: The size in Unicode (UCS2) characters of the buffer.

After the call: The actual number of Unicode (UCS2) characters extracted.

Buffer: Pointer to IGR_UCS2

Returns the string value of the property specified by Path for the given Item.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```

Error_Control_Block Error;
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,
&Error);

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)
    {
        for (IGR_LONG i = 0; i < PageCount; i++)
        {
            if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
            {
                IGR_LONG AnnotationCount = 0;
                IGR_Get_Page_Annotation_Count(PageHandle, &AnnotationCount, &Error);

                for (IGR_LONG j = 0; j < AnnotationCount; j++)
                {
                    IGR_LONG read = 1;
                    IGR_Annotation annotation;

                    if (IGR_Get_Page_Annotations(PageHandle, j, &read, &annotation,
&Error) == IGR_OK)
                    {
                        IGR_LONG charCount = 255;
                        IGR_UCS2 value[charCount];

                        IGR_Get_Page_Annotation_Str(&annotation,
                            _UCS2("title"),
                            &charCount,
                            &value[0],
                            &Error);
                    }
                }

                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
}

```

SEE ALSO

- [IGR_Annotation](#)
- [IGR_Get_Page_Annotations](#)
- [IGR_Get_Page_Annotation_Count](#)
- [IGR_Get_Page_Annotation_Long](#)

IGR_Get_Page_Annotations

IGR_Get_Page_Annotations copies references of page annotations into the user supplied array. The caller can iterate over all the page annotations by incrementing the Index parameter. For Office documents, comments will be extracted and returned in the page annotations array. For comments spanning multiple pages, the OFFICE_COMMENT_PAGE open document option determines if the first, last or all pages containing the comments should return an annotation.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Annotations (  
    HPAGE Page,  
    IGR_LONG Index,  
    IGR_LONG* Count,  
    IGR_Annotation* Items,  
    Error_Control_Block* Error);
```

PARAMETERS

PageHandle: HPAGE

Handle to a page, opened by a call to [IGR_Open_File](#).

Index: IGR_LONG

Offset of the first form element to return, 0 based.

Count: Pointer to IGR_LONG

Prior to the call: Set to the number of IGR_Annotation structures pointed to by the Items buffer.

After the call: Returns the number of items copied into the Items buffer.

Items: Pointer to IGR_Annotation

Pointer to a user allocated array of [IGR_Annotation](#) structures to be filled.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E](#) error codes.

SAMPLE CODE

C/C++

```
Error_Control_Block Error;
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,
&Error);

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)
    {
        for (IGR_LONG i = 0; i < PageCount; i++)
        {
            if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
            {
                IGR_LONG AnnotationCount = 0;
                IGR_Get_Page_Annotation_Count(PageHandle, &AnnotationCount, &Error);

                for (IGR_LONG j = 0; j < AnnotationCount; j++)
                {
                    IGR_LONG read = 1;
                    IGR_Annotation annotation;

                    if (IGR_Get_Page_Annotations(PageHandle, j, &read, &annotation,
&Error) == IGR_OK)
                    {
                        // ...
                    }
                }

                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
}
```

SEE ALSO

- [Annotation interface](#)
- [IGR_Annotation](#)
- [IGR_Get_Page_Annotation_Count](#)
- [IGR_Get_Page_Annotation_Long](#)
- [IGR_Get_Page_Annotation_Str](#)

IGR_Get_Page_Attribute

IGR_Get_Page_Attribute returns style or properties of an open page; see <page> under Structured XML for a full list of options.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Attribute(  
    HPAGE Page,  
    const IGR_UCS2* Name,  
    IGR_UCS2* Buffer,  
    IGR_LONG* BufferSize,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Page: Handle to Page

The handle to a page that was opened using [IGR_Open_Page](#).

Name: Unicode string (UCS2)

The name of the attribute to be extracted, see <page> under Structured XML for a full list of options.

Buffer: Unicode string (UCS2)

Application allocated memory block that will be filled with the next portion of text.

BufferSize: Pointer to IGR_LONG

Prior to the call: The size in Unicode (UCS2) characters of the buffer.

After the call: The actual number of Unicode (UCS2) characters extracted.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```

Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,
&ISYSError);

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSError) == IGR_OK)
    {
        for (IGR_LONG I = 1; I <= PageCount; i++)
        {
            if (IGR_Open_Page(DocHandle, i, &PageHandle, &ISYSError) == IGR_OK)
            {
                IGR_UCS2 Buffer[255];
                IGR_LONG BufferSize = 254;

                if (IGR_Get_Page_Attribute(PageHandle, _UCS2("SourceDpiX"), Buffer,
                    BufferSize, &ISYSError) == IGR_OK)
                {
                    // - Buffer contains the value for SourceDpiX
                }
                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
}

```

SEE ALSO

- [Page::GetAttribute method](#)

IGR_Get_Page_Count

IGR_Get_Page_Count returns the number of pages generated for an open document. This method only works on functions opened with IGR_FORMAT_IMAGE.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Count(
    IGR_LONG DocHandle,
    IGR_LONG* PageCount,
    Error_Control_Block* Error);
```

PARAMETERS

DocHandle: Unicode string (UCS2)

Handle to a document, opened by a call to [IGR_Open_File](#), [IGR_Open_File_Ex](#), [IGR_Open_Stream](#) or [IGR_Open_Stream_Ex](#).

PageCount: Pointer to IGR_LONG

Set to the number of pages contained within the document.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle, PageCount;
HPAGE PageHandle;

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &ISYSError);
if (RC == IGR_OK)
{
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSError) == IGR_OK)
```

```
{
    for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)
    {
        if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSERROR) == IGR_OK)
        {
            // Process Page Element

            IGR_Close_Page(PageHandle, &ISYSERROR);
        }
    }
    IGR_Close_File(DocHandle, &ISYSERROR);
}
```

SEE ALSO

- [IGR_Open_Page](#)

IGR_Get_Page_Dimensions

IGR_Get_Page_Dimensions returns the size of the given page in pixels.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Dimensions(  
    HPAGE pageHandle,  
    IGR_LONG* width,  
    IGR_LONG* height,  
    Error_Control_Block* error);
```

PARAMETERS

PageHandle: HPAGE

Handle to a page, opened by a call to [IGR_Open_Page](#).

Width: Pointer to IGR_LONG

Returns the width of the page in pixels.

Height: Pointer to IGR_LONG

Returns the height of the page in pixels.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;  
IGR_LONG Capabilities, DocType, DocHandle, PageCount, WordCount;  
HPAGE PageHandle;  
  
IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &ISYSError);
```

```
if (RC == IGR_OK)
{
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSERROR) == IGR_OK)
    {
        for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)
        {
            if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSERROR) == IGR_OK)
            {
                IGR_LONG Width(0), Height(0);
                IGR_Get_Page_Dimensions(PageHandle, &Width, &Height, &ISYSERROR);
            }
        }
    }
    IGR_Close_File(DocHandle, &ISYSERROR);
}
```

IGR_Get_Page_Element_First_Child

IGR_Get_Page_Element_First_Child fills `dest` with the first child of the `parent` element.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Page_Element_First_Child(  
    IGR_HPAGE page,  
    const struct IGR_Page_Element* parent,  
    struct IGR_Page_Element* dest,  
    Error_Control_Block* error  
)
```

PARAMETERS

`page`: IGR_HPAGE

Page element containing the page elements to extract.

`parent`: Pointer to IGR_Page_Element

Optional parent element from which the children elements are to be extracted.

`dest`: Pointer to IGR_Page_Element

Pointer an IGR_Page_Element. On input, the elements struct_size must be set. On output, the struct is filled with element data.

`error`: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Get_Page_Element_Next_Sibling

IGR_Get_Page_Element_Next_Sibling fills `dest` with the next sibling node of `element` element.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Page_Element_Next_Sibling(  
    IGR_HPAGE page,  
    const struct IGR_Page_Element* element,  
    struct IGR_Page_Element* dest,  
    Error_Control_Block* error  
)
```

PARAMETERS

`page`: IGR_HPAGE

Page element containing the page elements to extract.

`element`: Pointer to IGR_Page_Element

The element of the element where the sibling should be extracted.

`dest`: Pointer to IGR_Page_Element

Pointer an IGR_Page_Element. On input, the elements struct_size must be set. On output, the struct is filled with element data.

`error`: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

`Success`: IGR_LONG

Returns IGR_OK.

`Failure`: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Get_Page_Element_Root

IGR_Get_Page_Element_Root fills {dest} with the root element of the page.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Page_Element_Root(  
    IGR_HPAGE page,  
    struct IGR_Page_Element* dest,  
    Error_Control_Block* error  
)
```

PARAMETERS

page: IGR_HPAGE

Page element containing the page elements to extract.

dest: Pointer to IGR_Page_Element

Pointer an IGR_Page_Element. On input, the elements struct_size must be set. On output, the struct is filled with element data.

error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Get_Page_Element_Style

IGR_Get_Page_Element_Style returns the named style value from the page element.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Page_Element_Style(  
    IGR_HPAGE page,  
    const struct IGR_Page_Element* source,  
    const IGR_UCS2* name,  
    IGR_ULONG* char_count,  
    IGR_UCS2* buffer,  
    Error_Control_Block* error  
)
```

PARAMETERS

page: IGR_HPAGE

Page element containing the page elements to extract.

source: Pointer to IGR_Page_Element

The element of the element where the text should be extracted.

name: Pointer to IGR_UCS2

The name of the style to extractor.

char_count: Pointer to IGR_ULONG

On input, contains the number of elements in the {dest} array. On output, is set to the number of elements copied.

buffer: Pointer to IGR_UCS2

Pointer an IGR_UCS2 array.

error: Pointer to Error_Control_Block

Contains any error text.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

- [IGR_Get_Page_Element_Styles](#)

IGR_Get_Page_Element_Styles

IGR_Get_Page_Element_Styles calls the `callback` with each style for the given `source` element.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Page_Element_Styles(  
    IGR_HPAGE page,  
    const struct IGR_Page_Element* source,  
    IGR_PAGE_ELEMENT_STYLES_CALLBACK callback,  
    void* context,  
    Error_Control_Block* error  
)
```

PARAMETERS

`page`: IGR_HPAGE

Page element containing the page elements to extract.

`source`: Pointer to IGR_Page_Element

The element of the element where the styles should be extracted.

`callback`: IGR_PAGE_ELEMENT_STYLES_CALLBACK

User supplied callback where the style information is to be sent.

`context`: Pointer to void

User supplied callback context to pass when calling {callback}.

`error`: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Get_Page_Element_Text

IGR_Get_Page_Element_Text returns any text associated with the page element.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Page_Element_Text(  
    IGR_HPAGE page,  
    const struct IGR_Page_Element* source,  
    IGR_ULONG* char_count,  
    IGR_UCS2* buffer,  
    Error_Control_Block* error  
)
```

PARAMETERS

page: IGR_HPAGE

Page element containing the page elements to extract.

source: Pointer to IGR_Page_Element

The element of the element where the text should be extracted.

char_count: Pointer to IGR_ULONG

On input, contains the number of elements in the {dest} array. On output, is set to the number of elements copied.

buffer: Pointer to IGR_UCS2

Pointer an IGR_UCS2 array.

error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

IGR_Get_Page_Elements

IGR_Get_Page_Elements fills the `dest` buffer with `dest_count` page elements from the parent object, starting from the `first` index. If parent is NULL, the root element is copied into `dest`.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Page_Elements(
    IGR_HPAGE page,
    const struct IGR_Page_Element* parent,
    IGR_ULONG first,
    IGR_ULONG* dest_count,
    struct IGR_Page_Element* dest,
    Error_Control_Block* error);
```

PARAMETERS

`page`: HPAGE

Handle to a page, opened by a call to [IGR_Open_Page](#).

`parent`: *IGR_Page_Element

Pointer to parent element, or NULL to retrieve elements for the root of the page.

`first`: IGR_ULONG

Index of first element to extract.

`dest_count`: *IGR_ULONG

Must set to size of `dest` array when calling. Will be set to number of elements populated on return.

`dest`: *IGR_Page_Element

Pointer to `dest_count` array of `IGR_Page_Element`. On input, the first elements `struct_size` must be set. On output, the structures are filled with element data.

`Error`: Pointer to `Error_Control_Block`

Returns error details if the call fails.

RETURN VALUE

`Success`: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E](#) error codes.

SEE ALSO

- [IGR_Page_Element Struct](#)
- [IGR_PAGE_ELEMENT_TYPE](#)

IGR_Get_Page_Form_Element_Count

IGR_Get_Page_Form_Element_Count returns the number of form elements that are contained on the page.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Form_Element_Count (  
    HPAGE Page,  
    IGR_LONG* Count,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Page: Handle to Page

The handle to a page that was opened using [IGR_Open_Page](#).

Count: Pointer to IGR_LONG

Returns the number of form elements on the page on success.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block Error;  
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;  
  
IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,  
&ISYSError);  
  
if (RC == IGR_OK)  
{  
    IGR_LONG PageCount;  
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)  
    {
```

```
for (IGR_LONG i = 0; i < PageCount; i++)
{
    if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
    {
        IGR_LONG ElementCount = 0;
        IGR_Get_Page_Form_Element_Count(PageHandle, &ElementCount, &Error);

        for (IGR_LONG j = 0; j < ElementCount; j++)
        {
            IGR_LONG read = 1;
            IGR_Page_Form_Element FormElement;

            if (IGR_Get_Page_Form_Elements(PageHandle, j, &read, &FormElement,
                &Error) == IGR_OK)
            {
                // ...
            }
        }

        IGR_Close_Page(PageHandle, &ISYSError);
    }
}
}
```

SEE ALSO

- [FormElement interface](#)
- [IGR_Get_Page_Form_Elements](#)
- [IGR_Get_Page_Form_Element_Option_Str](#)
- [IGR_Get_Page_Form_Element_Str](#)

IGR_Get_Page_Form_Element_Option_Str

IGR_Get_Page_Form_Element_Option_Str copies string information from an existing form element's option into the user supplied buffer.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Form_Element_Option_Str(  
    const struct IGR_Page_Form_Element* Item,  
    IGR_LONG ValueType,  
    IGR_LONG OptionIndex,  
    IGR_LONG CharCount,  
    IGR_UCS2* Buffer,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Item: Pointer to IGR_Page_Form_Element

Pointer to a form element returned by a call to IHandle to a page, opened by a call to [IGR_Get_Page_Form_Elements](#).

ValueType: IGR_LONG

Indicates the type of value being requested, can be one of the following:

Name	Value
IGR_PAGE_FORM_ELEMENT_GET_NAME	0
IGR_PAGE_FORM_ELEMENT_GET_VALUE	1
IGR_PAGE_FORM_ELEMENT_GET_SELECTED	5

OptionIndex: IGR_LONG

Indicates the 0-based option the value is being request for.

CharCount: IGR_LONG

Set to the number of characters pointed to by Buffer.

Items: Pointer to IGR_UCS2

Pointer to a user allocated array of IGR_UCS2 to be filled.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block Error;
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,
&ISYSError);

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)
    {
        for (IGR_LONG i = 0; i < PageCount; i++)
        {
            if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
            {
                IGR_LONG ElementCount = 0;
                IGR_Get_Page_Form_Element_Count(PageHandle, &ElementCount, &Error);

                for (IGR_LONG j = 0; j < ElementCount; j++)
                {
                    IGR_LONG read = 1;
                    IGR_Page_Form_Element FormElement;

                    if (IGR_Get_Page_Form_Elements(PageHandle, j, &read, &FormElement,
&Error) == IGR_OK)
                    {
                        const IGR_LONG valueSize = 255;
                        IGR_UCS2 value[valueSize];
                        IGR_Get_Page_Form_Element_Str(&FormElement,
IGR_PAGE_FORM_ELEMENT_GET_VALUE,
valueSize,
&value[0],
&Error);
                    }
                }

                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
}
```

SEE ALSO

- [IGR_Get_Page_Form_Elements](#)
- [IGR_Get_Page_Form_Element_Count](#)
- [IGR_Get_Page_Form_Element_Str](#)

IGR_Get_Page_Form_Element_Str

IGR_Get_Page_Form_Element_Str copies string information from an existing form element into the user supplied buffer.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Form_Element_Str (
    const struct IGR_Page_Form_Element* Item,
    IGR_LONG ValueType,
    IGR_LONG CharCount,
    IGR_UCS2* Buffer,
    Error_Control_Block* ISYSError);
```

PARAMETERS

Item: Pointer to IGR_Page_Form_Element

Pointer to a form element returned by a call to IHandle to a page, opened by a call to [IGR_Get_Page_Form_Elements](#).

ValueType: IGR_LONG

Indicates the type of value being requested, can be one of the following:

Name	Value
IGR_PAGE_FORM_ELEMENT_GET_NAME	0
IGR_PAGE_FORM_ELEMENT_GET_VALUE	1
IGR_PAGE_FORM_ELEMENT_GET_CAPTION	2
IGR_PAGE_FORM_ELEMENT_GET_ACTION	3
IGR_PAGE_FORM_ELEMENT_GET_ACTION_DEST	4
IGR_PAGE_FORM_ELEMENT_GET_SELECTED	5
IGR_PAGE_FORM_ELEMENT_GET_FONT_NAME	6

CharCount: IGR_LONG

Set to the number of characters pointed to by Buffer.

Buffer: Pointer to IGR_UCS2

Pointer to a user allocated array of IGR_UCS2 to be filled.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```

Error_Control_Block Error;
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,
&ISYS_Error);

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)
    {
        for (IGR_LONG i = 0; i < PageCount; i++)
        {
            if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
            {
                IGR_LONG ElementCount = 0;
                IGR_Get_Page_Form_Element_Count(PageHandle, &ElementCount, &Error);

                for (IGR_LONG j = 0; j < ElementCount; j++)
                {
                    IGR_LONG read = 1;
                    IGR_Page_Form_Element FormElement;

                    if (IGR_Get_Page_Form_Elements(PageHandle, j, &read, &FormElement,
&Error) == IGR_OK)
                    {
                        const IGR_LONG valueSize = 255;
                        IGR_UCS2 value[valueSize];
                        IGR_Get_Page_Form_Element_Str(&FormElement,
IGR_PAGE_FORM_ELEMENT_GET_VALUE,
valueSize,
&value[0],
&Error);
                    }
                }
            }
        }
    }
}

```

```
        IGR_Close_Page(PageHandle, &ISYSError);  
    }  
}  
}
```

SEE ALSO

- [IGR_Get_Page_Form_Elements](#)
- [IGR_Get_Page_Form_Element_Count](#)
- [IGR_Get_Page_Form_Element_Option_Str](#)

IGR_Get_Page_Form_Elements

IGR_Get_Page_Form_Elements copies references of page form elements into the user supplied array. The caller can iterate over all the page form elements by incrementing the Index parameter.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Form_Elements (
    HPAGE Page,
    IGR_LONG Index,
    IGR_LONG* Count,
    IGR_Page_Form_Element* Items,
    Error_Control_Block* ISYSError);
```

PARAMETERS

PageHandle: HPAGE

Handle to a page, opened by a call to [IGR_Open_File](#).

Index: IGR_LONG

Offset of the first form element to return, 0 based.

Count: Pointer to IGR_LONG

Prior to the call: Set to the number of IGR_Page_Form_Element structures pointed to by the Items buffer.

After the call: Returns the number of items copied into the Items buffer.

Items: Pointer to IGR_Page_Form_Element

Pointer to a user allocated array of [IGR_Page_Form_Element](#) structures to be filled.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```

Error_Control_Block Error;
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,
&ISYSError);

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)
    {
        for (IGR_LONG i = 0; i < PageCount; i++)
        {
            if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
            {
                IGR_LONG ElementCount = 0;
                IGR_Get_Page_Form_Element_Count(PageHandle, &ElementCount, &Error);

                for (IGR_LONG j = 0; j < ElementCount; j++)
                {
                    IGR_LONG read = 1;
                    IGR_Page_Form_Element FormElement;

                    if (IGR_Get_Page_Form_Elements(PageHandle, j, &read, &FormElement,
&Error) == IGR_OK)
                    {
                        // ...
                    }
                }

                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
}

```

SEE ALSO

- [FormElement interface](#)
- [IGR_Page_Form_Element](#)
- [IGR_Get_Page_Form_Element_Count](#)
- [IGR_Get_Page_Form_Element_Option_Str](#)
- [IGR_Get_Page_Form_Element_Str](#)

IGR_Get_Page_Handle_File_Type

IGR_Get_Page_Handle_File_Type gets the type and the capabilities of an open page.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Page_Handle_File_Type(  
    IGR_HPAGE handle,  
    IGR_LONG* capabilities,  
    IGR_LONG* filetype,  
    Error_Control_Block* error  
)
```

PARAMETERS

handle: IGR_HPAGE

Handle to an open document.

capabilities: Pointer to IGR_LONG

Returns the Document Capabilities as a bit field.

filetype: Pointer to IGR_LONG

Returns the Document Format Code of the document.

error: Pointer to Error_Control_Block

Contains any error text.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

- [IGR_Get_Handle_File_Type](#)
- [Document Format Codes](#)
- [Document Capability Codes](#)

IGR_Get_Page_Hyperlink_Count

IGR_Get_Page_Hyperlink_Count returns the number of hyperlinks that are contained on the page.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Hyperlink_Count (
    HPAGE Page,
    IGR_LONG* Count,
    Error_Control_Block* ISYSError);
```

PARAMETERS

Page: Handle to Page

The handle to a page that was opened using [IGR_Open_Page](#).

Count: Pointer to IGR_LONG

Returns the number of hyperlinks on the page on success.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block Error;
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,
&ISYSError);

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)
    {
```

```
for (IGR_LONG i = 0; i < PageCount; i++)
{
    if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
    {
        IGR_LONG HyperlinkCount = 0;
        IGR_Get_Page_Hyperlink_Count(PageHandle, &HyperlinkCount, &Error);

        for (IGR_LONG j = 0; j < HyperlinkCount; j++)
        {
            IGR_LONG read = 1;
            IGR_Hyperlink hyperlink;

            if (IGR_Get_Page_Hyperlinks(PageHandle, j, &read, &hyperlink,
                &Error) == IGR_OK)
            {
                // ...
            }
        }

        IGR_Close_Page(PageHandle, &ISYSError);
    }
}
}
```

SEE ALSO

- [IGR_Hyperlink](#)

IGR_Get_Page_Hyperlink_Str

IGR_Get_Page_Hyperlink_Str copies string information from an existing hyperlink into the user supplied buffer.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Hyperlink_Str (  
    const struct IGR_Hyperlink* Item,  
    IGR_LONG ValueType,  
    IGR_LONG CharCount,  
    IGR_UCS2* Buffer,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Item: Pointer to IGR_Hyperlink

Pointer to a hyperlink returned by a call to [IGR_Get_Page_Hyperlinks](#).

ValueType: IGR_LONG

Indicates the type of value being requested, can be one of the following:

IGR_HYPERLINK_GET_VALUE_REF 0

IGR_HYPERLINK_GET_VALUE_URI 1

CharCount: IGR_LONG

Set to the number of characters pointed to by Buffer.

Buffer: Pointer to IGR_UCS2

Pointer to a user allocated array of IGR_UCS2 to be filled.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```

Error_Control_Block Error;
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,
&ISYSError);

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)
    {
        for (IGR_LONG i = 0; i < PageCount; i++)
        {
            if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
            {
                IGR_LONG HyperlinkCount = 0;
                IGR_Get_Page_Form_Element_Count(PageHandle, &HyperlinkCount, &Error);

                for (IGR_LONG j = 0; j < HyperlinkCount; j++)
                {
                    IGR_LONG read = 1;
                    IGR_Hyperlink hyperlink;

                    if (IGR_Get_Page_Hyperlinks(PageHandle, j, &read, &hyperlink,
&Error) == IGR_OK)
                    {
                        const IGR_LONG valueSize = 255;
                        IGR_UCS2 value[valueSize];
                        IGR_Get_Page_Hyperlink_Str(&hyperlink,
IGR_HYPERLINK_GET_VALUE_URI,
valueSize,
&value[0],
&Error);
                    }
                }

                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
}

```

SEE ALSO

- [IGR_Hyperlink](#)

IGR_Get_Page_Hyperlinks

IGR_Get_Page_Hyperlinks copies references of page hyperlinks into the user supplied array. The caller can iterate over all the page hyperlinks by incrementing the Index parameter.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Hyperlinks (  
    HPAGE Page,  
    IGR_LONG Index,  
    IGR_LONG* Count,  
    IGR_Hyperlink* Items,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

PageHandle: HPAGE

Handle to a page, opened by a call to [IGR_Open_File](#).

Index: IGR_LONG

Offset of the hyperlink to return, 0 based.

Count: Pointer to IGR_LONG

Prior to the call: Set to the number of [IGR_Hyperlink](#) structures pointed to by the Items buffer.

After the call: Returns the number of items copied into the Items buffer.

Items: Pointer to IGR_Hyperlink

Pointer to a user allocated array of [IGR_Hyperlink](#) structures to be filled.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```

Error_Control_Block Error;
IGR_LONG Capabilities, DocType, DocHandle, PageHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle,
&ISYSError);

if (RC == IGR_OK)
{
    IGR_LONG PageCount;
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &Error) == IGR_OK)
    {
        for (IGR_LONG i = 0; i < PageCount; i++)
        {
            if (IGR_Open_Page(DocHandle, i, &PageHandle, &Error) == IGR_OK)
            {
                IGR_LONG HyperlinkCount = 0;
                IGR_Get_Page_Hyperlink_Count(PageHandle, &HyperlinkCount, &Error);

                for (IGR_LONG j = 0; j < HyperlinkCount; j++)
                {
                    IGR_LONG read = 1;
                    IGR_Hyperlink hyperlink;

                    if (IGR_Get_Page_Hyperlink(PageHandle, j, &read, &hyperlink,
&Error) == IGR_OK)
                    {
                        // ...
                    }
                }

                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
}

```

SEE ALSO

- [IGR_Hyperlink](#)

IGR_Get_Page_Image_Entry

IGR_Get_Page_Image_Entry enumerates the set of images on an open page.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Page_Image_Entry(  
    HPAGE pageHandle,  
    IGR_UCS2* id,  
    IGR_UCS2* name,  
    IGR_LONGLONG* date,  
    IGR_LONGLONG* size,  
    Error_Control_Block* error  
)
```

PARAMETERS

pageHandle: HPAGE

Handle to a page, opened by a call to [IGR_Open_File](#).

id: Pointer to IGR_UCS2

Application allocated memory block of 8192 bytes that will be filled with up to 4096 Unicode characters that specify the unique ID of the next sub-document.

name: Pointer to IGR_UCS2

Application allocated memory block of 2048 bytes that will be filled with up to 1024 Unicode characters that specify the name of the sub-document.

date: Pointer to IGR_LONGLONG

Returns the date and time of the sub-document in [FILETIME](#) format.

size: Pointer to IGR_LONGLONG

Returns the size in bytes of the sub-document.

error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

IGR_Get_Page_Images_Enumerator

`IGR_Get_Page_Images_Enumerator` creates a new enumerator for page images. Unlike `IGR_Get_Page_Image_Entry`, it facilitates the creation of multiple enumerators, enabling the reprocessing of images within an open page. Remember to free a subfile enumerator by calling `IGR_Subfiles_Close` once it is no longer needed.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Page_Images_Enumerator(  
    IGR_LONG docHandle,  
    HSUBFILES* enumerator,  
    Error_Control_Block* error  
)
```

PARAMETERS

`docHandle`: `IGR_LONG`

Is a handle to a file, opened by a call to `IGR_Open_File`, `IGR_Open_Stream` or equivalent.

`enumerator`: Pointer to `HSUBFILES`

Returns a handle to be used in subsequent enumerator calls.

`error`: Pointer to `Error_Control_Block`

Returns error details if the call fails.

RETURN VALUE

`Success`: `IGR_LONG`

Returns `IGR_OK`.

`Failure`: `IGR_LONG`

Returns one of the possible [IGR_E error codes](#).

IGR_Get_Page_Pixels

The IGR_Get_Page_Pixels method provides low-level access to a rectangular portion of the page and provides a temporary buffer that you can use to read the pixel data in a specified format.

The caller must call IGR_Free_Page_Pixels when done processing the pixel data.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Pixels (  
    HPAGE pageHandle,  
    const struct IGR_Rect* sourceRect,  
    const struct IGR_Size* destSize,  
    IGR_ULONG flags,  
    const IGR_UCS2* options,  
    IGR_OPEN_BITMAP_PIXEL_TYPE pixel_format,  
    struct IGR_Page_Pixels* pixel_data,  
    Error_Control_Block* error);
```

PARAMETERS

pageHandle: HPAGE

Handle to a page opened by a call to [IGR_Open_Page](#).

sourceRect: Pointer to IGR_Rect

Pointer to a rectangle that specifies the section of the page to copy.

May be NULL to copy the entire page.

destSize: Pointer to IGR_Size

Pointer to a size that specifies the requested output size for the copied pixels.

flags: IGR_LONG

A set of flags that indicate processing options.

IGR_GET_PAGE_PIXELS_BUFFER_ALLOCATED: indicates that (name) buffer is preallocated by the caller and should be filled.

options: Pointer to Unicode String

Extended processing options as name=value, semicolon separated.

pixel_format: IGR_OPEN_BITMAP_PIXEL_TYPE

Integer that specifies the format of the pixel data in the temporary buffer. The pixel format of the temporary buffer does not have to be the same as the pixel format as the page.

`pixel_data: Pointer to IGR_Page_Pixels`

If `IGR_GET_PAGE_PIXELS_BUFFER_ALLOCATED` is not set, `pixel_data` serves only as an output parameter. The `scanline0` data member is set to a pointer to a temporary buffer, which is filled with the requested pixel data.

If `IGR_GET_PAGE_PIXELS_BUFFER_ALLOCATED` is set, the caller must initial the width, height, stride and `scanline0` members of the struct with the details of the pre-allocated buffer.

`error: Pointer to Error_Control_Block`

Returns error details if the call fail.

RETURN VALUE

`Success: IGR_LONG`

Returns `IGR_OK`.

`Failure: IGR_LONG`

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

- [IGR_Free_Page_Pixels](#)
- [IGR_Page_Pixels](#)

IGR_Get_Page_Text

IGR_Get_Page_Text extracts the text of a previously opened page of a document.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Text(  
    HPAGE PageHandle,  
    IGR_UCS2* Buffer,  
    IGR_LONG* Size,  
    Error_Control_Block* error);
```

PARAMETERS

PageHandle: HPAGE

Handle to a page, opened by a call to [IGR_Open_Page](#).

Buffer: Unicode string (UCS2)

Application allocated memory block that is to be populated with the next portion of text.

Size: Pointer to IGR_LONG

Prior to the call: The size in Unicode characters of the buffer.

After the call: The actual number of Unicode characters extracted.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Success and the end of the document was reached: IGR_LONG

Returns IGR_NO_MORE.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

IGR_Get_Page_Text operates on the same concept as [IGR_Get_Text](#); the caller should keep calling IGR_Get_Page_Text until it returns IGR_E_NO_MORE error.

SEE ALSO

- [IGR_Get_Text](#)

IGR_Get_Page_Word_Count

IGR_Get_Page_Word_Count returns the number of words of the given page.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Word_Count(  
    HPAGE PageHandle,  
    IGR_LONG* WordCount,  
    Error_Control_Block* error);
```

PARAMETERS

PageHandle: HPAGE

Handle to a page, opened by a call to [IGR_Open_File](#).

WordCount: Pointer to IGR_LONG

Returns the number of words on the page on success.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;  
IGR_LONG Capabilities, DocType, DocHandle, PageCount, WordCount;  
HPAGE PageHandle;  
  
IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &ISYSError);  
if (RC == IGR_OK)  
{  
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSError) == IGR_OK)  
    {  
        for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)  
        {
```

```
    if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSError) == IGR_OK)
    {
        // Process Page Element
        IGR_Get_Page_Word_Count(PageHandle, &WordCount, &ISYSError);
        IGR_Close_Page(PageHandle, &ISYSError);
    }
}
IGR_Close_File(DocHandle, &ISYSError);
}
```

SEE ALSO

- [IGR_Get_Page_Words](#)

IGR_Get_Page_Words

IGR_Get_Page_Words copies references of page words into the user supplied array. The caller can iterate over all the page words by incrementing the Index parameter.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Page_Words(  
    HPAGE PageHandle,  
    IGR_LONG Index,  
    IGR_LONG *Count,  
    IGR_Page_Word* Words,  
    Error_Control_Block* Error);
```

PARAMETERS

PageHandle: HPAGE

Handle to a page, opened by a call to [IGR_Open_File](#).

Index: IGR_LONG

Offset of the first word to return, 0 based.

Count: Pointer to IGR_LONG

Prior to the call: Set to the number of Words structures pointed to by the Words buffer.

After the call: Returns the number of Words copied into the Words buffer.

Words: Pointer to IGR_Page_Word

Pointer to a user allocated array of [IGR_Page_Word](#) structures to be filled.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```

Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle, PageCount, WordCount;
HPAGE PageHandle;

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType,
    &DocHandle, &ISYSError);
if (RC == IGR_OK)
{
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSError) == IGR_OK)
    {
        for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)
        {
            if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSError) == IGR_OK)
            {
                // Buffer to hold the word records
                IGR_Page_Word words[255];

                IGR_LONG WordIndex = 0;
                IGR_LONG Count = sizeof(words) / sizeof(words[0]);
                while (IGR_Get_Page_Words(PageHandle, Index, &Count, words,
                    ISYSError) == 0)
                {
                    for (IGR_LONG i = 0; i < Count; i++)
                    {
                        // Process the word record
                    }
                    WordIndex += Count;
                    Count = sizeof(words) / sizeof(words[0]);
                }
                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
    IGR_Close_File(DocHandle, &ISYSError);
}

```

SEE ALSO

- [IGR_Get_Page_Word_Count](#)

IGR_Get_Stream_Type

IGR_Get_Stream_Type gets the type and the capabilities of a given stream object.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Stream_Type(  
    IGR_Stream *Stream,  
    IGR_LONG* Capabilities,  
    IGR_LONG* DocType,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Stream: Pointer to an IGR_Stream
(an stream object)

The stream can be either user implemented, or created using the [IGR_Make_Stream_From_File](#) and [IGR_Make_Stream_From_Memory](#) utility functions.

Capabilities: Pointer to IGR_LONG

Returns the Document Capabilities as a bit field.

DocType: Pointer to IGR_LONG

Returns the [Document Format Code](#) of the document.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;  
IGR_LONG Capabilities, DocType;
```

```
IGR_LONG RC = IGR_Get_Stream_Type(pStream, &Capabilities, &DocType, &ISYS_Error);  
if (RC == IGR_OK)  
{  
    if (DocType == 25)  
    {  
        // Document is an MS Word document  
    }  
}
```

ADDITIONAL INFORMATION

If the document has the IGR_FILE_SUPPORTS_TEXT capability, text may be directly extracted from the document by calling [IGR_Get_Text](#) (e.g. a Word document).

If the document has the IGR_FILE_SUPPORTS_SUBFILES capability, then it is a container for other documents and it is valid to enumerate and/or extract it's sub-documents.

It is valid for a document to have both capabilities (for example email message documents have their own text and also can have attached documents).

Document Filters also has the ability to identify certain document formats, without being able to extract content. In this situation, the capabilities will be returned as 0. See [Document Format Codes](#) for a list of these formats on page .

Compound documents can include other compound documents, for example an MSG with a ZIP attachment, which contain ZIPs and MSGs. The calling application can navigate as far down as needed.

IGR_Get_Subfile_Entry

IGR_Get_Subfile_Entry enumerates the sub-documents contained in a previously opened compound document, such as message documents (MSG) or archive documents (ZIP).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Subfile_Entry(  
    IGR_LONG DocHandle,  
    IGR_UCS2* ID,  
    IGR_UCS2* Name,  
    IGR_LONGLONG* FileDate,  
    IGR_LONGLONG* FileSize,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

DocHandle: IGR_LONG

Is a handle to a file, opened by a call to [IGR_Open_File](#) , [IGR_Open_Stream](#) or equivalent.

ID: Unicode string (UCS2)

Application allocated memory block of 8192 bytes that will be filled with up to 4096 Unicode characters that specify the unique ID of the next sub-document.

Name: Unicode string (UCS2)

Application allocated memory block of 2048 bytes that will be filled with up to 1024 Unicode characters that specify the name of the sub-document.

FileDate: Pointer to INT64

Returns the date and time of the sub-document in [FILETIME](#) format.

FileSize: Pointer to INT64

Returns the size in bytes of the sub-document.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Success and the end of the document was reached: IGR_LONG

Returns IGR_NO_MORE.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```

1 Error_Control_Block ISYSError;
2 IGR_UCS2 ID[4096], Name[1024];
3 INT64 FileDate, FileSize;
4
5 while (true)
6 {
7     IGR_LONG rc = IGR_Get_Subfile_Entry(DocHandle, ID, Name, &FileDate, &FileSize,
8         &ISYSError);
9     if (rc != IGR_OK)
10     {
11         if (rc != IGR_NO_MORE)
12             // ReportError(rc);
13         break;
14     }
15     rc = IGR_Extract_Subfile(DocHandle, ID, _UCS2("TEMP.DAT"), &ISYSError);
16     if (rc != IGR_OK)
17         // ReportError(rc);
18     else
19         // DoSomethingWithTheFile("TEMP.DAT", ID, Name);
20 }
```

ADDITIONAL INFORMATION

The previously opened document must have the IGR_FILE_SUPPORTS_SUBFILES capability.

After a successful call to [IGR_Open_File](#) / [IGR_Open_Stream](#), each call to [IGR_Get_Subfile_Entry](#) will retrieve information about the next sub-document contained in the compound document, referenced by DocHandle. To traverse all the sub-documents, the application will need to call this method in a loop until IGR_NO_MORE is returned.

Note that the null-terminating character will also be copied to the ID and Name parameters. The Name parameter could be an empty string if the ID of the sub-document is not available. If the function succeeds, the ID is guaranteed not to be empty and will be unique among all traversed sub-documents retrieved from the document. The returned ID can be used in a call to [IGR_Extract_Subfile](#) to save the binary content of the sub-document to disk.

If the date of the sub-document is not available, the parameter FileDate will be set to 0, otherwise it will be populated in FILETIME format.

If the size of the sub-document is not available, the parameter FileSize will be set to 0.

SEE ALSO

- [IGR_Get_Subfile_Entry_Ex](#)

IGR_Get_Subfile_Entry_Ex

IGR_Get_Subfile_Entry_Ex enumerates the sub-documents contained in a previously opened compound document, such as message documents (MSG) or archive documents (ZIP).

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Subfile_Entry_Ex(  
    IGR_LONG docHandle,  
    struct IGR_Subfile_Info* result,  
    Error_Control_Block* error);
```

PARAMETERS

`docHandle`: IGR_LONG

Is a handle to a file, opened by a call to [IGR_Open_File](#), [IGR_Open_Stream](#) or equivalent.

`result`: Pointer to IGR_Subfile_Info

Pointer to a user allocated [IGR_Subfile_Info](#) to be populated with the subfile information.

`error`: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

`Success`: IGR_LONG

Returns IGR_OK.

`Success and the end of the document was reached`: IGR_LONG

Returns IGR_NO_MORE.

`Failure`: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```

1 Error_Control_Block error = { 0 };
2 IGR_Subfile_Info subfile = { 0 };
3 IGR_RETURN_CODE res;
4
5 std::array<IGR_UCS2, 4096> id;
6 std::array<IGR_UCS2, 1024> name;
7 std::array<IGR_UCS2, 4096> comment;
8
9 subfile.struct_size = sizeof(subfile);
10 subfile.id = &id[0];
11 subfile.id_size = id.size();
12
13 subfile.name = &name[0];
14 subfile.name_size = name.size();
15
16 subfile.comment = &comment[0];
17 subfile.comment_size = comment.size();
18
19 while ((res = IGR_Get_Subfile_Entry_Ex(dochandle, &subfile, &error)) == IGR_OK)
20 {
21     res = IGR_Extract_Subfile(DocHandle, ID, _UCS2("TEMP.DAT"), &ISYSError);
22     if (res != IGR_OK)
23         // ReportError(rc);
24     else
25         // DoSomethingWithTheFile("TEMP.DAT", ID, Name);
26 }

```

ADDITIONAL INFORMATION

The previously opened document must have the IGR_FILE_SUPPORTS_SUBFILES capability.

After a successful call to [IGR_Open_File](#) / [IGR_Open_Stream](#), each call to [IGR_Get_Subfile_Entry_Ex](#) will retrieve information about the next sub-document contained in the compound document, referenced by DocHandle. To traverse all the sub-documents, the application will need to call this method in a loop until IGR_NO_MORE is returned.

Note that the null-terminating character will also be copied to the result.ID and result.Name parameters. The Name parameter could be an empty string if the ID of the sub-document is not available. If the function succeeds, the ID is guaranteed not to be empty and will be unique among all traversed sub-documents retrieved from the document. The returned ID can be used in a call to [IGR_Extract_Subfile](#) to save the binary content of the sub-document to disk.

If the date of the sub-document is not available, the parameter FileDate will be set to 0, otherwise it will be populated in FILETIME format.

If the size of the sub-document is not available, the parameter FileSize will be set to 0.

SEE ALSO

- [IGR_Get_Subfile_Entry](#)

IGR_Get_Subfiles_Enumerator

`IGR_Get_Subfiles_Enumerator` creates a new enumerator for subfiles. Unlike `IGR_Get_Subfile_Entry`, it facilitates the creation of multiple enumerators, enabling the reprocessing of subfiles within an open document. Remember to free a subfile enumerator by calling `IGR_Subfiles_Close` once it is no longer needed.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Get_Subfiles_Enumerator(  
    IGR_LONG docHandle,  
    HSUBFILES* enumerator,  
    Error_Control_Block* error);
```

PARAMETERS

`handle`: IGR_LONG

Is a handle to a file, opened by a call to `IGR_Open_File`, `IGR_Open_Stream` or equivalent.

`enumerator`: HSUBFILES

Returns a handle to be used in subsequent enumerator calls.

`error`: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

`Success`: IGR_LONG

Returns IGR_OK.

`Failure`: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```

1 Error_Control_Block error = { 0 };
2 IGR_Subfile_Info subfile = { 0 };
3 IGR_RETURN_CODE res;
4 HSUBFILES subs;
5
6 std::array<IGR_UCS2, 4096> id;
7 std::array<IGR_UCS2, 1024> name;
8 std::array<IGR_UCS2, 4096> comment;
9
10 subfile.struct_size = sizeof(subfile);
11 subfile.id = &id[0];
12 subfile.id_size = id.size();
13
14 subfile.name = &name[0];
15 subfile.name_size = name.size();
16
17 subfile.comment = &comment[0];
18 subfile.comment_size = comment.size();
19
20 if ((res = IGR_Get_Subfiles_Enumerator(docHandle, &subs, &error)) == IGR_OK)
21 {
22     while ((res = IGR_Subfiles_Next_Ex(subs, &subfile, &error)) == IGR_OK)
23     {
24         res = IGR_Extract_Subfile(DocHandle, ID, _UCS2("TEMP.DAT"), &ISYSError);
25         if (res != IGR_OK)
26             // ReportError(rc);
27         else
28             // DoSomethingWithTheFile("TEMP.DAT", ID, Name);
29     }
30     IGR_Subfiles_Close(subs, &error);
31 }

```

SEE ALSO

- [IGR_Subfiles_Close](#)
- [IGR_Subfiles_Next_Ex](#)
- [IGR_Subfiles_Next](#)
- [IGR_Subfiles_Reset](#)

IGR_Get_Text

IGR_Get_Text extracts the text of previously opened document.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Get_Text(  
    IGR_LONG DocHandle,  
    IGR_UCS2* Buffer,  
    IGR_LONG* BufferSize,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

DocHandle: IGR_LONG

Handle to a document, opened by a call to [IGR_Open_File](#), [IGR_Open_File_Ex](#), [IGR_Open_Stream](#), [IGR_Open_Stream_Ex](#) or [IGR_Open_Ex](#).

Buffer: Unicode string (UCS2)

Application allocated memory block that will be filled with the next portion of text.

BufferSize: Pointer to IGR_LONG

Prior to the call: The size in Unicode (UCS2) characters of the buffer.

After the call: The actual number of Unicode (UCS2) characters extracted.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Success and the end of the document was reached: IGR_LONG

Returns IGR_NO_MORE.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```
Error_Control_Block ISYSError;
IGR_UCS2 Buffer[BUFFER_SIZE+1];
IGR_LONG Size, rc;

while (true)
{
    Size = BUFFER_SIZE;
    rc = IGR_Get_Text(DocHandle, Buffer, &Size, &ISYSError);
    if (rc != IGR_OK)
    {
        if (rc != IGR_NO_MORE)
            // ReportError(rc);
        break;
    }
    Buffer[Size] = 0;
    // DoSomethingWithTheText(Buffer);
}
```

ADDITIONAL INFORMATION

The previously opened document must have the IGR_FILE_SUPPORTS_TEXT capability.

Note The populated buffer will not be null-terminated. If required, a null terminator may be explicitly added to the buffer at position BufferSize as shown in the Sample code (above).

After a successful call to [IGR_Open_File](#), [IGR_Open_File_Ex](#), [IGR_Open_Stream](#) or [IGR_Open_Stream_Ex](#), the document pointer is set to the beginning of the text to be returned. Each call to [IGR_Get_Text](#) will retrieve the next portion of the text and a maximum of BufferSize characters will be copied to Buffer. To extract the whole text, the application will need to call [IGR_Get_Text](#) in a loop until the function returns IGR_NO_MORE.

Text returned may contain markup characters that your application will need to process.

IGR_Make_Output_Canvas

IGR_Make_Output_Canvas creates a new canvas that is used for rendering page content. The output data will be written to the file specified in Filename. To write to memory or stream, see IGR_Make_Output_Canvas_On.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Make_Output_Canvas(
    IGR_LONG Type,
    const IGR_UCS2* Filename,
    const IGR_UCS2* Options,
    HCANVAS* CanvasHandle,
    Error_Control_Block* error);
```

PARAMETERS

Type: IGR_LONG

Indicates the type of canvas object to create, can be one of the following:

Name	Value	Description
IGR_DEVICE_HTML	6	Create a single or multipage HTML5
IGR_DEVICE_IMAGE_BMP	4	Create a single BMP per page
IGR_DEVICE_IMAGE_BRK	17	Create a single or multipage Brooktrout FAX
IGR_DEVICE_IMAGE_DCX	19	Create a single or multipage DCX
IGR_DEVICE_IMAGE_EPS	13	Create a single EPS (Encapsulated PostScript) per page
IGR_DEVICE_IMAGE_GIF	21	Create a single GIF per page
IGR_DEVICE_IMAGE_JPEG2000	20	Create a single JPEG2000 per page
IGR_DEVICE_IMAGE_JPG	1	Create a single JPG per page
IGR_DEVICE_IMAGE_PBM	7	Create a single PBM per page
IGR_DEVICE_IMAGE_PCX	18	Create a single PCX per page
IGR_DEVICE_IMAGE_PDF	2	Create a single or multipage PDF
IGR_DEVICE_IMAGE_PGM	8	Create a single PGM per page

Name	Value	Description
IGR_DEVICE_IMAGE_PNG	0	Create a single PNG per page
IGR_DEVICE_IMAGE_PPM	9	Create a single PPM per page
IGR_DEVICE_IMAGE_PS	14	Create a single or multipage PostScript
IGR_DEVICE_IMAGE_SVG	12	Create a single SVG per page
IGR_DEVICE_IMAGE_TGA	16	Create a single TGA per page
IGR_DEVICE_IMAGE_TIF	3	Create a single or multipage TIF
IGR_DEVICE_IMAGE_WEBP	10	Create a single WEBP per page
IGR_DEVICE_IMAGE_WEBSAFE	15	Create a single image per page, where the format is determined based on the palette and number of colors.
IGR_DEVICE_IMAGE_XPS	11	Create a single or multipage XPS
IGR_DEVICE_JSON	22	Create a single or multipage structured JSON
IGR_DEVICE_PDF	2	Create a single or multipage PDF (Alias for IGR_DEVICE_IMAGE_PDF)
IGR_DEVICE_XML	5	Create a single or multipage structured XML

Filename: Unicode string (UCS2)

Destination filename where the output is written.

CanvasHandle: Pointer to HCANVAS

Returns a handle to be used in subsequent canvas calls.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle, PageCount;
HPAGE PageHandle;

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &ISYSError);
if (RC == IGR_OK)
{
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSError) == IGR_OK)
    {
        for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)
        {
            if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSError) == IGR_OK)
            {
                HCANVAS CanvasHandle;
                if (IGR_Make_Output_Canvas(IGR_DEVICE_IMAGE_PNG, L"page.png",
                    &CanvasHandle, &ISYSError) == IGR_OK)
                {
                    IGR_Render_Page(PageHandle, CanvasHandle, &ISYSError);
                    IGR_Close_Canvas(CanvasHandle, &ISYSError);
                }
                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
    IGR_Close_File(DocHandle, &ISYSError);
}
```

ADDITIONAL INFORMATION

Some canvas objects allow multiple pages to be rendered to the same file, PDF for example. In this circumstance, create the canvas object outside of the loop and call [IGR_Render_Page](#) to the one canvas object. For output formats that support multiple pages, you may choose to write multiple input documents to a single output document.

SEE ALSO

- [IGR_Make_Output_Canvas_On](#)

IGR_Make_Output_Canvas_On

IGR_Make_Output_Canvas_On creates a new canvas that is used for rendering page content, the output data will be written to the stream specified. The Stream must be a caller created IGR_Writable_Stream derivative.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Make_Output_Canvas_On(
    IGR_LONG Type,
    IGR_Writable_Stream* Stream,
    const IGR_UCS2* options,
    HCANVAS* CanvasHandle,
    Error_Control_Block* error);
```

PARAMETERS

Type: IGR_LONG

Indicates the type of canvas object to create. can be one of the following:

Name	Value	Description
IGR_DEVICE_HTML	6	Create a single or multipage HTML5
IGR_DEVICE_IMAGE_BMP	4	Create a single BMP per page
IGR_DEVICE_IMAGE_BRK	17	Create a single or multipage Brooktrout FAX
IGR_DEVICE_IMAGE_DCX	19	Create a single or multipage DCX
IGR_DEVICE_IMAGE_EPS	13	Create a single EPS (Encapsulated PostScript) per page
IGR_DEVICE_IMAGE_GIF	21	Create a single GIF per page
IGR_DEVICE_IMAGE_JPEG2000	20	Create a single JPEG2000 per page
IGR_DEVICE_IMAGE_JPG	1	Create a single JPG per page
IGR_DEVICE_IMAGE_PBM	7	Create a single PBM per page
IGR_DEVICE_IMAGE_PCX	18	Create a single PCX per page
IGR_DEVICE_IMAGE_PDF	2	Create a single or multipage PDF
IGR_DEVICE_IMAGE_PGM	8	Create a single PGM per page

Name	Value	Description
IGR_DEVICE_IMAGE_PNG	0	Create a single PNG per page
IGR_DEVICE_IMAGE_PPM	9	Create a single PPM per page
IGR_DEVICE_IMAGE_PS	14	Create a single or multipage PostScript
IGR_DEVICE_IMAGE_SVG	12	Create a single SVG per page
IGR_DEVICE_IMAGE_TGA	16	Create a single TGA per page
IGR_DEVICE_IMAGE_TIF	3	Create a single or multipage TIF
IGR_DEVICE_IMAGE_WEBP	10	Create a single WEBP per page
IGR_DEVICE_IMAGE_WEBSAFE	15	Create a single image per page, where the format is determined based on the palette and number of colors.
IGR_DEVICE_IMAGE_XPS	11	Create a single or multipage XPS
IGR_DEVICE_JSON	22	Create a single or multipage structured JSON
IGR_DEVICE_PDF	2	Create a single or multipage PDF (Alias for IGR_DEVICE_IMAGE_PDF)
IGR_DEVICE_XML	5	Create a single or multipage structured XML

Stream: Pointer to IGR_Writable_Stream

A caller provided stream object where the output data is to be written. It is the caller's responsibility to create and destroy the stream.

CanvasHandle: Pointer to HCANVAS

Returns a handle to be used in subsequent canvas calls.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

To use `IGR_Make_Output_Canvas_On`, the caller must create an object/record that derives from `IGR_Writable_Stream`. The implementation must dynamically grow memory, as the amount for data that will be written is not known at creation time.

SEE ALSO

- [IGR_Make_Output_Canvas](#)
- [IGR_Writable_Stream](#)

IGR_Make_Stream_From_File

IGR_Make_Stream_From_File creates a stream based on a file for use with the document stream functions.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Make_Stream_From_File(  
    const IGR_UCS2* FileName,  
    IGR_LONG Flags,  
    IGR_Stream **Stream,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

FileName: Unicode string (UCS2)

Path to the document to be opened.

Flags: IGR_LONG

A bit field of options that affect the behavior of the stream object.

FILE_FLAG_DELETE_ON_CLOSE *Value:* 0x4000000 *_Description:* _Indicates the document specified in FileName should be deleted when the stream object is closed.

Stream: Pointer to an IGR_Stream pointer

A system allocated memory stream structure will be returned.

It is the caller's responsibility to free the stream object by calling Stream->Close()

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType;
IGR_Stream *pStream;

if (IGR_Make_Stream_From_File(_UCS2("TEST.DOC"), 0, &pStream, &ISYSError) == IGR_OK)
{
    if (IGR_Get_Stream_Type(pStream, &Capabilities, &DocType, &ISYSError) == IGR_OK)
    {
        if (DocType == 25)
        {
            // Document is an MS Word document
        }
    }
    pStream->Close(pStream);
}
```

SEE ALSO

- [IGR_Open_Stream](#)
- [IGR_Get_Stream_Type](#)
- [IGR_Stream](#)

IGR_Make_Stream_From_Functions

`IGR_Make_Stream_From_Functions` creates a new IGR stream based off the context and functions. This is functionally identical to constructing the `IGR_Stream` structure by hand; this function can be used by languages that do not have precise control over memory layout.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Make_Stream_From_Functions(
    void* context,
    IGR_LONG flags,
    IGR_ULONG(CALLBACK* seek)(void* context, IGR_LONGLONG offset, IGR_ULONG whence),
    IGR_ULONG(CALLBACK* read)(void* context, void* buffer, IGR_ULONG buffer_size),
    IGR_ULONG(CALLBACK* write)(void* context, const void* buffer, IGR_ULONG buffer_size),
    IGR_LONG(CALLBACK* action)(void* context, int actionId, void* action_data),
    void (CALLBACK* destruct)(void* context),
    struct IGR_Stream** stream,
    Error_Control_Block* error
)
```

PARAMETERS

`context`: Pointer to void

Context is a user-provided block that is passed to each callback function.

`flags`: `IGR_LONG`

A bit field of options that affect the behavior of the stream object.

`seek`: Pointer to func

Function pointer to required Seek method.

`read`: Pointer to func

Function pointer to required Read method.

`write`: Pointer to func

Function pointer to optional Write method.

`action`: Pointer to func

Function pointer to optional Action method.

`destruct`: Pointer to

Function pointer to required destructor that can be used to dispose of *context* memory.

stream: Pointer to

A system allocated memory stream structure will be returned. It is the caller's responsibility to free the stream object by calling `stream->Close()` .

error: Pointer to `Error_Control_Block`

Returns error details if the call fails.

RETURN VALUE

Success: `IGR_LONG`

Returns `IGR_OK`.

Failure: `IGR_LONG`

Returns one of the possible [IGR_E error codes](#).

IGR_Make_Stream_From_Memory

IGR_Make_Stream_From_Memory creates a stream based on a memory buffer for use with the document stream functions.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Make_Stream_From_Memory(  
    void * Data,  
    IGR_LONG DataSize,  
    void * Destructor  
    IGR_Stream **Stream,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Data: Pointer

A pointer to a user allocated memory buffer that contains the binary document you wish to work with.

DataSize: IGR_LONG

Indicates the size of the buffer pointed to by Data.

Destructor: Pointer

Optional function pointer that will be called when the stream object is closed, giving your application the ability to free the memory buffer or perform other cleanup routines. Specify NULL if unused.

The destructor must take the following form:

```
void __cdecl Destruct(void *data);
```

Stream: Pointer to an IGR_Stream pointer

A system allocated memory stream structure will be returned.

It is the caller's responsibility to free the stream object by calling:

```
stream->Close()
```

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```
void __cdecl FreeMyBuffer(void *p)
{
    delete[] p;
}

Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType;
IGR_Stream *pStream;

if (IGR_Make_Stream_From_Memory(myBuffer, myBufferSize, &FreeMyBuffer, &pStream,
    &ISYSError) == IGR_OK)
{
    if (IGR_Get_Stream_Type(pStream, &Capabilities, &DocType, &ISYSError) == IGR_OK)
    {
        if (DocType == 25)
        {
            // Document is an MS Word document
        }
    }
    pStream->Close(pStream);
}
```

SEE ALSO

- [IGR_Open_Stream](#)
- [IGR_Get_Stream_Type](#)
- [IGR_Stream](#)

IGR_Multiplex

The IGR_Multiplex function is used to supply extensible functionality to the Document Filters API. It is used as a general purpose extension mechanism to avoid disturbing the published Document Filters API.

PROTOTYPE

C/C++

```
void IGR_Multiplex(
    IGR_LONG Function,
    size_t* Parameter1,
    size_t* Parameter2,
    Error_Control_Block* ISYSError);
```

PARAMETERS

Function: IGR_LONG

The function to perform as listed in the following Function Chart.

Parameter1: Pointer to size_t

Used for some functions.

Parameter2: Pointer to size_t

Used for some functions.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

FUNCTION CHART

Function	Parameter1	Parameter2	Purpose
IGR_Multi_Set_Code_Page (1)	Codepage	Unused	Specifies the default character set when the character set cannot be determined.
IGR_Multi_Set_Temp_Path (186)	Temp Path	Unused	Specifies the temp path to use.

RETURN VALUE

None

SAMPLE CODE**C/C++**

```
Error_Control_Block ISYSError;  
size_t L1 = 1251, L2 = 0;  
IGR_Multiplex(IGR_Multi_Set_Code_Page, &L1, &L2, &ISYSError);
```

IGR_Open_DDB

Opens a document from an OS based, device-dependant bitmap (Windows-Only).

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Open_DDB(  
    void* bitmap,  
    void* palette,  
    IGR_LONG flags,  
    const IGR_UCS2* options,  
    IGR_LONG* capabilities,  
    IGR_LONG* file_type,  
    IGR_LONG* handle,  
    Error_Control_Block* error  
)
```

PARAMETERS

bitmap: Pointer to void

Provide a handle to the OS bitmap (HBITMAP).

palette: Pointer to void

Provide a handle to an optional OS palette (HPALETTE).

flags: IGR_LONG

Specifies what type of data is returned from subsequent calls to the IGR_Get_Text function. These Open Document Flags affect the verbosity or the format of the extracted data.

options: Pointer to IGR_UCS2

Extended processing options, used when converting the document to HTML. The Open Document Options are expressed as Name=Value with a semicolon delimiter.

capabilities: Pointer to IGR_LONG

Returns the Document Capabilities as a bit-field.

file_type: Pointer to IGR_LONG

Returns the Document Format Code of the document.

handle: Pointer to IGR_LONG

Returns a handle to be used in subsequent calls.

`error: Pointer to Error_Control_Block`

Returns error details if the call fails.

RETURN VALUE

`Success: IGR_LONG`

Returns IGR_OK.

`Failure: IGR_LONG`

Returns one of the possible [IGR_E error codes](#).

IGR_Open_DIB

IGR_Open_DIB opens a document for content extraction or enumeration of sub-documents from an existing in-memory image.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Open_DIB(  
    const struct IGR_Open_DIB_Info* bitmap,  
    IGR_LONG flags,  
    const IGR_UCS2* options,  
    IGR_LONG* capabilities,  
    IGR_LONG* file_type,  
    IGR_LONG* handle,  
    Error_Control_Block* error);
```

PARAMETERS

Bitmap: const struct IGR_Open_DIB_Info

Pointer to a caller-provided IGR_Open_DIB_Info structure that contains details on how to decode the in-memory image.

Flags: IGR_LONG

Specifies what type of data is returned from subsequent calls to the [IGR_Get_Text](#) function. These Open Document Flags affect the verbosity or the format of the extracted data.

Options: Unicode string (UCS2)

Extended processing options, used when converting the document to HTML. The Open Document Options are expressed as Name=Value with a semicolon delimiter.

Capabilities: Pointer to IGR_LONG

Returns the Document Capabilities as a bit field.

DocType: Pointer to IGR_LONG

Returns the Document Format Code of the document.

DocHandle: Pointer to IGR_LONG

Returns a handle to be used in subsequent calls.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

- [IGR_Open_DIB_Info](#)

IGR_Open_Ex

IGR_Open_Ex opens a document for text extraction or enumeration of sub-documents, and takes an [IGR_OPEN_CALLBACK](#) parameter that allows for interaction with the API user during API calls.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Open_Ex(  
    IGR_OPEN_FROM source_type,  
    const void* source,  
    IGR_LONG flags,  
    const IGR_UCS2* options,  
    IGR_LONG* capabilities,  
    IGR_LONG* file_type,  
    void* reserved,  
    IGR_OPEN_CALLBACK callback,  
    void* callback_context,  
    IGR_LONG* handle,  
    Error_Control_Block* error);
```

PARAMETERS

source_type: IGR_OPEN_FROM

The type of source, that can be either a UTF-16 string containing the filename, or an IGR_Stream record. See [IGR_OPEN_FROM](#).

source: Pointer to VOID

The pointer to the source based on the source_type parameter. If the source is an IGR_Stream, the stream MUST remain valid for the life-type of the document handle returned.

flags: IGR_LONG

Specifies what type of data is returned from subsequent calls to the [IGR_Get_Text](#) function. These Open Document Flags affect the verbosity or the format of the extracted data.

options: Unicode string (UCS2)

Extended processing options, used when converting the document to HTML. The Open Document Options are expressed as Name=Value with a semicolon delimiter.

capabilities: Pointer to IGR_LONG

Returns the Document Capabilities as a bit field.

file_type: Pointer to IGR_LONG

Returns the [Document Format Code](#) of the document.

`reserved: Pointer to VOID`

MUST be NULL. Reserved for future use.

`callback: IGR_OPEN_CALLBACK`

[IGR_OPEN_CALLBACK](#) that will be called for specific events while processing the document. MUST remain valid for the lifetime of the document handle returned.

`callback_context: Pointer to VOID`

Contextual information that will be passed back when the callback is called. MUST remain valid for the lifetime of the document handle returned.

`handle: Pointer to IGR_LONG`

Returns a handle to be used in subsequent calls.

`error: Pointer to Error_Control_Block`

Returns error details if the call fails.

RETURN VALUE

`Success: IGR_LONG`

Returns IGR_OK.

`Failure: IGR_LONG`

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
struct OpenExCallbackContext
{
    OpenExCallbackContext();
    bool isCancellationRequested();
};

IGR_LONG IGR_EXPORT OpenExCallback(IGR_OPEN_CALLBACK_ACTION action, void* payload, void* context)
{
    if (action == IGR_OPEN_CALLBACK_ACTION_HEARTBEAT)
    {
        if (OpenExCallbackContext* callbackContext = static_cast<OpenExCallbackContext*>(context))
        {
            if (callbackContext->isCancellationRequested())
                return IGR_CANCELLED;
        }
    }
}
```

```

    }
}
else if (action == IGR_OPEN_CALLBACK_ACTION_PASSWORD)
{
    IGR_Open_Callback_Action_Password*
passwordStruct(static_cast<IGR_Open_Callback_Action_Password*>(payload));

    if (!passwordStruct || passwordStruct->struct_size != sizeof(*passwordStruct))
        return IGR_E_ERROR;

    // Read string from passwordStruct->id to identify which password is being requested

    // Get password string for password being requested

    // Write password string to passwordStruct->password
}
else if (action == IGR_OPEN_CALLBACK_ACTION_LOCALIZE)
{
    IGR_Open_Callback_Action_Localize*
localizeStruct(static_cast<IGR_Open_Callback_Action_Localize*>(payload));

    if (!localizeStruct || localizeStruct->struct_size != sizeof(*localizeStruct))
        return IGR_E_ERROR;

    // Read string from localizeStruct->original to identify the string to be replaced

    // Get replacement string

    // Write replacement string to localizeStruct->replacement
}
return IGR_OK;
}

int main(int argc, char** argv)
{
    Error_Control_Block ISYSError;
    IGR_LONG Capabilities, DocType, DocHandle;
    OpenExCallbackContext CallbackContext;

    IGR_LONG RC = IGR_Open_Ex(IGR_OPEN_FROM_FILENAME_UTF16, _UCS2("TEST.DOC"),
    IGR_BODY_AND_META | IGR_FORMAT_HTML, _UCS2("IMAGEPATH=C:\\Temp"),
    &Capabilities, &DocType, NULL,
    OpenExCallback, &CallbackContext, &DocHandle, &ISYSError);

    if (RC == IGR_OK)
    {
        // Extract document text or sub-documents...
        IGR_Close_File(DocHandle, &ISYSError);
    }
}

```

ADDITIONAL INFORMATION

The call will establish a link to the document and populates a handle. The handle can be used to extract the text by calling [IGR_Get_Text](#), generate page images with [IGR_Open_Page](#), or enumerate and extract the sub-documents by calls to [IGR_Get_Subfile_Entry](#) and [IGR_Extract_Subfile](#) respectively.

The application must call [IGR_Close_File](#) when finished using the document.

If a (non-NULL) [IGR_OPEN_CALLBACK](#) is provided, the callbacks can happen any time from the point of [IGR_Open_Ex](#) being called through to its handle being closed by [IGR_Close_File](#).

See [IGR_OPEN_CALLBACK](#) for more information on the specifics of each callback action.

SEE ALSO

- [IGR_OPEN_CALLBACK](#)
- [IGR_Get_File_Type](#)
- [IGR_Open_File_Ex](#)
- [IGR_Open_Stream_Ex](#)

IGR_Open_File

IGR_Open_File opens a document for content extraction or enumeration of sub-documents.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Open_File(  
    const IGR_UCS2* FileName,  
    IGR_LONG Flags,  
    IGR_LONG* Capabilities,  
    IGR_LONG* DocType,  
    IGR_LONG* DocHandle,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

FileName: Unicode string (UCS2)

Path to the document to be opened.

Flags: IGR_LONG

Specifies what type of data is returned from subsequent calls to the [IGR_Get_Text](#) function. These Open Document Flags affect the verbosity or the format of the extracted data.

Capabilities: Pointer to IGR_LONG

Returns the Document Capabilities as a bit field.

DocType: Pointer to IGR_LONG

Returns the Document Format Code of the document.

DocHandle: Pointer to IGR_LONG

Returns a handle to be used in subsequent calls.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E](#) error codes.

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle;

IGR_LONG RC = IGR_Open_File(_UCS2("TEST.DOC"), IGR_BODY_AND_META, &Capabilities, &DocType, &DocHandle,
&ISYSError);

if (RC == IGR_OK)
{
    // Extract document text or sub-documents...
    IGR_Close_File(DocHandle, &ISYSError);
}
```

ADDITIONAL INFORMATION

The call will establish a link to the document and populates a handle. The handle can be used to extract the text by calling [IGR_Get_Text](#), generate page images with [IGR_Open_Page](#), or enumerate and extract the sub-documents by calls to [IGR_Get_Subfile_Entry](#) and [IGR_Extract_Subfile](#) respectively.

The application must call [IGR_Close_File](#) when finished using the document.

SEE ALSO

- [IGR_Open_File_Ex](#)
- [IGR_Open_Stream](#)
- [IGR_Get_File_Type](#)

IGR_Open_File_Ex

IGR_Open_File_Ex opens a document for content extraction or enumeration of sub-documents **and** controls the output format, including converting the source document to HTML.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Open_File_Ex(  
    const IGR_UCS2* FileName,  
    IGR_LONG Flags,  
    const IGR_UCS2* Options,  
    IGR_LONG* Capabilities,  
    IGR_LONG* DocType,  
    IGR_LONG* DocHandle,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

FileName: Unicode string (UCS2)

Path to the document to be opened.

Flags: IGR_LONG

Specifies what type of data is returned from subsequent calls to the [IGR_Get_Text](#) function. These Open Document Flags affect the verbosity or the format of the extracted data.

Options: Unicode string (UCS2)

Extended processing options, used when converting the document to HTML. The Open Document Options are expressed as Name=Value with a semicolon delimiter.

Capabilities: Pointer to IGR_LONG

Returns the Document Capabilities as a bit field.

DocType: Pointer to IGR_LONG

Returns the Document Format Code of the document.

DocHandle: Pointer to IGR_LONG

Returns a handle to be used in subsequent calls.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE**C/C++**

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle;

IGR_LONG RC = IGR_Open_File_Ex(_UCS2("TEST.DOC"), IGR_BODY_AND_META | IGR_FORMAT_HTML,
    _UCS2("IMAGEPATH=C:\\Temp"), &Capabilities, &DocType, &DocHandle, &ISYSError);

if (RC == IGR_OK)
{
    // Extract document text or sub-documents...
    IGR_Close_File(DocHandle, &ISYSError);
}
```

ADDITIONAL INFORMATION

The call will establish a link to the document and populates a handle. The handle can be used to extract the text by calling [IGR_Get_Text](#), generate page images with [IGR_Open_Page](#), or enumerate and extract the sub-documents by calls to [IGR_Get_Subfile_Entry](#) and [IGR_Extract_Subfile](#) respectively.

The application must call [IGR_Close_File](#) when finished using the document.

SEE ALSO

- [IGR_Open_File](#)
- [IGR_Get_File_Type](#)

IGR_Open_Page

IGR_Open_Page gives access to page specific content for documents opened using the IGR_FORMAT_IMAGE flag, including page words, images, and structured XML.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Open_Page(  
    IGR_LONG DocHandle,  
    IGR_LONG PageIndex,  
    HPAGE* PageHandle,  
    Error_Control_Block* error);
```

PARAMETERS

DocHandle: IGR_LONG

Handle to a document, opened by a call to [IGR_Open_File](#), [IGR_Open_File_Ex](#), [IGR_Open_Stream](#) or [IGR_Open_Stream_Ex](#).

PageIndex: IGR_LONG

0-based page number to the page to open.

PageHandle: Pointer to HPAGE

Returns a handle to be used in subsequent page calls.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;  
IGR_LONG Capabilities, DocType, DocHandle, PageCount;
```

```
HPAGE PageHandle;

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &ISYSSError);
if (RC == IGR_OK)
{
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSSError) == IGR_OK)
    {
        for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)
        {
            if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSSError) == IGR_OK)
            {
                // Process Page Element

                IGR_Close_Page(PageHandle, &ISYSSError);
            }
        }
    }
    IGR_Close_File(DocHandle, &ISYSSError);
}
```

ADDITIONAL INFORMATION

The call will load resources associated with the page that can then be used in calls to [IGR_Get_Page_Word_Count](#), [IGR_Get_Page_Words](#), [IGR_Get_Page_Dimensions](#), [IGR_Get_Page_Text](#), and [IGR_Render_Page](#).

The application must call [IGR_Close_Page](#) when finished using the page. All pages must be closed before calling [IGR_Close_File](#).

SEE ALSO

- [IGR_Get_Page_Word_Count](#)
- [IGR_Get_Page_Words](#)
- [IGR_Get_Page_Dimensions](#)
- [IGR_Get_Page_Text](#)
- [IGR_Render_Page](#)
- [IGR_Close_Page](#)

IGR_Open_Stream

IGR_Open_Stream opens a document from a stream object for content extraction or enumeration of contained sub-documents.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Open_Stream(  
    IGR_Stream *Stream,  
    IGR_LONG Flags,  
    IGR_LONG* Capabilities,  
    IGR_LONG* DocType,  
    IGR_LONG* DocHandle,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Stream: Pointer to an IGR_Stream
(a stream object)

The stream can be either user implemented, or created using the [IGR_Make_Stream_From_File](#) and [IGR_Make_Stream_From_Memory](#) utility functions.

Flags: IGR_LONG

Specifies what type of data is returned from subsequent calls to the [IGR_Get_Text](#) function. These Open Document Flags affect the verbosity or the format of the extracted data.

Capabilities: Pointer to IGR_LONG

Returns the Document Capabilities as a bit field.

DocType: Pointer to IGR_LONG

Returns the [Document Format Code](#) of the document.

DocHandle: Pointer to IGR_LONG

Returns a handle to be used in subsequent calls.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle;

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_BODY_AND_META, &Capabilities, &DocType, &DocHandle, &ISYSError);

if (RC == IGR_OK)
{
    // Extract document text or sub-documents...
    IGR_Close_File(DocHandle, &ISYSError);
}
```

ADDITIONAL INFORMATION

The call will establish a link to the document and populates a handle. The handle can be used to extract the text by calling [IGR_Get_Text](#), generate page images with [IGR_Open_Page](#), or enumerate and extract the sub-documents by calls to [IGR_Get_Subfile_Entry](#) and [IGR_Extract_Subfile](#) respectively.

The application must call [IGR_Close_File](#) when finished using the document.

SEE ALSO

- [IGR_Open_File](#)
- [IGR_Get_File_Type](#)
- [IGR_Stream](#)

IGR_Open_Stream_Ex

IGR_Open_Stream_Ex opens a document from a stream object for content extraction or enumeration of contained sub-documents **and** controls the output format.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Open_Stream_Ex(  
    IGR_Stream *Stream,  
    IGR_LONG Flags,  
    const IGR_UCS2* Options,  
    IGR_LONG* Capabilities,  
    IGR_LONG* DocType,  
    IGR_LONG* DocHandle,  
    Error_Control_Block* ISYSError);
```

PARAMETERS

Stream: Pointer to an IGR_Stream
(a stream object)

The stream can be either user implemented, or created using the [IGR_Make_Stream_From_File](#) and [IGR_Make_Stream_From_Memory](#) utility functions.

Flags: IGR_LONG

Specifies what type of data is returned from subsequent calls to the [IGR_Get_Text](#) function. These Open Document Flags affect the verbosity or the format of the extracted data.

Options: Unicode string (UCS2)

Extended processing options, used when converting the document. The Open Document Options are expressed as Name=Value with a semicolon delimiter.

Capabilities: Pointer to IGR_LONG

Returns the Document Capabilities as a bit field.

DocType: Pointer to IGR_LONG

Returns the [Document Format Code](#) of the document.

DocHandle: Pointer to IGR_LONG

Returns a handle to be used in subsequent calls.

ISYSError: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle;

IGR_LONG RC = IGR_Open_Stream_Ex(pStream, IGR_BODY_AND_META | IGR_FORMAT_HTML,
    _UCS2("IMAGEPATH=C:\\Temp"), &Capabilities, &DocType, &DocHandle, &ISYSError);

if (RC == IGR_OK)
{
    // Extract document text or sub-documents...
    IGR_Close_File(DocHandle, &ISYSError);
}
```

ADDITIONAL INFORMATION

The call will establish a link to the document and populates a handle. The handle can be used to extract the text by calling [IGR_Get_Text](#), generate page images with [IGR_Open_Page](#), or enumerate and extract the sub-documents by calls to [IGR_Get_Subfile_Entry](#) and [IGR_Extract_Subfile](#) respectively.

The application must call [IGR_Close_File](#) when finished using the document.

SEE ALSO

- [IGR_Open_File](#)
- [IGR_Get_File_Type](#)
- [IGR_Stream](#)

IGR_Redact_Page_Text

IGR_Redact_Page_Text removes the words and blacks out the location for the specified range from the page.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Redact_Page_Text(  
    HPAGE page,  
    IGR_LONG from,  
    IGR_LONG to,  
    Error_Control_Block* Error);
```

PARAMETERS

Page: HPAGE

Handle to a page, opened by a call to [IGR_Open_Page](#).

From: IGR_LONG

The index of the first word to redact.

To: IGR_LONG

The index to the last word to redact.

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

It should be assumed that redacted content will not persist between closing and re-opening a page. To create a redacted Image, PDF or HTML file, open a page, perform the redaction and render the page to a canvas before closing it.

The API allows for redacting single words or a run or range of words. When redacting a range, whitespace between the words will also be redacted.

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle, PageCount;
HPAGE PageHandle;

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &ISYSError);
if (RC == IGR_OK)
{
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSError) == IGR_OK)
    {
        for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)
        {
            if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSError) == IGR_OK)
            {
                IGR_Redact_Page_Text(PageHandle, 0, 15, &ISYSError);

                HCANVAS CanvasHandle;
                if (IGR_Make_Output_Canvas(IGR_DEVICE_IMAGE_PNG, L"page.png",
                    &CanvasHandle, &ISYSError) == IGR_OK)
                {
                    IGR_Render_Page(PageHandle, CanvasHandle, &ISYSError);
                    IGR_Close_Canvas(CanvasHandle, &ISYSError);
                }
                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
    IGR_Close_File(DocHandle, &ISYSError);
}
```

SEE ALSO

- [IGR_Open_Page](#)
- [IGR_Make_Output_Canvas](#)

IGR_Render_Page

IGR_Render_Page draws the page content into the specified output canvas.

PROTOTYPE

C/C++

```
IGR_LONG IGR_Render_Page(  
    HPAGE Page,  
    HCANVAS Canvas,  
    Error_Control_Block* Error);
```

PARAMETERS

Page: HPAGE

Handle to a page, opened by a call to [IGR_Open_Page](#).

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SAMPLE CODE

C/C++

```
Error_Control_Block ISYSError;  
IGR_LONG Capabilities, DocType, DocHandle, PageCount;  
HPAGE PageHandle;  
  
IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &ISYSError);  
if (RC == IGR_OK)  
{
```

```
if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSLError) == IGR_OK)
{
    for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)
    {
        if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSLError) == IGR_OK)
        {
            HCANVAS CanvasHandle;
            if (IGR_Make_Output_Canvas(IGR_DEVICE_IMAGE_PNG, L"page.png",
                &CanvasHandle, &ISYSLError) == IGR_OK)
            {
                IGR_Render_Page(PageHandle, CanvasHandle, &ISYSLError);
                IGR_Close_Canvas(CanvasHandle, &ISYSLError);
            }
            IGR_Close_Page(PageHandle, &ISYSLError);
        }
    }
}
IGR_Close_File(DocHandle, &ISYSLError);
}
```

SEE ALSO

- [IGR_Open_Page](#)
- [IGR_Make_Output_Canvas](#)
- [IGR_Render_Page_Ex](#)

IGR_Render_Page_Ex

IGR_Render_Page_Ex draws the page content into the specified output canvas

PROTOTYPE

C/C++

```
IGR_LONG IGR_Render_Page_Ex(  
    HPAGE Page,  
    HCANVAS Canvas,  
    IGR_UCS2* Options,  
    const struct IGR_Render_Page_Properties* Properties,  
    Error_Control_Block* Error);
```

PARAMETERS

Page: HPAGE

Handle to a page, opened by a call to [IGR_Open_Page](#).

Canvas: HCANVAS

Handle to a canvas, opened by a call to [IGR_Make_Output_Canvas](#).

Options: Unicode string (UCS2)

Extended processing options used when converting the page. The [Open Document Options](#) are expressed as Name=Value with a semicolon delimiter.

Properties: Pointer to const IGR_Render_Page_Properties

Page specific render properties

Error: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

ADDITIONAL INFORMATION

Note The drawing API is available for bitmap and PDF outputs only. Drawing onto an HTML5 output is not supported.

SAMPLE CODE**C/C++**

```

Error_Control_Block ISYSError;
IGR_LONG Capabilities, DocType, DocHandle, PageCount;
HPAGE PageHandle;
IGR_Render_Page_Properties RenderProperties;
RenderProperties.struct_size = sizeof(IGR_Render_Page_Properties);

IGR_LONG RC = IGR_Open_Stream(pStream, IGR_FORMAT_IMAGE, &Capabilities, &DocType, &DocHandle, &ISYSError);
if (RC == IGR_OK)
{
    if (IGR_Get_Page_Count(DocHandle, &PageCount, &ISYSError) == IGR_OK)
    {
        for (IGR_LONG PageIndex = 0; PageIndex < PageCount; PageIndex++)
        {
            if (IGR_Open_Page(DocHandle, PageIndex, &PageHandle, &ISYSError) == IGR_OK)
            {
                HCANVAS CanvasHandle;
                if (IGR_Make_Output_Canvas(IGR_DEVICE_IMAGE_PNG, L"page.png",
                    &CanvasHandle, &ISYSError) == IGR_OK)
                {
                    IGR_Render_Page_Ex(PageHandle, CanvasHandle,
                        UCS2("WATERMARK=EXAMPLE;GRAPHIC_EFFECT=GRAYSCALE"),
                        &RenderProperties, &ISYSError);
                    IGR_Close_Canvas(CanvasHandle, &ISYSError);
                }
                IGR_Close_Page(PageHandle, &ISYSError);
            }
        }
    }
    IGR_Close_File(DocHandle, &ISYSError);
}

```

SEE ALSO

- [IGR_Open_Page](#)
- [IGR_Make_Output_Canvas](#)
- [IGR_Render_Page](#)
- [IGR_Render_Page_Properties](#)

IGR_Subfiles_Close

`IGR_Subfiles_Close` releases the resources associated with an enumeration generated by `IGR_Get_Subfiles_Enumerator` or a similar function. It is imperative to free enumerators before closing the owning document.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Subfiles_Close(  
    HSUBFILES handle,  
    Error_Control_Block* error);
```

PARAMETERS

`enumerator`: HSUBFILES

Handle to an enumerator opened by `IGR_Get_Subfiles_Enumerator`.

`error`: Pointer to `Error_Control_Block`

Returns error details if the call fails.

RETURN VALUE

`Success`: IGR_LONG

Returns IGR_OK.

`Failure`: IGR_LONG

Returns one of the possible `IGR_E` error codes.

SAMPLE CODE**C++**

```
1 Error_Control_Block error;
2 HSUBFILES subs;
3 std::array<IGR_UCS2, 4096> id;
4 std::array<IGR_UCS2, 1024> name;
5 IGR_LONGLONG date, size;
6 IGR_RETURN_CODE res;
7
8 if ((res = IGR_Get_Subfiles_Enumerator(docHandle, &subs, &error)) == IGR_OK)
9 {
10     while ((res = IGR_Subfiles_Next(subs, &id[0], &name[0], &date, &size, &error)) == IGR_OK)
11     {
12         res = IGR_Extract_Subfile(docHandle, id, _UCS2("TEMP.DAT"), &error);
13         if (res != IGR_OK)
14             // ReportError(res);
15         else
16             // DoSomethingWithTheFile("TEMP.DAT", id, name);
17     }
18     IGR_Subfiles_Close(subs, &error);
19 }
```

SEE ALSO

- [IGR_Get_Subfiles_Enumerator](#)
- [IGR_Subfiles_Close](#)

IGR_Subfiles_Next

`IGR_Subfiles_Next` iterates through the sub-documents within a subfile enumeration obtained from `IGR_Get_Subfiles_Enumerator` or a similar function.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Subfiles_Next(  
    HSUBFILES handle,  
    IGR_UCS2* id,  
    IGR_UCS2* name,  
    IGR_LONGLONG* date,  
    IGR_LONGLONG* size,  
    Error_Control_Block* error);
```

PARAMETERS

`handle: IGR_LONG`

Is a handle to a file, opened by a call to [IGR_Open_File](#).

`id: Unicode string (UCS2)`

Application allocated memory block of 8192 bytes that will be filled with up to 4096 Unicode characters that specify the unique id of the next sub-document.

`name: Unicode string (UCS2)`

Application allocated memory block of 2048 bytes that will be filled with up to 1024 Unicode characters that specify the name of the sub-document.

`date: Pointer to INT64`

Returns the date and time of the sub-document in [FILETIME](#) format.

`size: Pointer to INT64`

Returns the size in bytes of the sub-document.

`error: Pointer to Error_Control_Block`

Returns error details if the call fails.

RETURN VALUE

`Success: IGR_LONG`

Returns IGR_OK.

Success and the end of the document was reached: IGR_LONG

Returns IGR_NO_MORE.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C++

```

1 Error_Control_Block error;
2 HSUBFILES subs;
3 std::array<IGR_UCS2, 4096> id;
4 std::array<IGR_UCS2, 1024> name;
5 IGR_LONGLONG date, size;
6 IGR_RETURN_CODE res;
7
8 if ((res = IGR_Get_Subfiles_Enumerator(docHandle, &subs, &error)) == IGR_OK)
9 {
10     while ((res = IGR_Subfiles_Next(subs, &id[0], &name[0], &date, &size, &error)) == IGR_OK)
11     {
12         res = IGR_Extract_Subfile(docHandle, id, _UCS2("TEMP.DAT"), &error);
13         if (res != IGR_OK)
14             // ReportError(res);
15         else
16             // DoSomethingWithTheFile("TEMP.DAT", id, name);
17     }
18     IGR_Subfiles_Close(subs, &error);
19 }
```

ADDITIONAL INFORMATION

The previously opened document must have the IGR_FILE_SUPPORTS_SUBFILES capability.

After a successful call to [IGR_Get_Subfiles_Enumerator](#), each call to [IGR_Subfiles_Next](#) will retrieve information about the next sub-document contained in the compound document, referenced by DocHandle. To traverse all the sub-documents, the application will need to call this method in a loop until IGR_NO_MORE is returned.

Note that the null-terminating character will also be copied to the id and name parameters. The name parameter could be an empty string if the id of the sub-document is not available. If the function succeeds, the id is guaranteed not to be empty and will be unique among all traversed sub-documents retrieved from the document. The returned id can be used in a call to [IGR_Extract_Subfile](#) to save the binary content of the sub-document to disk.

If the date of the sub-document is not available, the parameter FileDate will be set to 0, otherwise it will be populated in FILETIME format.

If the size of the sub-document is not available, the parameter FileSize will be set to 0.

SEE ALSO

- [IGR_Get_Subfiles_Enumerator](#)
- [IGR_Subfiles_Close](#)
- [IGR_Subfiles_Next_Ex](#)

IGR_Subfiles_Next_Ex

`IGR_Subfiles_Next_Ex` iterates through the sub-documents within a subfile enumeration obtained from `IGR_Get_Subfiles_Enumerator` or a similar function.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Subfiles_Next_Ex(  
    HSUBFILES enumerator,  
    struct IGR_Subfile_Info* result,  
    Error_Control_Block* error);
```

PARAMETERS

`enumerator`: HSUBFILES

Handle to an enumerator opened by `IGR_Get_Subfiles_Enumerator`.

`result`: Pointer to `IGR_Subfile_Info`

Pointer to a user allocated `IGR_Subfile_Info` to be populated with the subfile information.

`error`: Pointer to `Error_Control_Block`

Returns error details if the call fails.

RETURN VALUE

`Success`: IGR_LONG

Returns IGR_OK.

`Failure`: IGR_LONG

Returns one of the possible `IGR_E` error codes.

SAMPLE CODE

C/C++

```

1 Error_Control_Block error = { 0 };
2 IGR_Subfile_Info subfile = { 0 };
3 IGR_RETURN_CODE res;
4 HSUBFILES subs;
5
6 std::array<IGR_UCS2, 4096> id;
7 std::array<IGR_UCS2, 1024> name;
8 std::array<IGR_UCS2, 4096> comment;
9
10 subfile.struct_size = sizeof(subfile);
11 subfile.id = &id[0];
12 subfile.id_size = id.size();
13
14 subfile.name = &name[0];
15 subfile.name_size = name.size();
16
17 subfile.comment = &comment[0];
18 subfile.comment_size = comment.size();
19
20 if ((res = IGR_Get_Subfiles_Enumerator(docHandle, &subs, &error)) == IGR_OK)
21 {
22     while ((res = IGR_Subfiles_Next_Ex(subs, &subfile, &error)) == IGR_OK)
23     {
24         res = IGR_Extract_Subfile(DocHandle, ID, _UCS2("TEMP.DAT"), &ISYS_Error);
25         if (res != IGR_OK)
26             // ReportError(rc);
27         else
28             // DoSomethingWithTheFile("TEMP.DAT", ID, Name);
29     }
30     IGR_Subfiles_Close(subs, &error);
31 }

```

ADDITIONAL INFORMATION

The previously opened document must have the IGR_FILE_SUPPORTS_SUBFILES capability.

After a successful call to [IGR_Get_Subfiles_Enumerator](#), each call to [IGR_Get_Subfile_Entry_Ex](#) will retrieve information about the next sub-document contained in the compound document, referenced by DocHandle. To traverse all the sub-documents, the application will need to call this method in a loop until IGR_NO_MORE is returned.

Note that the null-terminating character will also be copied to the result.ID and result.Name parameters. The Name parameter could be an empty string if the ID of the sub-document is not available. If the function succeeds, the ID is guaranteed not to be empty and will be unique among all traversed sub-documents retrieved from the document. The returned ID can be used in a call to [IGR_Extract_Subfile](#) to save the binary content of the sub-document to disk.

If the date of the sub-document is not available, the parameter FileDate will be set to 0, otherwise it will be populated in FILETIME format.

If the size of the sub-document is not available, the parameter FileSize will be set to 0.

SEE ALSO

- [IGR_Get_Subfile_Entry](#)
- [IGR_Get_Subfiles_Enumerator](#)
- [IGR_Subfiles_Next](#)

IGR_Subfiles_Reset

`IGR_Subfiles_Reset` resets an enumeration to its initial state, positioning it at the beginning. Subsequently, invoke `IGR_Subfiles_Next` to commence reprocessing the document. Notably, invoking `IGR_Subfiles_Reset` is unnecessary for the initial iteration.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Subfiles_Reset(  
    HSUBFILES enumerator,  
    Error_Control_Block* error);
```

PARAMETERS

`enumerator`: HSUBFILES

Handle to an enumerator opened by [IGR_Get_Subfiles_Enumerator](#).

`error`: Pointer to Error_Control_Block

Returns error details if the call fails.

RETURN VALUE

`Success`: IGR_LONG

Returns IGR_OK.

`Failure`: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

- [IGR_Get_Subfiles_Enumerator](#)
- [IGR_Subfiles_Close](#)

IGR_Text_Compare_Close

Closes the text comparison enumerator and frees associated resources.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Text_Compare_Close(  
    IGR_HTEXTCOMPARE enumerator,  
    Error_Control_Block* error  
)
```

PARAMETERS

enumerator: IGR_HTEXTCOMPARE

The text comparison enumerator to close.

error: Pointer to Error_Control_Block

Contains any error text.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C++

```

1  #include <iostream>
2  #include <string>
3  #include "DocumentFiltersObjects.h"
4
5  int main() {
6      try {
7          // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8          Hyland::DocFilters::Api api;
9
10         // Initialize the DocumentFilters object
11         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
12
13         // Open the documents using OpenExtractor in Paginated mode
14         Hyland::DocFilters::Extractor doc1 = api.OpenExtractor("original.docx",
Hyland::DocFilters::OpenMode::Paginated);
15         Hyland::DocFilters::Extractor doc2 = api.OpenExtractor("revision.docx",
Hyland::DocFilters::OpenMode::Paginated);
16
17         // Compare the documents
18         Hyland::DocFilters::CompareResults compare = doc1.Compare(doc2);
19
20         // Iterate through the differences
21         while (compare.MoveNext()) {
22             Hyland::DocFilters::CompareResultDifference diff = compare.GetCurrent();
23
24             // Work with the diff object
25             // ... Your code to process the differences goes here ...
26
27             // Example: Print the type and text of the difference
28             std::wcout << L"Difference Type: " << (int)diff.getType() << std::endl;
29             std::wcout << L"Difference Text: " << diff.getText() << std::endl;
30         }
31
32         // Close the documents (not explicitly shown in the C# sample, but good practice)
33         doc1.Close();
34         doc2.Close();
35
36     } catch (const std::exception& ex) {
37         std::cerr << "Error: " << ex.what() << std::endl;
38         return 1; // Indicate an error occurred
39     }
40
41     return 0;
42 }

```

SEE ALSO

- [IGR_Text_Compare_Documents](#)
- [IGR_Text_Compare_Elements](#)
- [IGR_Text_Compare_Pages](#)

IGR_Text_Compare_Difference_Dispose

Disposes of a difference result returned by the text comparison enumerator.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Text_Compare_Difference_Dispose(  
    struct IGR_Compare_Documents_Difference* result,  
    Error_Control_Block* error  
)
```

PARAMETERS

result: Pointer to IGR_Compare_Documents_Difference

The difference result to dispose of.

error: Pointer to Error_Control_Block

Contains any error text.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C++

```

1  #include <iostream>
2  #include <string>
3  #include "DocumentFiltersObjects.h"
4
5  int main() {
6      try {
7          // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8          Hyland::DocFilters::Api api;
9
10         // Initialize the DocumentFilters object
11         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
12
13         // Open the documents using OpenExtractor in Paginated mode
14         Hyland::DocFilters::Extractor doc1 = api.OpenExtractor("original.docx",
Hyland::DocFilters::OpenMode::Paginated);
15         Hyland::DocFilters::Extractor doc2 = api.OpenExtractor("revision.docx",
Hyland::DocFilters::OpenMode::Paginated);
16
17         // Compare the documents
18         Hyland::DocFilters::CompareResults compare = doc1.Compare(doc2);
19
20         // Iterate through the differences
21         while (compare.MoveNext()) {
22             Hyland::DocFilters::CompareResultDifference diff = compare.GetCurrent();
23
24             // Work with the diff object
25             // ... Your code to process the differences goes here ...
26
27             // Example: Print the type and text of the difference
28             std::wcout << L"Difference Type: " << (int)diff.getType() << std::endl;
29             std::wcout << L"Difference Text: " << diff.getText() << std::endl;
30         }
31
32         // Close the documents (not explicitly shown in the C# sample, but good practice)
33         doc1.Close();
34         doc2.Close();
35
36     } catch (const std::exception& ex) {
37         std::cerr << "Error: " << ex.what() << std::endl;
38         return 1; // Indicate an error occurred
39     }
40
41     return 0;
42 }

```

SEE ALSO

- [IGR_Text_Compare_Documents](#)
- [IGR_Text_Compare_Pages](#)
- [IGR_Text_Compare_Elements](#)
- [IGR_Compare_Documents_Difference](#) data type

IGR_Text_Compare_Documents

Compares two documents and returns an enumerator for iterating through the differences.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Text_Compare_Documents(  
    const struct IGR_Text_Compare_Document_Source* doc1,  
    const struct IGR_Text_Compare_Document_Source* doc2,  
    const struct IGR_Text_Compare_Settings* settings,  
    IGR_HTEXTCOMPARE* enumerator,  
    Error_Control_Block* error  
)
```

PARAMETERS

doc1: Pointer to IGR_Text_Compare_Document_Source

The first document to compare.

doc2: Pointer to IGR_Text_Compare_Document_Source

The second document to compare.

settings: Pointer to IGR_Text_Compare_Settings

Settings for text comparison.

enumerator: Pointer to IGR_HTEXTCOMPARE

Pointer to store the enumerator for iterating through differences.

error: Pointer to Error_Control_Block

Contains any error text.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C++

```

1  #include <iostream>
2  #include <string>
3  #include "DocumentFiltersObjects.h"
4
5  int main() {
6      try {
7          // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8          Hyland::DocFilters::Api api;
9
10         // Initialize the DocumentFilters object
11         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
12
13         // Open the documents using OpenExtractor in Paginated mode
14         Hyland::DocFilters::Extractor doc1 = api.OpenExtractor("original.docx",
Hyland::DocFilters::OpenMode::Paginated);
15         Hyland::DocFilters::Extractor doc2 = api.OpenExtractor("revision.docx",
Hyland::DocFilters::OpenMode::Paginated);
16
17         // Compare the documents
18         Hyland::DocFilters::CompareResults compare = doc1.Compare(doc2);
19
20         // Iterate through the differences
21         while (compare.MoveNext()) {
22             Hyland::DocFilters::CompareResultDifference diff = compare.GetCurrent();
23
24             // Work with the diff object
25             // ... Your code to process the differences goes here ...
26
27             // Example: Print the type and text of the difference
28             std::wcout << L"Difference Type: " << (int)diff.getType() << std::endl;
29             std::wcout << L"Difference Text: " << diff.getText() << std::endl;
30         }
31
32         // Close the documents (not explicitly shown in the C# sample, but good practice)
33         doc1.Close();
34         doc2.Close();
35
36     } catch (const std::exception& ex) {
37         std::cerr << "Error: " << ex.what() << std::endl;
38         return 1; // Indicate an error occurred
39     }
40
41     return 0;
42 }

```

SEE ALSO

- [IGR_Text_Compare_Close](#)
- [IGR_Text_Compare_Difference_Dispose](#)
- [IGR_Text_Compare_Elements](#)
- [IGR_Text_Compare_Next](#)
- [IGR_Text_Compare_Pages](#)

- [IGR_Text_Compare_Document_Source](#) data type
- [IGR_Text_Compare_Settings](#) data type

IGR_Text_Compare_Elements

Compares elements within pages of documents and returns an enumerator for iterating through the differences.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Text_Compare_Elements(  
    IGR_HPAGE page1,  
    const struct IGR_Page_Element* page1_ele_root,  
    IGR_HPAGE page2,  
    const struct IGR_Page_Element* page2_ele_root,  
    const struct IGR_Text_Compare_Settings* settings,  
    IGR_HTEXTCOMPARE* enumerator,  
    Error_Control_Block* error  
)
```

PARAMETERS

page1: IGR_HPAGE

The first page containing elements to compare.

page1_ele_root: Pointer to IGR_Page_Element

The root element of the first page to start comparison.

page2: IGR_HPAGE

The second page containing elements to compare.

page2_ele_root: Pointer to IGR_Page_Element

The root element of the second page to start comparison.

settings: Pointer to IGR_Text_Compare_Settings

Settings for text comparison.

enumerator: Pointer to IGR_HTEXTCOMPARE

Pointer to store the enumerator for iterating through differences.

error: Pointer to Error_Control_Block

Contains any error text.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C++

```

1  #include <DocumentFilters.h>
2  #include <string.h>
3  #include <string>
4
5  #define UCS2(src) reinterpret_cast<const IGR_UCS2 *>(std::u16string(src).c_str())
6
7  bool findSectionElement(IGR_HPAGE page, IGR_Page_Element& res)
8  {
9      struct helper_t
10     {
11         IGR_Page_Element result;
12
13         helper_t() : result()
14         {
15         }
16         static IGR_LONG CALLBACK cb(IGR_HPAGE page, const struct IGR_Page_Element* item, void* context)
17         {
18             if (item->type == IGR_PAGE_ELEMENT_TYPE_SECTION)
19             {
20                 static_cast<helper_t*>(context)->result = *item;
21                 return IGR_NO_MORE;
22             }
23             return IGR_OK;
24         }
25     };
26
27     Error_Control_Block ecb = { 0 };
28     helper_t h;
29
30     if (IGR_Enum_Page_Elements(page, NULL, 0, 0xffff, helper_t::cb, &h, &ecb) == IGR_OK)
31     {
32         res = h.result;
33         return true;
34     }
35     return false;
36 }
37
38 int main(int argc, char **argv)
39 {
40     Instance_Status_Block isb = {0};
41     Error_Control_Block ecb = {0};
42     IGR_SHORT df;
43     IGR_LONG caps, type, pageCount = 0;
44     IGR_HDOC doc1_handle = 0, doc2_handle = 0;
45     IGR_HPAGE doc1_page1_handle = 0, doc2_page1_handle = 0;
46     IGR_HTEXTCOMPARE compare = 0;
47     IGR_Page_Element doc1_page1_root = { sizeof(IGR_Page_Element) };
48     IGR_Page_Element doc2_page1_root = { sizeof(IGR_Page_Element) };
49
50     strncpy(isb.Licensee_ID1, "YOUR_LICENSE_KEY_HERE", sizeof(isb.Licensee_ID1) - 1);
51     Init_Instance(0, ".", &isb, &df, &ecb);
52
53     if ((res = IGR_Open_File_Ex(UCS2(u"original.doc"), IGR_FORMAT_IMAGE, UCS2(u""), &caps, &type,
54     &doc1_handle, &ecb)) == IGR_OK
55     && (res = IGR_Open_File_Ex(UCS2(u"revised.doc"), IGR_FORMAT_IMAGE, UCS2(u""), &caps, &type,
56     &doc2_handle, &ecb)) == IGR_OK

```



```

55     && (res = IGR_Open_Page(doc1_handle, 0, &doc1_page1_handle, &ecb)) == IGR_OK
56     && (res = IGR_Open_Page(doc2_handle, 0, &doc2_page1_handle, &ecb)) == IGR_OK
57     && findSectionElement(doc1_page1_handle, doc1_page1_root)
58     && findSectionElement(doc2_page1_handle, doc2_page1_root))
59
60     {
61         IGR_Text_Compare_Settings settings = { sizeof(IGR_Text_Compare_Settings) };
62         IGR_Compare_Documents_Difference diff = { sizeof(IGR_Compare_Documents_Difference) };
63
64         if ((ret = IGR_Text_Compare_Elements(doc1_page1_handle, &doc1_page1_root, doc2_page1_handle,
65         &doc2_page1_root, &settings, &compare, &ecb)) == IGR_OK)
66         {
67             while (IGR_Text_Compare_Next(compare, &diff, &ecb) == IGR_OK)
68             {
69                 // ... work with diff object
70
71                 IGR_Text_Compare_Difference_Dispose(&diff, &ecb);
72             }
73
74             IGR_Text_Compare_Close(compare, &ecb);
75         }
76
77         if (doc1_page1_handle)
78             IGR_Close_Page(doc1_page1_handle, &ecb);
79         if (doc2_page1_handle)
80             IGR_Close_Page(doc2_page1_handle, &ecb);
81         if (doc1_handle)
82             IGR_Close_File(doc1_handle, &ecb);
83         if (doc2_handle)
84             IGR_Close_File(doc2_handle, &ecb);
85         return 0;
86     }

```

SEE ALSO

- [IGR_Text_Compare_Close](#)
- [IGR_Text_Compare_Difference_Dispose](#)
- [IGR_Text_Compare_Documents](#)
- [IGR_Text_Compare_Next](#)
- [IGR_Text_Compare_Pages](#)
- [IGR_Text_Compare_Settings](#) data type
- [IGR_Page_Element](#) data type

IGR_Text_Compare_Next

Retrieves the next difference from the text comparison enumerator.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Text_Compare_Next(  
    IGR_HTEXTCOMPARE enumerator,  
    struct IGR_Compare_Documents_Difference* result,  
    Error_Control_Block* error  
)
```

PARAMETERS

enumerator: IGR_HTEXTCOMPARE

The text comparison enumerator.

result: Pointer to IGR_Compare_Documents_Difference

Pointer to store the next difference.

error: Pointer to Error_Control_Block

Contains any error text.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C++

```

1  #include <iostream>
2  #include <string>
3  #include "DocumentFiltersObjects.h"
4
5  int main() {
6      try {
7          // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8          Hyland::DocFilters::Api api;
9
10         // Initialize the DocumentFilters object
11         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
12
13         // Open the documents using OpenExtractor in Paginated mode
14         Hyland::DocFilters::Extractor doc1 = api.OpenExtractor("original.docx",
Hyland::DocFilters::OpenMode::Paginated);
15         Hyland::DocFilters::Extractor doc2 = api.OpenExtractor("revision.docx",
Hyland::DocFilters::OpenMode::Paginated);
16
17         // Compare the documents
18         Hyland::DocFilters::CompareResults compare = doc1.Compare(doc2);
19
20         // Iterate through the differences
21         while (compare.MoveNext()) {
22             Hyland::DocFilters::CompareResultDifference diff = compare.GetCurrent();
23
24             // Work with the diff object
25             // ... Your code to process the differences goes here ...
26
27             // Example: Print the type and text of the difference
28             std::wcout << L"Difference Type: " << (int)diff.getType() << std::endl;
29             std::wcout << L"Difference Text: " << diff.getText() << std::endl;
30         }
31
32         // Close the documents (not explicitly shown in the C# sample, but good practice)
33         doc1.Close();
34         doc2.Close();
35
36     } catch (const std::exception& ex) {
37         std::cerr << "Error: " << ex.what() << std::endl;
38         return 1; // Indicate an error occurred
39     }
40
41     return 0;
42 }

```

SEE ALSO

- [IGR_Text_Compare_Documents](#)
- [IGR_Text_Compare_Pages](#)
- [IGR_Text_Compare_Elements](#)
- [IGR_Compare_Documents_Difference](#)

IGR_Text_Compare_Pages

Compares two pages within documents and returns an enumerator for iterating through the differences.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Text_Compare_Pages(  
    IGR_HPAGE page1,  
    const struct IGR_FRect* page1_margins,  
    IGR_HPAGE page2,  
    const struct IGR_FRect* page2_margins,  
    const struct IGR_Text_Compare_Settings* settings,  
    IGR_HTEXTCOMPARE* enumerator,  
    Error_Control_Block* error  
)
```

PARAMETERS

page1: IGR_HPAGE

The first page to compare.

page1_margins: Pointer to IGR_FRect

The margins of the first page.

page2: IGR_HPAGE

The second page to compare.

page2_margins: Pointer to IGR_FRect

The margins of the second page.

settings: Pointer to IGR_Text_Compare_Settings

Settings for text comparison.

enumerator: Pointer to IGR_HTEXTCOMPARE

Pointer to store the enumerator for iterating through differences.

error: Pointer to Error_Control_Block

Contains any error text.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SAMPLE CODE

C++

```

1  #include <DocumentFilters.h>
2  #include <string.h>
3  #include <string>
4
5  #define UCS2(src) reinterpret_cast<const IGR_UCS2 *>(std::u16string(src).c_str())
6
7  int main(int argc, char **argv)
8  {
9      Instance_Status_Block isb = {0};
10     Error_Control_Block ecb = {0};
11     IGR_SHORT df;
12     IGR_LONG caps, type, pageCount = 0;
13     IGR_HDOC doc1_handle = 0, doc2_handle = 0;
14     IGR_HPAGE doc1_page1_handle = 0, doc2_page1_handle = 0;
15     IGR_HTEXTCOMPARE compare = 0;
16
17     strncpy(isb.Licensee_ID1, "YOUR_LICENSE_KEY_HERE", sizeof(isb.Licensee_ID1) - 1);
18     Init_Instance(0, ".", &isb, &df, &ecb);
19
20     if ((res = IGR_Open_File_Ex(UCS2(u"original.doc"), IGR_FORMAT_IMAGE, UCS2(u""), &caps, &type,
21 &doc1_handle, &ecb)) == IGR_OK
22     && (res = IGR_Open_File_Ex(UCS2(u"revised.doc"), IGR_FORMAT_IMAGE, UCS2(u""), &caps, &type,
23 &doc2_handle, &ecb)) == IGR_OK
24     && (res = IGR_Open_Page(doc1_handle, 0, &doc1_page1_handle, &ecb)) == IGR_OK
25     && (res = IGR_Open_Page(doc2_handle, 0, &doc2_page1_handle, &ecb)) == IGR_OK)
26     {
27         IGR_Text_Compare_Settings settings = { sizeof(IGR_Text_Compare_Settings) };
28         IGR_Compare_Documents_Difference diff = { sizeof(IGR_Compare_Documents_Difference) };
29
30         if ((ret = IGR_Text_Compare_Pages(doc1_page1_handle, nullptr, doc2_page1_handle, nullptr,
31 &settings, &compare, &ecb)) == IGR_OK)
32         {
33             while (IGR_Text_Compare_Next(compare, &diff, &ecb) == IGR_OK)
34             {
35                 // ... work with diff object
36
37                 IGR_Text_Compare_Difference_Dispose(&diff, &ecb);
38             }
39
40             IGR_Text_Compare_Close(compare, &ecb);
41         }
42
43         if (doc1_page1_handle)
44             IGR_Close_Page(doc1_page1_handle, &ecb);
45         if (doc2_page1_handle)
46             IGR_Close_Page(doc2_page1_handle, &ecb);
47         if (doc1_handle)
48             IGR_Close_File(doc1_handle, &ecb);
49         if (doc2_handle)
50             IGR_Close_File(doc2_handle, &ecb);
51         return 0;
52     }

```

SEE ALSO

- [IGR_Text_Compare_Close](#)

- [IGR_Text_Compare_Difference_Dispose](#)
- [IGR_Text_Compare_Documents](#)
- [IGR_Text_Compare_Elements](#)
- [IGR_Text_Compare_Next](#)
- [IGR_Text_Compare_Settings](#) data type
- [IGR_FRect](#) data type

IGR_Text_Compare_Reset

Resets the text comparison enumerator to its initial state.

PROTOTYPE

C/C++

```
IGR_RETURN_CODE IGR_Text_Compare_Reset(  
    IGR_HTEXTCOMPARE enumerator,  
    Error_Control_Block* error  
)
```

PARAMETERS

enumerator: IGR_HTEXTCOMPARE

The text comparison enumerator to reset.

error: Pointer to Error_Control_Block

Contains any error text.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

SEE ALSO

UTF8_to_Widechar

UTF8_to_Widechar converts a UTF-8 string into a UCS2 string.

PROTOTYPE

C/C++

```
void UTF8_to_Widechar(  
    const char* source,  
    IGR_UCS2* buffer,  
    IGR_LONG bufferSize  
)
```

PARAMETERS

source: Pointer to char

Contains a NULL terminated UTF-8 string.

buffer: Pointer to IGR_UCS2

Pointer to a buffer that will hold the converted text.

bufferSize: IGR_LONG

Indicates the size of the buffer, the system will copy up to this size including the NULL.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

UTF8_to_Widechar_Ex

UTF8_to_Widechar converts a UTF-8 string into a UCS2 string.

PROTOTYPE

C/C++

```
void UTF8_to_Widechar_Ex(  
    const char* source,  
    IGR_LONG sourceLen,  
    IGR_UCS2* buffer,  
    IGR_LONG bufferSize  
)
```

PARAMETERS

source: Pointer to char

Contains a NULL terminated UTF-8 string.

sourceLen: IGR_LONG

Indicate the size of the buffer pointed to by **source** , in bytes.

buffer: Pointer to IGR_UCS2

Pointer to a buffer that will hold the converted text.

bufferSize: IGR_LONG

Indicates the size of the buffer, the system will copy up to this size including the NULL.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

Widechar_to_UTF8

Widechar_to_UTF8 converts a UCS2 string into UTF-8.

PROTOTYPE

C/C++

```
void Widechar_to_UTF8(  
    const IGR_UCS2* source,  
    char* buffer,  
    IGR_LONG bufferSize  
)
```

PARAMETERS

source: Pointer to IGR_UCS2

Contains a NULL terminated UCS2 string.

buffer: Pointer to char

Pointer to a buffer that will hold the converted text.

bufferSize: IGR_LONG

Indicates the size of the buffer, the system will copy up to this size including the NULL.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

Widechar_to_UTF8_Ex

Widechar_to_UTF8 converts a UCS2 string into UTF-8

PROTOTYPE

C/C++

```
void Widechar_to_UTF8_Ex(  
    const IGR_UCS2* source,  
    IGR_LONG sourceLen,  
    char* buffer,  
    IGR_LONG bufferSize  
)
```

PARAMETERS

source: Pointer to IGR_UCS2

Contains a NULL terminated UCS2 string.

sourceLen: IGR_LONG

Indicate the size of the buffer pointed to by **source** , in bytes.

buffer: Pointer to char

Pointer to a buffer that will hold the converted text.

bufferSize: IGR_LONG

Indicates the size of the buffer, the system will copy up to this size including the NULL.

RETURN VALUE

Success: IGR_LONG

Returns IGR_OK.

Failure: IGR_LONG

Returns one of the possible [IGR_E error codes](#).

3.2.3 Structures

Overview

Error_Control_Block data type	Used to return error messages from the C API.
IGR_Annotation data type	The IGR_Annotation structure is used by the IGR_Get_Page_Annotations , IGR_Get_Page_Annotation_L , IGR_Get_Page_Annotation_Str methods to return information as annotation type, position, and dimensions for annotations given page.
IGR_Bookmark data type	The IGR_Bookmark structure is used by the IGR_Get_Bookmarks_Root method to return information about hierarchical navigation of a document.
IGR_CALLBACK data type	The IGR_CALLBACK data type represents the function signature of a callback from Document Filters.
IGR_Compare_Documents_Callback_Context data type	Context structure for the callback functions used in document comparison.
IGR_Compare_Documents_Difference data type	Structure representing a difference between compared documents.
IGR_Compare_Documents_Difference_Item data type	Structure representing a difference item in a compared document.
IGR_FPoint data type	Structure representing a size with signed floating-point width and height.
IGR_FRect data type	Structure representing a rectangle with floating-point coordinates. The coordinates are defined by the left, top, right, and bottom.
IGR_Hyperlink data type	The IGR_Hyperlink structure is used by the IGR_Get_Page_Hyperlinks method to return information such as position and destination for hyperlinks on a given page.
IGR_OPEN_CALLBACK data type	The IGR_OPEN_CALLBACK data type represents the function signature of a callback passed to IGR_Open_Ex .
IGR_Open_Callback_Action_Approve_External_Resource data type	The IGR_Open_Callback_Action_Approve_External_Resource structure is the payload argument for IGR_OPEN_CALLBACK .

	the action argument is <code>IGR_OPEN_CALLBACK_ACTION_APPROVE_EXTERNAL_RE</code>
IGR_Open_Callback_Action_Get_Resource_Stream data type	The <code>IGR_Open_Callback_Action_Get_Resource_St</code> structure is the payload argument for IGR_OPEN_CALLBACK when the action argument is <code>IGR_OPEN_CALLBACK_ACTION_GET_RESOURCE_STREAM</code>
IGR_Open_Callback_Action_Heartbeat data type	The <code>IGR_Open_Callback_Action_Heartbeat</code> structure is the argument for IGR_OPEN_CALLBACK when the action argument is <code>IGR_OPEN_CALLBACK_ACTION_HEARTBEAT</code>.
IGR_Open_Callback_Action_Localize data type	The <code>IGR_Open_Callback_Action_Localize</code> structure is the argument for IGR_OPEN_CALLBACK when the action argument is <code>IGR_OPEN_CALLBACK_ACTION_LOCALIZE</code>.
IGR_Open_Callback_Action_Log_Level data type	The <code>IGR_Open_Callback_Action_Log_Level</code> structure is the argument for IGR_OPEN_CALLBACK when the action argument is <code>IGR_OPEN_CALLBACK_ACTION_LOG_LEVEL</code>.
IGR_Open_Callback_Action_Log_Message data type	The <code>IGR_Open_Callback_Action_Log_Message</code> structure is the payload argument for IGR_OPEN_CALLBACK when the action argument is <code>IGR_OPEN_CALLBACK_ACTION_LOG_MESSAGE</code>.
IGR_Open_Callback_Action_OCR_Image data type	Structure that is payload argument for <code>IGR_OPEN_CALLBACK</code> when action argument is <code>IGR_OPEN_CALLBACK_ACTION_OCR_IMAGE</code>.
IGR_Open_Callback_Action_OCR_Image_Style_Info data type	Structure that is payload argument when calling <code>AddText</code> from <code>IGR_OPEN_CALLBACK</code> for <code>IGR_OPEN_CALLBACK_ACTION_OCR_IMAGE</code>.
IGR_Open_Callback_Action_Password data type	The <code>IGR_Open_Callback_Action_Password</code> structure is the argument for IGR_OPEN_CALLBACK when the action argument is <code>IGR_OPEN_CALLBACK_ACTION_PASSWORD</code>.
IGR_Open_DIB_Info data type	Structure representing information about an open Device Independent Bitmap (DIB). It includes details such as size, width, height, stride, pixel format, pixel data, palette information, and palette count.
IGR_Page_Element data type	<code>IGR_Open_DIB_Info</code> is used by IGR_Get_Page_Elements and related functions to enumerate the elements on a page.

IGR_Page_Form_Element data type	The IGR_Page_Form_Element structure is used by the IGR_Get_Page_Form_Elements method to return information such as position, dimensions, and text about the form items on a given page.
IGR_Page_Pixels data type	The IGR_Page_Pixels structure is used by IGR_Get_Page_Pixels to retrieve the pixel data for a given page.
IGR_Page_Word data type	The IGR_Page_Word structure is used by the IGR_Get_Page_Words method to return information such as position, dimensions, and text about the words on a given page.
IGR_QuadPoint data type	Structure representing the points for a rectangle, in clockwise order.
IGR_Rect data type	The IGR_Rect structure is used to specify the position of a rectangle.
IGR_Render_Page_Form_Values data type	The IGR_Render_Page_Form_Values structure is used with IGR_Render_Page_Properties to specify new values for form items.
IGR_Render_Page_Properties data type	The IGR_Render_Page_Properties structure is used by the IGR_Render_Page_Extents method and provides additional information for rendering a page to a canvas.
IGR_Render_Page_Redactions data type	Structure representing a redaction area for rendering on a page. A redaction area is defined by its position (x, y) and size (width, height), along with the color to use for rendering and additional reserved data.
IGR_SRect data type	Structure representing a rectangle with signed integer coordinates. The coordinates are defined by the left, top, right, and bottom values.
IGR_Size data type	The IGR_Size structure is used to specify the size of a rectangle.
IGR_Stream data type	The IGR_Stream data type allows for the creation of custom streams.
IGR_Subfile_Info data type	Used to return subfile information from the IGR_Get_Subfile_Entry_Ex C API.
IGR_T_ACTION_GET_STREAM_PART data type	When a second or later part of a multi-part archive is required for processing an Extended Stream, a callback will request the structure to be populated with a new stream instance.
IGR_Text_Compare_Document_Source data type	Structure representing a source document for text comparison.

IGR_Text_Compare_Settings data type	Settings structure for text comparison operations.
IGR_Writable_Stream data type	Similar to the IGR_Stream, the IGR_Writable_Stream data type allows for the creation of custom output streams. It's important to note, all methods must be implemented, not just the Write.
Instance_Status_Block data type	The Instance_Status_Block data type is a control block returned by Init_Instance with information about the installed Document engine.

Error_Control_Block data type

Used to return error messages from the C API.

DEFINITIONS

C/C++

```
struct Error_Control_Block
{
    char      Msg[238];      // Array of 238 null-terminated ANSI characters
    IGR_SHORT MsgId;        // error code
};
```

ADDITIONAL INFORMATION

The functions in the C API report errors via an error message control block. It consists of an array of 238 null-terminated ANSI characters, followed by a IGR_SHORT error code.

If the call worked, the error message is empty (the first character is zero).

Any C API function that fails will populate the error message block with additional information.

IGR_Annotation data type

The IGR_Annotation structure is used by the [IGR_Get_Page_Annotations](#), [IGR_Get_Page_Annotation_Long](#), and [IGR_Get_Page_Annotation_Str](#) methods to return information such as annotation type, position, and dimensions for annotations on a given page.

DEFINITIONS

C/C++

```
struct IGR_Annotation
{
    const void* reserved;
    IGR_LONG type;
    IGR_LONG flags;
    IGR_LONG x;
    IGR_LONG y;
    IGR_LONG width;
    IGR_LONG height;
    IGR_LONG reserved2[8];
};
```

DETAILS

reserved: Pointer to VOID

Reserved data

type: IGR_LONG

Contains the annotation type, see [IGR_ANNOTATION_TYPE](#).

flags: IGR_LONG

Is a bitmask containing specifying various characteristics of the annotation. See [IGR_ANNOTATION_FLAGS_TYPE](#).

x: IGR_LONG

Contains the X coordinate of the annotation's bounding box in pixels.

y: IGR_LONG

Contains the Y coordinate of the annotation's bounding box in pixels.

width: IGR_LONG

Contains the Width dimension of the annotation's bounding box in pixels.

height: IGR_LONG

Contains the Height dimension of the annotation's bounding box in pixels.

```
reserved2: IGR_LONG[8]
```

Reserved for future use.

SEE ALSO

- [IGR_Get_Page_Annotations](#)
- [IGR_Get_Page_Annotation_Count](#)
- [IGR_Get_Page_Annotation_Long](#)
- [IGR_Get_Page_Annotation_Str](#)

IGR_Bookmark data type

The IGR_Bookmark structure is used by the IGR_Get_Bookmarks_Root method to return information about the hierarchical navigation of a document.

DEFINITIONS

C/C++

```
struct IGR_Bookmark
{
    const void* reserved;
    IGR_LONG flags;
    IGR_UCS2 title[255];
    IGR_LONG action;
    IGR_UCS2 dest[1024];
    IGR_LONG level;
    IGR_LONG fit;
    IGR_LONG zoom;
    IGR_LONG x;
    IGR_LONG y;
    IGR_LONG width;
    IGR_LONG height;
    IGR_LONG color;
    IGR_LONG text_style;
    IGR_LONG reserved2[8];
};
```

DETAILS

`reserved`: Pointer to VOID

Reserved data

`flags`: Unicode String

Reserved for future use

`title`: Unicode String

Contains the Unicode String of the bookmark title

`action`: IGR_LONG

Contains the action that should be executed when a user selects the bookmark, can be one of:

- IGR_BOOKMARK_ACTION_GOTO: 0
- IGR_BOOKMARK_ACTION_LINK: 1

`dest`: Unicode String

Contains the Unicode String of the bookmark destination

`level: IGR_LONG`

Contains the indent level of the bookmark in the navigation tree.

`fit: IGR_LONG`

Contains the zoom/fit information which should be applied when a user selects the bookmark, can be one of:

- IGR_BOOKMARK_FIT_NONE : 0
- IGR_BOOKMARK_FIT_BOTH : 1
- IGR_BOOKMARK_FIT_WIDTH : 2
- IGR_BOOKMARK_FIT_HEIGHT : 3
- IGR_BOOKMARK_FIT_RECT : 4

`zoom: IGR_LONG`

Contains the zoom/fit information which should be applied when a user selects the bookmark

`x: IGR_LONG`

Contains the X coordinate of the word in pixels.

`y: IGR_LONG`

Contains the Y coordinate of the word in pixels.

`width: IGR_LONG`

Contains the Width dimension of the word in pixels.

`height: IGR_LONG`

Contains the Height dimension of the word in pixels.

`color: IGR_LONG`

Contains the color that the bookmark should be shown

`text_style: IGR_LONG`

Contains any text styling that should be applied when showing the bookmark title; see [Text Styles](#) for details.

`reserved2: IGR_LONG[8]`

Reserved for future use.

SEE ALSO

- [IGR_Get_Bookmark_Root](#)
- [IGR_Get_Bookmark_First_Child](#)
- [IGR_Get_Bookmark_Next_Child](#)

IGR_CALLBACK data type

The IGR_CALLBACK data type represents the function signature of a callback from Document Filters.

DEFINITION

```
typedef IGR_LONG (*IGR_CALLBACK)(int actionID, void* actionData, void* context);
```

DETAILS

Function return type: IGR_LONG

User-set error code return value from function.

actionID: int

Represents the action or request type required of the API user's code.

actionData: void pointer

A structure, dependant on the actionID, for the API user to populate with return data.

context: void pointer

A context pointer for the API user to utilise in any way.

ADDITIONAL INFORMATION

API User callback function signatures must match the signature of this typedef. The nature of actionID and actionData are specific to the function generating the callback. The API user passes the context pointer to Document Filters and it is returned when the callback is invoked. This can be employed by the API user for any purpose.

SEE ALSO

- [IGR_Extend_Stream](#)

IGR_Compare_Documents_Callback_Context data type

Context structure for the callback functions used in document comparison.

PROTOTYPE

C/C++

```
struct IGR_Compare_Documents_Callback_Context
{
    IGR_ULONG struct_size;
    IGR_ULONG doc_index;
    IGR_LONG doc_handle;
    IGR_ULONG page_index;
    IGR_HPAGE page_handle;
}
```

DETAILS

struct_size: IGR_ULONG

Size of this structure.

doc_index: IGR_ULONG

Index to the document being processed. Will be either 0 or 1.

doc_handle: IGR_LONG

The handle to the document being processed.

page_index: IGR_ULONG

The index to the current page being processed.

page_handle: IGR_HPAGE

The handle to the current page being processed.

IGR_Compare_Documents_Difference data type

Structure representing a difference between compared documents.

PROTOTYPE

C/C++

```
struct IGR_Compare_Documents_Difference
{
    IGR_ULONG struct_size;
    IGR_ULONG type;
    IGR_ULONG doc_source;
    IGR_ULONG original_page_index;
    IGR_ULONG revised_page_index;
    IGR_ULONG item_count;
    const struct IGR_Compare_Documents_Difference_Item* items;
}
```

DETAILS

struct_size: IGR_ULONG

Size of this structure.

type: IGR_ULONG

The type of difference.

doc_source: IGR_ULONG

The document where the difference occurred.

original_page_index: IGR_ULONG

The page index in the original/left document of the first item in the hit list.

revised_page_index: IGR_ULONG

The page index in the revised/right document of the first item in the hit list.

item_count: IGR_ULONG

The number of difference items pointed to by items.

items: Pointer to IGR_Compare_Documents_Difference_Item

Pointer to the first rectangle.

IGR_Compare_Documents_Difference_Item data type

Structure representing a difference item in a compared document.

PROTOTYPE

C/C++

```
struct IGR_Compare_Documents_Difference_Item
{
    IGR_ULONG struct_size;
    IGR_ULONG page_index;
    struct IGR_FRect bounds;
    const IGR_UCS2* text;
}
```

DETAILS

struct_size: IGR_ULONG

Size of this structure.

page_index: IGR_ULONG

The index of the page that generated the diff.

bounds: struct IGR_FRect

The bounding box of the word, in pixels.

text: Pointer to IGR_UCS2

Null terminated UCS2 text buffer.

IGR_FPoint data type

Structure representing a size with signed floating-point width and height.

PROTOTYPE

C/C++

```
struct IGR_FPoint
{
    IGR_FLOAT x;
    IGR_FLOAT y;
}
```

DETAILS

x: IGR_FLOAT

X position.

y: IGR_FLOAT

Y position.

IGR_FRect data type

Structure representing a rectangle with floating-point coordinates. The coordinates are defined by the left, top, right, and bottom edges.

PROTOTYPE

C/C++

```
struct IGR_FRect
{
    IGR_FLOAT left;
    IGR_FLOAT top;
    IGR_FLOAT right;
    IGR_FLOAT bottom;
}
```

DETAILS

`left: IGR_FLOAT`

Left edge coordinate.

`top: IGR_FLOAT`

Top edge coordinate.

`right: IGR_FLOAT`

Right edge coordinate.

`bottom: IGR_FLOAT`

Bottom edge coordinate.

IGR_Hyperlink data type

The IGR_Hyperlink structure is used by the [IGR_Get_Page_Hyperlinks](#) method to return information such as position and destination for hyperlinks on a given page.

DEFINITIONS

C/C++

```
struct IGR_Hyperlink
{
    const void * reserved;
    IGR_LONG type;
    IGR_LONG x;
    IGR_LONG y;
    IGR_LONG width;
    IGR_LONG height;
    IGR_LONG page_number;
    IGR_LONG dest_fit;
    IGR_LONG dest_zoom;
    IGR_LONG dest_left;
    IGR_LONG dest_top;
    IGR_LONG dest_right;
    IGR_LONG dest_bottom;
    IGR_LONG flags;
    IGR_LONG reserved2[8];
};
```

DETAILS

reserved: Pointer to const void

Reserved for internal use

type: IGR_LONG

Indicates the type of action to execute when the link is clicked. Can be one or more of:

- IGR_HYPERLINK_ACTION_GOTO : 0x1
- IGR_HYPERLINK_ACTION_URI : 0x2

x: IGR_LONG

The x pixel position of the hyperlink

y: IGR_LONG

The y pixel position of the hyperlink

width: IGR_LONG

The width in pixels of the hyperlink

`height: IGR_LONG`

The height in pixels of the hyperlink

`page_number: IGR_LONG`

Indicates the page number to show if the type is IGR_HYPERLINK_ACTION_GOTO

`dest_fit: IGR_LONG`

Indicates the page fit that should be applied when navigating to the destination. See PDF Reference for additional information. One of:

- IGR_HYPERLINK_FIT_XYZ : 0
- IGR_HYPERLINK_FIT_FIT : 1
- IGR_HYPERLINK_FIT_FITH : 2
- IGR_HYPERLINK_FIT_FITV : 3
- IGR_HYPERLINK_FIT_FITR : 4
- IGR_HYPERLINK_FIT_FITB : 5
- IGR_HYPERLINK_FIT_FITBH : 6
- IGR_HYPERLINK_FIT_FITBV : 7

`dest_zoom: IGR_LONG`

Indicates the zoom factor to apply when navigating

`dest_left: IGR_LONG`

Indicates left position in pixels that should be applied when navigating to the destination, fit value specific

`dest_top: IGR_LONG`

Indicates top position in pixels that should be applied when navigating to the destination, fit value specific

`dest_right: IGR_LONG`

Indicates right position in pixels that should be applied when navigating to the destination, fit value specific

`dest_bottom: IGR_LONG`

Indicates bottom position in pixels that should be applied when navigating to the destination, fit value specific

`flags: IGR_LONG`

Flags specifying which destination coordinates are provided. See PDF Reference for additional information. One or more of:

- `IGR_HYPERLINK_FLAGS_CHANGE_LEFT` : 0x1
- `IGR_HYPERLINK_FLAGS_CHANGE_TOP` : 0x2
- `IGR_HYPERLINK_FLAGS_CHANGE_ZOOM` : 0x4

`reserved2: IGR_LONG[8]`

Reserved for future use.

SEE ALSO

- [IGR_Get_Page_Hyperlink_Count](#)
- [IGR_Get_Page_Hyperlinks](#)
- [IGR_Get_Page_Hyperlink_Str](#)

IGR_OPEN_CALLBACK data type

The IGR_OPEN_CALLBACK data type represents the function signature of a callback passed to [IGR_Open_Ex](#).

DEFINITION

```
typedef IGR_LONG(IGR_EXPORT* IGR_OPEN_CALLBACK)(IGR_OPEN_CALLBACK_ACTION action, void* payload, void* context);
```

DETAILS

Function return type: IGR_LONG

User-set error code return value from function. The interpretation of the return value is dependent on the type of action.

action: IGR_OPEN_CALLBACK_ACTION

The type of action that prompted the calling of the callback.

Name	Value	Description
IGR_OPEN_CALLBACK_ACTION_HEARTBEAT	0	Specifies the type of callback action is heartbeat.
IGR_OPEN_CALLBACK_ACTION_PASSWORD	1	Specifies the type of callback action is password.
IGR_OPEN_CALLBACK_ACTION_LOCALIZE	2	Specifies the type of callback action is localize.
IGR_OPEN_CALLBACK_ACTION_LOG_LEVEL	3	Specifies the type of callback action is log-level.

Name	Value	Description
IGR_OPEN_CALLBACK_ACTION_LOG_MESSAGE	4	Specifies the type of callback action is log-message.
IGR_OPEN_CALLBACK_ACTION_APPROVE_EXTERNAL_RESOURCE	5	Specifies the type of callback action is approve-external-resource.
IGR_OPEN_CALLBACK_ACTION_GET_RESOURCE_STREAM	6	Specifies the type of callback action is get-resource-stream.
IGR_OPEN_CALLBACK_ACTION_OCR_IMAGE	7	Specifies the type of callback action is ocr-image.

payload: Pointer to VOID

A structure that can be modified to provide information that will be used to process the document. The type of the structure corresponds to the type of action, e.g. if action is IGR_OPEN_CALLBACK_ACTION_HEARTBEAT, then payload is a IGR_Open_Callback_Action_Heartbeat structure.

action	payload type
IGR_OPEN_CALLBACK_ACTION_HEARTBEAT	IGR_Open_Callback_Action_Heartbeat
IGR_OPEN_CALLBACK_ACTION_PASSWORD	IGR_Open_Callback_Action_Password
IGR_OPEN_CALLBACK_ACTION_LOCALIZE	IGR_Open_Callback_Action_Localize

action	payload type
IGR_OPEN_CALLBACK_ACTION_LOG_LEVEL	IGR_Open_Callback_Action_Log_Level
IGR_OPEN_CALLBACK_ACTION_LOG_MESSAGE	IGR_Open_Callback_Action_Log_Message
IGR_OPEN_CALLBACK_ACTION_APPROVE_EXTERNAL_RESOURCE	IGR_Open_Callback_Action_Approve_Ext
IGR_OPEN_CALLBACK_ACTION_GET_RESOURCE_STREAM	IGR_Open_Callback_Action_Get_Resource
IGR_OPEN_CALLBACK_ACTION_OCR_IMAGE	IGR_Open_Callback_Action_OCR_Image

context: Pointer to VOID

Contextual information that was passed to [IGR_Open_Ex](#) as the `callback_context` argument.

ADDITIONAL INFORMATION

[IGR_Open_Ex](#) callback function signatures must match the signature of this typedef.

The API user passes the context pointer to Document Filters and it is returned when the callback is invoked. This can be employed by the API user for any purpose.

Heartbeat

A heartbeat callback allows for the API user to request cancellation of an operation.

For example, an API user can measure the time of an API call and request cancellation if 30 seconds has passed without completion.

A return value of anything other than `IGR_OK` from the callback function is considered a request for cancellation.

If a Document Filters API call exits early without success because of cancellation, it will return with code `IGR_CANCELLED`.

Password

A password callback allows for the API user to provide a password string that will be used to decode a file, e.g. a password-protected archive.

It provides the API user with an id that specifies if the password being requested is for the top-level document or for a subfile, and allows the API user to fill a buffer with a password string meant to satisfy the password request.

The password callback for a specific id will repeatedly be called until the password is correct, allowing for multiple passwords to be tried in succession, unless any of the following conditions is true:

- Anything other than `IGR_OK` is returned from the callback function
- On return from the callback function, the password string buffer is empty

- On return from the callback function, the password string buffer is filled with the same password string returned from the previous password callback

Localize

A localize callback allows for the API user to replace hard-coded English string values, specifically in metadata.

It provides the API user with the original value, i.e. the hard-coded English value, and allows them to fill a buffer with its replacement.

The original value is used without replacement if any of the following conditions is true:

- Anything other than IGR_OK is returned from the callback function
- The replacement buffer is empty

Log Level

The log level determines the verbosity of log messages for a given module in Hyland Document Filters. When a logger is created, the calling application can configure the log level to enable or disable logging for that module.

Since logging introduces performance overhead—particularly with high verbosity levels—applications may wish to limit the logs they capture to essential information.

Log Message

A log message provides diagnostic information about the operation of Hyland Document Filters. Each log message includes:

- Module Name – Identifies the component or subsystem generating the message.
- Log Level – Indicates the severity or importance of the message (e.g., DEBUG, INFO, WARNING, ERROR).
- Message Content – Describes the event or condition being logged.

Approve External Resource

An **Approve External Resource** message is sent when `HTTP_EXTERNAL_IMAGES` is enabled for HTML-based documents. This message allows the application to approve or reject the downloading of image files from external sources.

- If the application permits the request, the image will be downloaded and processed.
- If the application returns `IGR_CANCELLED`, the image download will be blocked.
- This mechanism provides control over external content loading, which can be useful for security, performance, or compliance reasons.

Get Resource Stream

A **Get Resource Stream** message is sent when an image is embedded in an HTML-based document, allowing the application to provide a stream for reading the image bytes. This callback is triggered before **Approve External Resource** and before the image is automatically downloaded.

OCR Image

A **OCR Image** message is sent when OCR is enabled, and `OCR_ENGINE=CUSTOM`. This gives you the ability to hook into an external OCR engine and provide the words.

SEE ALSO

- [IGR_Open_Ex](#)
- [IGR_OPEN_CALLBACK_ACTION](#)
- [IGR_Open_Callback_Action_Approve_External_Resource](#)
- [IGR_Open_Callback_Action_Get_Resource_Stream](#)
- [IGR_Open_Callback_Action_Heartbeat](#)
- [IGR_Open_Callback_Action_Localize](#)
- [IGR_Open_Callback_Action_Log_Level](#)
- [IGR_Open_Callback_Action_Log_Message](#)
- [IGR_Open_Callback_Action_OCR_Image](#)
- [IGR_Open_Callback_Action_Password](#)

IGR_Open_Callback_Action_Approve_External_Resource data type

The `IGR_Open_Callback_Action_Approve_External_Resource` structure is the payload argument for `IGR_OPEN_CALLBACK` when the `action` argument is `IGR_OPEN_CALLBACK_ACTION_APPROVE_EXTERNAL_RESOURCE`.

DEFINITIONS

C/C++

```
struct IGR_Open_Callback_Action_Approve_External_Resource
{
    // Indicate the size of this structure; must be populated.
    IGR_ULONG struct_size;

    // Original URL string of the external resource (e.g., image URL).
    IGR_UCS2 url[1024];
};
```

DETAILS

`struct_size: IGR_ULONG`

Specifies the size of this structure in bytes. Used to determine available fields.

`url: IGR_UCS2`

The url of the resource that is about to be requested. Returning `IGR_CANCELLED` prevents the resource from downloading.

SEE ALSO

- [IGR_OPEN_CALLBACK](#)
- [IGR_OPEN_CALLBACK_ACTION](#)
- [IGR_Open_Ex](#)

IGR_Open_Callback_Action_Get_Resource_Stream data type

The `IGR_Open_Callback_Action_Get_Resource_Stream` structure is the payload argument for `IGR_OPEN_CALLBACK` when the `action` argument is `IGR_OPEN_CALLBACK_ACTION_GET_RESOURCE_STREAM`.

PROTOTYPE

C/C++

```
struct IGR_Open_Callback_Action_Get_Resource_Stream
{
    IGR_ULONG struct_size;
    IGR_UCS2 url[1024];
    struct IGR_Stream* result;
}
```

DETAILS

`struct_size`: `IGR_ULONG`

Specifies the size of this structure in bytes. Used to determine available fields.

`url`: `IGR_UCS2`

The url of the resource that is about to be requested. Returning `IGR_CANCELLED` prevents the resource from downloading.

`result`: Pointer to `IGR_Stream`

The stream where the resource bytes can be read.

SEE ALSO

- [IGR_OPEN_CALLBACK](#)
- [IGR_OPEN_CALLBACK_ACTION](#)
- [IGR_Open_Ex](#)

IGR_Open_Callback_Action_Heartbeat data type

The IGR_Open_Callback_Action_Heartbeat structure is the payload argument for [IGR_OPEN_CALLBACK](#) when the [action](#) argument is IGR_OPEN_CALLBACK_ACTION_HEARTBEAT.

DEFINITIONS

C/C++

```
struct IGR_Open_Callback_Action_Heartbeat
{
    IGR_ULONG struct_size;
};
```

DETAILS

`struct_size: IGR_ULONG`

Specifies the size of this structure in bytes. Used to determine available fields.

SEE ALSO

- [IGR_OPEN_CALLBACK](#)
- [IGR_OPEN_CALLBACK_ACTION](#)
- [IGR_Open_Ex](#)

IGR_Open_Callback_Action_Localize data type

The `IGR_Open_Callback_Action_Localize` structure is the payload argument for [IGR_OPEN_CALLBACK](#) when the `action` argument is `IGR_OPEN_CALLBACK_ACTION_LOCALIZE`.

DEFINITIONS

C/C++

```
struct IGR_Open_Callback_Action_Localize
{
    IGR_ULONG struct_size;
    IGR_ULONG string_id;
    IGR_UCS2 original[1024];
    IGR_UCS2 replacement[1024];
};
```

DETAILS

`struct_size: IGR_ULONG`

Specifies the size of this structure in bytes. Used to determine available fields.

`string_id: IGR_ULONG`

Indicates the string resource id for the string being localized. This number is fixed, and will not change across builds. See [IGR_STRING_ID](#) for a list of strings.

`original: IGR_UCS2`

The original unlocalized string.

`replacement: IGR_UCS2`

Buffer where user populates the localized replacement value.

SEE ALSO

- [IGR_OPEN_CALLBACK](#)
- [IGR_OPEN_CALLBACK_ACTION](#)
- [IGR_Open_Ex](#)

IGR_Open_Callback_Log_Level data type

The IGR_Open_Callback_Action_Log_Level structure is the payload argument for [IGR_OPEN_CALLBACK](#) when the [action](#) argument is IGR_OPEN_CALLBACK_ACTION_LOG_LEVEL.

DEFINITIONS

C/C++

```
struct IGR_Open_Callback_Action_Log_Level
{
    // [out] Indicate the size of this structure; must be populated.
    IGR_ULONG struct_size;

    // Indicates the module that the log level is being requested.
    char module[16];

    // User sets with the desired log level for the module. Defaults to IGR_LOG_LEVEL_NONE
    IGR_LOG_LEVEL_TYPE result;
};
```

DETAILS

`struct_size: IGR_ULONG`

Specifies the size of this structure in bytes. Used to determine available fields.

`module: char[16]`

The module name that the log-level is requested for.

`result: IGR_LOG_LEVEL_TYPE`

The user-set log-level for the module.

Name	Value
IGR_LOG_LEVEL_NONE	0
IGR_LOG_LEVEL_CRITICAL	1
IGR_LOG_LEVEL_ERROR	2
IGR_LOG_LEVEL_WARNING	3
IGR_LOG_LEVEL_INFORMATION	4
IGR_LOG_LEVEL_DEBUG	5

Name	Value
IGR_LOG_LEVEL_TRACE	6

SEE ALSO

- [IGR_OPEN_CALLBACK](#)
- [IGR_OPEN_CALLBACK_ACTION](#)
- [IGR_Open_Ex](#)

IGR_Open_Callback_Action_Log_Message data type

The IGR_Open_Callback_Action_Log_Message structure is the payload argument for IGR_OPEN_CALLBACK when the `action` argument is `IGR_OPEN_CALLBACK_ACTION_LOG_MESSAGE`.

DEFINITIONS

C/C++

```
struct IGR_Open_Callback_Action_Log_Message
{
    // Indicates the size of this structure; must be populated.
    IGR_ULONG struct_size;

    // Indicates the module that generated the message.
    char module[16];

    // Indicates the log level of the message.
    IGR_LOG_LEVEL_TYPE log_level;

    // Contains the log message.
    char message[512];
};
```

DETAILS

`struct_size`: IGR_ULONG

Specifies the size of this structure in bytes. Used to determine available fields.

`module`: char[16]

The name of the module that generated the log message.

`log_level`: IGR_LOG_LEVEL_TYPE

The log level of the message.

Name	Value
IGR_LOG_LEVEL_NONE	0
IGR_LOG_LEVEL_CRITICAL	1
IGR_LOG_LEVEL_ERROR	2
IGR_LOG_LEVEL_WARNING	3
IGR_LOG_LEVEL_INFORMATION	4

Name	Value
IGR_LOG_LEVEL_DEBUG	5
IGR_LOG_LEVEL_TRACE	6

`message: char[512]`

The log message represented as utf-8.

SEE ALSO

- [IGR_OPEN_CALLBACK](#)
- [IGR_OPEN_CALLBACK_ACTION](#)
- [IGR_Open_Ex](#)
- [IGR_Open_Callback_Log_Level](#)
- [IGR_LOG_LEVEL_TYPE](#)

IGR_Open_Callback_Action_OCR_Image data type

Structure that is payload argument for IGR_OPEN_CALLBACK when action argument is IGR_OPEN_CALLBACK_ACTION_OCR_IMAGE.

PROTOTYPE

C/C++

```
struct IGR_Open_Callback_Action_OCR_Image
{
    IGR_ULONG struct_size;
    void* reserved;
    const struct IGR_Open_DIB_Info* source_image_pixels;
    IGR_ULONG source_page_index;
    struct IGR_Rect source_rect;
    IGR_UCS2 source_name[128];
    IGR_RETURN_CODE(IGR_EXPORT* GetMetadata)(const struct IGR_Open_Callback_Action_OCR_Image* action, const
IGR_UCS2* name, IGR_UCS2* dest, size_t dest_size);
    IGR_RETURN_CODE(IGR_EXPORT* SaveImage)(const struct IGR_Open_Callback_Action_OCR_Image* action, const
IGR_UCS2* filename, const IGR_UCS2* mimetype);
    IGR_RETURN_CODE(IGR_EXPORT* StartBlock)(const struct IGR_Open_Callback_Action_OCR_Image* action,
IGR_ULONG block_type, const struct IGR_QuadPoint* rect);
    IGR_RETURN_CODE(IGR_EXPORT* EndBlock)(const struct IGR_Open_Callback_Action_OCR_Image* action, IGR_ULONG
block_type);
    IGR_RETURN_CODE(IGR_EXPORT* AddText)(const struct IGR_Open_Callback_Action_OCR_Image* action, const
IGR_UCS2* text, const struct IGR_QuadPoint* rect, IGR_ULONG flags, const struct
IGR_Open_Callback_Action_OCR_Image_Style_Info* font_style);
    IGR_RETURN_CODE(IGR_EXPORT* AddTextWithOffsets)(const struct IGR_Open_Callback_Action_OCR_Image* action,
const IGR_UCS2* text, const IGR_FLOAT* offsets, const struct IGR_QuadPoint* rect, IGR_ULONG flags, const
struct IGR_Open_Callback_Action_OCR_Image_Style_Info* font_style);
    IGR_RETURN_CODE(IGR_EXPORT* Reorient)(const struct IGR_Open_Callback_Action_OCR_Image* action, IGR_FLOAT
degrees);
}
```

DETAILS

`struct_size`: IGR_ULONG

[in] Indicate the size of this structure; must be populated.

`reserved`: Pointer to void

Reserved for internal use.

`source_image_pixels`: Pointer to IGR_Open_DIB_Info

The image data to process.

`source_page_index`: IGR_ULONG

The page-index of the page containing the image.

`source_rect: IGR_Rect`

The rectangle of the image in the source document.

`source_name: IGR_UCS2[128]`

The name of the source image, if known.

`GetMetadata: Function Pointer`

Returns EXIF based metadata about the document.

`SaveImage: Function Pointer`

Save the inbound image to a file with the specified mime type.

`StartBlock: Function Pointer`

Starts a new block element in the output.

`EndBlock: Function Pointer`

Close the current block element in the output.

`AddText: Function Pointer`

Add text to the output.

`AddTextWithOffsets: Function Pointer`

Add text, factoring in the offsets, to the output. There must be one offset for each character in the text.

`Reorient: Function Pointer`

Replace the original image with the processed image.

IGR_Open_Callback_Action_OCR_Image_Style_Info data type

Structure that is payload argument when calling AddText from the IGR_OPEN_CALLBACK for IGR_OPEN_CALLBACK_ACTION_OCR_IMAGE.

PROTOTYPE

C/C++

```
struct IGR_Open_Callback_Action_OCR_Image_Style_Info
{
    IGR_ULONG struct_size;
    IGR_FLOAT font_size;
    IGR_UCS2 font_family[256];
    IGR_ULONG text_style;
}
```

DETAILS

`struct_size: IGR_ULONG`

Indicate the size of this structure; must be populated.

`font_size: IGR_FLOAT`

The size of the font as reported by the OCR engine.

`font_family: IGR_UCS2[256]`

The font family as reported by the OCR engine.

`text_style: IGR_ULONG`

The font style, using the IGR_TEXT_STYLE_ bit field.

IGR_Open_Callback_Action_Password data type

The IGR_Open_Callback_Action_Password structure is the payload argument for [IGR_OPEN_CALLBACK](#) when the [action](#) argument is IGR_OPEN_CALLBACK_ACTION_PASSWORD.

DEFINITIONS

C/C++

```
struct IGR_Open_Callback_Action_Password
{
    IGR_ULONG struct_size;
    IGR_UCS2 id[4096];
    IGR_UCS2 password[4096];
};
```

DETAILS

struct_size: IGR_ULONG

Specifies the size of this structure in bytes. Used to determine available fields.

id: IGR_UCS2

Subfile Id indicating which subfile the password is requested for. May be empty to indicate the top level document password is requested. Null terminated.

password: IGR_UCS2

Password string to be used to decode the requested document or subfile. Must be null terminated.

SEE ALSO

- [IGR_OPEN_CALLBACK](#)
- [IGR_OPEN_CALLBACK_ACTION](#)
- [IGR_Open_Ex](#)

IGR_Open_DIB_Info data type

Structure representing information about an open Device Independent Bitmap (DIB). It includes details such as size, flags, width, height, stride, pixel format, pixel data, palette information, and palette count.

PROTOTYPE

C/C++

```
struct IGR_Open_DIB_Info
{
    IGR_ULONG struct_size;
    IGR_ULONG flags;
    IGR_ULONG width;
    IGR_ULONG height;
    IGR_ULONG stride;
    IGR_LONG pixel_format;
    const void* pixel_data;
    IGR_ULONG pixel_data_size;
    const IGR_LONG* palette;
    IGR_ULONG palette_count;
}
```

DETAILS

struct_size: IGR_ULONG

Size of this structure, must be populated.

flags: IGR_ULONG

Flags indicating specific characteristics.

width: IGR_ULONG

Width of the bitmap, in pixels.

height: IGR_ULONG

Height of the bitmap, in pixels.

stride: IGR_ULONG

Number of bytes per scanline.

pixel_format: IGR_LONG

Format of the pixel data.

pixel_data: Pointer to void

Pointer to the raw pixel data.

`pixel_data_size: IGR_ULONG`

Size of the buffer in `pixel_data`.

`palette: Pointer to IGR_LONG`

Pointer to the RGB palette information.

`palette_count: IGR_ULONG`

Number of colors in the palette.

IGR_Page_Element data type

IGR_Open_DIB_Info is used by [IGR_Get_Page_Elements](#) and related functions to enumerate the elements on a page.

DEFINITIONS

C/C++

```
struct IGR_Page_Element
{
    IGR_ULONG struct_size;
    const void* reserved;
    IGR_PAGE_ELEMENT_TYPE type;
    IGR_ULONG depth;
    IGR_ULONG flags;

    struct IGR_FRect pos;

    IGR_ULONG rotation;
};
```

DETAILS

struct_size: IGR_ULONG

The size of the structure, in bytes. The caller must set this member to sizeof(IGR_Page_Element).

reserved: void

Reserved for future use.

type: IGR_PAGE_ELEMENT_TYPE

The type of element being enumerated. See [IGR_PAGE_ELEMENT_TYPE](#) for details.

depth: IGR_ULONG

The depth of the element in the element stack.

flags: IGR_ULONG

Reserved for future use.

pos: IGR_FRect

The absolute position of the element on the page, in points.

rotation: IGR_ULONG

Any rotation applied to the element.

SEE ALSO

- [IGR_Get_Page_Elements](#)
- [IGR_PAGE_ELEMENT_TYPE](#)

IGR_Page_Form_Element data type

The IGR_Page_Form_Element structure is used by the [IGR_Get_Page_Form_Elements](#) method to return information such as position, dimensions, and text about the form items on a given page.

DEFINITIONS

C/C++

```
struct IGR_Page_Form_Element
{
    const void* reserved;
    IGR_UCS2 name[255];
    IGR_LONG x;
    IGR_LONG y;
    IGR_LONG width;
    IGR_LONG height;
    IGR_LONG type;
    IGR_LONG flags;
    IGR_LONG fontSize;
    IGR_LONG fontStyle;
    IGR_LONG rotation;
    IGR_LONG textAlignment;
    IGR_LONG selectedItem;
    IGR_LONG option_count;
    IGR_LONG reserved2[8];
};
```

DETAILS

reserved: Pointer to VOID

Reserved data

name: Unicode String

Contains the Unicode String of the form element name

x: IGR_LONG

Contains the X coordinate of the word in pixels.

y: IGR_LONG

Contains the Y coordinate of the word in pixels.

width: IGR_LONG

Contains the Width dimension of the word in pixels.

height: IGR_LONG

Contains the Height dimension of the word in pixels.

`type: IGR_LONG`

Contains the form element type, see [IGR_PAGE_FORM_ELEMENT_TYPE](#).

`flags: IGR_LONG`

Is a bitmask containing any style information known about the form element. See [IGR_PAGE_FORM_ELEMENT_FLAG_TYPE](#).

`fontSize: IGR_LONG`

Contains the font size that should be used when presenting single or multiline text boxes.

`fontStyle: IGR_LONG`

Is a bitmask containing any style information known about the word; see [Text Styles](#) for details.

`rotation: IGR_LONG`

Contains any rotation to apply to the element, in 90 degree increments.

`textAlignment: IGR_LONG`

Contains the text alignment that should be used when presenting single or multiline text boxes, one of:

- `IGR_TEXT_ALIGN_LEFT : 0`
- `IGR_TEXT_ALIGN_RIGHT : 1`
- `IGR_TEXT_ALIGN_CENTER : 2`

`selectedItem: IGR_LONG`

Contains the index of the selected item for list boxes or dropdown boxes

`option_count: IGR_LONG`

Contains the number of items available for list boxes or dropdown boxes.

`reserved2: IGR_LONG[8]`

Reserved for future use.

SEE ALSO

- [IGR_Get_Page_Form_Elements](#)

IGR_Page_Pixels data type

The IGR_Page_Pixels structure is used by [IGR_Get_Page_Pixels](#) to retrieve the pixel data for a given page.

DEFINITIONS

C/C++

```
struct IGR_Page_Pixels
{
    IGR_ULONG width;
    IGR_ULONG height;
    IGR_LONG stride;
    IGR_OPEN_BITMAP_PIXEL_TYPE pixel_format;
    IGR_OPEN_BITMAP_FLAGS_TYPE flags;
    void* scanline0;
    IGR_ULONG palette_count;
    IGR_LONG palette[256];
    void* reserved;
};
```

DETAILS

width: IGR_ULONG

Number of pixel in one scan line of the extracted pixels.

height: IGR_ULONG

Number of scanlines extracted.

stride: IGR_ULONG

Offset, in bytes, between consecutive scan lines of the page. If the stride is positive, the scan lines are top-down. If the stride is negative, the scan lines are bottom-up.

pixel_format: IGR_OPEN_BITMAP_PIXEL_TYPE

Integer that specifies the pixel format of the extracted pixels.

flags: IGR_OPEN_BITMAP_FLAGS_TYPE

Integer that specifies any pixel format flags

scanline0: Pointer

Pointer to the first scan line extracted.

palette_count: IGR_ULONG

Indicates the number of palette colors extracted, if any

`palette: Array of 256 IGR_LONG`

Indicates the palette colors extracted

`reserved: Pointer`

Reserved for internal use

IGR_Page_Word data type

The IGR_Page_Word structure is used by the [IGR_Get_Page_Words](#) method to return information such as position, dimensions, and text about the words on a given page.

DEFINITIONS

C/C++

```
struct IGR_Page_Word
{
    int x;
    int y;
    int width;
    int height;
    int style;
    const IGR_UCS2 *word;
    int wordLength;
    int charOffset;
};
```

DETAILS

X: Integer

Contains the X coordinate of the word in pixels.

Y: Integer

Contains the Y coordinate of the word in pixels.

Width: Integer

Contains the Width dimension of the word in pixels.

Height: Integer

Contains the Height dimension of the word in pixels.

Style: Integer

Is a bitmask containing any style information known about the word; see [Text Styles](#) for details.

Word: Unicode String

Contains the Unicode String of the word.

WordLength: Integer

Contains the length of the word pointed to by Word.

CharOffset: Integer

Contains the character offset of the word in the current page.

SEE ALSO

- [IGR_Get_Page_Words](#)

IGR_QuadPoint data type

Structure representing the points for a rectangle, in clockwise order.

PROTOTYPE

C/C++

```
struct IGR_QuadPoint
{
    union {
        struct {
            struct IGR_FPoint upperLeft;
            struct IGR_FPoint upperRight;
            struct IGR_FPoint lowerRight;
            struct IGR_FPoint lowerLeft;
        };
        struct IGR_FPoint points[4];
    };
};
```

DETAILS

`upperLeft: IGR_FPoint`

Upper Left position of the quad rect.

`upperRight: IGR_FPoint`

Upper Right position of the quad rect.

`lowerRight: IGR_FPoint`

Upper Lower Right position of the quad rect.

`lowerLeft: IGR_FPoint`

Upper Lower Left position of the quad rect.

IGR_Rect data type

The IGR_Rect structure is used to specify the position of a rectangle.

DEFINITIONS

C/C++

```
struct IGR_Rect
{
    IGR_ULONG left;
    IGR_ULONG top;
    IGR_ULONG right;
    IGR_ULONG bottom;
};
```

DETAILS

`left: IGR_ULONG`

Left positional value of the rectangle

`top: IGR_ULONG`

Top positional value of the rectangle

`right: IGR_ULONG`

Right positional value of the rectangle

`bottom: IGR_ULONG`

Bottom positional value of the rectangle

IGR_Render_Page_Form_Values data type

The IGR_Render_Page_Form_Values structure is used within [IGR_Render_Page_Properties](#) to specify new values for form values.

DEFINITIONS

C/C++

```
struct IGR_Render_Page_Form_Values
{
    const IGR_UCS2 * name;
    const IGR_UCS2 * value;
    int selected;
};
```

DETAILS

name: Pointer to const IGR_UCS2

Name of the form value to update

value: Pointer to const IGR_UCS2

Value to be replaced. If the form element type is a multi-select list box this may contain the name of multiple elements separated by space

selected: int

For elements of type Check Box or Radio Button a value of 1 sets the state to On and a value of 0 sets the state to Off

SEE ALSO

- [IGR_Render_Page_Ex](#)
- [IGR_Render_Page_Properties](#)

IGR_Render_Page_Properties data type

The IGR_Render_Page_Properties structure is used by the [IGR_Render_Page_Ex](#) method and provides additional information for rendering a page to a canvas.

DEFINITIONS

C/C++

```
struct IGR_Render_Page_Properties
{
    IGR_ULONG struct_size;
    IGR_ULONG form_value_count;
    const struct IGR_Render_Page_Form_Values * form_values;
    IGR_REDACT_FLAGS_TYPE redaction_flags;
    IGR_ULONG redaction_count;
    const struct IGR_Render_Page_Redactions * redactions;
    struct IGR_Rect source_rect;
    struct IGR_Rect dest_rect;
};
```

DETAILS

struct_size: IGR_ULONG

Size of this structure used to determine available fields. Set to sizeof(IGR_Render_Page_Properties) before using this structure.

form_value_count: IGR_ULONG

Number of elements in the form_values array

form_values: const struct IGR_Render_Page_Form_Values *

Pointer to an array of const struct [IGR_Render_Page_Form_Values](#) objects matching the length specified in form_value_count

IGR_REDACT_FLAGS_TYPE: IGR_ULONG

Indicates the type of action to execute when redacting. Can be one or more of:

- IGR_REDACT_FLAGS_DO_NOT_REDACT_TEXT : 0x1
- IGR_REDACT_FLAGS_DO_NOT_REDACT_IMAGES : 0x2
- IGR_REDACT_FLAGS_ALWAYS_RASTER : 0x4
- IGR_REDACT_FLAGS_NEVER_RASTER : 0x8
- IGR_REDACT_FLAGS_DO_NOT_DRAW_REGION : 0x10

`redaction_count: IGR_ULONG`

Number of items in the redaction array

`redactions: IGR_Render_Page_Redactions`

Pointer to an array of const struct [IGR_Render_Page_Redactions](#) objects matching the length specified in `redaction_count`

`source_rect: IGR_Rect`

Source rect measured in pixels

`dest_rect: IGR_Rect`

Destination rect measured in pixels

SEE ALSO

- [IGR_Render_Page_Ex](#)
- [IGR_Render_Page_Form_Values](#)
- [IGR_Render_Page_Redactions](#)
- [IGR_Rect](#)

IGR_Render_Page_Redactions data type

Structure representing a redaction area for rendering on a page. The redaction area is defined by its position (x, y) and size (width, height), along with the color to use for rendering and additional reserved data.

PROTOTYPE

C/C++

```
struct IGR_Render_Page_Redactions
{
    IGR_LONG x;
    IGR_LONG y;
    IGR_LONG width;
    IGR_LONG height;
    IGR_ULONG color;
    IGR_ULONG reserved;
}
```

DETAILS

x: IGR_LONG

X-coordinate of the redaction area.

y: IGR_LONG

Y-coordinate of the redaction area.

width: IGR_LONG

Width of the redaction area.

height: IGR_LONG

Height of the redaction area.

color: IGR_ULONG

Color type used for rendering the redaction.

reserved: IGR_ULONG

Reserved data for future use.

IGR_SRect data type

Structure representing a rectangle with signed integer coordinates. The coordinates are defined by the left, top, right, and bottom edges.

PROTOTYPE

C/C++

```
struct IGR_SRect
{
    IGR_LONG left;
    IGR_LONG top;
    IGR_LONG right;
    IGR_LONG bottom;
}
```

DETAILS

`left: IGR_LONG`

Left edge coordinate.

`top: IGR_LONG`

Top edge coordinate.

`right: IGR_LONG`

Right edge coordinate.

`bottom: IGR_LONG`

Bottom edge coordinate.

IGR_Size data type

The IGR_Size structure is used to specify the size of a rectangle.

DEFINITIONS

C/C++

```
struct IGR_Size
{
    IGR_ULONG width;
    IGR_ULONG height;
};
```

DETAILS

width: IGR_ULONG

The width of the item in pixels.

height: IGR_ULONG

The height of the item in pixels.

IGR_Stream data type

The IGR_Stream data type allows for the creation of custom input streams.

DEFINITIONS

C/C++

```
struct IGR_Stream
{
    unsigned long (CALLBACK* Open) (IGR_Stream*);
    unsigned long (CALLBACK* Read) (IGR_Stream*, void *, unsigned long);
    LONGLONG (CALLBACK* Seek) (IGR_Stream*, LONGLONG, unsigned long);
    void (CALLBACK* Close) (IGR_Stream*);
};
```

DETAILS

Open: Function Pointer

A pointer to an application supplied function that will be called when Document Filters attempts to **open** the document.

Read: Function Pointer

A pointer to an application supplied function that will be called when Document Filters attempts to **read** the document.

Seek: Function Pointer

A pointer to an application supplied function that will be called when Document Filters attempts to **seek** within the document.

Close: Function Pointer

A pointer to an application supplied function that will be called when Document Filters attempts to **close** the document.

ADDITIONAL INFORMATION

The helper functions [IGR_Make_Stream_From_File](#) and [IGR_Make_Stream_From_Memory](#) are provided to enable applications to easily instantiate streams from these common sources. The stream created by these functions must be freed by the application calling the stream's close function.

All functions above need to be implemented to stream documents from custom sources.

SEE ALSO

- [IGR_Open_Stream](#)
- [IGR_Open_Stream_Ex](#)

IGR_Subfile_Info data type

Used to return subfile information from the IGR_Get_Subfile_Entry_Ex C API.

DEFINITIONS

C/C++

```
struct IGR_Subfile_Info
{
    IGR_ULONG struct_size;
    IGR_ULONG flags;
    IGR_UCS2* id;
    IGR_ULONG id_size;
    IGR_UCS2* name;
    IGR_ULONG name_size;
    IGR_UCS2* comment;
    IGR_ULONG comment_size;
    IGR_ULONGLONG size;
    IGR_ULONGLONG date;
};
```

DETAILS

struct_size: IGR_ULONG

Indicate the size of this structure, must be populated

flags: IGR_ULONG

Bitmask of [IGR_SUBFILE_INFO_FLAG_TYPE](#) values.

id: IGR_UCS2*

Buffer to be populated with the id, up to id_size characters.

id_size: IGR_ULONG

The number of IGR_UCS2 characters pointed to by id.

name: IGR_UCS2*

Buffer to be populated with the name, up to name_size characters.

name_size: IGR_ULONG;

The number of IGR_UCS2 characters pointed to by name.

comment: IGR_UCS2* ;

Buffer to be populated with the comment, up to `comment_size` characters.

`comment_size: IGR_ULONG`

The number of `IGR_UCS2` characters pointed to by comment.

`size: IGR_ULONGLONG`

Contains the size in bytes of the sub-document.

`date: IGR_ULONGLONG`

Contains the date and time of the sub-document in FILETIME format.

ADDITIONAL INFORMATION

`struct_size` must be set to `sizeof(IGR_Subfile_Info)`.

The caller must assign a buffer and it's size to retrieve strings for id, name and comments. To ignore any of these value, assign it's buffer to NULL and its size to 0.

SEE ALSO

- [IGR_Get_Subfile_Entry_Ex](#)

IGR_T_ACTION_GET_STREAM_PART data type

When a second or later part of a multi-part archive is required as part of processing an Extended Stream, a callback will request this structure to be populated with a new stream instance.

DEFINITION

C/C++

```
struct IGR_T_ACTION_GET_STREAM_PART
{
    uint32_t size;
    const wchar_t* partName;
    const wchar_t* partFullName;
    int partIndex;
    IGR_Stream* istr;
};
```

DETAILS

size: uint32_t

(in) The size of this structure in bytes.

partName: const wchar_t*

(in) The file name, if available (may be null).

partFullName: const wchar_t*

(in) The file name and path, if available (may be null).

partIndex: int

(in) The number of the file part required (guaranteed to be correct).

istr: IGR_Stream*

(out) User creates the requested stream and stores the pointer here.

ADDITIONAL INFORMATION

This structure is allocated by Document Filters and passed to the API user's code through a callback with actionID = IGR_ACTION_GET_STREAM_PART. The user should create a stream for the requested file part and store a pointer to that stream in the istr member of the struct.

SEE ALSO

- [IGR_Extend_Stream](#)

IGR_Text_Compare_Document_Source data type

Structure representing a source document for text comparison.

PROTOTYPE

C/C++

```
struct IGR_Text_Compare_Document_Source
{
    IGR_ULONG struct_size;
    IGR_LONG doc_handle;
    IGR_ULONG doc_first_page;
    IGR_ULONG doc_page_count;
    struct IGR_FRect doc_margins;
}
```

DETAILS

`struct_size: IGR_ULONG`

Size of this structure.

`doc_handle: IGR_LONG`

The handle to an open document.

`doc_first_page: IGR_ULONG`

The index to the first page to compare from document.

`doc_page_count: IGR_ULONG`

The number of pages from `doc1_first_page` to compare.

`doc_margins: struct IGR_FRect`

The margins, in pts, to exclude from the comparison for document.

IGR_Text_Compare_Settings data type

Settings structure for text comparison operations.

PROTOTYPE

C/C++

```
struct IGR_Text_Compare_Settings
{
    IGR_ULONG struct_size;
    IGR_ULONG compare_type;
    IGR_ULONG flags;
    void* user_context;
    IGR_ULONG comparison_window;
    IGR_ULONG comparison_overlap;
    void* reserved1;
    void* reserved2;
    void* reserved3;
    void* reserved4;
    IGR_RETURN_CODE(CALLBACK* OpenPage) (void* context, const struct IGR_Compare_Documents_Callback_Context*
info, IGR_ULONG page_index, IGR_HPAGE* page_handle);
    IGR_RETURN_CODE(CALLBACK* ClosePage) (void* context, const struct
IGR_Compare_Documents_Callback_Context* info, IGR_HPAGE page_handle);
    IGR_RETURN_CODE(CALLBACK* GetPageMargins)(void* context, const struct
IGR_Compare_Documents_Callback_Context* info, struct IGR_FRect* margins);
    IGR_RETURN_CODE(CALLBACK* GetPageArea)(void* context, const struct
IGR_Compare_Documents_Callback_Context* info, struct IGR_FRect* page_area);
    IGR_RETURN_CODE(CALLBACK* GetNextPageIndex) (void* context, const struct
IGR_Compare_Documents_Callback_Context* info, IGR_ULONG* page_index);
    IGR_RETURN_CODE(CALLBACK* HandleError) (void* context, const struct
IGR_Compare_Documents_Callback_Context* info, const Error_Control_Block* ecb);
}
```

DETAILS

struct_size: IGR_ULONG

Size of this structure.

compare_type: IGR_ULONG

The type of comparison to perform.

flags: IGR_ULONG

Flags that control the comparison logic.

user_context: Pointer to void

Pointer to user supplied context that is passed as /context to each of the callbacks.

`comparison_window: IGR_ULONG`

Reserved.

`comparison_overlap: IGR_ULONG`

Reserved.

`reserved1: Pointer to void`

Reserved for future use.

`reserved2: Pointer to void`

Reserved for future use.

`reserved3: Pointer to void`

Reserved for future use.

`reserved4: Pointer to void`

Reserved for future use.

`OpenPage: Pointer to Callback`

Optional callback that is called to provide when the page resource is required.

`ClosePage: Pointer to Callback`

Optional callback that is called to provide when the page resource can be destroyed.

`GetPageMargins: Pointer to Callback`

Optional callback that is called to provide the margins for a given page. User can modify the left, top, right, margins of the page. Units are in pixels.

`GetPageArea: Pointer to Callback`

Optional callback that is called with the page area where text is to be diffed.

`GetNextPageIndex: Pointer to Callback`

Optional callback that is called to provide the next page number to process.

`HandleError: Pointer to Callback`

Optional callback that is called to when an error is encountered.

IGR_Writable_Stream data type

Similar to the IGR_Stream, the IGR_Writable_Stream data type allows for the creation of custom output streams. It's important to note, all methods must be implemented, not just the Write.

DEFINITIONS

C/C++

```
struct IGR_Writable_Stream
{
    struct IGR_Stream Base;
    unsigned long (CALLBACK* Write)(IGR_Writable_Stream* stream, void * buffer, unsigned long bufferSize);
};
```

DETAILS

Base.Open: Function Pointer

A pointer to an application supplied function that will be called when Document Filters attempts to **open** the document.

Base.Read: Function Pointer

A pointer to an application supplied function that will be called when Document Filters attempts to **read** the document.

Base.Seek: Function Pointer

A pointer to an application supplied function that will be called when Document Filters attempts to **seek** within the document.

Base.Close: Function Pointer

A pointer to an application supplied function that will be called when Document Filters attempts to **close** the document.

Write: Function Pointer

A pointer to an application supplied function that will be called when Document Filters attempts to **write** to the output.

ADDITIONAL INFORMATION

All functions above need to be implemented to stream to custom output sources.

SEE ALSO

- [IGR_Make_Output_Canvas_On](#)

Instance_Status_Block data type

The Instance_Status_Block data type is a control block returned by [Init_Instance](#) with information about the installed Document Filters engine.

DEFINITIONS

C/C++

```
struct Instance_Status_Block
{
    char DLL_Version[16];
    short Reserved;
    short License_Type;
    char Licensee_ID1[40];
    char Licensee_ID2[40];
};
```

SEE ALSO

- [Init_Instance](#)

3.2.4 Constants

Overview

IGR_RETURN_CODE	Indicates the success or failure of a call.
IGR_ANNOTATION_FLAGS_TYPE	Contains the annotation flags.
IGR_ANNOTATION_TYPE	Contains the annotation type.
IGR_BRUSH_STYLE	Contains the brush types available when calling <code>IGR_Canvas_SetBrush</code> or <code>Canvas.SetBrush</code> .
IGR_CANVAS_TYPE	Canvas Types specify the type of output to create when rendering a page. They are passed to the IGR_Make_Output_Canvas and DocumentFilters::MakeOutputCanvas functions.
IGR_CAPABILITY	Document Capabilities are a bitmask representing the operations that may be performed on a document.
IGR_CHAR_CODE	These character codes are output by Document Filters when using the <code>IGR_Get_Text</code> function and <code>Extractor.GetText</code> .
IGR_COMPARE_DOCUMENTS_COMPARE_TYPE	Contains the list of document comparison types.
IGR_COMPARE_DOCUMENTS_DIFFERENCE_SOURCE_TYPE	Contains the list of comparison difference source types.
IGR_COMPARE_DOCUMENTS_DIFFERENCE_TYPE	Contains the list of comparison difference types.
IGR_COMPARE_DOCUMENTS_FLAGS_TYPE	Contains the list of comparison flags.
IGR_FILETYPE_CATEGORY	

	The File Type Category provides extra classification of a file type. It is returned by calls to IGR_Get_Format_Attribute .
IGR_FONT_STYLE	These font styles are used with the IGR_Canvas_SetFont method.
IGR_GET_PAGE_PIXELS_FLAGS_TYPE	Contains the flags available when calling IGR_Page_Get_Pixels .
IGR_LOG_LEVEL_TYPE	IGR_LOG_LEVEL_TYPE is used by IGR_Open_Callback_Action_Log_Level when determining the log-level for a module.
IGR_OPEN_BITMAP_FLAGS_TYPE	IGR_OPEN_BITMAP_FLAGS_TYPE is used by IGR_Open_DIB_Info and IGR_Open_DIB when creating a Document Filters document from an existing in-memory image.
IGR_OPEN_BITMAP_PIXEL_TYPE	IGR_OPEN_BITMAP_PIXEL_TYPE is used by IGR_Open_DIB_Info and IGR_Open_DIB when creating a Document Filters document from an existing in-memory image; or IGR_Get_Page_Pixels when accessing pixels of a page.
IGR_OPEN_CALLBACK_ACTION	IGR_OPEN_CALLBACK_ACTION is used by IGR_OPEN_CALLBACK and IGR_Open_Ex when opening a document for text extraction or enumeration of sub-documents.
IGR_OPEN_FLAGS	Open Document Flags specify the type of content that is to be retrieved from the document. The flags are a bitwise combination of the values below.
IGR_OPEN_FROM	IGR_OPEN_FROM is used by IGR_Open_Ex when opening a

	document for text extraction or enumeration of sub-documents.
IGR_PAGE_ELEMENT_TYPE	Contains the list of form element types.
IGR_PAGE_FORM_ELEMENT_FLAG_TYPE	Contains the form element type flags.
IGR_PAGE_FORM_ELEMENT_GET_TYPE	Contains the options when requesting form element data.
IGR_PAGE_FORM_ELEMENT_TYPE	Contains the form element type flags.
IGR_PEN_STYLE	Contains the pen types available when calling <code>IGR_Canvas_SetPen</code> or <code>Canvas.SetPen</code> .
IGR_STRING_ID	Contains the list of strings that can be localized.
IGR_SUBFILE_INFO_FLAG_TYPE	Contains the subfile flag information.
IGR_TEXT_STYLE	Contains the text style types available when calling <code>IGR_Canvas_DrawText</code> or <code>Canvas.DrawText</code> .

Result Codes

SUCCESSFUL RESULT CODES

Name	Value
IGR_OK	0
IGR_NO_MORE	19

UNSUCCESSFUL RESULT CODES

Name	Value	Description
IGR_E_OPEN_ERROR	1	Open error
IGR_E_WRONG_TYPE	2	Attempt to open file with wrong type
IGR_E_IN_USE	3	File is locked
IGR_E_NOT_READABLE	4	File is not readable
IGR_E_PASSWORD	5	File is encrypted
IGR_E_NOT_FOUND	10	File not found
IGR_E_WRITE_ERROR	11	Unable to write file
IGR_E_NOT_VALID_FOR_THIS_CLASS	12	Operation is not valid for this type of object
IGR_E_ERROR	13	General error
IGR_E_INVALID_HANDLE	14	Invalid handle
IGR_E_INVALID_POINTER	15	Invalid pointer
IGR_E_INVALID_PARAMETER	16	Invalid parameter
IGR_E_FILE_CORRUPT	17	Corrupt file
IGR_E_OUT_OF_MEMORY	20	Out of memory
IGR_E_BAD_ERROR	21	Bad error
IGR_E_WRONG_THREAD	22	Method called from wrong thread
IGR_E_TOO_MANY_HANDLES	23	Too many open handles

Name	Value	Description
IGR_E_FONTS_NOT_FOUND	24	Fonts not found
IGR_E_VERSION_MISMATCH	25	Version mismatch
IGR_E_ACTION_NOT_SUPPORTED	26	Action not supported
IGR_CANCELLED	27	Operation has been cancelled

IGR_ANNOTATION_FLAGS_TYPE

Name	Value	Description
IGR_ANNOTATION_FLAGS_INVISIBLE	0x001	Indicates the annotation should not be displayed if the annotation type is unknown.
IGR_ANNOTATION_FLAGS_HIDDEN	0x002	Indicates the annotation should not be displayed nor printed nor allowed to interact with the user.
IGR_ANNOTATION_FLAGS_PRINT	0x004	Indicates the annotation should be printed when the page is printed.
IGR_ANNOTATION_FLAGS_NOZOOM	0x008	Indicates the annotation's appearance should not be scaled to match the magnification of the page.
IGR_ANNOTATION_FLAGS_NOROTATE	0x010	Indicates the annotation's appearance should not be rotated to match the rotation of the page.
IGR_ANNOTATION_FLAGS_NOVIEW	0x020	Indicates the annotation should not be displayed nor allowed to interact with the user.
IGR_ANNOTATION_FLAGS_READONLY	0x040	Indicates the annotation should not be allowed to interact with the user.
IGR_ANNOTATION_FLAGS_LOCKED	0x080	Indicates the annotation should not be allowed to be deleted and its properties should not be allowed to be modified by the user.

Name	Value	Description
IGR_ANNOTATION_FLAGS_TOGGLENOVIEW	0x100	Indicates the interpretation of the annotation's NoView flag should be inverted for certain events.
IGR_ANNOTATION_FLAGS_LOCKEDCONTENTS	0x200	Indicates the annotation's contents should not be allowed to be modified by the user.

IGR_ANNOTATION_TYPE

Name	Value	Description
IGR_ANNOTATION_UNKNOWN	0	Indicates the annotation type is unknown.
IGR_ANNOTATION_TEXT	1	Indicates the annotation is a Text annotation.
IGR_ANNOTATION_LINK	2	Indicates the annotation is a Link annotation.
IGR_ANNOTATION_FREETEXT	3	Indicates the annotation is a Free Text annotation.
IGR_ANNOTATION_LINE	4	Indicates the annotation is a Line annotation.
IGR_ANNOTATION_SQUARE	5	Indicates the annotation is a Square annotation
IGR_ANNOTATION_CIRCLE	6	Indicates the annotation is a Circle annotation.
IGR_ANNOTATION_POLYGON	7	Indicates the annotation is a Polygon annotation.
IGR_ANNOTATION_POLYLINE	8	Indicates the annotation is a Polyline annotation.
IGR_ANNOTATION_HIGHLIGHT	9	Indicates the annotation is a Highlight annotation.
IGR_ANNOTATION_UNDERLINE	10	Indicates the annotation is a Underline annotation.
IGR_ANNOTATION_SQUIGGLE	11	Indicates the annotation is a Squiggly-underline annotation.
IGR_ANNOTATION_STRIKEOUT	12	Indicates the annotation is a Strikeout annotation.
IGR_ANNOTATION_STAMP	13	

Name	Value	Description
		Indicates the annotation is a Rubber Stamp annotation.
IGR_ANNOTATION_CARET	14	Indicates the annotation is a Caret annotation.
IGR_ANNOTATION_INK	15	Indicates the annotation is a Ink annotation.
IGR_ANNOTATION_POPUP	16	Indicates the annotation is a Pop-up annotation.
IGR_ANNOTATION_FILEATTACHMENT	17	Indicates the annotation is a File Attachment annotation.
IGR_ANNOTATION_SOUND	18	Indicates the annotation is a Sound annotation.
IGR_ANNOTATION_MOVIE	19	Indicates the annotation is a Movie annotation.
IGR_ANNOTATION_WIDGET	20	Indicates the annotation is a Widget annotation.
IGR_ANNOTATION_SCREEN	21	Indicates the annotation is a Screen annotation.
IGR_ANNOTATION_PRINTERMARK	22	Indicates the annotation is a Printer's Mark annotation.
IGR_ANNOTATION_TRAPNET	23	Indicates the annotation is a Trap Network annotation.
IGR_ANNOTATION_WATERMARK	24	Indicates the annotation is a Watermark annotation.
IGR_ANNOTATION_3D	25	Indicates the annotation is a 3D annotation.
IGR_ANNOTATION_BARCODE	26	Indicates the annotation is a Barcode annotation.

Brush Styles

Name	Value	Description
IGR_BRUSH_NONE	0	Indicates that objects should not be filled with the brush.
IGR_BRUSH_SOLID	1	Indicates that objects should be filled with the specified brush color.

Canvas Types

Canvas Types specify the type of output to create when rendering a page. They are passed to the [IGR_Make_Output_Canvas](#) and [DocumentFilters::MakeOutputCanvas](#) functions.

Name	Value	Description
IGR_DEVICE_HTML	6	Create a single or multipage HTML5
IGR_DEVICE_IMAGE_BMP	4	Create a single BMP per page
IGR_DEVICE_IMAGE_BRK	17	Create a single or multipage Brooktrout FAX
IGR_DEVICE_IMAGE_DCX	19	Create a single or multipage DCX
IGR_DEVICE_IMAGE_EPS	13	Create a single EPS (Encapsulated PostScript) per page
IGR_DEVICE_IMAGE_GIF	21	Create a single GIF per page
IGR_DEVICE_IMAGE_JPEG2000	20	Create a single JPEG2000 per page
IGR_DEVICE_IMAGE_JPG	1	Create a single JPG per page
IGR_DEVICE_IMAGE_PBM	7	Create a single PBM per page
IGR_DEVICE_IMAGE_PCX	18	Create a single PCX per page
IGR_DEVICE_IMAGE_PDF	2	Create a single or multipage PDF
IGR_DEVICE_IMAGE_PGM	8	Create a single PGM per page
IGR_DEVICE_IMAGE_PNG	0	Create a single PNG per page
IGR_DEVICE_IMAGE_PPM	9	Create a single PPM per page
IGR_DEVICE_IMAGE_PS	14	Create a single or multipage PostScript
IGR_DEVICE_IMAGE_SVG	12	Create a single SVG per page
IGR_DEVICE_IMAGE_TGA	16	Create a single TGA per page
IGR_DEVICE_IMAGE_TIF	3	Create a single or multipage TIF
IGR_DEVICE_IMAGE_WEBP	10	Create a single WEBP per page

Name	Value	Description
IGR_DEVICE_IMAGE_WEBSAFE	15	Create a single image per page, where the format is determined based on the palette and number of colors.
IGR_DEVICE_IMAGE_XPS	11	Create a single or multipage XPS
IGR_DEVICE_JSON	22	Create a single or multipage structured JSON
IGR_DEVICE_PDF	2	Create a single or multipage PDF (Alias for IGR_DEVICE_IMAGE_PDF)
IGR_DEVICE_XML	5	Create a single or multipage structured XML

Document capabilities

Document Capabilities are a bitmask representing the operations that may be performed on a document.

Name	Value	Description
IGR_FILE_SUPPORTS_TEXT	1	The document contains text that may be retrieved by calls to IGR_Get_Text.
IGR_FILE_SUPPORTS_SUBFILES	2	The document may contain sub-documents that can be enumerated with IGR_Get_Subfile_Entry and extracted with IGR_Extract_Subfile
IGR_FILE_SUPPORTS_HDHTML	4	The document may be converted to high-definition, preserving the document layout, font colors, styles, tables, and images.

Character codes

These character codes are output by Document Filters when using the IGR_Get_Text function and Extractor.GetText.

Character Value	Description
1	Soft space
2	Reset normal Style
3	Start bold
4	Start italics
5	Start underlined
9	TAB
12	Page break
13	Line break
14	Paragraph break
15	Start of metadata
16	End of metadata

Note End of lines are always specified by character 13.

IGR_COMPARE_DOCUMENTS_COMPARE_TYPE

Name	Value	Description
IGR_COMPARE_DOCUMENTS_COMPARE_WORDS	0x0	Perform word-level comparison.

IGR_COMPARE_DOCUMENTS_DIFFERENCE_SOURCE_TYPE

Name	Value	Description
IGR_COMPARE_DOCUMENTS_DIFFERENCE_SOURCE_ORIGINAL	0x0	Diff was generated for the original document.
IGR_COMPARE_DOCUMENTS_DIFFERENCE_SOURCE_REVISED	0x1	Diff was generated for the revised document.
IGR_COMPARE_DOCUMENTS_DIFFERENCE_SOURCE_BOTH	0x2	Diff was generated against both documents.

IGR_COMPARE_DOCUMENTS_DIFFERENCE_TYPE

Name	Value	Description
IGR_COMPARE_DOCUMENTS_DIFFERENCE_EQUAL	0	Emitted when IGR_COMPARE_DOCUMENTS_F flag is set; indicates that the text appears in both documents.
IGR_COMPARE_DOCUMENTS_DIFFERENCE_INSERT	1	Indicates that the text was inserted into the document.
IGR_COMPARE_DOCUMENTS_DIFFERENCE_DELETE	2	Indicates that the text was deleted from the original document.
IGR_COMPARE_DOCUMENTS_DIFFERENCE_FORMATTING	3	Indicates that formatting was changed.
IGR_COMPARE_DOCUMENTS_DIFFERENCE_NEXT_BATCH	0xFF	Indicates that the comparer is about to process a new batch of content to compare.

IGR_COMPARE_DOCUMENTS_FLAGS_TYPE

Name	Value
IGR_COMPARE_DOCUMENTS_FLAGS_EQUALS	0x1
IGR_COMPARE_DOCUMENTS_FLAGS_MOVES	0x10
IGR_COMPARE_DOCUMENTS_FLAGS_FORMATTING	0x20
IGR_COMPARE_DOCUMENTS_FLAGS_NO_COMMENTS	0x40
IGR_COMPARE_DOCUMENTS_FLAGS_NO_CASE	0x80
IGR_COMPARE_DOCUMENTS_FLAGS_NO_WHITESPACE	0x100
IGR_COMPARE_DOCUMENTS_FLAGS_NO_PUNCTUATION	0x200
IGR_COMPARE_DOCUMENTS_FLAGS_NO_TABLES	0x400
IGR_COMPARE_DOCUMENTS_FLAGS_NO_HEADERS	0x800
IGR_COMPARE_DOCUMENTS_FLAGS_NO_FOOTERS	0x1000
IGR_COMPARE_DOCUMENTS_FLAGS_NO_FOOTNOTES	0x2000
IGR_COMPARE_DOCUMENTS_FLAGS_NO_TEXTBOXES	0x4000
IGR_COMPARE_DOCUMENTS_FLAGS_NO_FIELDS	0x8000
IGR_COMPARE_DOCUMENTS_FLAGS_NO_HEADERS_FOOTERS	(IGR_COMPARE_DOCUMENTS_FLAGS_NO_HEADERS IGR_COMPARE_DOCUMENTS_FLAGS_NO_FOOTERS)

IGR_FILETYPE_CATEGORY type

The File Type Category provides extra classification of a file type. It is returned by calls to [IGR_Get_Format_Attribute](#).

Each value that is devisable by 100 with no remainder is a top-level category, where values with a remainder is a sub-category.

For example, `IGR_FILETYPE_GRAPHIC = 1000` is a top-level; where `IGR_FILETYPE_GRAPHIC_RASTER = 1001` is a sub-category of `IGR_FILETYPE_GRAPHIC`.

Name	Value
IGR_FILETYPE_UNKNOWN	0
IGR_FILETYPE_TEXT	100
IGR_FILETYPE_TEXT_MARKUP	101
IGR_FILETYPE_WORD_PROCESSOR	200
IGR_FILETYPE_SPREADSHEET	300
IGR_FILETYPE_PRESENTATION	400
IGR_FILETYPE_OFFICE_OTHER	500
IGR_FILETYPE_CAD	600
IGR_FILETYPE_ARCHIVE	700
IGR_FILETYPE_SYSTEM	800
IGR_FILETYPE_DATABASE	900
IGR_FILETYPE_GRAPHIC	1000
→ IGR_FILETYPE_GRAPHIC_RASTER	1001
→ IGR_FILETYPE_GRAPHIC_VECTOR	1002
IGR_FILETYPE_EMAIL	1100
→ IGR_FILETYPE_EMAIL_MESSAGE	1101
→ IGR_FILETYPE_EMAIL_CONTAINER	1102

Name	Value
→ IGR_FILETYPE_EMAIL_CONTACT	1103
→ IGR_FILETYPE_EMAIL_CALEDAR	1104
→ IGR_FILETYPE_EMAIL_OTHER	1105
IGR_FILETYPE_MULTIMEDIA	1200
→ IGR_FILETYPE_MULTIMEDIA_VIDEO	1201
→ IGR_FILETYPE_MULTIMEDIA_AUDIO	1202
IGR_FILETYPE_PUBLISHING	1300
IGR_FILETYPE_APPDATA	2000

Font styles

These font styles are used with the IGR_Canvas_SetFont method.

Character Value	Description
0x01	FONT_STYLE_BOLD
0x02	FONT_STYLE_ITALICS
0x04	FONT_STYLE_UNDERLINE
0x08	FONT_STYLE_STRIKEOUT
0x10	FONT_STYLE_SERIF
0x20	FONT_STYLE_MONO
0x40	FONT_STYLE_RTL

IGR_GET_PAGE_PIXELS_FLAGS_TYPE

Name	Value	Description
IGR_GET_PAGE_PIXELS_FLAGS_BUFFER_ALLOCATED	0x00010000	Indicates that the caller has provided the memory that is to be populated.

IGR_LOG_LEVEL_TYPE

IGR_LOG_LEVEL_TYPE is used by [IGR_Open_Callback_Action_Log_Level](#) when determining the log-level for a module.

Name	Value
IGR_LOG_LEVEL_NONE	0
IGR_LOG_LEVEL_CRITICAL	1
IGR_LOG_LEVEL_ERROR	2
IGR_LOG_LEVEL_WARNING	3
IGR_LOG_LEVEL_INFORMATION	4
IGR_LOG_LEVEL_DEBUG	5
IGR_LOG_LEVEL_TRACE	6

IGR_OPEN_BITMAP_FLAGS_TYPE

IGR_OPEN_BITMAP_FLAGS_TYPE is used by [IGR_Open_DIB_Info](#) and [IGR_Open_DIB](#) when creating a Document Filters document from an existing in-memory image.

Name	Value	Description
IGR_OPEN_BITMAP_FLAGS_MIN_IS_BLACK	1	For 1 bit images, indicates if 0 is black or white.
IGR_OPEN_BITMAP_FLAGS_BOTTOM_UP	2	Specifies that the image is to be processed bottom-up.

IGR_OPEN_BITMAP_PIXEL_TYPE

IGR_OPEN_BITMAP_PIXEL_TYPE is used by [IGR_Open_DIB_Info](#) and [IGR_Open_DIB](#) when creating a Document Filters document from an existing in-memory image; or [IGR_Get_Page_Pixels](#) when accessing pixels of a page.

Name	Value	Description
IGR_OPEN_BITMAP_PIXEL_1BPP_INDEXED	1	Specifies the image is 1 bit per pixel, indexed.
IGR_OPEN_BITMAP_PIXEL_4BPP_INDEXED	2	Specifies the image is 4 bits per pixel, indexed.
IGR_OPEN_BITMAP_PIXEL_8BPP_INDEXED	3	Specifies the image is 8 bits per pixel, indexed.
IGR_OPEN_BITMAP_PIXEL_16BPP_565_RGB	4	Specifies the image is 16 bits per pixel; 5 bits for red and blue, and 6 for green.
IGR_OPEN_BITMAP_PIXEL_16BPP_565_BGR	5	Specifies the image is 16 bits per pixel; 5 bits for red and blue, and 6 for green.
IGR_OPEN_BITMAP_PIXEL_16BPP_4444_ARGB	6	Specifies the image is 16 bit per pixel; 4 bits for alpha, red, green and blue.
IGR_OPEN_BITMAP_PIXEL_16BPP_4444_BGRA	7	Specifies the image is 16 bit per pixel; 4 bits for alpha, red, green and blue.
IGR_OPEN_BITMAP_PIXEL_24BPP_888_RGB	8	Specifies the image is 24 bit per pixel; 8 bits for red, green and blue.
IGR_OPEN_BITMAP_PIXEL_24BPP_888_BGR	9	Specifies the image is 24 bit per pixel; 8 bits for red, green and blue.
IGR_OPEN_BITMAP_PIXEL_32BPP_8888_ARGB	10	Specifies the image is 32 bit per pixel; 8 bits for alpha, red, green and blue.

Name	Value	Description
IGR_OPEN_BITMAP_PIXEL_32BPP_8888_BGRA	11	Specifies the image is 32 bit per pixel; 8 bits for alpha, red, green and blue.
IGR_OPEN_BITMAP_PIXEL_32BPP_8888_RGBA	12	Specifies the image is 32 bit per pixel; 8 bits for alpha, red, green and blue.
IGR_OPEN_BITMAP_PIXEL_32BPP_8888_ABGR	13	Specifies the image is 32 bit per pixel; 8 bits for alpha, red, green and blue.

IGR_OPEN_CALLBACK_ACTION

IGR_OPEN_CALLBACK_ACTION is used by [IGR_OPEN_CALLBACK](#) and [IGR_Open_Ex](#) when opening a document for text extraction or enumeration of sub-documents.

Name	Value	Description
IGR_OPEN_CALLBACK_ACTION_HEARTBEAT	0	Specifies the type of callback action is heartbeat.
IGR_OPEN_CALLBACK_ACTION_PASSWORD	1	Specifies the type of callback action is password.
IGR_OPEN_CALLBACK_ACTION_LOCALIZE	2	Specifies the type of callback action is localize.
IGR_OPEN_CALLBACK_ACTION_LOG_LEVEL	3	Specifies the type of callback action is log-level.
IGR_OPEN_CALLBACK_ACTION_LOG_MESSAGE	4	Specifies the type of callback action is log-message.
IGR_OPEN_CALLBACK_ACTION_APPROVE_EXTERNAL_RESOURCE	5	Specifies the type of callback action is approve-

Name	Value	Description
IGR_OPEN_CALLBACK_ACTION_GET_RESOURCE_STREAM	6	external-resource. Specifies the type of callback action is get-resource-stream.
IGR_OPEN_CALLBACK_ACTION_OCR_IMAGE	7	Specifies the type of callback action is ocr-image.

Open Document Flags

Open Document Flags specify the type of content that is to be retrieved from the document. The flags are a bitwise combination of the values below.

Name	Value	Description
IGR_BODY_ONLY	0	Only the document text will be returned.
IGR_META_ONLY	1	Only the metadata will be returned.
IGR_BODY_AND_META	2	Both the metadata and the document text will be returned.
IGR_FORMAT_TEXT	0x00010000	The document will be converted to plain text.
IGR_FORMAT_HTML	0x00020000	The document will be converted to HTML.
IGR_FORMAT_HDHTML_IFYOUCAN	0x00040000	Will perform an HD/Classic HTML conversion if the format supports HD, otherwise it will remain in text mode. Note This differs from IGR_FORMAT_HTML, which will “upscale” non-HD formats.
IGR_FORMAT_IMAGE	0x00050000	The document will be converted, via HD paginated mode, to be rendered to an output device.
IGR_FORMAT_MARKDOWN	0x00080000	The document will be converted to markdown.

IGR_OPEN_FROM

IGR_OPEN_FROM is used by [IGR_Open_Ex](#) when opening a document for text extraction or enumeration of sub-documents.

Name	Value	Description
IGR_OPEN_FROM_FILENAME_UTF16	0	Specifies the source is a UTF-16 string containing the filename.
IGR_OPEN_FROM_STREAM	1	Specifies the source is an IGR_Stream record.

IGR_PAGE_ELEMENT_TYPE

Name	Value
IGR_PAGE_ELEMENT_TYPE_PAGE	1
IGR_PAGE_ELEMENT_TYPE_SECTION	2
IGR_PAGE_ELEMENT_TYPE_COLUMN	3
IGR_PAGE_ELEMENT_TYPE_PARAGRAPH	4
IGR_PAGE_ELEMENT_TYPE_LINE	5
IGR_PAGE_ELEMENT_TYPE_RUN	6
IGR_PAGE_ELEMENT_TYPE_WORD	7
IGR_PAGE_ELEMENT_TYPE_HEADER	8
IGR_PAGE_ELEMENT_TYPE_FOOTER	9
IGR_PAGE_ELEMENT_TYPE_IMAGE	10
IGR_PAGE_ELEMENT_TYPE_TABLE	11
IGR_PAGE_ELEMENT_TYPE_TABLE_ROW	12
IGR_PAGE_ELEMENT_TYPE_TABLE_CELL	13
IGR_PAGE_ELEMENT_TYPE_FORM	14
IGR_PAGE_ELEMENT_TYPE_FORM_ELEMENT	15
IGR_PAGE_ELEMENT_TYPE_FLOAT	16
IGR_PAGE_ELEMENT_TYPE_GRAPHIC	17
IGR_PAGE_ELEMENT_TYPE_TEXT_BOX	18

IGR_PAGE_FORM_ELEMENT_FLAG_TYPE

Name	Value	Description
IGR_PAGE_FORM_ELEMENT_FLAG_READONLY	1	Indicates the form element is read-only.
IGR_PAGE_FORM_ELEMENT_FLAG_REQUIRED	2	Indicates the form element is marked as required.
IGR_PAGE_FORM_ELEMENT_FLAG_COMB	4	Indicates that each character should be it's own box.
IGR_PAGE_FORM_ELEMENT_FLAG_CHECKED	8	Indicates the form element is checked.

IGR_PAGE_FORM_ELEMENT_GET_TYPE

Name	Value	Description
IGR_PAGE_FORM_ELEMENT_GET_NAME	0	Requests the form elements name.
IGR_PAGE_FORM_ELEMENT_GET_VALUE	1	Requests the form elements current value.
IGR_PAGE_FORM_ELEMENT_GET_CAPTION	2	Requests the form elements caption.
IGR_PAGE_FORM_ELEMENT_GET_ACTION	3	Requests the form elements Action.
IGR_PAGE_FORM_ELEMENT_GET_ACTION_DEST	4	Requests the form elements Action Destination.
IGR_PAGE_FORM_ELEMENT_GET_SELECTED	5	Requests the form elements currently selected value.
IGR_PAGE_FORM_ELEMENT_GET_FONT_NAME	6	Requests the form elements font name.

IGR_PAGE_FORM_ELEMENT_TYPE

Name	Value	Description
IGR_PAGE_FORM_ELEMENT_TYPE_BUTTON	0	Indicates the form element is a push button.
IGR_PAGE_FORM_ELEMENT_TYPE_RADIOBUTTON	1	Indicates the form element is a radio button.
IGR_PAGE_FORM_ELEMENT_TYPE_CHECKBOX	2	Indicates the form element is a check box.
IGR_PAGE_FORM_ELEMENT_TYPE_FILESELECT	3	Indicates the form element is a file selection box.
IGR_PAGE_FORM_ELEMENT_TYPE_MULTILINE_TEXT	4	Indicates the form element is a multiline text input.
IGR_PAGE_FORM_ELEMENT_TYPE_SINGLELINE_TEXT	5	Indicates the form element is a single line text input.
IGR_PAGE_FORM_ELEMENT_TYPE_COMBOBOX	6	Indicates the form element is a a drop down box.
IGR_PAGE_FORM_ELEMENT_TYPE_LISTBOX	7	Indicates the form element is a list box.
IGR_PAGE_FORM_ELEMENT_TYPE_SIGNATURE	8	Indicates the form element is a signature.
IGR_PAGE_FORM_ELEMENT_TYPE_PASSWORD	9	Indicates the form element is a password input.

Pen Styles

Name	Value	Description
IGR_PEN_NONE	0	Indicates that objects should not be outlined with the pen.
IGR_PEN_SOLID	1	Indicates that objects should be outlined with the specified pen color.

IGR_STRING_ID

The following strings can be localized.

ID	Name	Default English V
10000	METADATA_ABSTRACT	Abstract
10001	METADATA_ACCEPTED	Accepted
10002	METADATA_ACCRUE_AT	Accrue At
10003	METADATA_ACTIVE	Active
10004	METADATA_ACTUAL_COST	Actual Co
10005	METADATA_ACTUAL_DURATION	Actual Du
10006	METADATA_ACTUAL_FINISH	Actual Fi
10007	METADATA_ACTUAL_FRAME_SIZE	Actual Fr
10008	METADATA_ACTUAL_OVERTIME_COST	Actual Ov
10009	METADATA_ACTUAL_OVERTIME_WORK	Actual Ov
10010	METADATA_ACTUAL_START	Actual St
10011	METADATA_ACTUAL_WORK	Actual Wo
10012	METADATA_ACWP	ACWP
10013	METADATA_ADDRESS	Address
10014	METADATA_ADDRESSEE	Addressee
10015	METADATA_ALBUM	Album
10016	METADATA_ALBUM_ARTIST	Album Art
10017	METADATA_ALBUM_MOVIE_SHOW_TITLE	Album/Mov
10018	METADATA_ALBUM_SORT_ORDER	Album sor
10019	METADATA_APP_VERSION	App Versi

ID	Name	Default English V
10020	METADATA_APPLICATION	Applicati
10021	METADATA_APPNAME	AppName
10022	METADATA_ARTIST	Artist
10023	METADATA_ASPECT_RATIO	Aspect Ra
10024	METADATA_ASSIGNMENT	Assignmen
10025	METADATA_ASSIGNMENT_DELAY	Assignmen
10026	METADATA_ASSIGNMENT_OWNER	Assignmen
10027	METADATA_ASSIGNMENT_UNITS	Assignmen
10028	METADATA_ATTACHED_PICTURE	Attached
10029	METADATA_ATTACHMENTS	Attachmen
10030	METADATA_AUDIO_CHANNELS	Audio Cha
10031	METADATA_AUDIO_ENCRYPTION	Audio enc
10032	METADATA_AUDIO_FORMAT	Audio For
10033	METADATA_AUDIO_SAMPLES_PER_SECOND	Audio Sam
10034	METADATA_AUTHOR	Author
10035	METADATA_AUTOCAD_VERSION	AutoCAD v
10036	METADATA_AVAILABLE_FROM	Available
10037	METADATA_AVAILABLE_TO	Available
10038	METADATA_AVERAGE_BIT_RATE	Average B
10039	METADATA_BAND_ORCHESTRA_ACCOMPANIMENT	Band/orch
10040	METADATA_BASE_CALENDAR	Base Cale
10041	METADATA_BASELINE_BUDGET_COST	Baseline
10042	METADATA_BASELINE_BUDGET_WORK	Baseline

ID	Name	Default English V
10043	METADATA_BASELINE_COST	Baseline
10044	METADATA_BASELINE_DELIVERABLE_FINISH	Baseline
10045	METADATA_BASELINE_DELIVERABLE_START	Baseline
10046	METADATA_BASELINE_DURATION	Baseline
10047	METADATA_BASELINE_ESTIMATED_DURATION	Baseline
10048	METADATA_BASELINE_ESTIMATED_FINISH	Baseline
10049	METADATA_BASELINE_ESTIMATED_START	Baseline
10050	METADATA_BASELINE_FINISH	Baseline
10051	METADATA_BASELINE_FIXED_COST	Baseline
10052	METADATA_BASELINE_FIXED_COST_ACCRUAL	Baseline
10053	METADATA_BASELINE_N_BUDGET_COST	Baseline{
10054	METADATA_BASELINE_N_BUDGET_WORK	Baseline{
10055	METADATA_BASELINE_N_COST	Baseline{
10056	METADATA_BASELINE_N_DELIVERABLE_FINISH	Baseline{ Finish
10057	METADATA_BASELINE_N_DELIVERABLE_START	Baseline{
10058	METADATA_BASELINE_N_DURATION	Baseline{
10059	METADATA_BASELINE_N_ESTIMATED_DURATION	Baseline{ Duration
10060	METADATA_BASELINE_N_ESTIMATED_FINISH	Baseline{
10061	METADATA_BASELINE_N_ESTIMATED_START	Baseline{
10062	METADATA_BASELINE_N_FINISH	Baseline{
10063	METADATA_BASELINE_N_FIXED_COST	Baseline{
10064	METADATA_BASELINE_N_FIXED_COST_ACCRUAL	

ID	Name	Default English V
		Baseline{ Accrual
10065	METADATA_BASELINE_N_START	Baseline{
10066	METADATA_BASELINE_N_WORK	Baseline{
10067	METADATA_BASELINE_START	Baseline
10068	METADATA_BASELINE_WORK	Baseline
10069	METADATA_BCC	BCC
10070	METADATA_BCWP	BCWP
10071	METADATA_BCWS	BCWS
10072	METADATA_BITRATE	Bitrate
10073	METADATA_BITS_PER_CHANNEL	Bits Per
10074	METADATA_BITS_PER_PIXEL	Bits Per
10075	METADATA_BITS_PER_SAMPLE	Bits per
10076	METADATA_BODY_FORMAT	Body Form
10077	METADATA_BOOKING_TYPE	Booking T
10078	METADATA_BPM	BPM
10079	METADATA_BPM_BEATS_PER_MINUTE_	BPM (beat
10080	METADATA_BUDGET	Budget
10081	METADATA_BUDGET_COST	Budget Co
10082	METADATA_BUDGET_WORK	Budget Wo
10083	METADATA_BUILD_NUMBER_CREATED	Build Num
10084	METADATA_BUILD_NUMBER_EDITED	Build Num
10085	METADATA_BUSINESS_FAX	Business
10086	METADATA_BUSINESS_PHONE	Business

ID	Name	Default English V
10087	METADATA_BYLINE	Byline
10088	METADATA_BYLINE_TITLE	Byline Ti
10089	METADATA_BYTE_COUNT	Byte Coun
10090	METADATA_BYTES	Bytes
10091	METADATA_CALENDAR	Calendar
10092	METADATA_CALENDAR_GUID	Calendar
10093	METADATA_CAN_BE_STORED_ON_DISK	Can Be St
10094	METADATA_CAN_LEVEL	Can Level
10095	METADATA_CAPTION	Caption
10096	METADATA_CATEGORY	Category
10097	METADATA_CC	CC
10098	METADATA_CELL_COUNT	Cell Coun
10099	METADATA_CHANNELS	Channels
10100	METADATA_CHARACTER_COUNT	Character
10101	METADATA_CHARACTERS_WITH_SPACES_COUNT	Character
10102	METADATA_CITY	City
10103	METADATA_CODE	Code
10104	METADATA_CODEC_FLAVOR	Codec Fla
10105	METADATA_CODEC_FRAME_SIZE	Codec Fra
10106	METADATA_COLOR	Color
10107	METADATA_COLOR_COUNT	Color Cou
10108	METADATA_COLOR_MAP_TYPE	Color Map
10109	METADATA_COLOR_MODE	Color Mod

ID	Name	Default English V
10110	METADATA_COLOR_PLANES	Color Pla
10111	METADATA_COLORS	Colors
10112	METADATA_COMMENT	Comment
10113	METADATA_COMMENTS	Comments
10114	METADATA_COMMERCIAL_FRAME	Commercia
10115	METADATA_COMMERCIAL_INFORMATION	Commercia
10116	METADATA_COMPANY	Company
10117	METADATA_COMPILATION	Compilati
10118	METADATA_COMPLETE	Complete
10119	METADATA_COMPLETE_THROUGH	Complete
10120	METADATA_COMPOSER	Composer
10121	METADATA_COMPRESSION	Compressi
10122	METADATA_CONDUCTOR_PERFORMER_REFINEMENT	Conductor refinemen
10123	METADATA_CONFIRMED	Confirmed
10124	METADATA_CONSTRAINT_DATE	Constrain
10125	METADATA_CONSTRAINT_TYPE	Constrain
10126	METADATA_CONTACT	Contact
10127	METADATA_CONTENT_CLASS	Content-c
10128	METADATA_CONTENT_GROUP_DESCRIPTION	Content g
10129	METADATA_CONTENT_TYPE	Content t
10130	METADATA_COPYRIGHT	Copyright
10131	METADATA_COPYRIGHT_FLAG	Copyright
10132	METADATA_COPYRIGHT_LEGAL_INFORMATION	Copyright

ID	Name	Default English V
10133	METADATA_COPYRIGHT_MESSAGE	Copyright
10134	METADATA_COPYRIGHT_NOTICE	Copyright
10135	METADATA_COST	Cost
10136	METADATA_COST_CENTER	Cost Cent
10137	METADATA_COST_N	Cost{}
10138	METADATA_COST_PER_USE	Cost Per
10139	METADATA_COST_RATE_TABLE	Cost Rate
10140	METADATA_COST_VARIANCE	Cost Vari
10141	METADATA_COUNTRY	Country
10142	METADATA_COUNTRY_CODE	Country C
10143	METADATA_CPI	CPI
10144	METADATA_CREATED	Created
10145	METADATA_CREATED_DATE	Created D
10146	METADATA_CREATION_DATE	Creation
10147	METADATA_CREATOR	Creator
10148	METADATA_CREATOR_TOOL	Creator T
10149	METADATA_CREDIT	Credit
10150	METADATA_CRITICAL	Critical
10151	METADATA_CV	CV
10152	METADATA_CVPERC	CV%
10153	METADATA_DATE	Date
10154	METADATA_DATE_CREATED	Date crea
10155	METADATA_DATE_MODIFIED	Date modi

ID	Name	Default English V
10156	METADATA_DATE_N	Date{}
10157	METADATA_DEADLINE	Deadline
10158	METADATA_DEFAULT_ASSIGNMENT_OWNER	Default A
10159	METADATA_DELIVERABLE_FINISH	Deliverab
10160	METADATA_DELIVERABLE_GUID	Deliverab
10161	METADATA_DELIVERABLE_NAME	Deliverab
10162	METADATA_DELIVERABLE_START	Deliverab
10163	METADATA_DELIVERABLE_TYPE	Deliverab
10164	METADATA_DEPTH	Depth
10165	METADATA_DESCRIPTION	Descripti
10166	METADATA_DIMENSIONS	Dimension
10167	METADATA_DISK_NUMBER	Disk numb
10168	METADATA_DOCUMENT_NAME	Document
10169	METADATA_DPI	DPI
10170	METADATA_DRAWING_COUNT	Drawing C
10171	METADATA_DRAWING_ID	Drawing I
10172	METADATA_DRAWING_VERSION	Drawing v
10173	METADATA_DUE_DATE	Due Date
10174	METADATA_DURATION	Duration
10175	METADATA_DURATION_N	Duration{
10176	METADATA_DURATION_VARIANCE	Duration
10177	METADATA_EAC	EAC
10178	METADATA_EARLY_FINISH	Early Fin

ID	Name	Default English V
10179	METADATA_EARLY_START	Early Sta
10180	METADATA_EARNED_VALUE_METHOD	Earned Va
10181	METADATA_EDITING_CYCLES	Editing C
10182	METADATA_EDITING_DURATION	Editing D
10183	METADATA_EDITOR	Editor
10184	METADATA_EFFORT_DRIVEN	Effort Dr
10185	METADATA_EMAIL	Email
10186	METADATA_EMAIL_ADDRESS	Email Add
10187	METADATA_ENCODED_BY	Encoded b
10188	METADATA_ENCODER	Encoder
10189	METADATA_ENCODING_TIME	Encoding
10190	METADATA_ENCRYPTION_METHOD_REGISTRATION	Encryptio registrat
10191	METADATA_END_TIME	End Time
10192	METADATA_ENTERPRISE	Enterpris
10193	METADATA_ENTERPRISE_BASE_CALENDAR	Enterpris
10194	METADATA_ENTERPRISE_COST_N	Enterpris
10195	METADATA_ENTERPRISE_DATE_N	Enterpris
10196	METADATA_ENTERPRISE_DURATION_N	Enterpris
10197	METADATA_ENTERPRISE_FLAG_N	Enterpris
10198	METADATA_ENTERPRISE_NUMBER_N	Enterpris
10199	METADATA_ENTERPRISE_REQUIRED_VALUES	Enterpris
10200	METADATA_ENTERPRISE_TEAM_MEMBER	Enterpris
10201	METADATA_ENTERPRISE_TEXT_N	Enterpris

ID	Name	Default English V
10202	METADATA_ENTERPRISE_UNIQUE_ID	Enterpris
10203	METADATA_EQUALIZATION	Equalizat
10204	METADATA_ERROR_MESSAGE	Error Mes
10205	METADATA_ESTIMATED	Estimated
10206	METADATA_ESTIMATED_DURATION	Estimated
10207	METADATA_EVENT_TIMING_CODES	Event tim
10208	METADATA_EXPOSURE_PROGRAM	Exposure
10209	METADATA_EXPOSURE_TIME	Exposure
10210	METADATA_EXTENTS	Extents
10211	METADATA_EXTERNAL_TASK	External
10212	METADATA_F_STOP	F-Stop
10213	METADATA_FILE_AS	File As
10214	METADATA_FILE_IS_LIVE_BROADCAST	The File
10215	METADATA_FILE_OWNER_LICENSEE	File owne
10216	METADATA_FILE_TYPE	File type
10217	METADATA_FILE_VERSION	File Vers
10218	METADATA_FINISH	Finish
10219	METADATA_FINISH_N	Finish{}
10220	METADATA_FINISH_SLACK	Finish SL
10221	METADATA_FINISH_VARIANCE	Finish Va
10222	METADATA_FINISHDATE	FinishDat
10223	METADATA_FIXED_COST	Fixed Cos
10224	METADATA_FIXED_COST_ACCRUAL	Fixed Cos

ID	Name	Default English V
10225	METADATA_FLAG_N	Flag{}
10226	METADATA_FLAGS	Flags
10227	METADATA_FLASH	Flash
10228	METADATA_FOCAL_LENGTH	Focal Len
10229	METADATA_FORMAT	Format
10230	METADATA_FRAME_COUNT	Frame Cou
10231	METADATA_FRAMES_PER_SECOND	Frames/se
10232	METADATA_FREE_SLACK	Free Slac
10233	METADATA_FROM	From
10234	METADATA_FULL_NAME	Full Name
10235	METADATA_GENERAL_ENCAPSULATED_OBJECT	General e
10236	METADATA_GENERATOR	Generator
10237	METADATA_GENERIC	Generic
10238	METADATA_GENRE	Genre
10239	METADATA_GROUP	Group
10240	METADATA_GROUP_BY_SUMMARY	Group By
10241	METADATA_GROUP_IDENTIFICATION_REGISTRATION	Group ide registrat
10242	METADATA_GROUPING	Grouping
10243	METADATA_GUID	GUID
10244	METADATA_HANDLER	Handler
10245	METADATA_HEADLINE	Headline
10246	METADATA_HEIGHT	Height
10247	METADATA_HIDDEN_COUNT	Hidden Co

ID	Name	Default English V
10248	METADATA_HIDDEN_SLIDES	Hidden SL
10249	METADATA_HIDE_BAR	Hide Bar
10250	METADATA_HOME_PHONE	Home Phon
10251	METADATA_HYPERLINK	Hyperlink
10252	METADATA_HYPERLINK_ADDRESS	Hyperlink
10253	METADATA_HYPERLINK_HREF	Hyperlink
10254	METADATA_HYPERLINK_SUBADDRESS	Hyperlink
10255	METADATA_ID	ID
10256	METADATA_ID3V2_GENRE_ACAPELLA	Acapella
10257	METADATA_ID3V2_GENRE_ACID	Acid
10258	METADATA_ID3V2_GENRE_ACID_JAZZ	Acid Jazz
10259	METADATA_ID3V2_GENRE_ACID_PUNK	Acid Punk
10260	METADATA_ID3V2_GENRE_ACOUSTIC	Acoustic
10261	METADATA_ID3V2_GENRE_ALTERNATIVE	Alternati
10262	METADATA_ID3V2_GENRE_ALTERNATIVE_ROCK	Alternati
10263	METADATA_ID3V2_GENRE_AMBIENT	Ambient
10264	METADATA_ID3V2_GENRE_ANIME	Anime
10265	METADATA_ID3V2_GENRE_AVANTGARDE	Avantgard
10266	METADATA_ID3V2_GENRE_BALLAD	Ballad
10267	METADATA_ID3V2_GENRE_BASS	Bass
10268	METADATA_ID3V2_GENRE_BEAT	Beat
10269	METADATA_ID3V2_GENRE_BEBOB	Bebob
10270	METADATA_ID3V2_GENRE_BIG_BAND	Big Band

ID	Name	Default English V
10271	METADATA_ID3V2_GENRE_BLACK_METAL	Black Met
10272	METADATA_ID3V2_GENRE_BLUEGRASS	Bluegrass
10273	METADATA_ID3V2_GENRE_BLUES	Blues
10274	METADATA_ID3V2_GENRE_BOOTY_BASS	Booty Bas
10275	METADATA_ID3V2_GENRE_BRITPOP	BritPop
10276	METADATA_ID3V2_GENRE_CABARET	Cabaret
10277	METADATA_ID3V2_GENRE_CELTIC	Celtic
10278	METADATA_ID3V2_GENRE_CHAMBER_MUSIC	Chamber M
10279	METADATA_ID3V2_GENRE_CHANSON	Chanson
10280	METADATA_ID3V2_GENRE_CHORUS	Chorus
10281	METADATA_ID3V2_GENRE_CHRISTIAN_GANGSTA_RAP	Christian
10282	METADATA_ID3V2_GENRE_CHRISTIAN_RAP	Christian
10283	METADATA_ID3V2_GENRE_CHRISTIAN_ROCK	Christian
10284	METADATA_ID3V2_GENRE_CLASSIC_ROCK	Classic R
10285	METADATA_ID3V2_GENRE_CLASSICAL	Classical
10286	METADATA_ID3V2_GENRE_CLUB	Club
10287	METADATA_ID3V2_GENRE_CLUB_HOUSE	Club - Ho
10288	METADATA_ID3V2_GENRE_COMEDY	Comedy
10289	METADATA_ID3V2_GENRE_CONTEMPORARY_CHRISTIAN	Contempor
10290	METADATA_ID3V2_GENRE_COUNTRY	Country
10291	METADATA_ID3V2_GENRE_CROSSOVER	Crossover
10292	METADATA_ID3V2_GENRE_CULT	Cult
10293	METADATA_ID3V2_GENRE_DANCE	Dance

ID	Name	Default English V
10294	METADATA_ID3V2_GENRE_DANCE_HALL	Dance Ha1
10295	METADATA_ID3V2_GENRE_DARKWAVE	Darkwave
10296	METADATA_ID3V2_GENRE_DEATH_METAL	Death Met
10297	METADATA_ID3V2_GENRE_DISCO	Disco
10298	METADATA_ID3V2_GENRE_DREAM	Dream
10299	METADATA_ID3V2_GENRE_DRUM_BASS	Drum & Ba
10300	METADATA_ID3V2_GENRE_DRUM_SOLO	Drum Solo
10301	METADATA_ID3V2_GENRE_DUET	Duet
10302	METADATA_ID3V2_GENRE_EASY_LISTENING	Easy List
10303	METADATA_ID3V2_GENRE_ELECTRONIC	Electroni
10304	METADATA_ID3V2_GENRE_ETHNIC	Ethnic
10305	METADATA_ID3V2_GENRE_EURO_HOUSE	Euro-Hous
10306	METADATA_ID3V2_GENRE_EURO_TECHNO	Euro-Tech
10307	METADATA_ID3V2_GENRE_EURODANCE	Eurodance
10308	METADATA_ID3V2_GENRE_FAST_FUSION	Fast Fusi
10309	METADATA_ID3V2_GENRE_FOLK	Folk
10310	METADATA_ID3V2_GENRE_FOLK_ROCK	Folk-Rock
10311	METADATA_ID3V2_GENRE_FOLKLORE	Folklore
10312	METADATA_ID3V2_GENRE_FREESTYLE	Freestyle
10313	METADATA_ID3V2_GENRE_FUNK	Funk
10314	METADATA_ID3V2_GENRE_FUSION	Fusion
10315	METADATA_ID3V2_GENRE_GAME	Game
10316	METADATA_ID3V2_GENRE_GANGSTA	Gangsta

ID	Name	Default English V
10317	METADATA_ID3V2_GENRE_GOA	Goa
10318	METADATA_ID3V2_GENRE_GOSPEL	Gospel
10319	METADATA_ID3V2_GENRE_GOTHIC	Gothic
10320	METADATA_ID3V2_GENRE_GOTHIC_ROCK	Gothic Ro
10321	METADATA_ID3V2_GENRE_GRUNGE	Grunge
10322	METADATA_ID3V2_GENRE_HARD_ROCK	Hard Rock
10323	METADATA_ID3V2_GENRE_HARDCORE	Hardcore
10324	METADATA_ID3V2_GENRE_HEAVY_METAL	Heavy Met
10325	METADATA_ID3V2_GENRE_HIP_HOP	Hip-Hop
10326	METADATA_ID3V2_GENRE_HOUSE	House
10327	METADATA_ID3V2_GENRE_HUMOUR	Humor
10328	METADATA_ID3V2_GENRE_INDIE	Indie
10329	METADATA_ID3V2_GENRE_INDUSTRIAL	Industria
10330	METADATA_ID3V2_GENRE_INSTRUMENTAL	Instrumen
10331	METADATA_ID3V2_GENRE_INSTRUMENTAL_POP	Instrumen
10332	METADATA_ID3V2_GENRE_INSTRUMENTAL_ROCK	Instrumen
10333	METADATA_ID3V2_GENRE_JAZZ	Jazz
10334	METADATA_ID3V2_GENRE_JAZZ_FUNK	Jazz+Funk
10335	METADATA_ID3V2_GENRE_JPOP	JPop
10336	METADATA_ID3V2_GENRE_JUNGLE	Jungle
10337	METADATA_ID3V2_GENRE_LATIN	Latin
10338	METADATA_ID3V2_GENRE_LO_FI	Lo-Fi
10339	METADATA_ID3V2_GENRE_MEDITATIVE	Meditativ

ID	Name	Default English V
10340	METADATA_ID3V2_GENRE_MERENGUE	Merengue
10341	METADATA_ID3V2_GENRE_METAL	Metal
10342	METADATA_ID3V2_GENRE_MUSICAL	Musical
10343	METADATA_ID3V2_GENRE_NATIONAL_FOLK	National
10344	METADATA_ID3V2_GENRE_NATIVE_US	Native US
10345	METADATA_ID3V2_GENRE_NEGERPUNK	Negerpunk
10346	METADATA_ID3V2_GENRE_NEW_AGE	New Age
10347	METADATA_ID3V2_GENRE_NEW_WAVE	New Wave
10348	METADATA_ID3V2_GENRE_NOISE	Noise
10349	METADATA_ID3V2_GENRE_OLDIES	Oldies
10350	METADATA_ID3V2_GENRE_OPERA	Opera
10351	METADATA_ID3V2_GENRE_OTHER	Other
10352	METADATA_ID3V2_GENRE_POLKA	Polka
10353	METADATA_ID3V2_GENRE_POLSK_PUNK	Polsk Punk
10354	METADATA_ID3V2_GENRE_POP	Pop
10355	METADATA_ID3V2_GENRE_POP_FOLK	Pop-Folk
10356	METADATA_ID3V2_GENRE_POP_FUNK	Pop/Funk
10357	METADATA_ID3V2_GENRE_PORN_GROOVE	Porn Groove
10358	METADATA_ID3V2_GENRE_POWER_BALLAD	Power Ballad
10359	METADATA_ID3V2_GENRE_PRANKS	Pranks
10360	METADATA_ID3V2_GENRE_PRIMUS	Primus
10361	METADATA_ID3V2_GENRE_PROGRESSIVE_ROCK	Progressive Rock
10362	METADATA_ID3V2_GENRE_PSYCHEDELIC	Psychedelic

ID	Name	Default English V
10363	METADATA_ID3V2_GENRE_PSYCHEDELIC_ROCK	Psychedel
10364	METADATA_ID3V2_GENRE_PUNK	Punk
10365	METADATA_ID3V2_GENRE_PUNK_ROCK	Punk Rock
10366	METADATA_ID3V2_GENRE_RAP	Rap
10367	METADATA_ID3V2_GENRE_RAVE	Rave
10368	METADATA_ID3V2_GENRE_RB	R&B
10369	METADATA_ID3V2_GENRE_REGGAE	Reggae
10370	METADATA_ID3V2_GENRE_RETRO	Retro
10371	METADATA_ID3V2_GENRE_REVIVAL	Revival
10372	METADATA_ID3V2_GENRE_RHYTHMIC_SOUL	Rhythmic
10373	METADATA_ID3V2_GENRE_ROCK	Rock
10374	METADATA_ID3V2_GENRE_ROCK_ROLL	Rock & Ro
10375	METADATA_ID3V2_GENRE_SALSA	Salsa
10376	METADATA_ID3V2_GENRE_SAMBA	Samba
10377	METADATA_ID3V2_GENRE_SATIRE	Satire
10378	METADATA_ID3V2_GENRE_SHOWTUNES	Showtunes
10379	METADATA_ID3V2_GENRE_SKA	Ska
10380	METADATA_ID3V2_GENRE_SLOW_JAM	Slow Jam
10381	METADATA_ID3V2_GENRE_SLOW_ROCK	Slow Rock
10382	METADATA_ID3V2_GENRE_SONATA	Sonata
10383	METADATA_ID3V2_GENRE_SOUL	Soul
10384	METADATA_ID3V2_GENRE_SOUND_CLIP	Sound Cli
10385	METADATA_ID3V2_GENRE_SOUNDTRACK	Soundtrac

ID	Name	Default English V
10386	METADATA_ID3V2_GENRE_SOUTHERN_ROCK	Southern
10387	METADATA_ID3V2_GENRE_SPACE	Space
10388	METADATA_ID3V2_GENRE_SPEECH	Speech
10389	METADATA_ID3V2_GENRE_SWING	Swing
10390	METADATA_ID3V2_GENRE_SYMPHONIC_ROCK	Symphonic
10391	METADATA_ID3V2_GENRE_SYMPHONY	Symphony
10392	METADATA_ID3V2_GENRE_SYNTHPOP	Synthpop
10393	METADATA_ID3V2_GENRE_TANGO	Tango
10394	METADATA_ID3V2_GENRE_TECHNO	Techno
10395	METADATA_ID3V2_GENRE_TECHNO_INDUSTRIAL	Techno-In
10396	METADATA_ID3V2_GENRE_TERROR	Terror
10397	METADATA_ID3V2_GENRE_THRASH_METAL	Thrash Me
10398	METADATA_ID3V2_GENRE_TOP_40	Top 40
10399	METADATA_ID3V2_GENRE_TRAILER	Trailer
10400	METADATA_ID3V2_GENRE_TRANCE	Trance
10401	METADATA_ID3V2_GENRE_TRIBAL	Tribal
10402	METADATA_ID3V2_GENRE_TRIP_HOP	Trip-Hop
10403	METADATA_ID3V2_GENRE_VOCAL	Vocal
10404	METADATA_IDENTIFIER	Identifie
10405	METADATA_IGNORE_RESOURCE_CALENDAR	Ignore Re
10406	METADATA_IGNORE_WARNINGS	Ignore Wa
10407	METADATA_IM_ADDRESS	IM Addres
10408	METADATA_IMAGE_CODEC	Image Cod

ID	Name	Default English V
10409	METADATA_IMAGE_COUNT	Image Cou
10410	METADATA_IMAGE_FORMAT	Image For
10411	METADATA_IMAGE_RESOLUTION	Image Res
10412	METADATA_IMAGE_TYPE	Image typ
10413	METADATA_IMPORT	Import
10414	METADATA_INACTIVE	Inactive
10415	METADATA_INDICATORS	Indicator
10416	METADATA_INITIAL_AUTHOR	Initial A
10417	METADATA_INITIAL_KEY	Initial k
10418	METADATA_INITIALS	Initials
10419	METADATA_INTERLACED	Interlace
10420	METADATA_INTERNET_HEADERS	Internet
10421	METADATA_INTERNET_RADIO_STATION_NAME	Internet
10422	METADATA_INTERNET_RADIO_STATION_OWNER	Internet
10423	METADATA_INTERPRETED_REMIXED_OR_OTHERWISE_MODIFIED_BY	Interpret otherwise
10424	METADATA_INVOLVED_PEOPLE_LIST	Involved
10425	METADATA_ISO_SPEED	ISO Speed
10426	METADATA_ISRC_INTERNATIONAL_STANDARD_RECORDING_CODE_	ISRC (int standard
10427	METADATA_ISYS_MACROS	isys:Macr
10428	METADATA_ISYS_SUBTYPE	isys:SubT
10429	METADATA_JOB_TITLE	Job Title
10430	METADATA_KEYWORD	Keyword

ID	Name	Default English V
10431	METADATA_KEYWORDS	Keywords
10432	METADATA_LANG	Lang
10433	METADATA_LANGUAGE	Language
10434	METADATA_LANGUAGES	Language(
10435	METADATA_LAST_ACCESSED_BY	Last Acce
10436	METADATA_LAST_AUTHOR	Last Auth
10437	METADATA_LAST_MODIFIED_DATE	Last Modi
10438	METADATA_LAST_PRINTED	Last Prin
10439	METADATA_LAST_PRINTED_BY	Last Prin
10440	METADATA_LAST_SAVED_BY	Last Save
10441	METADATA_LAST_SAVED_DATE	Last Save
10442	METADATA_LAST_SAVED_PATH	Last Save
10443	METADATA_LATE_FINISH	Late Fini
10444	METADATA_LATE_START	Late Star
10445	METADATA_LAYERS	Layers
10446	METADATA_LEAD_PERFORMERS	Lead perf
10447	METADATA_LENGTH	Length
10448	METADATA_LEVEL_ASSIGNMENTS	Level Ass
10449	METADATA_LEVELING_CAN_SPLIT	Leveling
10450	METADATA_LEVELING_DELAY	Leveling
10451	METADATA_LICENSE	License
10452	METADATA_LINE_COUNT	Line Coun
10453	METADATA_LINES	Lines

ID	Name	Default English V
10454	METADATA_LINKED_FIELDS	Linked Fi
10455	METADATA_LINKED_INFORMATION	Linked in
10456	METADATA_LINKS_DIRTY	Links Dir
10457	METADATA_LINKS_UP_TO_DATE	Links Up
10458	METADATA_LOCATION	Location
10459	METADATA_LYRICIST_TEXT_WRITER	Lyricist/
10460	METADATA_LYRICS	Lyrics
10461	METADATA_MAKE	Make
10462	METADATA_MANAGER	Manager
10463	METADATA_MARKED	Marked
10464	METADATA_MATERIAL_LABEL	Material
10465	METADATA_MAX_BIT_RATE	Max Bit R
10466	METADATA_MAX_UNITS	Max Units
10467	METADATA_MEDIA_TYPE	Media Typ
10468	METADATA_MESSAGE_ID	Message-I
10469	METADATA_METADATA_DATE	Metadata
10470	METADATA_MILESTONE	Milestone
10471	METADATA_MIME_TYPE	MIME Type
10472	METADATA_MM_CLIP_COUNT	MM Clip C
10473	METADATA_MM_CLIPS	MM Clips
10474	METADATA_MOBILE	Mobile
10475	METADATA_MODE	Mode
10476	METADATA_MODEL	Model

ID	Name	Default English V
10477	METADATA_MODIFICATION_DATE	Modificat
10478	METADATA_MPEG_LOCATION_LOOKUP_TABLE	MPEG loca
10479	METADATA_MUSIC_CD_IDENTIFIER	Music CD
10480	METADATA_NAME	Name
10481	METADATA_NEWSGROUP	Newsgroup
10482	METADATA_NOTE_COUNT	Note Coun
10483	METADATA_NOTES	Notes
10484	METADATA_NUMBER	Number
10485	METADATA_NUMBER_N	Number{}
10486	METADATA_NUMBER_OF_CHARACTERS	Number of
10487	METADATA_NUMBER_OF_IMAGES	Number of
10488	METADATA_NUMBER_OF_PAGES	Number of
10489	METADATA_NUMBER_OF_RESOLUTIONS	Number of
10490	METADATA_NUMBER_OF_STREAMS	Number Of
10491	METADATA_NUMBER_OF_WORDS	Number of
10492	METADATA_OBJECT_COUNT	Object Co
10493	METADATA_OBJECT_NAME	Object Na
10494	METADATA_OBJECTS	Objects
10495	METADATA_OFFICIAL_ARTIST_PERFORMER_WEBPAGE	Official webpage
10496	METADATA_OFFICIAL_AUDIO_FILE_WEBPAGE	Official
10497	METADATA_OFFICIAL_AUDIO_SOURCE_WEBPAGE	Official webpage
10498	METADATA_OFFICIAL_INTERNET_RADIO_STATION_HOMEPAGE	

ID	Name	Default English V
		Official station h
10499	METADATA_OLEOBJECT_COUNT	OLE Objec
10500	METADATA_OPERATOR	Operator
10501	METADATA_ORIGIN	Origin
10502	METADATA_ORIGINAL_ALBUM_MOVIE_SHOW_TITLE	Original title
10503	METADATA_ORIGINAL_ARTISTS	Original
10504	METADATA_ORIGINAL_DATETIME	Original
10505	METADATA_ORIGINAL_FILENAME	Original
10506	METADATA_ORIGINAL_LYRICISTS	Original
10507	METADATA_ORIGINAL_RELEASE_TIME	Original
10508	METADATA_ORIGINAL_RELEASE_YEAR	Original
10509	METADATA_OUTLINE_CODE_N	Outline C
10510	METADATA_OUTLINE_LEVEL	Outline L
10511	METADATA_OUTLINE_NUMBER	Outline N
10512	METADATA_OVERALLOCATED	Overalloc
10513	METADATA_OVERTIME_COST	Overtime
10514	METADATA_OVERTIME_RATE	Overtime
10515	METADATA_OVERTIME_WORK	Overtime
10516	METADATA_OWNER	Owner
10517	METADATA_OWNERSHIP_FRAME	Ownership
10518	METADATA_PAGE_COUNT	Page Coun
10519	METADATA_PAGE_DIMENSIONS	Page Dime

ID	Name	Default English V
10520	METADATA_PAGE_NAME	Page Name
10521	METADATA_PAGES	Pages
10522	METADATA_PARAGRAPH_COUNT	Paragraph
10523	METADATA_PARAGRAPHS	Paragraph
10524	METADATA_PARAGRAPHS_COUNT	Paragraph
10525	METADATA_PART_OF_A_SET	Part of a
10526	METADATA_PAYMENT	Payment
10527	METADATA_PEAK	Peak
10528	METADATA_PERC_COMPLETE	% Complet
10529	METADATA_PERC_WORK_COMPLETE	% Work Co
10530	METADATA_PERCENT_WORK_COMPLETE	% Work Co
10531	METADATA_PERFECTPLAY	PerfectPl (extra bu
10532	METADATA_PERFORMER_SORT_ORDER	Performer
10533	METADATA_PHONETICS	Phonetics
10534	METADATA_PHOTO_CD_FILE_TYPE	Photo CD
10535	METADATA_PHYSICAL_PERC_COMPLETE	Physical
10536	METADATA_PIXEL_DIMENSIONS	Pixel Dim
10537	METADATA_PLACEHOLDER	Placehold
10538	METADATA_PLAY_COUNTER	Play coun
10539	METADATA_PLAYBACK_DELAY	Playback
10540	METADATA_PLAYBACK_OFFSET	Playback
10541	METADATA_PLAYLIST_DELAY	Playlist
10542	METADATA_PODCAST	Podcast

ID	Name	Default English V
10543	METADATA_PODCAST_URL	Podcast U
10544	METADATA_POPULARIMETER	Popularim
10545	METADATA_POSITION_SYNCHRONIZATION_FRAME	Position frame
10546	METADATA_PR	PR
10547	METADATA_PERCENT_COMPLETE	% Complet
10548	METADATA_PREDECESSORS	Predecess
10549	METADATA_PRELEVELED_FINISH	Prelevele
10550	METADATA_PRELEVELED_START	Prelevele
10551	METADATA_PREROLL_OFFSET	Preroll 0
10552	METADATA_PRESENTATION_FORMAT	Presentat
10553	METADATA_PRESENTATION_TARGET	Presentat
10554	METADATA_PRINTED_PAGE_COUNT	Printed P
10555	METADATA_PRIORITY	Priority
10557	METADATA_PRIVATE_FRAME	Private f
10558	METADATA_PRODUCER	Producer
10559	METADATA_PROGID	ProgId
10560	METADATA_PROJECT	Project
10561	METADATA_PUBLISH	Publish
10562	METADATA_PUBLISHER	Publisher
10563	METADATA_PUBLISHERS_OFFICIAL_WEBPAGE	Publisher
10564	METADATA_PURCHASE_DATE	Purchase
10565	METADATA_RATING	Rating
10566	METADATA_RATING_ADVISORY	Rating/Ad

ID	Name	Default English V
10567	METADATA_RECOMMENDED_BUFFER_SIZE	Recommend
10568	METADATA_RECORDING_DATES	Recording
10569	METADATA_RECORDS	Records
10570	METADATA_RECURRING	Recurring
10571	METADATA_REF_SERVICE	Ref. Serv
10572	METADATA_REGULAR_WORK	Regular W
10573	METADATA_REJECTED	Rejected
10574	METADATA_RELATIVE_VOLUME_ADJUSTMENT	Relative
10575	METADATA_RELEASE_TIME	Release t
10576	METADATA_REMAINING_COST	Remaining
10577	METADATA_REMAINING_DURATION	Remaining
10578	METADATA_REMAINING_OVERTIME_COST	Remaining
10579	METADATA_REMAINING_OVERTIME_WORK	Remaining
10580	METADATA_REMAINING_WORK	Remaining
10581	METADATA_REMINDER	Reminder
10582	METADATA_REQUEST_DEMAND	Request/D
10583	METADATA_RESOLUTION_UNIT	Resolutio
10584	METADATA_RESOLUTIONX	Resolutio
10585	METADATA_RESOLUTIONY	Resolutio
10586	METADATA_RESOURCE_GROUP	Resource
10587	METADATA_RESOURCE_INITIALS	Resource
10588	METADATA_RESOURCE_NAMES	Resource
10589	METADATA_RESOURCE_PHONETICS	Resource

ID	Name	Default English V
10590	METADATA_RESOURCE_TYPE	Resource
10591	METADATA_RESPONSE_PENDING	Response
10592	METADATA_RESUME	Resume
10593	METADATA_REVERB	Reverb
10594	METADATA_REVISION	Revision
10595	METADATA_REVISION_NUMBER	Revision
10596	METADATA_RIGHTS	Rights
10597	METADATA_ROLLUP	Rollup
10598	METADATA_ROTATION_ANGLE	Rotation
10599	METADATA_ROW_COUNT	Row Count
10600	METADATA_SAMPLE_RATE	Sample Ra
10601	METADATA_SAMPLES_PER_SECOND	Samples/s
10602	METADATA_SCALE_CROP	Scale Cro
10603	METADATA_SCALE_MODE	Scale Mod
10604	METADATA_SCAN_DIRECTION	Scan Dire
10605	METADATA_SCHEDULED_DURATION	Scheduled
10606	METADATA_SCHEDULED_FINISH	Scheduled
10607	METADATA_SCHEDULED_START	Scheduled
10608	METADATA_SECURITY	Security
10609	METADATA_SENDER	Sender
10610	METADATA_SENT	Sent
10611	METADATA_SENTENCE_COUNT	Sentence
10612	METADATA_SHARED_DOCUMENT	Shared Do

ID	Name	Default English V
10613	METADATA_SHUTTER_SPEED	Shutter S
10614	METADATA_SIZE	Size
10615	METADATA_SLIDE_COUNT	Slide Cou
10616	METADATA_SLIDES	Slides
10617	METADATA_SOFTWARE	Software
10618	METADATA_SOFTWARE_HARDWARE_AND_SETTINGS_USED_FOR_ENCODING	Software/ settings
10619	METADATA_SOUND_FILE	Sound Fil
10620	METADATA_SOURCE	Source
10621	METADATA_SPEC_INSTR	Spec. Ins
10622	METADATA_SPI	SPI
10623	METADATA_STANDARD_RATE	Standard
10624	METADATA_START	Start
10625	METADATA_START_DATE	Start Dat
10626	METADATA_START_N	Start{}
10627	METADATA_START_SLACK	Start Sla
10628	METADATA_START_TIME	Start Tim
10629	METADATA_START_VARIANCE	Start Var
10630	METADATA_STARTDATE	StartDate
10631	METADATA_STATE	State
10632	METADATA_STATUS	Status
10633	METADATA_STATUS_INDICATOR	Status In
10634	METADATA_STATUS_MANAGER	Status Ma
10635	METADATA_STOP	Stop

ID	Name	Default English V
10636	METADATA_SUB_LOCATION	Sub Locat
10637	METADATA_SUBFILE_FORMAT_NAME	Subfile F
10638	METADATA_SUBJECT	Subject
10639	METADATA_SUBPROJECT_FILE	Subprojec
10640	METADATA_SUBPROJECT_READ_ONLY	Subprojec
10641	METADATA_SUBTITLE_DESCRIPTION_REFINEMENT	Subtitle/ refinemen
10642	METADATA_SUCCESSORS	Successor
10643	METADATA_SUMMARY	Summary
10644	METADATA_SUMMARY_PROGRESS	Summary P
10645	METADATA_SUPPLEMENTAL_CATEGORIES	Supplemen
10646	METADATA_SV	SV
10647	METADATA_SVPERC	SV%
10648	METADATA_SYLLABLE_COUNT	Syllable
10649	METADATA_SYNCHRONIZED_LYRIC_TEXT	Synchroni
10650	METADATA_SYNCHRONIZED_TEMPO_CODES	Synchroni
10651	METADATA_TABLE_COUNT	Table Cou
10652	METADATA_TAGGING_TIME	Tagging t
10653	METADATA_TASK_CALENDAR	Task Cale
10654	METADATA_TASK_CALENDAR_GUID	Task Cale
10655	METADATA_TASK_MODE	Task Mode
10656	METADATA_TASK_NAME	Task Name
10657	METADATA_TASK_OUTLINE_NUMBER	Task Outl
10658	METADATA_TASK_SUMMARY_NAME	Task Summ

ID	Name	Default English V
10659	METADATA_TCPI	TCPI
10660	METADATA_TEAM_ASSIGNMENT_POOL	Team Assi
10661	METADATA_TEAMSTATUS_PENDING	TeamStatu
10662	METADATA_TEMPLATE	Template
10663	METADATA_TERMS_OF_USE	Terms of
10664	METADATA_TEXT_N	Text{}
10665	METADATA_THREAD_INDEX	Thread-In
10666	METADATA_THREAD_TOPIC	Thread-To
10667	METADATA_THUMBNAIL	Thumbnail
10668	METADATA_TIFF_ALIASELAYERMETADATA	AliasLaye
10669	METADATA_TIFF_ANALOGBALANCE	AnalogBal
10670	METADATA_TIFF_ANTIALIASSTRENGTH	AntiAlias
10671	METADATA_TIFF_APERTUREVALUE	ApertureV
10672	METADATA_TIFF_ARTIST	Artist
10673	METADATA_TIFF_ASSHOTNEUTRAL	AsShotNeu
10674	METADATA_TIFF_ASSHOTWHITEXY	AsShotWhi
10675	METADATA_TIFF_BADFAXLINES	BadFaxLin
10676	METADATA_TIFF_BASELINEEXPOSURE	BaselineE
10677	METADATA_TIFF_BASELINENOISE	BaselineN
10678	METADATA_TIFF_BASELINESHARPNESS	BaselineS
10679	METADATA_TIFF_BAYERGREENSPLIT	BayerGree
10680	METADATA_TIFF_BESTQUALITYSCALE	BestQuali
10681	METADATA_TIFF_BITSPERSAMPLE	BitsPerSa

ID	Name	Default English V
10682	METADATA_TIFF_BLACKLEVEL	BlackLeve
10683	METADATA_TIFF_BLACKLEVELDELTAH	BlackLeve
10684	METADATA_TIFF_BLACKLEVELDELTAV	BlackLeve
10685	METADATA_TIFF_BLACKLEVELREPEATDIM	BlackLeve
10686	METADATA_TIFF_BRIGHTNESSVALUE	Brightnes
10687	METADATA_TIFF_CALIBRATIONILLUMINANT1	Calibrati
10688	METADATA_TIFF_CALIBRATIONILLUMINANT2	Calibrati
10689	METADATA_TIFF_CAMERACALIBRATION1	CameraCal
10690	METADATA_TIFF_CAMERACALIBRATION2	CameraCal
10691	METADATA_TIFF_CAMERASERIALNUMBER	CameraSer
10692	METADATA_TIFF_CELLLENGTH	CellLengt
10693	METADATA_TIFF_CELLWIDTH	CellWidth
10694	METADATA_TIFF_CFALAYOUT	CFALayout
10695	METADATA_TIFF_CFAPATTERN	CFAPatter
10696	METADATA_TIFF_CFAPLANECOLOR	CFAPlaneC
10697	METADATA_TIFF_CHROMABLURRADIUS	ChromaBlu
10698	METADATA_TIFF_CLEANFAXDATA	CleanFaxD
10699	METADATA_TIFF_CLIPPATH	ClipPath
10700	METADATA_TIFF_CODINGMETHODS	CodingMet
10701	METADATA_TIFF_COLORMAP	ColorMap
10702	METADATA_TIFF_COLORMATRIX1	ColorMatr
10703	METADATA_TIFF_COLORMATRIX2	ColorMatr
10704	METADATA_TIFF_COLORSPACE	ColorSpac

ID	Name	Default English V
10705	METADATA_TIFF_COMPONENTSCONFIGURATION	Component
10706	METADATA_TIFF_COMPRESSEDBITSPPERPIXEL	Compressed
10707	METADATA_TIFF_COMPRESSION	Compression
10708	METADATA_TIFF_CONSECUTIVEBADFAXLINES	Consecutive
10709	METADATA_TIFF_CONTRAST	Contrast
10710	METADATA_TIFF_COPYRIGHT	Copyright
10711	METADATA_TIFF_CUSTOMRENDERED	CustomRendered
10712	METADATA_TIFF_DATETIME	DateTime
10713	METADATA_TIFF_DATETIMEDIGITIZED	DateTimeDigitized
10714	METADATA_TIFF_DATETIMEORIGINAL	DateTimeOriginal
10715	METADATA_TIFF_DECODE	Decode
10716	METADATA_TIFF_DEFAULTCROPORIGIN	DefaultCropOrigin
10717	METADATA_TIFF_DEFAULTCROPSIZE	DefaultCropSize
10718	METADATA_TIFF_DEFAULTIMAGECOLOR	DefaultImageColor
10719	METADATA_TIFF_DEFAULTSCALE	DefaultScale
10720	METADATA_TIFF_DEVICESETTINGDESCRIPTION	DeviceSettingDescription
10721	METADATA_TIFF_DIGITALZOOMRATIO	DigitalZoomRatio
10722	METADATA_TIFF_DNGBACKWARDVERSION	DNGBackwardVersion
10723	METADATA_TIFF_DNGPRIVATEDATA	DNGPrivateData
10724	METADATA_TIFF_DNGVERSION	DNGVersion
10725	METADATA_TIFF_DOCUMENTNAME	DocumentName
10726	METADATA_TIFF_DOTRANGE	DotRange
10727	METADATA_TIFF_EXIFIFD	ExifIFD

ID	Name	Default English V
10728	METADATA_TIFF_EXIFVERSION	ExifVersi
10729	METADATA_TIFF_EXPOSUREBIASVALUE	ExposureB
10730	METADATA_TIFF_EXPOSUREINDEX	ExposureI
10731	METADATA_TIFF_EXPOSUREMODE	ExposureM
10732	METADATA_TIFF_EXPOSUREPROGRAM	ExposureP
10733	METADATA_TIFF_EXPOSURETIME	ExposureT
10734	METADATA_TIFF_EXTRASAMPLES	ExtraSamp
10735	METADATA_TIFF_FAXPROFILE	FaxProfil
10736	METADATA_TIFF_FILESOURCE	FileSourc
10737	METADATA_TIFF_FILLORDER	FillOrder
10738	METADATA_TIFF_FLASH	Flash
10739	METADATA_TIFF_FLASHENERGY	FlashEner
10740	METADATA_TIFF_FLASHPIXVERSION	FlashpixV
10741	METADATA_TIFF_FNUMBER	FNumber
10742	METADATA_TIFF_FOCALLENGTH	FocalLeng
10743	METADATA_TIFF_FOCALLENGTHIN35MMFILM	FocalLeng
10744	METADATA_TIFF_FOCALPLANERESOLUTIONUNIT	FocalPlan
10745	METADATA_TIFF_FOCALPLANEXRESOLUTION	FocalPlan
10746	METADATA_TIFF_FOCALPLANEYRESOLUTION	FocalPlan
10747	METADATA_TIFF_FREEBYTECOUNTS	FreeByteC
10748	METADATA_TIFF_FREEOFFSETS	FreeOffse
10749	METADATA_TIFF_GAINCONTROL	GainContr
10750	METADATA_TIFF_GDAL_METADATA	GDAL_META

ID	Name	Default English V
10751	METADATA_TIFF_GDAL_NODATA	GDAL_NODA
10752	METADATA_TIFF_GEOASCIIIPARAMSTAG	GeoAsciiP
10753	METADATA_TIFF_GEODOUBLEPARAMSTAG	GeoDouble
10754	METADATA_TIFF_GEOKEYDIRECTORYTAG	GeoKeyDir
10755	METADATA_TIFF_GLOBALPARAMETERSIFD	GlobalPar
10756	METADATA_TIFF_GPSALTITUDE	GPSAltitu
10757	METADATA_TIFF_GPSALTITUDEREF	GPSAltitu
10758	METADATA_TIFF_GPSAREAINFORMATION	GPSAreaIn
10759	METADATA_TIFF_GPSDATESTAMP	GPSDateSt
10760	METADATA_TIFF_GPSDESTBEARING	GPSDestBe
10761	METADATA_TIFF_GPSDESTBEARINGREF	GPSDestBe
10762	METADATA_TIFF_GPSDESTDISTANCE	GPSDestDi
10763	METADATA_TIFF_GPSDESTDISTANCEREF	GPSDestDi
10764	METADATA_TIFF_GPSDESTLATITUDE	GPSDestLa
10765	METADATA_TIFF_GPSDESTLATITUDEREF	GPSDestLa
10766	METADATA_TIFF_GPSDESTLONGITUDE	GPSDestLo
10767	METADATA_TIFF_GPSDESTLONGITUDEREF	GPSDestLo
10768	METADATA_TIFF_GPSDIFFERENTIAL	GPSDiffer
10769	METADATA_TIFF_GPSDOP	GPSDOP
10770	METADATA_TIFF_GPSIFD	GPSIFD
10771	METADATA_TIFF_GPSIMGDIRECTION	GPSImgDir
10772	METADATA_TIFF_GPSIMGDIRECTIONREF	GPSImgDir
10773	METADATA_TIFF_GPSLATITUDE	GPSLatitu

ID	Name	Default English V
10774	METADATA_TIFF_GPSLATITUDEREF	GPSLatitu
10775	METADATA_TIFF_GPSLONGITUDE	GPSLongit
10776	METADATA_TIFF_GPSLONGITUDEREF	GPSLongit
10777	METADATA_TIFF_GPSMAPDATUM	GPSMapDat
10778	METADATA_TIFF_GPSMEASUREMODE	GPSMeasur
10779	METADATA_TIFF_GPSPROCESSINGMETHOD	GPSProces
10780	METADATA_TIFF_GPSSATELLITES	GPSSatell
10781	METADATA_TIFF_GPSSPEED	GPSSpeed
10782	METADATA_TIFF_GPSSPEEDREF	GPSSpeedR
10783	METADATA_TIFF_GPSSTATUS	GPSStatus
10784	METADATA_TIFF_GPSTIMESTAMP	GPSTimeSt
10785	METADATA_TIFF_GPSTRACK	GPSTrack
10786	METADATA_TIFF_GPSTRACKREF	GPSTrackR
10787	METADATA_TIFF_GPSVERSIONID	GPSVersio
10788	METADATA_TIFF_GRAYRESPONSECURVE	GrayRespo
10789	METADATA_TIFF_GRAYRESPONSEUNIT	GrayRespo
10790	METADATA_TIFF_HALFTONEHINTS	HalftoneH
10791	METADATA_TIFF_HOSTCOMPUTER	HostCompu
10792	METADATA_TIFF_HYLAFAXFAXRECVPARAMS	HylaFAXFa
10793	METADATA_TIFF_HYLAFAXFAXRECVMETHOD	HylaFAXFa
10794	METADATA_TIFF_HYLAFAXFAXSUBADDRESS	HylaFAXFa
10795	METADATA_TIFF_ICCPROFILE	ICCProfil
10796	METADATA_TIFF_IMAGEDESCRIPTION	ImageDesc

ID	Name	Default English V
10797	METADATA_TIFF_IMAGEID	ImageID
10798	METADATA_TIFF_IMAGELAYER	ImageLayer
10799	METADATA_TIFF_IMAGELENGTH	ImageLength
10800	METADATA_TIFF_IMAGESOURCEDATA	ImageSourceData
10801	METADATA_TIFF_IMAGEUNIQUEID	ImageUniqueID
10802	METADATA_TIFF_IMAGEWIDTH	ImageWidth
10803	METADATA_TIFF_INDEXED	Indexed
10804	METADATA_TIFF_INGRFLAGREGISTERS	INGRFlagRegisters
10805	METADATA_TIFF_INGRPACKETDATATAG	INGRPacketDataTag
10806	METADATA_TIFF_INKNAMES	InkNames
10807	METADATA_TIFF_INKSET	InkSet
10808	METADATA_TIFF_INTEROPERABILITYIFD	InteroperabilityIFD
10809	METADATA_TIFF_IPTC	IPTC
10810	METADATA_TIFF_IRASBTRANSFORMATIONMATRIX	IrasBTransformationMatrix
10811	METADATA_TIFF_ISOSPEEDRATINGS	ISO Speed Ratings
10812	METADATA_TIFF_JPEGACTABLES	JPEGACTables
10813	METADATA_TIFF_JPEGDCTABLES	JPEGDCTables
10814	METADATA_TIFF_JPEGINTERCHANGEFORMAT	JPEGInterchangeFormat
10815	METADATA_TIFF_JPEGINTERCHANGEFORMATLENGTH	JPEGInterchangeFormatLength
10816	METADATA_TIFF_JPEGLOSSLESSPREDICTORS	JPEGLosslessPredictors
10817	METADATA_TIFF_JPEGPOINTTRANSFORMS	JPEGPointTransforms
10818	METADATA_TIFF_JPEGPROC	JPEGProc
10819	METADATA_TIFF_JPEGQTABLES	JPEGQTables

ID	Name	Default English V
10820	METADATA_TIFF_JPEGRESTARTINTERVAL	JPEGResta
10821	METADATA_TIFF_JPEGTABLES	JPEGTable
10822	METADATA_TIFF_LENSINFO	LensInfo
10823	METADATA_TIFF_LIGHTSOURCE	LightSour
10824	METADATA_TIFF_LINEARIZATIONTABLE	Lineariza
10825	METADATA_TIFF_LINEARRESPONSELIMIT	LinearRes
10826	METADATA_TIFF_LOCALIZEDCAMERAMODEL	Localized
10827	METADATA_TIFF_M_JPG	M_JPG
10828	METADATA_TIFF_M_SOF0	M_SOF0
10829	METADATA_TIFF_M_SOF1	M_SOF1
10830	METADATA_TIFF_M_SOF10	M_SOF10
10831	METADATA_TIFF_M_SOF11	M_SOF11
10832	METADATA_TIFF_M_SOF13	M_SOF13
10833	METADATA_TIFF_M_SOF14	M_SOF14
10834	METADATA_TIFF_M_SOF15	M_SOF15
10835	METADATA_TIFF_M_SOF2	M_SOF2
10836	METADATA_TIFF_M_SOF3	M_SOF3
10837	METADATA_TIFF_M_SOF5	M_SOF5
10838	METADATA_TIFF_M_SOF6	M_SOF6
10839	METADATA_TIFF_M_SOF7	M_SOF7
10840	METADATA_TIFF_M_SOF9	M_SOF9
10841	METADATA_TIFF_MAKE	Make
10842	METADATA_TIFF_MAKERNOTE	MakerNote

ID	Name	Default English V
10843	METADATA_TIFF_MAKERNOTESAFETY	MakerNote
10844	METADATA_TIFF_MAXAPERTUREVALUE	MaxApertu
10845	METADATA_TIFF_MAXSAMPLEVALUE	MaxSample
10846	METADATA_TIFF_MDCOLORTABLE	MDColorTa
10847	METADATA_TIFF_MDFILETAG	MDFileTag
10848	METADATA_TIFF_MDFILEUNITS	MDFileUni
10849	METADATA_TIFF_MDLABNAME	MDLabName
10850	METADATA_TIFF_MDPREPPDATE	MDPrepDat
10851	METADATA_TIFF_MDPREPTIME	MDPrepTim
10852	METADATA_TIFF_MDSAMPLEINFO	MDSampleI
10853	METADATA_TIFF_MDSCALEPIXEL	MDScalePi
10854	METADATA_TIFF_METERINGMODE	MeteringM
10855	METADATA_TIFF_MINSAMPLEVALUE	MinSample
10856	METADATA_TIFF_MODEL	Model
10857	METADATA_TIFF_MODELPIXELSCALETAG	ModelPixe
10858	METADATA_TIFF_MODELTIEPOINTTAG	ModelTiep
10859	METADATA_TIFF_MODELTRANSFORMATIONTAG	ModelTran
10860	METADATA_TIFF_MODENUMBER	ModeNumbe
10861	METADATA_TIFF_NEWSUBFILETYPE	NewSubfil
10862	METADATA_TIFF_NUMBEROFINKS	NumberOfI
10863	METADATA_TIFF_OCEAPPLICATIONSELECTOR	OceApplic
10864	METADATA_TIFF_OCEIDENTIFICATIONNUMBER	OceIdenti
10865	METADATA_TIFF_OCEIMAGELOGICCHARACTERISTICS	OceImageL

ID	Name	Default English V
10866	METADATA_TIFF_OCESCANJOBDESCRIPTION	OceScanjo
10867	METADATA_TIFF_OECF	OECF
10868	METADATA_TIFF_OPIPROXY	OPIProxy
10869	METADATA_TIFF_ORIENTATION	Orientati
10870	METADATA_TIFF_PAGENAME	PageName
10871	METADATA_TIFF_PAGENUMBER	PageNumbe
10872	METADATA_TIFF_PHOTOMETRICINTERPRETATION	Photometr
10873	METADATA_TIFF_PHOTOSHOP	Photoshop
10874	METADATA_TIFF_PIXELXDIMENSION	PixelXDim
10875	METADATA_TIFF_PIXELYDIMENSION	PixelYDim
10876	METADATA_TIFF_PLANARCONFIGURATION	PlanarCon
10877	METADATA_TIFF_PREDICTOR	Predictor
10878	METADATA_TIFF_PRIMARYCHROMATICITIES	PrimaryCh
10879	METADATA_TIFF_PROFILETYPE	ProfileTy
10880	METADATA_TIFF_REDUCTIONMATRIX1	Reduction
10881	METADATA_TIFF_REDUCTIONMATRIX2	Reduction
10882	METADATA_TIFF_REFERENCEBLACKWHITE	Reference
10883	METADATA_TIFF_RELATEDSOUNDFILE	RelatedSo
10884	METADATA_TIFF_RESOLUTIONUNIT	Resolutio
10885	METADATA_TIFF_ROWSPERSTRIP	RowsPerSt
10886	METADATA_TIFF_RSAUTHOR	RSAuthor
10887	METADATA_TIFF_RSCOMMENTS	RSComment
10888	METADATA_TIFF_RSKEYWORDS	RSKeyword

ID	Name	Default English V
10889	METADATA_TIFF_RSSUBJECT	RSSubject
10890	METADATA_TIFF_RSTITLE	RSTitle
10891	METADATA_TIFF_SAMPLEFORMAT	SampleFor
10892	METADATA_TIFF_SAMPLESPERPIXEL	SamplesPe
10893	METADATA_TIFF_SATURATION	Saturatio
10894	METADATA_TIFF_SCENECAPTURETYPE	SceneCapt
10895	METADATA_TIFF_SCENETYPE	SceneType
10896	METADATA_TIFF_SENSINGMETHOD	SensingMe
10897	METADATA_TIFF_SHARPNESS	Sharpness
10898	METADATA_TIFF_SHUTTERSPEEDVALUE	ShutterSp
10899	METADATA_TIFF_SMAXSAMPLEVALUE	SMaxSampl
10900	METADATA_TIFF_SMINSAMPLEVALUE	SMinSampl
10901	METADATA_TIFF_SOFTWARE	Software
10902	METADATA_TIFF_SPATIALFREQUENCYRESPONSE	SpatialFr
10903	METADATA_TIFF_SPECTRALSENSITIVITY	SpectralS
10904	METADATA_TIFF_STRIPBYTECOUNTS	StripByte
10905	METADATA_TIFF_STRIPOFFSETS	StripOffs
10906	METADATA_TIFF_STRIPROWCOUNTS	StripRowC
10907	METADATA_TIFF_SUBFILETYPE	SubfileTy
10908	METADATA_TIFF_SUBIFDS	SubIFDs
10909	METADATA_TIFF_SUBJECTAREA	SubjectAr
10910	METADATA_TIFF_SUBJECTDISTANCE	SubjectDi
10911	METADATA_TIFF_SUBJECTDISTANCERANGE	SubjectDi

ID	Name	Default English V
10912	METADATA_TIFF_SUBJECTLOCATION	SubjectLo
10913	METADATA_TIFF_SUBSECTIME	SubsecTim
10914	METADATA_TIFF_SUBSECTIMEDIGITIZED	SubsecTim
10915	METADATA_TIFF_SUBSECTIMEORIGINAL	SubsecTim
10916	METADATA_TIFF_T4OPTIONS	T4Options
10917	METADATA_TIFF_T6OPTIONS	T6Options
10918	METADATA_TIFF_TARGETPRINTER	TargetPri
10919	METADATA_TIFF_THRESHHOLDING	Threshho1
10920	METADATA_TIFF_TILEBYTECOUNTS	TileByteC
10921	METADATA_TIFF_TILELENGTH	TileLengt
10922	METADATA_TIFF_TILEOFFSETS	TileOffse
10923	METADATA_TIFF_TILEWIDTH	TileWidth
10924	METADATA_TIFF_TRANSFERFUNCTION	TransferF
10925	METADATA_TIFF_TRANSFERRANGE	TransferR
10926	METADATA_TIFF_UNIQUECAMERAMODEL	UniqueCam
10927	METADATA_TIFF_USERCOMMENT	UserComme
10928	METADATA_TIFF_VERSIONYEAR	VersionYe
10929	METADATA_TIFF_WANGANNOATION	WangAnnot
10930	METADATA_TIFF_WHITEBALANCE	WhiteBala
10931	METADATA_TIFF_WHITELEVEL	WhiteLeve
10932	METADATA_TIFF_WHITEPOINT	WhitePoin
10933	METADATA_TIFF_XCLIPPATHUNITS	XClipPath
10934	METADATA_TIFF_XMP	XMP

ID	Name	Default English V
10935	METADATA_TIFF_XPOSITION	XPosition
10936	METADATA_TIFF_XRESOLUTION	XResoluti
10937	METADATA_TIFF_YCBCRCOEFFICIENTS	YCbCrCoef
10938	METADATA_TIFF_YCBCRPOSITIONING	YCbCrPosi
10939	METADATA_TIFF_YCBCRSUBSAMPLING	YCbCrSubS
10940	METADATA_TIFF_YCLIPPATHUNITS	YClipPath
10941	METADATA_TIFF_YPOSITION	YPosition
10942	METADATA_TIFF_YRESOLUTION	YResoluti
10943	METADATA_TILES	Tiles
10944	METADATA_TIME	Time
10945	METADATA_TIME_CREATED	Time Crea
10946	METADATA_TIMEZONE	Timezone
10947	METADATA_TITLE	Title
10948	METADATA_TITLE_COMPOSER	Title Com
10949	METADATA_TITLE_SONGNAME_CONTENT_DESCRIPTION	Title/son descripti
10950	METADATA_TITLE_SORT_ORDER	Title sor
10951	METADATA_TO	To
10952	METADATA_TOTAL_EDITING_TIME	Total Edi
10953	METADATA_TOTAL_FRAMES	Total Fra
10954	METADATA_TOTAL_SLACK	Total Sla
10955	METADATA_TRACK_NUMBER	Track num
10956	METADATA_TRANSMISSION_REFERENCE	Transmiss
10957	METADATA_TV_EPISODE_NUMBER	TV Episod

ID	Name	Default English V
10958	METADATA_TV_NETWORK_NAME	TV Network
10959	METADATA_TV_SEASON	TV Season
10960	METADATA_TV_SHOW_NAME	TV Show N
10961	METADATA_TYPE	Type
10962	METADATA_TYPIST	Typist
10963	METADATA_UNAVAILABLE	<Unavaila
10964	METADATA_UNIQUE_FILE_IDENTIFIER	Unique fi
10965	METADATA_UNIQUE_ID	Unique ID
10966	METADATA_UNIQUE_ID_PREDECESSORS	Unique ID
10967	METADATA_UNIQUE_ID_SUCCESSORS	Unique ID
10968	METADATA_UNIT_DIMENSIONS	Unit Dime
10969	METADATA_UNIT_OF_MEASUREMENT	Unit of M
10970	METADATA_UNSYNCHRONIZED_LYRIC_TEXT_TRANSCRIPTION	Unsynchro transcrip
10971	METADATA_UPDATE_NEEDED	Update Ne
10972	METADATA_USER_DEFINED_TEXT_INFORMATION_FRAME	User defi informati
10973	METADATA_USER_DEFINED_URL_LINK_FRAME	User defi
10974	METADATA_VAC	VAC
10975	METADATA_VERSION	Version
10976	METADATA_VIDEO_BITCOUNT	Video Bit
10977	METADATA_VIDEO_DIMENSIONS	Video Dim
10978	METADATA_VIDEO_N_DIMENSIONS	Video {}
10979	METADATA_VIDEO_PLANES	Video Pla

ID	Name	Default English V
10980	METADATA_VIDEO_SIZE	Video Siz
10981	METADATA_VISUAL_CLASS	Visual Cl
10982	METADATA_WARNING	Warning
10983	METADATA_WBS	WBS
10984	METADATA_WBS_PREDECESSORS	WBS Prede
10985	METADATA_WBS_SUCCESSORS	WBS Succ
10986	METADATA_WEB_PAGE	Web Page
10987	METADATA_WIDTH	Width
10988	METADATA_WINDOW_SIZE	Window Si
10989	METADATA_WINDOWS_USER_ACCOUNT	Windows U
10990	METADATA_WORD_COUNT	Word Coun
10991	METADATA_WORK	Work
10992	METADATA_WORK_CONTOUR	Work Cont
10993	METADATA_WORK_VARIANCE	Work Vari
10994	METADATA_WORKGROUP	Workgroup
10995	METADATA_XRESOLUTION	XResoluti
10996	METADATA_YEAR	Year
10997	METADATA_YRESOLUTION	YResoluti
10998	METADATA_AUTHORSHIP	Authorshi
10999	METADATA_BACK_LIGHT	Back ligh
11000	METADATA_BRIGHTNESS_VALUE	Brightnes
11001	METADATA_CAMERA_MANUFACTURER_NAME	Camera ma
11002	METADATA_CAMERA_MODEL_NAME	Camera mo

ID	Name	Default English V
11003	METADATA_CAMERA_SERIAL_NUMBER	Camera se
11004	METADATA_CAPTION_TEXT	Caption t
11005	METADATA_CAPTURE_DATE	Capture d
11006	METADATA_CFA_PATTERN	CFA patte
11007	METADATA_CONTENT_DESCRIPTION_NOTES	Content d
11008	METADATA_CREATION_PATH_VECTOR	Creation
11009	METADATA_DATE_OF_THE_ORIGINAL_IMAGE	Date of t
11010	METADATA_DEFAULT_DISPLAY_HEIGHT	Default d
11011	METADATA_DEFAULT_DISPLAY_WIDTH	Default d
11012	METADATA_DISPLAY_HEIGHT_WIDTH_UNITS	Display h
11013	METADATA_EVENTS_IN_THE_IMAGE	Events in
11014	METADATA_EXPOSURE_BIAS_VALUE	Exposure
11015	METADATA_EXPOSURE_INDEX	Exposure
11016	METADATA_F_NUMBER	F-number
11017	METADATA_FILE_SOURCE	File sour
11018	METADATA_FILM_BRAND	Film bran
11019	METADATA_FILM_CATEGORY	Film cate
11020	METADATA_FILM_FRAME_NUMBER	Film fram
11021	METADATA_FILM_ROLL_NUMBER	Film roll
11022	METADATA_FILM_SIZE	Film size
11023	METADATA_FLASH_ENERGY	Flash ene
11024	METADATA_FLASH_RETURN	Flash ret
11025	METADATA_FOCAL_PLANE_RESOLUTION_UNIT	Focal pla

ID	Name	Default English V
11026	METADATA_FOCAL_PLANE_X_RESOLUTION	Focal pla
11027	METADATA_FOCAL_PLANE_Y_RESOLUTION	Focal pla
11028	METADATA_GROUP_CAPTION	Group cap
11029	METADATA_HIGHEST_RESOLUTION_HEIGHT	Highest r
11030	METADATA_HIGHEST_RESOLUTION_WIDTH	Highest r
11031	METADATA_INTELLECTUAL_PROPERTY_NOTES	Intellect
11032	METADATA_ISO_SPEED_RATINGS	ISO speed
11033	METADATA_LEGAL_BROKER_FOR_THE_DIGITAL_IMAGE	Legal bro digital i
11034	METADATA_LEGAL_BROKER_FOR_THE_ORIGINAL_IMAGE	Legal bro original
11035	METADATA_MAXIMUM_APERTURE_VALUE	Maximum a
11036	METADATA_METERING_MODE	Metering
11037	METADATA_OECF	OECF
11038	METADATA_ORIGINAL_DOCUMENT_SIZE	Original
11039	METADATA_ORIGINAL_MEDIUM	Original
11040	METADATA_ORIGINAL_SCANNED_IMAGE_SIZE	Original
11041	METADATA_PEOPLE_IN_THE_IMAGE	People in
11042	METADATA_PER_PICTURE_NOTES	Per pictu
11043	METADATA_PLACES_IN_THE_IMAGE	Places in
11044	METADATA_SCAN_DATE	Scan date
11045	METADATA_SCAN_OPERATOR_ID	Scan oper
11046	METADATA_SCAN_SOFTWARE_REVISION_DATE	Scan soft
11047	METADATA_SCAN_SOFTWARE	Scan soft

ID	Name	Default English V
11048	METADATA_SCANNER_MANUFACTURER_NAME	Scanner m
11049	METADATA_SCANNER_MODEL_NAME	Scanner m
11050	METADATA_SCANNER_PIXEL_SIZE	Scanner p
11051	METADATA_SCANNER_SERIAL_NUMBER	Scanner s
11052	METADATA_SCENE_ILLUMINANT	Scene ill
11053	METADATA_SCENE_TYPE	Scene typ
11054	METADATA_SENSING_METHOD	Sensing m
11055	METADATA_SERVICE_BUREAU_ORGANIZATION_NAME	Service b name
11056	METADATA_SHARPNESS_APPROXIMATION	Sharpness
11057	METADATA_SOFTWARE_NAME_MANUFACTURER_RELEASE	Software Release
11058	METADATA_SPATIAL_FREQUENCY_RESPONSE	Spatial f
11059	METADATA_SPECIAL_EFFECTS_OPTICAL_FILTER	Special e filter
11060	METADATA_SPECTRAL_SENSITIVITY	Spectral
11061	METADATA_SUBJECT_DISTANCE	Subject d
11062	METADATA_SUBJECT_LOCATION	Subject l
11063	METADATA_TEST_TARGET_IN_THE_IMAGE	Test targ
11064	METADATA_THINGS_IN_THE_IMAGE	Things in
11065	METADATA_TYPE_OF_ORIGINAL	Type of o
11066	METADATA_USER_DEFINED_ID	User defi
11067	METADATA_VALUE	Value
11068	METADATA_UNTITLED	Untitled

ID	Name	Default English V
20001	CONTENT_SLIDE_BEGIN	==== Slid
20002	CONTENT_STREAM_WITH_NUMBER	Stream {0:
20003	CONTENT_IMAGE_WITH_NUMBER	Image {0:
20004	CONTENT_PAGE_WITH_NUMBER	Page {0:>
20005	CONTENT_DIMENSIONS	{0} x {1}
20006	CONTENT_AVI_STREAM_WITH_NUMBER	Stream #{

IGR_SUBFILE_INFO_FLAG_TYPE

Name	Value	Description
IGR_SUBFILE_INFO_FLAG_PASSWORD_PROTECTED	0x0002	Indicates the file requires a password to extract.
IGR_SUBFILE_INFO_FLAG_HAS_COMMENT	0x0004	Indicates the file has a comment attached.

Text Styles

Name	Value	Description
IGR_TEXT_STYLE_BOLD	1	Indicates the text should be rendered bold.
IGR_TEXT_STYLE_ITALIC	2	Indicates the text should be rendered italics.
IGR_TEXT_STYLE_UNDERLINE	4	Indicates the text should be rendered underlined.

3.3 Object reference

3.3.1 Overview

Annotation	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation, Page.GetNextAnnotation, or Page.GetAnnotationCount method.
AnnotationAction	Represents the base class for Link actions.
AnnotationActionGoTo	Represents a goto action that scrolls to a specified location in the document.
AnnotationActionGoToRemote	Represents a goto remote action that opens another document.
AnnotationActionNamed	Represents a named action that can be executed.
AnnotationActionUri	Represents a URI action that opens another document.
AnnotationAppearanceStream	Optional appearance data that controls how the annotation is rendered. Appearance streams enable the annotation to be presented visually in different ways to reflect its interactions with the user. Each appearance stream is a byte stream of a Document Filters supported image format. See Supported Formats, Raster image and Vector image for a list of supported formats. Please note only the SVG format is preserved as an SVG when creating appearance streams. All other formats will be rasterized. Appearance streams are currently for writing only and are not read from PDFs.
AnnotationAppearanceStreams	Represents the appearance streams for an annotation.
AnnotationAztec	Represents an Aztec barcode annotation.
AnnotationBarcode	Implements Annotation Interface. Represents the base class for all barcode annotations.
AnnotationBorderStyle	Represents the border appearance of the annotation.
AnnotationCode128	Represents a Code128 barcode annotation.
AnnotationCode39	Represents a Code39 barcode annotation.

AnnotationDataMatrix	Represents a DataMatrix barcode annotation.
AnnotationDefaultAppearance	Represents the default appearance values for the annotation, independent of any specific appearance stream.
AnnotationEllipse	A circle or ellipse annotation displays an ellipse on the page. When opened, it displays a pop-up window containing the text of the associated note.
AnnotationFreeText	A free text annotation displays text directly on the page. Unlike an ordinary text annotation, a free text annotation has no open or closed state; instead of being displayed in a pop-up window, the text is always visible.
AnnotationGS1_128	Represents a GS1-128 barcode annotation.
AnnotationHighlight	Highlight annotations are used to highlight text content on a page.
AnnotationInk	An ink annotation represents a freehand "scribble" composed of one or more disjoint paths. When opened, it displays a pop-up window containing the text of the associated note.
AnnotationLine	A line annotation displays a single straight line on the page. When opened, it displays a pop-up window containing the text of the associated note.
AnnotationLineEndings	Represents the line ending of the annotation.
AnnotationLink	A link annotation represents either a hypertext link to a destination elsewhere in the document or an action to be performed.
AnnotationNamedDestination	Represents a named destination in the document.
AnnotationNote	Represents the base class for note annotations.
AnnotationNoteList	A list of AnnotationNote objects.
AnnotationPDF417	Represents a PDF417 barcode annotation.
AnnotationPoint	A single point in the document's user space units. For .NET instead see <code>System.Drawing.Point</code> .
AnnotationPointList	A list of AnnotationPoint objects.
AnnotationPointListList	A list containing AnnotationPointList objects.
AnnotationPolygon	Polygon annotations display closed polygons on the page. Such polygons may have any number of vertices connected by straight lines. Polyline

	annotations are similar to polygons, except that the first and last vertex are not implicitly connected.
AnnotationPolyline	Polygon annotations display closed polygons on the page. Such polygons may have any number of vertices connected by straight lines. Polyline annotations are similar to polygons, except that the first and last vertex are not implicitly connected.
AnnotationPopup	A pop-up annotation displays text in a pop-up window for entry and editing. It typically does not appear alone but is associated with a markup annotation, its parent annotation, and is used for editing the parent's text. It has no appearance stream or associated actions of its own.
AnnotationQRCode	Represents a QR code barcode annotation.
AnnotationRect	The AnnotationRect interface represents an annotation's bounding box.
AnnotationRectangle	A rectangle annotation displays a rectangle on the page. When opened, it displays a pop-up window containing the text of the associated note.
AnnotationSquiggly	Squiggly annotations are used to underline text content on a page with a squiggly line.
AnnotationStamp	A rubber stamp annotation displays text or graphics intended to look as if they were stamped on the page with a rubber stamp. When opened, it displays a pop-up window containing the text of the associated note.
AnnotationStickyNote	A text annotation represents a "sticky note" attached to a point in the document. When closed, the annotation appears as an icon; when open, it displays a pop-up window containing the text of the note in a font and size chosen by the viewer application. Text annotations do not scale and rotate with the page; they behave as if the NoZoom and NoRotate annotation flags were always set.
AnnotationStrikeout	Strikeout annotations are used to strike through text content on a page.
AnnotationTextMarkup	Represents the base class for text markup annotations.
AnnotationUnderline	Underline annotations are used to underline text content on a page.
Bookmark	The Bookmark interface allows for the extraction of the hierarchical navigation data for a document. Bookmarks are either extracted directly from the document, or generated from the heading style information

	To obtain this interface, call the Extractor.GetRootBookmark method.
Canvas	The Canvas interface allows rendering of pages to a variety of output devices, including HD HTML, PNG, and PDF. The Canvas object also allows post-processing / image manipulation of output such as annotations, redaction, bates stamping, or general drawing. To obtain this interface, call the DocumentFilters.MakeOutputCanvas or DocumentFilters.MakeOutputCanvasOnStream methods. Note The drawing API is available for bitmap and PDF outputs only.
CompareDocumentSettings	The CompareDocumentSettings controls the compare settings for a document.
CompareDocumentSource	The CompareDocumentSource interface is used when comparing two documents, allowing you to set an extractor and provide optional settings.
CompareResultDifference	The CompareResultDifference represents a single diff in a document or page comparison.
CompareResultDifferenceDetail	The CompareResultDifferenceDetail interface contains information about the words that make up a CompareResultDifference , including the text, page index and bounding box.
CompareResults	The CompareResults interface is the returned object from a call to Extractor::Compare or Page::Compare and is used to enumerate the differences between the two entities.
CompareSettings	The CompareSettings instance allows you to set the comparison type and flags when comparing documents or pages.
DocumentFilters	DocumentFilters (formerly IFileReaders) is the primary factory object in the Document Filters Object Library. You will need to create and initialize an instance of this object to start using the API. It is recommended to define the object in the application scope and create and initialize it only once. The samples for each method are provided in JScript and assume a global instance of the DocumentFilters factory object that is already created and initialized.
Extractor	

	The Extractor interface allows you to extract the content of a document and/or enumerate its sub-documents, such as email attachments and ZIP archives. To obtain this interface, call the DocumentFilters.GetExtractor method. The Extractor interface contains the following methods and properties.
FormElement	The FormElement interface represents a single form control for a given page. FormElements are currently supported for static PDF documents.
FormElementOption	The FormElement interface represents a single form control for a given page. FormElements are currently support for static PDF documents.
FormKeyValue	The FormKeyValue interface is used to provided updated values for Forms. This interface can be created directly.
Hyperlink	The Hyperlink interface represents a single hyperlink on a given page. To obtain this interface, call the Page.GetNextHyperlink method.
IGRFormat	Represents a file-type as returned by GetSupportedFormats .
IGROption	Represents a configuration option as returned by GetAvailableOptions .
Page	The Page interface represents a single page in an image laid-out document. The page allows access to the words on a page, as well as the ability to render it onto a canvas such as TIFF, PNG, or PDF. To obtain this interface, call the Extractor::GetPage.
PageElement	Represents a page element in a document.
RenderPageProperties	The RenderPageProperties interface provides additional information for rendering a page to a canvas. This interface can be created directly.
SubFile	The SubFile interface is a descendant of Extractor, allowing work with sub-documents, extracted from a parent document, by calling the parent's Extractor::GetFirstSubFile and Extractor::GetNextSubFile methods. Open the sub-document associated with an instance of SubFile, in the same way as described for Extractor, allowing processing of sub-

	documents to any depth. This means that text can be extracted and/or sub-documents contained in this SubFile maybe enumerated.
Word	The Word interface allows extraction of words and their bounding boxes when in paginated image mode. To obtain this interface, call the Page.GetFirstWord, Page.GetNextWord methods, or Page.Words property.

3.3.2 Annotation

Annotation interface

The Annotation interface represents a single annotation on a given page.

To obtain this interface, call the [Page.GetFirstAnnotation](#), [Page.GetNextAnnotation](#), or [Page.GetAnnotationCount](#) method.

Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
Annotation::Name property	Indicates the name of the annotation.
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.

Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

Annotation::Annotate method

Applies the annotation to the provided canvas. This is the equivalent of calling the [Canvas::Annotate](#) method.

PROTOTYPE

COM

```
HRESULT Annotate([in] ICanvas2* canvas);
```

Annotation::Appearance property

Contains the appearance streams for the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.AppearanceStreams Appearance { get; set; }
```

JAVA

```
AnnotationAppearanceStreams getAppearance() throws IGRException;  
void setAppearance(AnnotationAppearanceStreams value) throws IGRException;
```

PYTHON

```
@property  
def Appearance(self) -> AnnotationAppearanceStreams  
  
@Appearance.setter  
def Appearance(self, value: AnnotationAppearanceStreams)
```

C++17

```
AnnotationAppearanceStreams getAppearance() const;  
void setAppearance(AnnotationAppearanceStreams value);
```

COM

```
[propget] HRESULT Appearance([out, retval] AnnotationAppearanceStreams* *result);  
[propput] HRESULT Appearance([in] AnnotationAppearanceStreams* value);
```

RETURN VALUE

[AnnotationAppearanceStreams](#)

SEE ALSO

- [AnnotationAppearanceStreams](#)

Annotation::Border property

Indicates the border style of the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.BorderStyle Border { get; set; }
```

JAVA

```
AnnotationBorderStyle getBorder() throws IGRException;  
void setBorder(AnnotationBorderStyle value) throws IGRException;
```

PYTHON

```
@property  
def Border(self) -> AnnotationBorderStyle  
  
@Border.setter  
def Border(self, value: AnnotationBorderStyle)
```

C++17

```
AnnotationBorderStyle getBorder() const;  
void setBorder(AnnotationBorderStyle value);
```

COM

```
[propget] HRESULT Border([out, retval] AnnotationBorderStyle* *result);  
[propput] HRESULT Border([in] AnnotationBorderStyle* value);
```

RETURN VALUE

[AnnotationBorderStyle](#)

SEE ALSO

- [AnnotationBorderStyle](#)

Annotation::Color property

Indicates the color of the annotation. Color used for the following purposes:

- The background of the annotation's icon when closed
- The title bar of the annotation's pop-up window
- The border of a link annotation

.NET

```
System.Drawing.Color Color { get; set; }
```

JAVA

```
string getColor() throws IGRException;  
void setColor(string value) throws IGRException;
```

PYTHON

```
@property  
def Color(self) -> string  
  
@Color.setter  
def Color(self, value: string)
```

C++17

```
Hyland.DocFilters.Color getColor() const;  
void setColor(Hyland.DocFilters.Color value);
```

COM

```
[propget] HRESULT Color([out, retval] BSTR *result);  
[propput] HRESULT Color([in] BSTR value);
```

RETURN VALUE

```
string
```

Annotation::DateCreated property

Indicates the date the annotation was created.

.NET

```
System.DateTime DateCreated { get; set; }
```

JAVA

```
java.util.Date getDateCreated() throws IGRException;  
void setDateCreated(java.util.Date value) throws IGRException;
```

PYTHON

```
@property  
def DateCreated(self) -> datetime  
  
@DateCreated.setter  
def DateCreated(self, value: datetime)
```

C++17

```
Hyland.DocFilters.DateTime getDateCreated() const;  
void setDateCreated(Hyland.DocFilters.DateTime value);
```

COM

```
[propget] HRESULT DateCreated([out, retval] date* *result);  
[propput] HRESULT DateCreated([in] date* value);
```

RETURN VALUE

date

Annotation::DateModified property

Indicates the date the annotation was last modified.

.NET

```
System.DateTime DateModified { get; set; }
```

JAVA

```
java.util.Date getDateModified() throws IGRException;  
void setDateModified(java.util.Date value) throws IGRException;
```

PYTHON

```
@property  
def DateModified(self) -> datetime  
  
@DateModified.setter  
def DateModified(self, value: datetime)
```

C++17

```
Hyland.DocFilters.DateTime getDateModified() const;  
void setDateModified(Hyland.DocFilters.DateTime value);
```

COM

```
[propget] HRESULT DateModified([out, retval] date* *result);  
[propput] HRESULT DateModified([in] date* value);
```

RETURN VALUE

date

Annotation::Flags property

Indicates the flags of the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.Flags Flags { get; set; }
```

JAVA

```
int getFlags() throws IGRException;  
void setFlags(int value) throws IGRException;
```

PYTHON

```
@property  
def Flags(self) -> int  
  
@Flags.setter  
def Flags(self, value: int)
```

C++17

```
int getFlags() const;  
void setFlags(int value);
```

COM

```
[propget] HRESULT Flags([out, retval] int *result);  
[propput] HRESULT Flags([in] int value);
```

RETURN VALUE

`int` : See [IGR_ANNOTATION_FLAGS_TYPE](#) for details.

SEE ALSO

- [IGR_ANNOTATION_FLAGS_TYPE](#)

Annotation::Intent property

String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.

.NET

```
string Intent { get; set; }
```

JAVA

```
string getIntent() throws IGRException;  
void setIntent(string value) throws IGRException;
```

PYTHON

```
@property  
def Intent(self) -> string  
  
@Intent.setter  
def Intent(self, value: string)
```

C++17

```
std::wstring getIntent() const;  
void setIntent(const std::wstring& value);
```

COM

```
[propget] HRESULT Intent([out, retval] BSTR *result);  
[propput] HRESULT Intent([in] BSTR value);
```

RETURN VALUE

```
string
```


Annotation::Name property

Indicates the name of the annotation.

.NET

```
string Name { get; set; }
```

JAVA

```
string getName() throws IGRException;  
void setName(string value) throws IGRException;
```

PYTHON

```
@property  
def Name(self) -> string  
  
@Name.setter  
def Name(self, value: string)
```

C++17

```
std::wstring getName() const;  
void setName(const std::wstring& value);
```

COM

```
[propget] HRESULT Name([out, retval] BSTR *result);  
[propput] HRESULT Name([in] BSTR value);
```

RETURN VALUE

```
string
```

Annotation::Opacity property

The constant opacity value to be used in painting the annotation

.NET

```
System.Double Opacity { get; set; }
```

JAVA

```
System.Double getOpacity() throws IGRException;  
void setOpacity(System.Double value) throws IGRException;
```

PYTHON

```
@property  
def Opacity(self) -> System.Double  
  
@Opacity.setter  
def Opacity(self, value: System.Double)
```

C++17

```
System.Double getOpacity() const;  
void setOpacity(System.Double value);
```

COM

```
[propget] HRESULT Opacity([out, retval] System.Double* *result);  
[propput] HRESULT Opacity([in] System.Double* value);
```

RETURN VALUE

System.Double

Annotation::Popup property

Pop-up annotation for entering or editing the text associated with this annotation.

.NET

```
Hyland.DocumentFilters.Annotations.Popup Popup { get; set; }
```

JAVA

```
AnnotationPopup getPopup() throws IGRException;  
void setPopup(AnnotationPopup value) throws IGRException;
```

PYTHON

```
@property  
def Popup(self) -> AnnotationPopup  
  
@Popup.setter  
def Popup(self, value: AnnotationPopup)
```

C++17

```
AnnotationPopup getPopup() const;  
void setPopup(AnnotationPopup value);
```

COM

```
[propget] HRESULT Popup([out, retval] AnnotationPopup* *result);  
[propput] HRESULT Popup([in] AnnotationPopup* value);
```

RETURN VALUE

[AnnotationPopup](#)

SEE ALSO

- [AnnotationPopup](#)

Annotation::Rect property

Indicates the rectangle of the annotation.

.NET

```
System.Drawing.Rectangle Rect { get; set; }
```

JAVA

```
AnnotationRect getRect() throws IGRException;  
void setRect(AnnotationRect value) throws IGRException;
```

PYTHON

```
@property  
def Rect(self) -> AnnotationRect  
  
@Rect.setter  
def Rect(self, value: AnnotationRect)
```

C++17

```
AnnotationRect getRect() const;  
void setRect(AnnotationRect value);
```

COM

```
[propget] HRESULT Rect([out, retval] AnnotationRect* *result);  
[propput] HRESULT Rect([in] AnnotationRect* value);
```

RETURN VALUE

[AnnotationRect](#)

SEE ALSO

- [AnnotationRectangle](#)

Annotation::Replies property

Contains the list of replies to the annotation.

.NET

```
List<Hyland.DocumentFilters.Annotations.Note> Replies { get; set; }
```

JAVA

```
AnnotationNoteList getReplies() throws IGRException;  
void setReplies(AnnotationNoteList value) throws IGRException;
```

PYTHON

```
@property  
def Replies(self) -> AnnotationNoteList  
  
@Replies.setter  
def Replies(self, value: AnnotationNoteList)
```

C++17

```
std::vector<AnnotationNote> getReplies() const;  
void setReplies(std::vector<AnnotationNote> value);
```

COM

```
[propget] HRESULT Replies([out, retval] AnnotationNoteList* *result);  
[propput] HRESULT Replies([in] AnnotationNoteList* value);
```

RETURN VALUE

[AnnotationNoteList](#)

SEE ALSO

- [AnnotationNoteList](#)

Annotation::Subject property

Text representing a short description of the subject being addressed by the annotation.

.NET

```
string Subject { get; set; }
```

JAVA

```
string getSubject() throws IGRException;  
void setSubject(string value) throws IGRException;
```

PYTHON

```
@property  
def Subject(self) -> string  
  
@Subject.setter  
def Subject(self, value: string)
```

C++17

```
std::wstring getSubject() const;  
void setSubject(const std::wstring& value);
```

COM

```
[propget] HRESULT Subject([out, retval] BSTR *result);  
[propput] HRESULT Subject([in] BSTR value);
```

RETURN VALUE

```
string
```

Annotation::Text property

String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.

.NET

```
string Text { get; set; }
```

JAVA

```
string getText() throws IGRException;  
void setText(string value) throws IGRException;
```

PYTHON

```
@property  
def Text(self) -> string  
  
@Text.setter  
def Text(self, value: string)
```

C++17

```
std::wstring getText() const;  
void setText(const std::wstring& value);
```

COM

```
[proppget] HRESULT Text([out, retval] BSTR *result);  
[propput] HRESULT Text([in] BSTR value);
```

RETURN VALUE

```
string
```

Annotation::Type property

Indicates the type of the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.Type Type { get; set; }
```

JAVA

```
int getType() throws IGRException;  
void setType(int value) throws IGRException;
```

PYTHON

```
@property  
def Type(self) -> int  
  
@Type.setter  
def Type(self, value: int)
```

C++17

```
int getType() const;  
void setType(int value);
```

COM

```
[propget] HRESULT Type([out, retval] int *result);  
[propput] HRESULT Type([in] int value);
```

RETURN VALUE

`int` : See [IGR_ANNOTATION_TYPE](#) for details.

SEE ALSO

- [IGR_ANNOTATION_TYPE](#)

3.3.3 AnnotationAction

AnnotationAction interface

Represents the base class for Link actions.

AnnotationAction::Type property	Indicates the action type.
AnnotationAction::NewWindow property	Indicates if the action is performed in a new window.

Action::Type property

Indicates the action type.

.NET

```
string Type { get; set; }
```

JAVA

```
string getType() throws IGRException;
void setType(string value) throws IGRException;
```

PYTHON

```
@property
def Type(self) -> string

@Type.setter
def Type(self, value: string)
```

C++17

```
std::wstring getType() const;
void setType(const std::wstring& value);
```

COM

```
[propget] HRESULT Type([out, retval] BSTR *result);
[propput] HRESULT Type([in] BSTR value);
```

RETURN VALUE

string : Indicates the action type.

Type	AnnotationAction object
GoTo	AnnotationActionGoTo
GoToR	AnnotationActionGoToRemote
URI	AnnotationActionURI
Named	AnnotationActionNamed

Action::NewWindow property

Indicates if the action is performed in a new window.

.NET

```
bool NewWindow { get; set; }
```

JAVA

```
boolean getNewWindow() throws IGRException;  
void setNewWindow(boolean value) throws IGRException;
```

PYTHON

```
@property  
def NewWindow(self) -> bool  
  
@NewWindow.setter  
def NewWindow(self, value: bool)
```

C++17

```
bool getNewWindow() const;  
void setNewWindow(bool value);
```

COM

```
[propget] HRESULT NewWindow([out, retval] VARIANT_BOOL *result);  
[propput] HRESULT NewWindow([in] VARIANT_BOOL value);
```

RETURN VALUE

bool

3.3.4 AnnotationActionGoTo

AnnotationActionGoTo interface

Represents a goto action that scrolls to a specified location in the document.

AnnotationActionGoTo::Name property	Indicates the named location to scroll to.
AnnotationActionGoTo::Page property	Indicates the page number to scroll to.
AnnotationActionGoTo::Rect property	Indicates the rectangle to scroll to.
AnnotationActionGoTo::Zoom property	Indicates the zoom level to set once scrolled.

INHERITED FROM ANNOTATIONACTION

AnnotationAction::AnnotationAction interface	Represents the base class for Link actions.
AnnotationAction::Type property	Indicates the action type.
AnnotationAction::NewWindow property	Indicates if the action is performed in a new window.

AnnotationActionGoTo::Name property

Indicates the named location to scroll to.

.NET

```
string Name { get; set; }
```

JAVA

```
string getName() throws IGRException;  
void setName(string value) throws IGRException;
```

PYTHON

```
@property  
def Name(self) -> string  
  
@Name.setter  
def Name(self, value: string)
```

C++17

```
std::wstring getName() const;  
void setName(const std::wstring& value);
```

COM

```
[propget] HRESULT Name([out, retval] BSTR *result);  
[propput] HRESULT Name([in] BSTR value);
```

RETURN VALUE

```
string
```

AnnotationActionGoTo::Page property

Indicates the page number to scroll to.

.NET

```
int Page { get; set; }
```

JAVA

```
int getPage() throws IGRException;  
void setPage(int value) throws IGRException;
```

PYTHON

```
@property  
def Page(self) -> int  
  
@Page.setter  
def Page(self, value: int)
```

C++17

```
int getPage() const;  
void setPage(int value);
```

COM

```
[propget] HRESULT Page([out, retval] int *result);  
[propput] HRESULT Page([in] int value);
```

RETURN VALUE

```
int
```

AnnotationActionGoTo::Rect property

Indicates the rectangle to scroll to.

.NET

```
System.Drawing.Rectangle Rect { get; set; }
```

JAVA

```
AnnotationRect getRect() throws IGRException;  
void setRect(AnnotationRect value) throws IGRException;
```

PYTHON

```
@property  
def Rect(self) -> AnnotationRect  
  
@Rect.setter  
def Rect(self, value: AnnotationRect)
```

C++17

```
AnnotationRect getRect() const;  
void setRect(AnnotationRect value);
```

COM

```
[propget] HRESULT Rect([out, retval] AnnotationRect* *result);  
[propput] HRESULT Rect([in] AnnotationRect* value);
```

RETURN VALUE

[AnnotationRect](#)

SEE ALSO

- [AnnotationRect](#)

AnnotationActionGoTo::Zoom property

Indicates the zoom level to set once scrolled.

.NET

```
double Zoom { get; set; }
```

JAVA

```
double getZoom() throws IGRException;  
void setZoom(double value) throws IGRException;
```

PYTHON

```
@property  
def Zoom(self) -> double  
  
@Zoom.setter  
def Zoom(self, value: double)
```

C++17

```
double getZoom() const;  
void setZoom(double value);
```

COM

```
[propget] HRESULT Zoom([out, retval] double* *result);  
[propput] HRESULT Zoom([in] double* value);
```

RETURN VALUE

double

3.3.5 AnnotationActionGoToRemote

AnnotationActionGoToRemote interface

Represents a goto remote action that opens another document.

AnnotationActionGoToRemote::Filename property	Indicates the file to open.

INHERITED FROM ANNOTATIONACTIONGOTO

AnnotationActionGoTo::AnnotationActionGoTo interface	Represents a goto action that scrolls to a specified location in the document.
AnnotationActionGoTo::Name property	Indicates the named location to scroll to.
AnnotationActionGoTo::Page property	Indicates the page number to scroll to.
AnnotationActionGoTo::Rect property	Indicates the rectangle to scroll to.
AnnotationActionGoTo::Zoom property	Indicates the zoom level to set once scrolled.

AnnotationActionGoToRemote::Filename property

Indicates the file to open.

.NET

```
string Filename { get; set; }
```

JAVA

```
string getFilename() throws IGRException;  
void setFilename(string value) throws IGRException;
```

PYTHON

```
@property  
def Filename(self) -> string  
  
@Filename.setter  
def Filename(self, value: string)
```

C++17

```
std::wstring getFilename() const;  
void setFilename(const std::wstring& value);
```

COM

```
[propget] HRESULT Filename([out, retval] BSTR *result);  
[propput] HRESULT Filename([in] BSTR value);
```

RETURN VALUE

string

3.3.6 AnnotationActionNamed

AnnotationActionNamed interface

Represents a named action that can be executed.

AnnotationActionNamed::Name property	Indicates the name of the action.

INHERITED FROM ANNOTATIONACTION

AnnotationAction::AnnotationAction interface	Represents the base class for Link actions.
AnnotationAction::Type property	Indicates the action type.
AnnotationAction::NewWindow property	Indicates if the action is performed in a new window.

AnnotationActionNamed::Name property

Indicates the name of the action.

.NET

```
string Name { get; set; }
```

JAVA

```
string getName() throws IGRException;
void setName(string value) throws IGRException;
```

PYTHON

```
@property
def Name(self) -> string

@Name.setter
def Name(self, value: string)
```

C++17

```
std::wstring getName() const;
void setName(const std::wstring& value);
```

COM

```
[propget] HRESULT Name([out, retval] BSTR *result);
[propput] HRESULT Name([in] BSTR value);
```

RETURN VALUE

string : A string representing a predefined named action to be performed by the viewer.

Name	Action
NextPage	Go to the next page of the document.
PrevPage	Go to the previous page of the document.
FirstPage	Go to the first page of the document.
LastPage	Go to the last page of the document.

3.3.7 AnnotationActionUri

AnnotationActionUri interface

Represents a URI action that opens another document.

AnnotationActionUri::Uri property	Indicates the URI to open.

INHERITED FROM ANNOTATIONACTION

AnnotationAction::AnnotationAction interface	Represents the base class for Link actions.
AnnotationAction::Type property	Indicates the action type.
AnnotationAction::NewWindow property	Indicates if the action is performed in a new window.

AnnotationActionUri::Uri property

Indicates the URI to open.

.NET

```
string Uri { get; set; }
```

JAVA

```
string getUri() throws IGRException;  
void setUri(string value) throws IGRException;
```

PYTHON

```
@property  
def Uri(self) -> string  
  
@Uri.setter  
def Uri(self, value: string)
```

C++17

```
std::wstring getUri() const;  
void setUri(const std::wstring& value);
```

COM

```
[propget] HRESULT Uri([out, retval] BSTR *result);  
[propput] HRESULT Uri([in] BSTR value);
```

RETURN VALUE

```
string
```

3.3.8 AnnotationAppearanceStream

AnnotationAppearanceStream interface

Optional appearance data that controls how the annotation is rendered. Appearance streams enable the annotation to be presented visually in different ways to reflect its interactions with the user. Each appearance stream is a byte stream of a Document Filters supported image format. See Supported Formats, Raster image and Vector image for a list of supported formats. Please note only the SVG format is preserved as an SVG when creating appearance streams. All other formats will be rasterized.

Appearance streams are currently for writing only and are not read from PDFs.

AnnotationAppearanceStream::ContentType property	The content type of the Content stream.
AnnotationAppearanceStream::Encoding property	String describing the content encoding. This must be "Base64" unless the content of the stream is a SVG in which case this value may also be "none". For .NET this value is always "Base64".
AnnotationAppearanceStream::Content property	The content in a supported format, such as SVG.

AnnotationAppearanceStream::ContentType property

The content type of the Content stream.

.NET

```
string ContentType { get; set; }
```

JAVA

```
string getContentType() throws IGRException;  
void setContentType(string value) throws IGRException;
```

PYTHON

```
@property  
def ContentType(self) -> string  
  
@ContentType.setter  
def ContentType(self, value: string)
```

C++17

```
std::wstring getContentType() const;  
void setContentType(const std::wstring& value);
```

COM

```
[propget] HRESULT ContentType([out, retval] BSTR *result);  
[propput] HRESULT ContentType([in] BSTR value);
```

RETURN VALUE

```
string
```


AnnotationAppearanceStream::Encoding property

String describing the content encoding. This must be "Base64" unless the content of the stream is a SVG in which case this value may also be "none". For .NET this value is always "Base64".

.NET

```
string Encoding { get; set; }
```

JAVA

```
string getEncoding() throws IGRException;  
void setEncoding(string value) throws IGRException;
```

PYTHON

```
@property  
def Encoding(self) -> string  
  
@Encoding.setter  
def Encoding(self, value: string)
```

C++17

```
std::wstring getEncoding() const;  
void setEncoding(const std::wstring& value);
```

COM

```
[propget] HRESULT Encoding([out, retval] BSTR *result);  
[propput] HRESULT Encoding([in] BSTR value);
```

RETURN VALUE

```
string
```

AnnotationAppearanceStream::Content property

The content in a supported format, such as SVG.

.NET

```
System.IO.Stream Content { get; set; }
```

JAVA

```
string getContent() throws IGRException;  
void setContent(string value) throws IGRException;
```

PYTHON

```
@property  
def Content(self) -> string  
  
@Content.setter  
def Content(self, value: string)
```

C++17

```
std::wstring getContent() const;  
void setContent(const std::wstring& value);
```

COM

```
[propget] HRESULT Content([out, retval] BSTR *result);  
[propput] HRESULT Content([in] BSTR value);
```

RETURN VALUE

```
string
```

3.3.9 AnnotationAppearanceStreams

AnnotationAppearanceStreams interface

Represents the appearance streams for an annotation.

AnnotationAppearanceStreams::Normal property	Indicates the default appearance stream for the annotation.
AnnotationAppearanceStreams::RollOverProperty	Indicates the mouse-over appearance stream for the annotation.
AnnotationAppearanceStreams::Down property	Indicates the appearance stream for when the annotation is clicked.

AnnotationAppearanceStreams::Normal property

Indicates the default appearance stream for the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.AppearanceStream Normal { get; set; }
```

JAVA

```
AnnotationAppearanceStream getNormal() throws IGRException;  
void setNormal(AnnotationAppearanceStream value) throws IGRException;
```

PYTHON

```
@property  
def Normal(self) -> AnnotationAppearanceStream  
  
@Normal.setter  
def Normal(self, value: AnnotationAppearanceStream)
```

C++17

```
AnnotationAppearanceStream getNormal() const;  
void setNormal(AnnotationAppearanceStream value);
```

COM

```
[propget] HRESULT Normal([out, retval] AnnotationAppearanceStream* *result);  
[propput] HRESULT Normal([in] AnnotationAppearanceStream* value);
```

RETURN VALUE

[AnnotationAppearanceStream](#)

SEE ALSO

- [AnnotationAppearanceStream](#)

AnnotationAppearanceStreams::RollOverProperty

Indicates the mouse-over appearance stream for the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.AppearanceStream RollOver { get; set; }
```

JAVA

```
AnnotationAppearanceStream getRollOver() throws IGRException;  
void setRollOver(AnnotationAppearanceStream value) throws IGRException;
```

PYTHON

```
@property  
def RollOver(self) -> AnnotationAppearanceStream  
  
@RollOver.setter  
def RollOver(self, value: AnnotationAppearanceStream)
```

C++17

```
AnnotationAppearanceStream getRollOver() const;  
void setRollOver(AnnotationAppearanceStream value);
```

COM

```
[propget] HRESULT RollOver([out, retval] AnnotationAppearanceStream* *result);  
[propput] HRESULT RollOver([in] AnnotationAppearanceStream* value);
```

RETURN VALUE

[AnnotationAppearanceStream](#)

SEE ALSO

- [AnnotationAppearanceStream](#)

AnnotationAppearanceStreams::Down property

Indicates the appearance stream for when the annotation is clicked.

.NET

```
Hyland.DocumentFilters.Annotations.AppearanceStream Down { get; set; }
```

JAVA

```
AnnotationAppearanceStream getDown() throws IGRException;  
void setDown(AnnotationAppearanceStream value) throws IGRException;
```

PYTHON

```
@property  
def Down(self) -> AnnotationAppearanceStream  
  
@Down.setter  
def Down(self, value: AnnotationAppearanceStream)
```

C++17

```
AnnotationAppearanceStream getDown() const;  
void setDown(AnnotationAppearanceStream value);
```

COM

```
[propget] HRESULT Down([out, retval] AnnotationAppearanceStream* *result);  
[propput] HRESULT Down([in] AnnotationAppearanceStream* value);
```

RETURN VALUE

[AnnotationAppearanceStream](#)

SEE ALSO

- [AnnotationAppearanceStream](#)

3.3.10 AnnotationAztec

AnnotationAztec interface

Represents an Aztec barcode annotation.

INHERITED FROM ANNOTATIONBARCODE

AnnotationBarcode::AnnotationBarcode interface	Implements Annotation Interface. Represents the base class for all barcode annotations.
AnnotationBarcode::SubType property	Indicates the subtype of the barcode.
AnnotationBarcode::Caption property	Indicates the caption of the barcode.
AnnotationBarcode::Content property	Indicates the content of the barcode.
AnnotationBarcode::ErrorCorrectionLevel property	Indicates the error correction level of the barcode.
AnnotationBarcode::Margin property	Indicates the margin of the barcode.
AnnotationBarcode::BgColor property	Indicates the background color of the barcode.

3.3.11 AnnotationBarcode

AnnotationBarcode interface

Implements [Annotation](#) Interface. Represents the base class for all barcode annotations.

AnnotationBarcode::SubType property	Indicates the subtype of the barcode.
AnnotationBarcode::Caption property	Indicates the caption of the barcode.
AnnotationBarcode::Content property	Indicates the content of the barcode.
AnnotationBarcode::ErrorCorrectionLevel property	Indicates the error correction level of the barcode.
AnnotationBarcode::Margin property	Indicates the margin of the barcode.
AnnotationBarcode::BgColor property	Indicates the background color of the barcode.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.

Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
Annotation::Name property	Indicates the name of the annotation.
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationBarcode::SubType property

Indicates the subtype of the barcode.

.NET

```
string SubType { get; set; }
```

JAVA

```
string getSubType() throws IGRException;
void setSubType(string value) throws IGRException;
```

PYTHON

```
@property
def SubType(self) -> string

@SubType.setter
def SubType(self, value: string)
```

C++17

```
std::wstring getSubType() const;
void setSubType(const std::wstring& value);
```

COM

```
[propget] HRESULT SubType([out, retval] BSTR *result);
[propput] HRESULT SubType([in] BSTR value);
```

RETURN VALUE

`string` : String representing the barcode type. Can be one of:

Name	Annotation object type
aztec	AnnotationAztec
code39	AnnotationCode39
code93	AnnotationCode93
code128	AnnotationCode128
datamatrix	AnnotationDataMatrix
gs1128	AnnotationGS1_128

Name	Annotation object type
pdf417	AnnotationPDF417
qr	AnnotationQrCode

AnnotationBarcode::Caption property

Indicates the caption of the barcode.

.NET

```
string Caption { get; set; }
```

JAVA

```
string getCaption() throws IGRException;  
void setCaption(string value) throws IGRException;
```

PYTHON

```
@property  
def Caption(self) -> string  
  
@Caption.setter  
def Caption(self, value: string)
```

C++17

```
std::wstring getCaption() const;  
void setCaption(const std::wstring& value);
```

COM

```
[propget] HRESULT Caption([out, retval] BSTR *result);  
[propput] HRESULT Caption([in] BSTR value);
```

RETURN VALUE

```
string
```

AnnotationBarcode::Content property

Indicates the content of the barcode.

.NET

```
string Content { get; set; }
```

JAVA

```
string getContent() throws IGRException;  
void setContent(string value) throws IGRException;
```

PYTHON

```
@property  
def Content(self) -> string  
  
@Content.setter  
def Content(self, value: string)
```

C++17

```
std::wstring getContent() const;  
void setContent(const std::wstring& value);
```

COM

```
[propget] HRESULT Content([out, retval] BSTR *result);  
[propput] HRESULT Content([in] BSTR value);
```

RETURN VALUE

```
string
```

AnnotationBarcode::ErrorCorrectionLevel property

Indicates the error correction level of the barcode.

.NET

```
int ErrorCorrectionLevel { get; set; }
```

JAVA

```
int getErrorCorrectionLevel() throws IGRException;  
void setErrorCorrectionLevel(int value) throws IGRException;
```

PYTHON

```
@property  
def ErrorCorrectionLevel(self) -> int  
  
@ErrorCorrectionLevel.setter  
def ErrorCorrectionLevel(self, value: int)
```

C++17

```
int getErrorCorrectionLevel() const;  
void setErrorCorrectionLevel(int value);
```

COM

```
[propget] HRESULT ErrorCorrectionLevel([out, retval] int *result);  
[propput] HRESULT ErrorCorrectionLevel([in] int value);
```

RETURN VALUE

int

AnnotationBarcode::Margin property

Indicates the margin of the barcode.

.NET

```
int Margin { get; set; }
```

JAVA

```
int getMargin() throws IGRException;  
void setMargin(int value) throws IGRException;
```

PYTHON

```
@property  
def Margin(self) -> int  
  
@Margin.setter  
def Margin(self, value: int)
```

C++17

```
int getMargin() const;  
void setMargin(int value);
```

COM

```
[propget] HRESULT Margin([out, retval] int *result);  
[propput] HRESULT Margin([in] int value);
```

RETURN VALUE

int

AnnotationBarcode::BgColor property

Indicates the background color of the barcode.

.NET

```
System.Drawing.Color BgColor { get; set; }
```

JAVA

```
string getBgColor() throws IGRException;  
void setBgColor(string value) throws IGRException;
```

PYTHON

```
@property  
def BgColor(self) -> string  
  
@BgColor.setter  
def BgColor(self, value: string)
```

C++17

```
Hyland.DocFilters.Color getBgColor() const;  
void setBgColor(Hyland.DocFilters.Color value);
```

COM

```
[propget] HRESULT BgColor([out, retval] BSTR *result);  
[propput] HRESULT BgColor([in] BSTR value);
```

RETURN VALUE

```
string
```


3.3.12 AnnotationBorderStyle

AnnotationBorderStyle interface

Represents the border appearance of the annotation.

AnnotationBorderStyle::Width property	The width of the border.
AnnotationBorderStyle::Intensity property	The intensity of the border.
AnnotationBorderStyle::Type property	The type of border style.
AnnotationBorderStyle::Dash property	The dash pattern of the border.

AnnotationBorderStyle::Width property

The width of the border.

.NET

```
System.Double Width { get; set; }
```

JAVA

```
System.Double getWidth() throws IGRException;  
void setWidth(System.Double value) throws IGRException;
```

PYTHON

```
@property  
def Width(self) -> System.Double  
  
@Width.setter  
def Width(self, value: System.Double)
```

C++17

```
System.Double getWidth() const;  
void setWidth(System.Double value);
```

COM

```
[propget] HRESULT Width([out, retval] System.Double* *result);  
[propput] HRESULT Width([in] System.Double* value);
```

RETURN VALUE

System.Double

AnnotationBorderStyle::Intensity property

The intensity of the border.

.NET

```
double Intensity { get; set; }
```

JAVA

```
double getIntensity() throws IGRException;  
void setIntensity(double value) throws IGRException;
```

PYTHON

```
@property  
def Intensity(self) -> double  
  
@Intensity.setter  
def Intensity(self, value: double)
```

C++17

```
double getIntensity() const;  
void setIntensity(double value);
```

COM

```
[propget] HRESULT Intensity([out, retval] double* *result);  
[propput] HRESULT Intensity([in] double* value);
```

RETURN VALUE

double

AnnotationBorderStyle::Type property

The type of border style.

.NET

```
Hyland.DocumentFilters.Annotations.BorderStyleType Type { get; set; }
```

JAVA

```
int getType() throws IGRException;  
void setType(int value) throws IGRException;
```

PYTHON

```
@property  
def Type(self) -> int  
  
@Type.setter  
def Type(self, value: int)
```

C++17

```
int getType() const;  
void setType(int value);
```

COM

```
[propget] HRESULT Type([out, retval] int *result);  
[propput] HRESULT Type([in] int value);
```

RETURN VALUE

int : Must be one of the following:

Integer Value	Border Style Type
0	Unknown
1	Solid
2	Dashed
3	Beveled
4	Inset
5	Underline

AnnotationBorderStyle::Dash property

The dash pattern of the border.

.NET

```
List<int> Dash { get; set; }
```

JAVA

```
int[] getDash() throws IGRException;  
void setDash(int[] value) throws IGRException;
```

PYTHON

```
@property  
def Dash(self) -> int[]  
  
@Dash.setter  
def Dash(self, value: int[])
```

C++17

```
int[] getDash() const;  
void setDash(int[] value);
```

COM

```
[propget] HRESULT Dash([out, retval] int[]* *result);  
[propput] HRESULT Dash([in] int[]* value);
```

RETURN VALUE

```
int[]
```

3.3.13 AnnotationCode128

AnnotationCode128 interface

Represents a Code128 barcode annotation.

INHERITED FROM ANNOTATIONBARCODE

AnnotationBarcode::AnnotationBarcode interface	Implements Annotation Interface. Represents the base class for all barcode annotations.
AnnotationBarcode::SubType property	Indicates the subtype of the barcode.
AnnotationBarcode::Caption property	Indicates the caption of the barcode.
AnnotationBarcode::Content property	Indicates the content of the barcode.
AnnotationBarcode::ErrorCorrectionLevel property	Indicates the error correction level of the barcode.
AnnotationBarcode::Margin property	Indicates the margin of the barcode.
AnnotationBarcode::BgColor property	Indicates the background color of the barcode.

3.3.14 AnnotationCode39

AnnotationCode39 interface

Represents a Code39 barcode annotation.

INHERITED FROM ANNOTATIONBARCODE

AnnotationBarcode::AnnotationBarcode interface	Implements Annotation Interface. Represents the base class for all barcode annotations.
AnnotationBarcode::SubType property	Indicates the subtype of the barcode.
AnnotationBarcode::Caption property	Indicates the caption of the barcode.
AnnotationBarcode::Content property	Indicates the content of the barcode.
AnnotationBarcode::ErrorCorrectionLevel property	Indicates the error correction level of the barcode.
AnnotationBarcode::Margin property	Indicates the margin of the barcode.
AnnotationBarcode::BgColor property	Indicates the background color of the barcode.

3.3.15 AnnotationDataMatrix

AnnotationDataMatrix interface

Represents a DataMatrix barcode annotation.

INHERITED FROM ANNOTATIONBARCODE

AnnotationBarcode::AnnotationBarcode interface	Implements Annotation Interface. Represents the base class for all barcode annotations.
AnnotationBarcode::SubType property	Indicates the subtype of the barcode.
AnnotationBarcode::Caption property	Indicates the caption of the barcode.
AnnotationBarcode::Content property	Indicates the content of the barcode.
AnnotationBarcode::ErrorCorrectionLevel property	Indicates the error correction level of the barcode.
AnnotationBarcode::Margin property	Indicates the margin of the barcode.
AnnotationBarcode::BgColor property	Indicates the background color of the barcode.

3.3.16 AnnotationDefaultAppearance

AnnotationDefaultAppearance interface

Represents the default appearance values for the annotation, independent of any specific appearance stream.

AnnotationDefaultAppearance::FontName property	Indicates the font name.
AnnotationDefaultAppearance::FontSize property	Indicates the font size.
AnnotationDefaultAppearance::TextColor property	Indicates the text color.

AnnotationDefaultAppearance::FontName property

Indicates the font name.

.NET

```
string FontName { get; set; }
```

JAVA

```
string getFontName() throws IGRException;  
void setFontName(string value) throws IGRException;
```

PYTHON

```
@property  
def FontName(self) -> string  
  
@FontName.setter  
def FontName(self, value: string)
```

C++17

```
std::wstring getFontName() const;  
void setFontName(const std::wstring& value);
```

COM

```
[propget] HRESULT FontName([out, retval] BSTR *result);  
[propput] HRESULT FontName([in] BSTR value);
```

RETURN VALUE

```
string
```

AnnotationDefaultAppearance::FontSize property

Indicates the font size.

.NET

```
double FontSize { get; set; }
```

JAVA

```
double getFontSize() throws IGRException;  
void setFontSize(double value) throws IGRException;
```

PYTHON

```
@property  
def FontSize(self) -> double  
  
@FontSize.setter  
def FontSize(self, value: double)
```

C++17

```
double getFontSize() const;  
void setFontSize(double value);
```

COM

```
[propget] HRESULT FontSize([out, retval] double* *result);  
[propput] HRESULT FontSize([in] double* value);
```

RETURN VALUE

double

AnnotationDefaultAppearance::TextColor property

Indicates the text color.

.NET

```
System.Drawing.Color TextColor { get; set; }
```

JAVA

```
string getTextColor() throws IGRException;  
void setTextColor(string value) throws IGRException;
```

PYTHON

```
@property  
def TextColor(self) -> string  
  
@TextColor.setter  
def TextColor(self, value: string)
```

C++17

```
Hyland.DocFilters.Color getTextColor() const;  
void setTextColor(Hyland.DocFilters.Color value);
```

COM

```
[propget] HRESULT TextColor([out, retval] BSTR *result);  
[propput] HRESULT TextColor([in] BSTR value);
```

RETURN VALUE

```
string
```

3.3.17 AnnotationEllipse

AnnotationEllipse interface

A circle or ellipse annotation displays an ellipse on the page. When opened, it displays a pop-up window containing the text of the associated note.

AnnotationEllipse::InteriorColor property	Indicates the interior color of the annotation.
AnnotationEllipse::RectDifferences property	Indicates the rectangle differences of the annotation.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
	Indicates the name of the annotation.

Annotation::Name property	
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationEllipse::InteriorColor property

Indicates the interior color of the annotation.

.NET

```
System.Drawing.Color InteriorColor { get; set; }
```

JAVA

```
string getInteriorColor() throws IGRException;  
void setInteriorColor(string value) throws IGRException;
```

PYTHON

```
@property  
def InteriorColor(self) -> string  
  
@InteriorColor.setter  
def InteriorColor(self, value: string)
```

C++17

```
Hyland.DocFilters.Color getInteriorColor() const;  
void setInteriorColor(Hyland.DocFilters.Color value);
```

COM

```
[propget] HRESULT InteriorColor([out, retval] BSTR *result);  
[propput] HRESULT InteriorColor([in] BSTR value);
```

RETURN VALUE

string

AnnotationEllipse::RectDifferences property

Indicates the rectangle differences of the annotation.

.NET

```
System.Drawing.Rectangle RectDifferences { get; set; }
```

JAVA

```
AnnotationRect getRectDifferences() throws IGRException;
void setRectDifferences(AnnotationRect value) throws IGRException;
```

PYTHON

```
@property
def RectDifferences(self) -> AnnotationRect

@RectDifferences.setter
def RectDifferences(self, value: AnnotationRect)
```

C++17

```
AnnotationRect getRectDifferences() const;
void setRectDifferences(AnnotationRect value);
```

COM

```
[propget] HRESULT RectDifferences([out, retval] AnnotationRect* *result);
[propput] HRESULT RectDifferences([in] AnnotationRect* value);
```

RETURN VALUE

[AnnotationRect](#)

ADDITIONAL INFORMATION

A set of four numbers describing the numerical differences between two rectangles: the Rect entry of the annotation and a rectangle contained within that rectangle. The inner rectangle is where the annotation's text should be displayed. Any border styles and/or border effects specified by BS and BE entries, respectively, are applied to the border of the inner rectangle. The four numbers correspond to the differences in default user space between the left, top, right, and bottom coordinates of Rect and those of the inner rectangle, respectively. Each value must be greater than or equal to 0. The sum of the top and bottom differences must be less than the height of Rect, and the sum of the left and right differences must be less than the width of Rect.

SEE ALSO

- [AnnotationRect](#)

3.3.18 AnnotationFreeText

AnnotationFreeText interface

A free text annotation displays text directly on the page. Unlike an ordinary text annotation, a free text annotation has no open or closed state; instead of being displayed in a pop-up window, the text is always visible.

AnnotationFreeText::Alignment property	Indicates the text alignment of the annotation.
AnnotationFreeText::DefaultAppearance property	Indicates the default appearance values for the annotation.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
	Indicates the name of the annotation.

Annotation::Name property	
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationFreeText::Alignment property

Indicates the text alignment of the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.AlignmentType Alignment { get; set; }
```

JAVA

```
int getAlignment() throws IGRException;
void setAlignment(int value) throws IGRException;
```

PYTHON

```
@property
def Alignment(self) -> int

@Alignment.setter
def Alignment(self, value: int)
```

C++17

```
int getAlignment() const;
void setAlignment(int value);
```

COM

```
[propget] HRESULT Alignment([out, retval] int *result);
[propput] HRESULT Alignment([in] int value);
```

RETURN VALUE

int : Must be one of the following:

Integer Value	AnnotationAlignment
0	Left
1	Center
2	Right

AnnotationFreeText::DefaultAppearance property

Indicates the default appearance values for the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.DefaultAppearance DefaultAppearance { get; set; }
```

JAVA

```
AnnotationDefaultAppearance getDefaultAppearance() throws IGRException;  
void setDefaultAppearance(AnnotationDefaultAppearance value) throws IGRException;
```

PYTHON

```
@property  
def DefaultAppearance(self) -> AnnotationDefaultAppearance  
  
@DefaultAppearance.setter  
def DefaultAppearance(self, value: AnnotationDefaultAppearance)
```

C++17

```
AnnotationDefaultAppearance getDefaultAppearance() const;  
void setDefaultAppearance(AnnotationDefaultAppearance value);
```

COM

```
[propget] HRESULT DefaultAppearance([out, retval] AnnotationDefaultAppearance* *result);  
[propput] HRESULT DefaultAppearance([in] AnnotationDefaultAppearance* value);
```

RETURN VALUE

[AnnotationDefaultAppearance](#)

SEE ALSO

- [AnnotationDefaultAppearance](#)

3.3.19 AnnotationGS1_128

AnnotationGS1_128 interface

Represents a GS1-128 barcode annotation.

AnnotationGS1_128::Parts property	Contains the application identifiers for the barcode.
AnnotationGS1_128::AutoCaption property	Indicates if the barcode should be automatically captioned.
AnnotationGS1_128::AddPart method	Add a new part to the GS1-128 annotation

INHERITED FROM ANNOTATIONBARCODE

AnnotationBarcode::AnnotationBarcode interface	Implements Annotation Interface. Represents the base class for all barcode annotations.
AnnotationBarcode::SubType property	Indicates the subtype of the barcode.
AnnotationBarcode::Caption property	Indicates the caption of the barcode.
AnnotationBarcode::Content property	Indicates the content of the barcode.
AnnotationBarcode::ErrorCorrectionLevel property	Indicates the error correction level of the barcode.
AnnotationBarcode::Margin property	Indicates the margin of the barcode.
AnnotationBarcode::BgColor property	Indicates the background color of the barcode.

AnnotationGS1_128::Parts property

Contains the application identifiers for the barcode.

.NET

```
IDictionary<string, string> Parts { get; set; }
```

C++17

```
IDictionary<string, string> getParts() const;  
void setParts(IDictionary<string, string> value);
```

RETURN VALUE

```
IDictionary<string, string>
```

AnnotationGS1_128::AutoCaption property

Indicates if the barcode should be automatically captioned.

.NET

```
bool AutoCaption { get; set; }
```

JAVA

```
boolean getAutoCaption() throws IGRException;  
void setAutoCaption(boolean value) throws IGRException;
```

PYTHON

```
@property  
def AutoCaption(self) -> bool  
  
@AutoCaption.setter  
def AutoCaption(self, value: bool)
```

C++17

```
bool getAutoCaption() const;  
void setAutoCaption(bool value);
```

COM

```
[propget] HRESULT AutoCaption([out, retval] VARIANT_BOOL *result);  
[propput] HRESULT AutoCaption([in] VARIANT_BOOL value);
```

RETURN VALUE

bool

AnnotationGS1_128::AddPart method

Add a new part to the GS1-128 annotation

PROTOTYPE

.NET

```
void AddPart(string name, string value)
```

JAVA

```
void AddPart(string name, string value) throws IGRException;
```

PYTHON

```
def AddPart(self, name: string, value: string) -> void
```

C++17

```
void AddPart(const std::wstring& name, const std::wstring& value)
```

PARAMETERS

`name: string` : The part name

`value: string` : The part value

3.3.20 AnnotationHighlight

AnnotationHighlight interface

Highlight annotations are used to highlight text content on a page.

INHERITED FROM ANNOTATIONTEXTMARKUP

AnnotationTextMarkup::AnnotationTextMarkup interface	Represents the base class for text markup annotations.

3.3.21 AnnotationInk

AnnotationInk interface

An ink annotation represents a freehand "scribble" composed of one or more disjoint paths. When opened, it displays a pop-up window containing the text of the associated note.

AnnotationInk::Points property	The first entry (i.e. path) in .

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
Annotation::Name property	Indicates the name of the annotation.

Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationInk::Points property

The first entry (i.e. path) in .

.NET

```
IList<System.Drawing.Point> Points { get; set; }
```

JAVA

```
AnnotationPointListList getPoints() throws IGRException;  
void setPoints(AnnotationPointListList value) throws IGRException;
```

PYTHON

```
@property  
def Points(self) -> AnnotationPointListList  
  
@Points.setter  
def Points(self, value: AnnotationPointListList)
```

C++17

```
AnnotationPointListList getPoints() const;  
void setPoints(AnnotationPointListList value);
```

COM

```
[propget] HRESULT Points([out, retval] AnnotationPointListList* *result);  
[propput] HRESULT Points([in] AnnotationPointListList* value);
```

RETURN VALUE

AnnotationPointListList

3.3.22 AnnotationLine

AnnotationLine interface

A line annotation displays a single straight line on the page. When opened, it displays a pop-up window containing the text of the associated note.

AnnotationLine::InteriorColor property	Indicates the interior color of the annotation.
AnnotationLine::LineEndings property	Indicates the line endings of the annotation.
AnnotationLine::Points property	Indicates the points of the annotation.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.

Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
Annotation::Name property	Indicates the name of the annotation.
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationLine::InteriorColor property

Indicates the interior color of the annotation.

.NET

```
System.Drawing.Color InteriorColor { get; set; }
```

JAVA

```
string getInteriorColor() throws IGRException;  
void setInteriorColor(string value) throws IGRException;
```

PYTHON

```
@property  
def InteriorColor(self) -> string  
  
@InteriorColor.setter  
def InteriorColor(self, value: string)
```

C++17

```
Hyland.DocFilters.Color getInteriorColor() const;  
void setInteriorColor(Hyland.DocFilters.Color value);
```

COM

```
[propget] HRESULT InteriorColor([out, retval] BSTR *result);  
[propput] HRESULT InteriorColor([in] BSTR value);
```

RETURN VALUE

string

AnnotationLine::LineEndings property

Indicates the line endings of the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.LineEndings LineEndings { get; set; }
```

JAVA

```
AnnotationLineEndings getLineEndings() throws IGRException;  
void setLineEndings(AnnotationLineEndings value) throws IGRException;
```

PYTHON

```
@property  
def LineEndings(self) -> AnnotationLineEndings  
  
@LineEndings.setter  
def LineEndings(self, value: AnnotationLineEndings)
```

C++17

```
AnnotationLineEndings getLineEndings() const;  
void setLineEndings(AnnotationLineEndings value);
```

COM

```
[propget] HRESULT LineEndings([out, retval] AnnotationLineEndings* *result);  
[propput] HRESULT LineEndings([in] AnnotationLineEndings* value);
```

RETURN VALUE

[AnnotationLineEndings](#)

SEE ALSO

- [AnnotationLineEndings](#)

AnnotationLine::Points property

Indicates the points of the annotation.

.NET

```
IList<System.Drawing.Point> Points { get; set; }
```

JAVA

```
AnnotationPointList getPoints() throws IGRException;  
void setPoints(AnnotationPointList value) throws IGRException;
```

PYTHON

```
@property  
def Points(self) -> AnnotationPointList  
  
@Points.setter  
def Points(self, value: AnnotationPointList)
```

C++17

```
AnnotationPointList getPoints() const;  
void setPoints(AnnotationPointList value);
```

COM

```
[propget] HRESULT Points([out, retval] AnnotationPointList* *result);  
[propput] HRESULT Points([in] AnnotationPointList* value);
```

RETURN VALUE

[AnnotationPointList](#)

3.3.23 AnnotationLineEndings

AnnotationLineEndings interface

Represents the line ending of the annotation.

AnnotationLineEndings::Begin property	The type of line ending at the beginning.
AnnotationLineEndings::End property	The type of line ending at the end.

AnnotationLineEndings::Begin property

The type of line ending at the beginning.

.NET

```
Hyland.DocumentFilters.Annotations.LineEndingType Begin { get; set; }
```

JAVA

```
int getBegin() throws IGREException;  
void setBegin(int value) throws IGREException;
```

PYTHON

```
@property  
def Begin(self) -> int  
  
@Begin.setter  
def Begin(self, value: int)
```

C++17

```
int getBegin() const;  
void setBegin(int value);
```

COM

```
[propget] HRESULT Begin([out, retval] int *result);  
[propput] HRESULT Begin([in] int value);
```

RETURN VALUE

int : One of the following:

Name	Value
None	0
Square	1
Circle	2
Diamond	3
OpenArrow	4
ClosedArrow	5

Name	Value
Butt	6
RightOpenArrow	7
RightClosedArrow	8
Slash	9

AnnotationLineEndings::End property

The type of line ending at the end.

.NET

```
Hyland.DocumentFilters.Annotations.LineEndingType End { get; set; }
```

JAVA

```
int getEnd() throws IGREException;
void setEnd(int value) throws IGREException;
```

PYTHON

```
@property
def End(self) -> int

@End.setter
def End(self, value: int)
```

C++17

```
int getEnd() const;
void setEnd(int value);
```

COM

```
[propget] HRESULT End([out, retval] int *result);
[propput] HRESULT End([in] int value);
```

RETURN VALUE

int : One of the following:

Name	Value
None	0
Square	1
Circle	2
Diamond	3
OpenArrow	4
ClosedArrow	5

Name	Value
Butt	6
RightOpenArrow	7
RightClosedArrow	8
Slash	9

3.3.24 AnnotationLink

AnnotationLink interface

A link annotation represents either a hypertext link to a destination elsewhere in the document or an action to be performed.

AnnotationLink::Highlight property	Indicates the highlight type of the annotation.
AnnotationLink::Action property	Indicates the action to be performed when the annotation is activated.
AnnotationLink::Points property	Indicates the points of the annotation.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.

Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
Annotation::Name property	Indicates the name of the annotation.
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationLink::Highlight property

Indicates the highlight type of the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.HighlightType Highlight { get; set; }
```

JAVA

```
int getHighlight() throws IGREException;
void setHighlight(int value) throws IGREException;
```

PYTHON

```
@property
def Highlight(self) -> int

@Highlight.setter
def Highlight(self, value: int)
```

C++17

```
int getHighlight() const;
void setHighlight(int value);
```

COM

```
[propget] HRESULT Highlight([out, retval] int *result);
[propput] HRESULT Highlight([in] int value);
```

RETURN VALUE

int : One of the following:

Value	Highlight
0	Unknown
1	Invert
2	Outline
3	Push

AnnotationLink::Action property

Indicates the action to be performed when the annotation is activated.

.NET

```
Hyland.DocumentFilters.Annotations.Action Action { get; set; }
```

JAVA

```
AnnotationAction getAction() throws IGRException;  
void setAction(AnnotationAction value) throws IGRException;
```

PYTHON

```
@property  
def Action(self) -> AnnotationAction  
  
@Action.setter  
def Action(self, value: AnnotationAction)
```

C++17

```
AnnotationAction getAction() const;  
void setAction(AnnotationAction value);
```

COM

```
[propget] HRESULT Action([out, retval] AnnotationAction* *result);  
[propput] HRESULT Action([in] AnnotationAction* value);
```

RETURN VALUE

[AnnotationAction](#))

SEE ALSO

- [AnnotationAction](#)

AnnotationLink::Points property

Indicates the points of the annotation.

.NET

```
IList<System.Drawing.Point> Points { get; set; }
```

JAVA

```
AnnotationPointList getPoints() throws IGRException;  
void setPoints(AnnotationPointList value) throws IGRException;
```

PYTHON

```
@property  
def Points(self) -> AnnotationPointList  
  
@Points.setter  
def Points(self, value: AnnotationPointList)
```

C++17

```
AnnotationPointList getPoints() const;  
void setPoints(AnnotationPointList value);
```

COM

```
[propget] HRESULT Points([out, retval] AnnotationPointList* *result);  
[propput] HRESULT Points([in] AnnotationPointList* value);
```

RETURN VALUE

[AnnotationPointList](#)

SEE ALSO

- [AnnotationPointList](#)

3.3.25 AnnotationNamedDestination

AnnotationNamedDestination interface

Represents a named destination in the document.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
Annotation::Name property	Indicates the name of the annotation.
Annotation::Opacity property	The constant opacity value to be used in painting the annotation

Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

SAMPLE CODE**C#**

```

using var source = api.OpenExtractor(sourceFilename, OpenMode.Paginated);
using var canvas = api.MakeOutputCanvas(destFilename, CanvasType.PDF);

canvas.BlankPage((int)(8.5 * 96), (int)(11 * 96));
canvas.SetBrush(0, 1);
canvas.SetFont("Arial", 11, 0);

// create a simple table of contents
int y = 36;
for (int i = 0; i < source.PageCount; ++i)
{
    var text = $"Page {i + 1}";
    canvas.TextOut(36, y, text);
    canvas.Annotate(new Annotations.Link
    {
        Action = new Annotations.ActionGoTo
        {
            Name = $"sourcePage{i + 1}"
        },
        Rect = new System.Drawing.Rectangle(36, y, canvas.TextWidth(text), lineHeight)
    });
    y += (int) (lineHeight * 1.2);
}

// render the actual document and create named destinations
for (var i = 0; i < source.PageCount; ++i)
{
    using (var page = source.GetPage(i))
    {
        canvas.RenderPage(page);
        canvas.Annotate(new Annotations.NamedDestination
        {

```

```
        Name = $"sourcePage{i+1}",  
        Rect = new System.Drawing.Rectangle(0, 0, page.Width, page.Height),  
    });  
}
```

SEE ALSO

- [Link Interface](#)

3.3.26 AnnotationNote

AnnotationNote interface

Represents the base class for note annotations.

AnnotationNote::State property	Indicates the state of the annotation.
AnnotationNote::StateModel property	Indicates the state model of the annotation.
AnnotationNote::Author property	Indicates the author of the annotation.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.

Annotation::Name property	Indicates the name of the annotation.
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationNote::State property

Indicates the state of the annotation.

.NET

```
string State { get; set; }
```

JAVA

```
string getState() throws IGRException;
void setState(string value) throws IGRException;
```

PYTHON

```
@property
def State(self) -> string

@State.setter
def State(self, value: string)
```

C++17

```
std::wstring getState() const;
void setState(const std::wstring& value);
```

COM

```
[propget] HRESULT State([out, retval] BSTR *result);
[propput] HRESULT State([in] BSTR value);
```

RETURN VALUE

`string` : Paired with stateModel, state represents a model specific value.

StateModel	Valid states
Marked	Marked, Unmarked
Review	Accepted, Rejected, Cancelled, Completed, None

AnnotationNote::StateModel property

Indicates the state model of the annotation.

.NET

```
string StateModel { get; set; }
```

JAVA

```
string getStateModel() throws IGRException;  
void setStateModel(string value) throws IGRException;
```

PYTHON

```
@property  
def StateModel(self) -> string  
  
@StateModel.setter  
def StateModel(self, value: string)
```

C++17

```
std::wstring getStateModel() const;  
void setStateModel(const std::wstring& value);
```

COM

```
[propget] HRESULT StateModel([out, retval] BSTR *result);  
[propput] HRESULT StateModel([in] BSTR value);
```

RETURN VALUE

```
string
```

AnnotationNote::Author property

Indicates the author of the annotation.

.NET

```
string Author { get; set; }
```

JAVA

```
string getAuthor() throws IGRException;  
void setAuthor(string value) throws IGRException;
```

PYTHON

```
@property  
def Author(self) -> string  
  
@Author.setter  
def Author(self, value: string)
```

C++17

```
std::wstring getAuthor() const;  
void setAuthor(const std::wstring& value);
```

COM

```
[propget] HRESULT Author([out, retval] BSTR *result);  
[propput] HRESULT Author([in] BSTR value);
```

RETURN VALUE

```
string
```

3.3.27 AnnotationNoteList

AnnotationNoteList interface

A list of AnnotationNote objects.

AnnotationNoteList::Count method	Returns the number of elements.
AnnotationNoteList::Add method	Adds an element.
AnnotationNoteList::Remove method	Removes an element.
AnnotationNoteList::Clear method	Removes all elements from the list.
AnnotationNoteList::Item method	Returns a specific element from the list.

AnnotationNoteList::Count method

Returns the number of elements.

PROTOTYPE

.NET

```
int GetCount()
```

JAVA

```
int GetCount() throws IGREException;
```

PYTHON

```
def GetCount(self) -> int
```

C++17

```
int GetCount()
```

RETURN VALUE

`int` : The number of items in the list.

AnnotationNoteList::Add method

Adds an element.

PROTOTYPE

.NET

```
void Add(AnnotationNote note)
```

JAVA

```
void Add(AnnotationNote note) throws IGRException;
```

PYTHON

```
def Add(self, note: AnnotationNote) -> void
```

C++17

```
void Add(AnnotationNote note)
```

PARAMETERS

`note: AnnotationNote` : The item to add.

AnnotationNoteList::Remove method

Removes an element.

PROTOTYPE

.NET

```
void Remove(int index)
```

JAVA

```
void Remove(int index) throws IGREException;
```

PYTHON

```
def Remove(self, index: int) -> void
```

C++17

```
void Remove(int index)
```

PARAMETERS

`index: int` : The index of the item to be remove.

AnnotationNoteList::Clear method

Removes all elements from the list.

PROTOTYPE

.NET

```
void Clear()
```

JAVA

```
void Clear() throws IGRException;
```

PYTHON

```
def Clear(self) -> void
```

C++17

```
void Clear()
```


AnnotationNoteList::Item method

Returns a specific element from the list.

PROTOTYPE

.NET

```
AnnotationNote Get(int index)
```

JAVA

```
AnnotationNote Get(int index) throws IGRException;
```

PYTHON

```
def Get(self, index: int) -> AnnotationNote
```

C++17

```
AnnotationNote Get(int index)
```

PARAMETERS

`index: int` : The index of the item being requested.

RETURN VALUE

`AnnotationNote` : The requested item.

3.3.28 AnnotationPDF417

AnnotationPDF417 interface

Represents a PDF417 barcode annotation.

INHERITED FROM ANNOTATIONBARCODE

AnnotationBarcode::AnnotationBarcode interface	Implements Annotation Interface. Represents the base class for all barcode annotations.
AnnotationBarcode::SubType property	Indicates the subtype of the barcode.
AnnotationBarcode::Caption property	Indicates the caption of the barcode.
AnnotationBarcode::Content property	Indicates the content of the barcode.
AnnotationBarcode::ErrorCorrectionLevel property	Indicates the error correction level of the barcode.
AnnotationBarcode::Margin property	Indicates the margin of the barcode.
AnnotationBarcode::BgColor property	Indicates the background color of the barcode.

3.3.29 AnnotationPoint

AnnotationPoint interface

A single point in the document's user space units. For .NET instead see `System.Drawing.Point`.

<code>AnnotationPoint::X</code> property	The horizontal value of the point in the document's user space units.
<code>AnnotationPoint::Y</code> property	The vertical value of the point in the document's user space units.

AnnotationPoint::X property

The horizontal value of the point in the document's user space units.

.NET

```
int X { get; set; }
```

JAVA

```
int getX() throws IGRException;  
void setX(int value) throws IGRException;
```

PYTHON

```
@property  
def X(self) -> int  
  
@X.setter  
def X(self, value: int)
```

C++17

```
int getX() const;  
void setX(int value);
```

COM

```
[propget] HRESULT X([out, retval] int *result);  
[propput] HRESULT X([in] int value);
```

RETURN VALUE

```
int
```

AnnotationPoint::Y property

The vertical value of the point in the document's user space units.

.NET

```
int Y { get; set; }
```

JAVA

```
int getY() throws IGRException;  
void setY(int value) throws IGRException;
```

PYTHON

```
@property  
def Y(self) -> int  
  
@Y.setter  
def Y(self, value: int)
```

C++17

```
int getY() const;  
void setY(int value);
```

COM

```
[propget] HRESULT Y([out, retval] int *result);  
[propput] HRESULT Y([in] int value);
```

RETURN VALUE

```
int
```

3.3.30 AnnotationPointList

AnnotationPointList interface

A list of [AnnotationPoint](#) objects.

AnnotationPointList::Count method	Returns the number of elements.
AnnotationPointList::Add method	Adds an element.
AnnotationPointList::Remove method	Removes an element.
AnnotationPointList::Clear method	Removes all elements from the list.
AnnotationPointList::Item method	Returns a specific element from the list.

AnnotationPointList::Count method

Returns the number of elements.

PROTOTYPE

.NET

```
int GetCount()
```

JAVA

```
int GetCount() throws IGREException;
```

PYTHON

```
def GetCount(self) -> int
```

C++17

```
int GetCount()
```

RETURN VALUE

`int` : The number of items in the list.

AnnotationPointList::Add method

Adds an element.

PROTOTYPE

.NET

```
void Add(AnnotationPoint note)
```

JAVA

```
void Add(AnnotationPoint note) throws IGRException;
```

PYTHON

```
def Add(self, note: AnnotationPoint) -> void
```

C++17

```
void Add(AnnotationPoint note)
```

PARAMETERS

`note: AnnotationPoint` : The item to add.

AnnotationPointList::Remove method

Removes an element.

PROTOTYPE

.NET

```
void Remove(int index)
```

JAVA

```
void Remove(int index) throws IGREException;
```

PYTHON

```
def Remove(self, index: int) -> void
```

C++17

```
void Remove(int index)
```

PARAMETERS

`index: int` : The index of the item to be remove.

AnnotationPointList::Clear method

Removes all elements from the list.

PROTOTYPE

.NET

```
void Clear()
```

JAVA

```
void Clear() throws IGRException;
```

PYTHON

```
def Clear(self) -> void
```

C++17

```
void Clear()
```

AnnotationPointList::Item method

Returns a specific element from the list.

PROTOTYPE

.NET

```
AnnotationPoint Get(int index)
```

JAVA

```
AnnotationPoint Get(int index) throws IGRException;
```

PYTHON

```
def Get(self, index: int) -> AnnotationPoint
```

C++17

```
AnnotationPoint Get(int index)
```

PARAMETERS

`index: int` : The index of the item being requested.

RETURN VALUE

`AnnotationPoint` : The requested item.

3.3.31 AnnotationPointListList

AnnotationPointListList interface

A list containing [AnnotationPointList](#) objects.

AnnotationPointListList::Count method	Returns the number of elements.
AnnotationPointListList::Add method	Adds an element.
AnnotationPointListList::Remove method	Removes an element.
AnnotationPointListList::Clear method	Removes all elements from the list.
AnnotationPointListList::Item method	Returns a specific element from the list.

AnnotationPointListList::Count method

Returns the number of elements.

PROTOTYPE

.NET

```
int GetCount()
```

JAVA

```
int GetCount() throws IGREException;
```

PYTHON

```
def GetCount(self) -> int
```

C++17

```
int GetCount()
```

RETURN VALUE

`int` : The number of items in the list.

AnnotationPointListList::Add method

Adds an element.

PROTOTYPE

.NET

```
void Add(AnnotationPointList note)
```

JAVA

```
void Add(AnnotationPointList note) throws IGREException;
```

PYTHON

```
def Add(self, note: AnnotationPointList) -> void
```

C++17

```
void Add(AnnotationPointList note)
```

PARAMETERS

`note: AnnotationPointList` : The item to add.

AnnotationPointListList::Remove method

Removes an element.

PROTOTYPE

.NET

```
void Remove(int index)
```

JAVA

```
void Remove(int index) throws IGREException;
```

PYTHON

```
def Remove(self, index: int) -> void
```

C++17

```
void Remove(int index)
```

PARAMETERS

`index: int` : The index of the item to be remove.

AnnotationPointListList::Clear method

Removes all elements from the list.

PROTOTYPE

.NET

```
void Clear()
```

JAVA

```
void Clear() throws IGRException;
```

PYTHON

```
def Clear(self) -> void
```

C++17

```
void Clear()
```


AnnotationPointListList::Item method

Returns a specific element from the list.

PROTOTYPE

.NET

```
AnnotationPointList Get(int index)
```

JAVA

```
AnnotationPointList Get(int index) throws IGREException;
```

PYTHON

```
def Get(self, index: int) -> AnnotationPointList
```

C++17

```
AnnotationPointList Get(int index)
```

PARAMETERS

`index: int` : The index of the item being requested.

RETURN VALUE

`AnnotationPointList` : The requested item.

3.3.32 AnnotationPolygon

AnnotationPolygon interface

Polygon annotations display closed polygons on the page. Such polygons may have any number of vertices connected by straight lines. Polyline annotations are similar to polygons, except that the first and last vertex are not implicitly connected.

AnnotationPolygon::InteriorColor property	Indicates the interior color of the annotation.
AnnotationPolygon::Points property	Indicates the points of the annotation.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.

Annotation::Name property	Indicates the name of the annotation.
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationPolygon::InteriorColor property

Indicates the interior color of the annotation.

.NET

```
System.Drawing.Color InteriorColor { get; set; }
```

JAVA

```
string getInteriorColor() throws IGRException;  
void setInteriorColor(string value) throws IGRException;
```

PYTHON

```
@property  
def InteriorColor(self) -> string  
  
@InteriorColor.setter  
def InteriorColor(self, value: string)
```

C++17

```
Hyland.DocFilters.Color getInteriorColor() const;  
void setInteriorColor(Hyland.DocFilters.Color value);
```

COM

```
[propget] HRESULT InteriorColor([out, retval] BSTR *result);  
[propput] HRESULT InteriorColor([in] BSTR value);
```

RETURN VALUE

string

AnnotationPolygon::Points property

Indicates the points of the annotation.

.NET

```
IList<System.Drawing.Point> Points { get; set; }
```

JAVA

```
AnnotationPointList getPoints() throws IGRException;  
void setPoints(AnnotationPointList value) throws IGRException;
```

PYTHON

```
@property  
def Points(self) -> AnnotationPointList  
  
@Points.setter  
def Points(self, value: AnnotationPointList)
```

C++17

```
AnnotationPointList getPoints() const;  
void setPoints(AnnotationPointList value);
```

COM

```
[propget] HRESULT Points([out, retval] AnnotationPointList* *result);  
[propput] HRESULT Points([in] AnnotationPointList* value);
```

RETURN VALUE

[AnnotationPointList](#)

SEE ALSO

- [AnnotationPointList](#)

3.3.33 AnnotationPolyline

AnnotationPolyline interface

Polygon annotations display closed polygons on the page. Such polygons may have any number of vertices connected by straight lines. Polyline annotations are similar to polygons, except that the first and last vertex are not implicitly connected.

AnnotationPolyLine::Points property	Indicates the points of the annotation.
AnnotationPolyLine::LineEndings property	Indicates the line endings of the annotation.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.

Annotation::Name property	Indicates the name of the annotation.
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationPolyLine::Points property

Indicates the points of the annotation.

.NET

```
IList<System.Drawing.Point> Points { get; set; }
```

JAVA

```
AnnotationPointList getPoints() throws IGRException;  
void setPoints(AnnotationPointList value) throws IGRException;
```

PYTHON

```
@property  
def Points(self) -> AnnotationPointList  
  
@Points.setter  
def Points(self, value: AnnotationPointList)
```

C++17

```
AnnotationPointList getPoints() const;  
void setPoints(AnnotationPointList value);
```

COM

```
[propget] HRESULT Points([out, retval] AnnotationPointList* *result);  
[propput] HRESULT Points([in] AnnotationPointList* value);
```

RETURN VALUE

[AnnotationPointList](#)

SEE ALSO

- [AnnotationPointList](#)

AnnotationPolyLine::LineEndings property

Indicates the line endings of the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.LineEndings LineEndings { get; set; }
```

JAVA

```
AnnotationLineEndings getLineEndings() throws IGRException;  
void setLineEndings(AnnotationLineEndings value) throws IGRException;
```

PYTHON

```
@property  
def LineEndings(self) -> AnnotationLineEndings  
  
@LineEndings.setter  
def LineEndings(self, value: AnnotationLineEndings)
```

C++17

```
AnnotationLineEndings getLineEndings() const;  
void setLineEndings(AnnotationLineEndings value);
```

COM

```
[propget] HRESULT LineEndings([out, retval] AnnotationLineEndings* *result);  
[propput] HRESULT LineEndings([in] AnnotationLineEndings* value);
```

RETURN VALUE

AnnotationLineEndings

3.3.34 AnnotationPopup

AnnotationPopup interface

A pop-up annotation displays text in a pop-up window for entry and editing. It typically does not appear alone but is associated with a markup annotation, its parent annotation, and is used for editing the parent's text.

It has no appearance stream or associated actions of its own.

AnnotationPopup::Open property	Indicates whether the annotation is open/visible.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
	Indicates the name of the annotation.

Annotation::Name property	
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationPopup::Open property

Indicates whether the annotation is open/visible.

.NET

```
bool Open { get; set; }
```

JAVA

```
boolean getOpen() throws IGRException;  
void setOpen(boolean value) throws IGRException;
```

PYTHON

```
@property  
def Open(self) -> bool  
  
@Open.setter  
def Open(self, value: bool)
```

C++17

```
bool getOpen() const;  
void setOpen(bool value);
```

COM

```
[propget] HRESULT Open([out, retval] VARIANT_BOOL *result);  
[propput] HRESULT Open([in] VARIANT_BOOL value);
```

RETURN VALUE

bool

3.3.35 AnnotationQrCode

AnnotationQrCode interface

Represents a QR code barcode annotation.

INHERITED FROM ANNOTATIONBARCODE

AnnotationBarcode::AnnotationBarcode interface	Implements Annotation Interface. Represents the base class for all barcode annotations.
AnnotationBarcode::SubType property	Indicates the subtype of the barcode.
AnnotationBarcode::Caption property	Indicates the caption of the barcode.
AnnotationBarcode::Content property	Indicates the content of the barcode.
AnnotationBarcode::ErrorCorrectionLevel property	Indicates the error correction level of the barcode.
AnnotationBarcode::Margin property	Indicates the margin of the barcode.
AnnotationBarcode::BgColor property	Indicates the background color of the barcode.

3.3.36 AnnotationRect

AnnotationRect interface

The AnnotationRect interface represents an annotation's bounding box.

AnnotationRect::Left property	The Left property returns the left coordinate of the annotation's bounding box.
AnnotationRect::Top property	The Top property returns the top coordinate of the annotation's bounding box.
AnnotationRect::Right property	The Right property returns the right coordinate of the annotation's bounding box.
AnnotationRect::Bottom property	The Bottom property returns the bottom coordinate of the annotation's bounding box.

AnnotationRect::Left property

The Left property returns the left coordinate of the annotation's bounding box.

.NET

```
double Left { get; set; }
```

JAVA

```
double getLeft() throws IGRException;  
void setLeft(double value) throws IGRException;
```

PYTHON

```
@property  
def Left(self) -> double  
  
@Left.setter  
def Left(self, value: double)
```

C++17

```
double getLeft() const;  
void setLeft(double value);
```

COM

```
[propget] HRESULT Left([out, retval] double* *result);  
[propput] HRESULT Left([in] double* value);
```

RETURN VALUE

double

AnnotationRect::Top property

The Top property returns the top coordinate of the annotation's bounding box.

.NET

```
double Top { get; set; }
```

JAVA

```
double getTop() throws IGRException;  
void setTop(double value) throws IGRException;
```

PYTHON

```
@property  
def Top(self) -> double  
  
@Top.setter  
def Top(self, value: double)
```

C++17

```
double getTop() const;  
void setTop(double value);
```

COM

```
[propget] HRESULT Top([out, retval] double* *result);  
[propput] HRESULT Top([in] double* value);
```

RETURN VALUE

double

AnnotationRect::Right property

The Right property returns the right coordinate of the annotation's bounding box.

.NET

```
double Right { get; set; }
```

JAVA

```
double getRight() throws IGRException;  
void setRight(double value) throws IGRException;
```

PYTHON

```
@property  
def Right(self) -> double  
  
@Right.setter  
def Right(self, value: double)
```

C++17

```
double getRight() const;  
void setRight(double value);
```

COM

```
[propget] HRESULT Right([out, retval] double* *result);  
[propput] HRESULT Right([in] double* value);
```

RETURN VALUE

double

AnnotationRect::Bottom property

The Bottom property returns the bottom coordinate of the annotation's bounding box.

.NET

```
double Bottom { get; set; }
```

JAVA

```
double getBottom() throws IGRException;  
void setBottom(double value) throws IGRException;
```

PYTHON

```
@property  
def Bottom(self) -> double  
  
@Bottom.setter  
def Bottom(self, value: double)
```

C++17

```
double getBottom() const;  
void setBottom(double value);
```

COM

```
[propget] HRESULT Bottom([out, retval] double* *result);  
[propput] HRESULT Bottom([in] double* value);
```

RETURN VALUE

double

3.3.37 AnnotationRectangle

AnnotationRectangle interface

A rectangle annotation displays a rectangle on the page. When opened, it displays a pop-up window containing the text of the associated note.

AnnotationRectangle::InteriorColor property	Indicates the interior color of the annotation.
AnnotationRectangle::RectDifferences property	Indicates the rectangle differences of the annotation.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
	Indicates the name of the annotation.

Annotation::Name property	
Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationRectangle::InteriorColor property

Indicates the interior color of the annotation.

.NET

```
System.Drawing.Color InteriorColor { get; set; }
```

JAVA

```
string getInteriorColor() throws IGRException;  
void setInteriorColor(string value) throws IGRException;
```

PYTHON

```
@property  
def InteriorColor(self) -> string  
  
@InteriorColor.setter  
def InteriorColor(self, value: string)
```

C++17

```
Hyland.DocFilters.Color getInteriorColor() const;  
void setInteriorColor(Hyland.DocFilters.Color value);
```

COM

```
[propget] HRESULT InteriorColor([out, retval] BSTR *result);  
[propput] HRESULT InteriorColor([in] BSTR value);
```

RETURN VALUE

string

AnnotationRectangle::RectDifferences property

Indicates the rectangle differences of the annotation.

.NET

```
System.Drawing.Rectangle RectDifferences { get; set; }
```

JAVA

```
AnnotationRect getRectDifferences() throws IGRException;
void setRectDifferences(AnnotationRect value) throws IGRException;
```

PYTHON

```
@property
def RectDifferences(self) -> AnnotationRect

@RectDifferences.setter
def RectDifferences(self, value: AnnotationRect)
```

C++17

```
AnnotationRect getRectDifferences() const;
void setRectDifferences(AnnotationRect value);
```

COM

```
[propget] HRESULT RectDifferences([out, retval] AnnotationRect* *result);
[propput] HRESULT RectDifferences([in] AnnotationRect* value);
```

RETURN VALUE

[AnnotationRect](#)

ADDITIONAL INFORMATION

A set of four numbers describing the numerical differences between two rectangles: the Rect entry of the annotation and a rectangle contained within that rectangle. The inner rectangle is where the annotation's text should be displayed. Any border styles and/or border effects specified by BS and BE entries, respectively, are applied to the border of the inner rectangle. The four numbers correspond to the differences in default user space between the left, top, right, and bottom coordinates of Rect and those of the inner rectangle, respectively. Each value must be greater than or equal to 0. The sum of the top and bottom differences must be less than the height of Rect, and the sum of the left and right differences must be less than the width of Rect.

3.3.38 AnnotationSquiggly

AnnotationSquiggly interface

Squiggly annotations are used to underline text content on a page with a squiggly line.

INHERITED FROM ANNOTATIONTEXTMARKUP

AnnotationTextMarkup::AnnotationTextMarkup interface	Represents the base class for text markup annotations.

3.3.39 AnnotationStamp

AnnotationStamp interface

A rubber stamp annotation displays text or graphics intended to look as if they were stamped on the page with a rubber stamp. When opened, it displays a pop-up window containing the text of the associated note.

AnnotationStamp::DefaultAppearance property	Indicates the default appearance values for the annotation.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
Annotation::Name property	Indicates the name of the annotation.

Annotation::Opacity property	The constant opacity value to be used in painting the annotation
Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

AnnotationStamp::DefaultAppearance property

Indicates the default appearance values for the annotation.

.NET

```
Hyland.DocumentFilters.Annotations.DefaultAppearance DefaultAppearance { get; set; }
```

JAVA

```
AnnotationDefaultAppearance getDefaultAppearance() throws IGRException;
void setDefaultAppearance(AnnotationDefaultAppearance value) throws IGRException;
```

PYTHON

```
@property
def DefaultAppearance(self) -> AnnotationDefaultAppearance

@DefaultAppearance.setter
def DefaultAppearance(self, value: AnnotationDefaultAppearance)
```

C++17

```
AnnotationDefaultAppearance getDefaultAppearance() const;
void setDefaultAppearance(AnnotationDefaultAppearance value);
```

COM

```
[propget] HRESULT DefaultAppearance([out, retval] AnnotationDefaultAppearance* *result);
[propput] HRESULT DefaultAppearance([in] AnnotationDefaultAppearance* value);
```

RETURN VALUE

[AnnotationDefaultAppearance](#)

3.3.40 AnnotationStickyNote

AnnotationStickyNote interface

A text annotation represents a "sticky note" attached to a point in the document. When closed, the annotation appears as an icon; when open, it displays a pop-up window containing the text of the note in a font and size chosen by the viewer application. Text annotations do not scale and rotate with the page; they behave as if the NoZoom and NoRotate annotation flags were always set.

AnnotationStickyNote::Open property	Indicates if the annotation is open/visible.
AnnotationStickyNote::IconName property	Indicates the name of the icon to use when the annotation is closed.

INHERITED FROM ANNOTATIONNOTE

AnnotationNote::AnnotationNote interface	Represents the base class for note annotations.
AnnotationNote::State property	Indicates the state of the annotation.
AnnotationNote::StateModel property	Indicates the state model of the annotation.
AnnotationNote::Author property	Indicates the author of the annotation.

AnnotationStickyNote::Open property

Indicates if the annotation is open/visible.

.NET

```
bool Open { get; set; }
```

JAVA

```
boolean getOpen() throws IGRException;  
void setOpen(boolean value) throws IGRException;
```

PYTHON

```
@property  
def Open(self) -> bool  
  
@Open.setter  
def Open(self, value: bool)
```

C++17

```
bool getOpen() const;  
void setOpen(bool value);
```

COM

```
[propget] HRESULT Open([out, retval] VARIANT_BOOL *result);  
[propput] HRESULT Open([in] VARIANT_BOOL value);
```

RETURN VALUE

bool

AnnotationStickyNote::IconName property

Indicates the name of the icon to use when the annotation is closed.

.NET

```
string IconName { get; set; }
```

JAVA

```
string getIconName() throws IGRException;  
void setIconName(string value) throws IGRException;
```

PYTHON

```
@property  
def IconName(self) -> string  
  
@IconName.setter  
def IconName(self, value: string)
```

C++17

```
std::wstring getIconName() const;  
void setIconName(const std::wstring& value);
```

COM

```
[propget] HRESULT IconName([out, retval] BSTR *result);  
[propput] HRESULT IconName([in] BSTR value);
```

RETURN VALUE

```
string
```

3.3.41 AnnotationStrikeout

AnnotationStrikeout interface

Strikeout annotations are used to strike through text content on a page.

INHERITED FROM ANNOTATIONTEXTMARKUP

AnnotationTextMarkup::AnnotationTextMarkup interface	Represents the base class for text markup annotations.

3.3.42 AnnotationTextMarkup

AnnotationTextMarkup interface

Represents the base class for text markup annotations.

INHERITED FROM ANNOTATION

Annotation::Annotation interface	The Annotation interface represents a single annotation on a given page. To obtain this interface, call the Page.GetFirstAnnotation , Page.GetNextAnnotation , or Page.GetAnnotationCount method.
Annotation::Annotate method	Applies the annotation to the provided canvas. This is the equivalent of calling the Canvas::Annotate method.
Annotation::Appearance property	Contains the appearance streams for the annotation.
Annotation::Border property	Indicates the border style of the annotation.
Annotation::Color property	Indicates the color of the annotation.
Annotation::DateCreated property	Indicates the date the annotation was created.
Annotation::DateModified property	Indicates the date the annotation was last modified.
Annotation::Flags property	Indicates the flags of the annotation.
Annotation::Intent property	String containing a name describing the intent of the markup annotation. Intents allow viewer applications to distinguish between different uses and behaviors of a single markup annotation type. If this entry is not present or its value is the same as the annotation type, the annotation has no explicit intent and should behave in a generic manner in a viewer app.
Annotation::Name property	Indicates the name of the annotation.
Annotation::Opacity property	The constant opacity value to be used in painting the annotation

Annotation::Popup property	Pop-up annotation for entering or editing the text associated with this annotation.
Annotation::Rect property	Indicates the rectangle of the annotation.
Annotation::Replies property	Contains the list of replies to the annotation.
Annotation::Subject property	Text representing a short description of the subject being addressed by the annotation.
Annotation::Text property	String to be displayed for the annotation or, if this type of annotation does not display text, an alternate description of the annotation's contents in human-readable form.
Annotation::Type property	Indicates the type of the annotation.

3.3.43 AnnotationUnderline

AnnotationUnderline interface

Underline annotations are used to underline text content on a page.

INHERITED FROM ANNOTATIONTEXTMARKUP

AnnotationTextMarkup::AnnotationTextMarkup interface	Represents the base class for text markup annotations.

3.3.44 Bookmark

Bookmark interface

The Bookmark interface allows for the extraction of the hierarchical navigation data for a document. Bookmarks are either extracted directly from the document, or generated from the heading style information

To obtain this interface, call the [Extractor.GetRootBookmark](#) method.

Bookmark::Action property	The Action property returns the behavior when a user clicks the bookmark.
Bookmark::AddChild method	The AddMethod adds a new bookmark as a child of this bookmark.
Bookmark::Color property	Returns the color that the bookmark should be shown.
Bookmark::Destination property	The Destination property returns a Unicode string for the destination of this bookmark. The destination value is based on the Action type of the bookmark.
Bookmark::Fit property	The Fit property returns the zoom/fit level that should be applied when navigating to the destination.
Bookmark::GetFirstChild property	The GetFirstChild method returns the first child bookmark of the current node, or NULL if the node contains no children.
Bookmark::GetNextSibling property	The GetNextSibling method returns the next peer bookmark of the current node, or NULL if the node is the last in it's chain.
Bookmark::Height property	The Height property return the dimensions on the page where the user should be zoomed into. The dimension information is based on the DPI used when loading the page.
Bookmark::Level property	The Level property returns the depth in the hierarchy of the current bookmark.
Bookmark::PageIndex property	The PageIndex property returns the index of the page to navigate to when the Action is GoTo, otherwise it returns -1.
Bookmark::Rect property	Returns the destination rectangle of the bookmark.
Bookmark::TextStyle property	Returns any text styling that should be applied when showing the bookmark title; see Text Styles for details.
Bookmark::Title property	The Title property returns a Unicode string for the title of this bookmark.
Bookmark::Width property	

	The Width property return the dimensions on the page where the user should be zoomed into. The dimension information is based on the DPI used hen loading the page.
Bookmark::X property	The X property return the position on the page where the user should be scrolled to. The position information is based on the DPI used when loading the page.
Bookmark::Y property	The Y property return the position on the page where the user should be scrolled to. The position information is based on the DPI used when loading the page.
Bookmark::Zoom property	The Zoom property returns the zoom/fit level that should be applied when navigating to the destination.

Bookmark::Action property

The Action property returns the behavior when a user clicks the bookmark.

.NET

```
int Action { get; set; }
```

JAVA

```
int getAction() throws IGRException;
void setAction(int value) throws IGRException;
```

PYTHON

```
@property
def Action(self) -> int

@Action.setter
def Action(self, value: int)
```

C++17

```
int getAction() const;
void setAction(int value);
```

RETURN VALUE

`int` : Integer value containing the Action enumerated value, one of:

Name	Value
IGR_BOOKMARK_ACTION_GOTO	0x00
IGR_BOOKMARK_ACTION_LINK	0x01

SEE ALSO

- [Bookmark interface](#)

Bookmark::AddChild method

The AddMethod adds a new bookmark as a child of this bookmark.

PROTOTYPE

.NET

```
void AddChild(Bookmark bookmark)
```

JAVA

```
void AddChild(Bookmark bookmark) throws IGRException;
```

PYTHON

```
def AddChild(self, bookmark: Bookmark) -> void
```

C++17

```
void AddChild(Bookmark bookmark)
```

PARAMETERS

bookmark : [Bookmark](#) : The new bookmark to add

SEE ALSO

- [Bookmark interface](#)

Bookmark::Color property

Returns the color that the bookmark should be shown.

.NET

```
System.Drawing.Color Color { get; set; }
```

JAVA

```
int getColor() throws IGRException;  
void setColor(int value) throws IGRException;
```

PYTHON

```
@property  
def Color(self) -> int  
  
@Color.setter  
def Color(self, value: int)
```

C++17

```
Hyland.DocFilters.Color getColor() const;  
void setColor(Hyland.DocFilters.Color value);
```

RETURN VALUE

`int` : Returns the color that the bookmark should be shown.

SEE ALSO

- [Bookmark interface](#)

Bookmark::Destination property

The Destination property returns a Unicode string for the destination of this bookmark. The destination value is based on the Action type of the bookmark.

.NET

```
string Destination { get; set; }
```

JAVA

```
string getDestination() throws IGREException;  
void setDestination(string value) throws IGREException;
```

PYTHON

```
@property  
def Destination(self) -> string  
  
@Destination.setter  
def Destination(self, value: string)
```

C++17

```
std::wstring getDestination() const;  
void setDestination(const std::wstring& value);
```

RETURN VALUE

`string` : Unicode string containing the destination of the bookmark.

SEE ALSO

- [Bookmark interface](#)

Bookmark::Fit property

The Fit property returns the zoom/fit level that should be applied when navigating to the destination.

.NET

```
int Fit { get; set; }
```

JAVA

```
int getFit() throws IGRException;
void setFit(int value) throws IGRException;
```

PYTHON

```
@property
def Fit(self) -> int

@Fit.setter
def Fit(self, value: int)
```

C++17

```
int getFit() const;
void setFit(int value);
```

RETURN VALUE

`int` : Integer value containing the Fit enumerated value, one of:

Name	Value
IGR_BOOKMARK_FIT_NONE	0x0
IGR_BOOKMARK_FIT_BOTH	0x1
IGR_BOOKMARK_FIT_WIDTH	0x2
IGR_BOOKMARK_FIT_HEIGHT	0x3
IGR_BOOKMARK_FIT_RECT	0x4

SEE ALSO

- [Bookmark interface](#)

Bookmark::GetFirstChild property

The GetFirstChild method returns the first child bookmark of the current node, or NULL if the node contains no children.

PROTOTYPE

.NET

```
Bookmark GetFirstChild()
```

JAVA

```
Bookmark GetFirstChild() throws IGREException;
```

PYTHON

```
def GetFirstChild(self) -> Bookmark
```

C++17

```
Bookmark GetFirstChild()
```

RETURN VALUE

[Bookmark](#) : The GetFirstChild method returns the first child bookmark of the current node, or NULL if the node contains no children.

SEE ALSO

- [Bookmark interface](#)

Bookmark::GetNextSibling property

The GetNextSibling method returns the next peer bookmark of the current node, or NULL if the node is the last in it's chain.

PROTOTYPE

.NET

```
Bookmark GetNextSibling()
```

JAVA

```
Bookmark GetNextSibling() throws IGREException;
```

PYTHON

```
def GetNextSibling(self) -> Bookmark
```

C++17

```
Bookmark GetNextSibling()
```

RETURN VALUE

Bookmark : The GetNextSibling method returns the next peer bookmark of the current node, or NULL if the node is the last in it's chain.

SEE ALSO

- [Bookmark interface](#)

Bookmark::Height property

The Height property return the dimensions on the page where the user should be zoomed into. The dimension information is based on the DPI used when loading the page.

.NET

```
int Height { get; set; }
```

JAVA

```
int getHeight() throws IGRException;  
void setHeight(int value) throws IGRException;
```

PYTHON

```
@property  
def Height(self) -> int  
  
@Height.setter  
def Height(self, value: int)
```

C++17

```
int getHeight() const;  
void setHeight(int value);
```

RETURN VALUE

`int` : Integer containing the dimension in pixels.

SEE ALSO

- [Bookmark interface](#)

Bookmark::Level property

The Level property returns the depth in the hierarchy of the current bookmark.

.NET

```
int Level { get; set; }
```

JAVA

```
int getLevel() throws IGRException;  
void setLevel(int value) throws IGRException;
```

PYTHON

```
@property  
def Level(self) -> int  
  
@Level.setter  
def Level(self, value: int)
```

C++17

```
int getLevel() const;  
void setLevel(int value);
```

RETURN VALUE

`int` : The Level property returns the depth in the hierarchy of the current bookmark.

SEE ALSO

- [Bookmark interface](#)

Bookmark::PageIndex property

The PageIndex property returns the index of the page to navigate to when the Action is GoTo, otherwise it returns -1.

.NET

```
int PageIndex { get; set; }
```

JAVA

```
int getPageIndex() throws IGRException;  
void setPageIndex(int value) throws IGRException;
```

PYTHON

```
@property  
def PageIndex(self) -> int  
  
@PageIndex.setter  
def PageIndex(self, value: int)
```

C++17

```
int getPageIndex() const;  
void setPageIndex(int value);
```

RETURN VALUE

`int` : The PageIndex property returns the index of the page to navigate to when the Action is GoTo, otherwise it returns -1.

SEE ALSO

- [Bookmark interface](#)

Bookmark::Rect property

Returns the destination rectangle of the bookmark.

.NET

```
TextStyleFlags Rect { get; set; }
```

C++17

```
System.Drawing.Rect getRect() const;  
void setRect(System.Drawing.Rect value);
```

RETURN VALUE

`System.Drawing.Rect` : The destination rectangle.

SEE ALSO

- [Bookmark interface](#)

Bookmark::TextStyle property

Returns any text styling that should be applied when showing the bookmark title; see [Text Styles](#) for details.

.NET

```
TextStyleFlags TextStyle { get; set; }
```

JAVA

```
int getTextStyle() throws IGRException;  
void setTextStyle(int value) throws IGRException;
```

PYTHON

```
@property  
def TextStyle(self) -> int  
  
@TextStyle.setter  
def TextStyle(self, value: int)
```

C++17

```
int getTextStyle() const;  
void setTextStyle(int value);
```

RETURN VALUE

`int` : See [Text Styles](#) for details

SEE ALSO

- [Bookmark interface](#)

Bookmark::Title property

The Title property returns a Unicode string for the title of this bookmark.

.NET

```
string Title { get; set; }
```

JAVA

```
string getTitle() throws IGRException;  
void setTitle(string value) throws IGRException;
```

PYTHON

```
@property  
def Title(self) -> string  
  
@Title.setter  
def Title(self, value: string)
```

C++17

```
std::wstring getTitle() const;  
void setTitle(const std::wstring& value);
```

RETURN VALUE

`string` : Unicode string containing the title of the bookmark.

SEE ALSO

- [Bookmark interface](#)

Bookmark::Width property

The Width property return the dimensions on the page where the user should be zoomed into. The dimension information is based on the DPI used hen loading the page.

.NET

```
int Width { get; set; }
```

JAVA

```
int getWidth() throws IGRException;  
void setWidth(int value) throws IGRException;
```

PYTHON

```
@property  
def Width(self) -> int  
  
@Width.setter  
def Width(self, value: int)
```

C++17

```
int getWidth() const;  
void setWidth(int value);
```

RETURN VALUE

`int` : Integer containing the dimension in pixels.

SEE ALSO

- [Bookmark interface](#)

Bookmark::X property

The X property return the position on the page where the user should be scrolled to. The position information is based on the DPI used when loading the page.

.NET

```
int X { get; set; }
```

JAVA

```
int getX() throws IGRException;  
void setX(int value) throws IGRException;
```

PYTHON

```
@property  
def X(self) -> int  
  
@X.setter  
def X(self, value: int)
```

C++17

```
int getX() const;  
void setX(int value);
```

RETURN VALUE

`int` : Integer containing the coordinate in pixels.

SEE ALSO

- [Bookmark interface](#)

Bookmark::Y property

The Y property return the position on the page where the user should be scrolled to. The position information is based on the DPI used when loading the page.

.NET

```
int Y { get; set; }
```

JAVA

```
int getY() throws IGRException;  
void setY(int value) throws IGRException;
```

PYTHON

```
@property  
def Y(self) -> int  
  
@Y.setter  
def Y(self, value: int)
```

C++17

```
int getY() const;  
void setY(int value);
```

RETURN VALUE

`int` : Integer containing the coordinate in pixels.

SEE ALSO

- [Bookmark interface](#)

Bookmark::Zoom property

The Zoom property returns the zoom/fit level that should be applied when navigating to the destination.

.NET

```
double Zoom { get; set; }
```

JAVA

```
double getZoom() throws IGRException;  
void setZoom(double value) throws IGRException;
```

PYTHON

```
@property  
def Zoom(self) -> double  
  
@Zoom.setter  
def Zoom(self, value: double)
```

C++17

```
double getZoom() const;  
void setZoom(double value);
```

RETURN VALUE

`double` : The Zoom property returns the zoom/fit level that should be applied when navigating to the destination.

SEE ALSO

- [Bookmark interface](#)

3.3.45 Canvas

Canvas interface

The Canvas interface allows rendering of pages to a variety of output devices, including HD HTML, PNG, and PDF.

The Canvas object also allows post-processing / image manipulation of output such as annotations, redaction, bates stamping, or general drawing.

To obtain this interface, call the `DocumentFilters.MakeOutputCanvas` or `DocumentFilters.MakeOutputCanvasOnStream` methods.

Note The drawing API is available for bitmap and PDF outputs only.

Canvas::AddCustomMetadata method	Add a custom metadata field to the output canvas. This function MUST be called before the first call to <code>RenderPage</code> .
Canvas::Annotate method	The <code>Annotate</code> method draws the given annotation onto the current page. The annotation is delivered as a string of JSON text. The schema for Annotations can be found in the Document Filters installation directory in “Annotation-Schema.json”.
Canvas::AppendBookmark method	Appends only this current bookmark. This function does not enumerate the Children bookmarks.
Canvas::AppendBookmarkRecursive method	Appends this bookmark and the and it's children recursively.
Canvas::Arc method	The <code>Arc</code> method draws an arc on the image along the perimeter of the ellipse, bounded by the specified rectangle. It uses the current Pen.
Canvas::BlankPage method	Create a blank page on the canvas with the specified properties.
Canvas::Chord method	The <code>Chord</code> method draws a closed figure represented by the intersection of a line and an ellipse. The ellipse is bisected by a line that runs between X3, Y3 and X4, Y4 coordinates.
Canvas::ClearBookmarks method	Remove existing bookmarks from the canvas.
Canvas::Close method	Close the canvas object, disposing of any resources.
Canvas::DrawImage method	<code>DrawImage</code> renders an image from a buffer onto the Canvas.

Canvas::DrawScaleImage method	DrawScaleImage renders an image from a buffer onto the Canvas. The image is scaled to a specified size.
Canvas::Ellipse method	The Ellipse method draws the ellipse defined by a bounding rectangle on the canvas, outlined with the current pen and filled with the current brush.
Canvas::Handle property	Returns the internal HCANVAS handle for the canvas object.
Canvas::LineTo method	LineTo draws a line on the canvas from current pen position to the point specified by X and Y, and sets the pen position to (X, Y) coordinates.
Canvas::MoveTo method	MoveTo changes the current drawing position to the point (X, Y).
Canvas::Pie method	The Pie method draws a pie-shaped section of the ellipse on the canvas, bounded by the rectangle (X, Y) and (X2, Y2).
Canvas::Rect method	The Rect method draws a rectangle using the Brush and Pen of the canvas to fill and draw the border.
Canvas::RenderPage method	Render a page to the canvas using default options.
Canvas::RenderPages method	Render all pages of the extractor to the canvas.
Canvas::RoundRect method	RoundRect draws a rectangle with rounded corners on the canvas, outlined with the current pen and filled with the current brush.
Canvas::SetBrush method	SetBrush updates the current brush on the canvas with the given color and style. Brushes are used when drawing rectangles and text.
Canvas::SetFont method	SetFont specifies the font to be used when drawing text to the canvas. All subsequent calls to TextOut and MeasureText will use this font.
Canvas::SetOpacity method	SetOpacity sets the opacity or transparency for future drawing routines.
Canvas::SetPen method	SetPen updates the canvas pen with the specific color, width, and style.
Canvas::TextOut method	TextOut writes a string on the canvas, starting at (X, Y). It updates the pen position to the end of the string and uses the current font and brush.

Canvas::TextRect method	Writes a string inside a clipping rectangle, using the current brush and font.
Canvas::TextWidth method	Returns the width in pixels, of a string if rendered with the current font.
Canvas::TextHeight method	Returns the height in pixels, of a string if rendered with the current font.

Canvas::AddCustomMetadata method

Add a custom metadata field to the output canvas. This function MUST be called before the first call to RenderPage.

PROTOTYPE

.NET

```
void AddCustomMetadata(string name, string value)
```

JAVA

```
void AddCustomMetadata(string name, string value) throws IGREException;
```

PYTHON

```
def AddCustomMetadata(self, name: string, value: string) -> void
```

C++17

```
void AddCustomMetadata(const std::wstring& name, const std::wstring& value)
```

PARAMETERS

`name: string` : The value of the metadata field.

`value: string` : The name of the metadata field.

Canvas::Annotate method

The Annotate method draws the given annotation onto the current page. The annotation is delivered as a string of JSON text. The schema for Annotations can be found in the Document Filters installation directory in “Annotation-Schema.json”.

OVERLOADS

<code>Annotate(string)</code>	Add an annotation from a JSON.
<code>Annotate(Annotation)</code>	Add an annotation from an object.

ANNOTATE(STRING)

Prototype

.NET

```
void Annotate(string annotation)
```

JAVA

```
void Annotate(string annotation) throws IGREException;
```

PYTHON

```
def Annotate(self, annotation: string) -> void
```

C++17

```
void Annotate(const std::wstring& annotation)
```

Parameters

`annotation: string` : JSON representation of an annotation.

ANNOTATE(ANNOTATION)

Prototype

.NET

```
void Annotate(Annotation annotation)
```

JAVA

```
void Annotate(Annotation annotation) throws IGREException;
```

PYTHON

```
def Annotate(self, annotation: Annotation) -> void
```

C++17

```
void Annotate(Annotation annotation)
```

Parameters

annotation : **Annotation** : Object representation of an annotation.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var canvas = api.MakeOutputCanvas("barcode.png", CanvasType.PNG);
7 canvas.BlankPage(400, 400);
8 canvas.Annotate(new Hyland.DocumentFilters.Annotations.QrCode {
9     Caption = "Hello World",
10    Text = "http://www.hyland.com",
11    Rect = System.Drawing.Rectangle.FromLTRB(20, 20, 380, 380)
12 });
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class GenerateQRCode {
4     public static void main(String[] args) {
5         try {
6             // Initialize the API
7             Api api = new Api();
8             api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10            // Create an output canvas
11            try (Canvas canvas = api.MakeOutputCanvas("barcode.png", CanvasType.PNG.swigValue())) {
12                // Create a blank page
13                canvas.BlankPage(400, 400);
14
15                // Create and configure the QR code annotation
16                AnnotationQrCode qrCode = new AnnotationQrCode();
17                qrCode.setCaption("Hello World");
18                qrCode.setContent("http://www.hyland.com");
19                qrCode.setRect(new AnnotationRect(20, 20, 380, 380));
20
21                // Annotate the canvas
22                canvas.Annotate(qrCode);
23            }
24        } catch (IGRException e) {
25            System.err.println("Error: " + e.getMessage());
26        }
27    }
28 }
```

PYTHON

```

1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.MakeOutputCanvas("barcode.png", IGR_DEVICE_IMAGE_PNG) as canvas:
7     canvas.BlankPage(400, 400)
8
9     anno = AnnotationQrCode()
10    anno.Caption = "Hello World"
11    anno.Content = "http://www.hyland.com"
12    anno.Margin = 8
13    anno.ErrorCorrectionLevel = 3
14    anno.Rect = AnnotationRect.FromLTRB(20, 20, 380, 380)
15    canvas.Annotate(anno)

```

C++17

```

1 #include "DocumentFiltersObjects.h"
2
3 int main() {
4     try {
5         // Create and initialize the API object
6         Hyland::DocFilters::Api api;
7         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
8
9         // Create the output canvas
10        Hyland::DocFilters::Canvas canvas = api.MakeOutputCanvas("barcode.png",
Hyland::DocFilters::CanvasType::PNG);
11
12        // Set up a blank page
13        canvas.BlankPage(400, 400);
14
15        // Create and configure the QR code annotation
16        Hyland::DocFilters::AnnotationQrCode qrCode;
17        qrCode.setCaption(L"Hello World");
18        qrCode.setText(L"http://www.hyland.com");
19        qrCode.setRect(Hyland::DocFilters::RectI32::LTRB(20, 20, 380, 380));
20
21        // Add the QR code to the canvas
22        canvas.Annotate(qrCode);
23
24        // Close the canvas (ensures the image is saved)
25        canvas.Close();
26
27    } catch (const std::exception& e) {
28        std::cerr << "Error: " << e.what() << std::endl;
29        return 1; // Indicate an error
30    }
31
32    return 0; // Successful execution
33 }

```

Canvas::AppendBookmark method

Appends only this current bookmark.

This function does not enumerate the Children bookmarks.

PROTOTYPE

.NET

```
void AppendBookmark(Bookmark bookmark)
```

JAVA

```
void AppendBookmark(Bookmark bookmark) throws IGREException;
```

PYTHON

```
def AppendBookmark(self, bookmark: Bookmark) -> void
```

C++17

```
void AppendBookmark(Bookmark bookmark)
```

PARAMETERS

bookmark : **Bookmark** : The bookmark to add to the output canvas.

Canvas::AppendBookmarkRecursive method

Appends this bookmark and the and it's children recursively.

PROTOTYPE

.NET

```
void AppendBookmarkRecursive(Bookmark bookmark)
```

JAVA

```
void AppendBookmarkRecursive(Bookmark bookmark) throws IGRException;
```

PYTHON

```
def AppendBookmarkRecursive(self, bookmark: Bookmark) -> void
```

C++17

```
void AppendBookmarkRecursive(Bookmark bookmark)
```

PARAMETERS

`bookmark` : [Bookmark](#) : The bookmark to add to the output canvas.

Canvas::Arc method

The Arc method draws an arc on the image along the perimeter of the ellipse, bounded by the specified rectangle. It uses the current Pen.

PROTOTYPE

.NET

```
void Arc(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4)
```

JAVA

```
void Arc(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4) throws IGRException;
```

PYTHON

```
def Arc(self, x: int, y: int, x2: int, y2: int, x3: int, y3: int, x4: int, y4: int) -> void
```

C++17

```
void Arc(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4)
```

C++11

```
void Arc(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4)
```

COM

```
HRESULT Arc([in] int x, [in] int y, [in] int x2, [in] int y2, [in] int x3, [in] int y3, [in] int x4, [in] int y4)
```

PARAMETERS

x: int : Left-most coordinate of the bounding box.

y: int : Top-most coordinate of the bounding box.

x2: int : Right-most coordinate of the bounding box.

y2: int : Bottom-most coordinate of the bounding box.

x3: int : X coordinate of the start point.

y3: int : Y coordinate of the start point.

x4: int : X coordinate of the end point.

`y4: int` : Y coordinate of the end point.

Canvas::BlankPage method

Create a blank page on the canvas with the specified properties.

PROTOTYPE

.NET

```
void BlankPage(int width, int height, string options)
```

JAVA

```
void BlankPage(int width, int height, String options) throws IGRException;
```

PYTHON

```
def BlankPage(self, width: int, height: int, options: string) -> void
```

C++17

```
void BlankPage(int width, int height, const std::wstring& options)
```

C++11

```
void BlankPage(int width, int height, std::wstring options)
```

COM

```
HRESULT BlankPage([in] int width, [in] int height, [in] BSTR options)
```

PARAMETERS

width: int : The width of the new page, in pixels.

height: int : The height of the new page, in pixels.

options: string : Canvas processing options to use when creating the new page.

SAMPLE CODE

.NET

```

1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var canvas = api.MakeOutputCanvas("barcode.png", CanvasType.PNG);
7 canvas.BlankPage(400, 400);
8 canvas.Annotate(new Hyland.DocumentFilters.Annotations.QrCode {
9     Caption = "Hello World",
10    Text = "http://www.hyland.com",
11    Rect = System.Drawing.Rectangle.FromLTRB(20, 20, 380, 380)
12 });

```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class GenerateQRCode {
4     public static void main(String[] args) {
5         try {
6             // Initialize the API
7             Api api = new Api();
8             api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10            // Create an output canvas
11            try (Canvas canvas = api.MakeOutputCanvas("barcode.png", CanvasType.PNG.swigValue())) {
12                // Create a blank page
13                canvas.BlankPage(400, 400);
14
15                // Create and configure the QR code annotation
16                AnnotationQrCode qrCode = new AnnotationQrCode();
17                qrCode.setCaption("Hello World");
18                qrCode.setContent("http://www.hyland.com");
19                qrCode.setRect(new AnnotationRect(20, 20, 380, 380));
20
21                // Annotate the canvas
22                canvas.Annotate(qrCode);
23            }
24        } catch (IGRException e) {
25            System.err.println("Error: " + e.getMessage());
26        }
27    }
28 }

```

PYTHON

```

1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.MakeOutputCanvas("barcode.png", IGR_DEVICE_IMAGE_PNG) as canvas:
7     canvas.BlankPage(400, 400)
8
9     anno = AnnotationQrCode()
10    anno.Caption = "Hello World"
11    anno.Content = "http://www.hyland.com"
12    anno.Margin = 8
13    anno.ErrorCorrectionLevel = 3
14    anno.Rect = AnnotationRect.FromLTRB(20, 20, 380, 380)
15    canvas.Annotate(anno)

```

C++17

```

1 #include "DocumentFiltersObjects.h"
2
3 int main() {
4     try {
5         // Create and initialize the API object
6         Hyland::DocFilters::Api api;
7         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
8
9         // Create the output canvas
10        Hyland::DocFilters::Canvas canvas = api.MakeOutputCanvas("barcode.png",
Hyland::DocFilters::CanvasType::PNG);
11
12        // Set up a blank page
13        canvas.BlankPage(400, 400);
14
15        // Create and configure the QR code annotation
16        Hyland::DocFilters::AnnotationQrCode qrCode;
17        qrCode.setCaption(L"Hello World");
18        qrCode.setText(L"http://www.hyland.com");
19        qrCode.setRect(Hyland::DocFilters::RectI32::LTRB(20, 20, 380, 380));
20
21        // Add the QR code to the canvas
22        canvas.Annotate(qrCode);
23
24        // Close the canvas (ensures the image is saved)
25        canvas.Close();
26
27    } catch (const std::exception& e) {
28        std::cerr << "Error: " << e.what() << std::endl;
29        return 1; // Indicate an error
30    }
31
32    return 0; // Successful execution
33 }

```

Canvas::Chord method

The Chord method draws a closed figure represented by the intersection of a line and an ellipse. The ellipse is bisected by a line that runs between X3, Y3 and X4, Y4 coordinates.

PROTOTYPE

.NET

```
void Chord(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4)
```

JAVA

```
void Chord(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4) throws IGRException;
```

PYTHON

```
def Chord(self, x: int, y: int, x2: int, y2: int, x3: int, y3: int, x4: int, y4: int) -> void
```

C++17

```
void Chord(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4)
```

C++11

```
void Chord(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4)
```

COM

```
HRESULT Chord([in] int x, [in] int y, [in] int x2, [in] int y2, [in] int x3, [in] int y3, [in] int x4, [in] int y4)
```

PARAMETERS

x: int : Left-most coordinate of the bounding box.

y: int : Top-most coordinate of the bounding box.

x2: int : Right-most coordinate of the bounding box.

y2: int : Bottom-most coordinate of the bounding box.

x3: int : X coordinate of the start point.

y3: int : Y coordinate of the start point.

x4: int : X coordinate of the end point.

`y4: int` : Y coordinate of the end point.

Canvas::ClearBookmarks method

Remove existing bookmarks from the canvas.

PROTOTYPE

.NET

```
void ClearBookmarks()
```

JAVA

```
void ClearBookmarks() throws IGRException;
```

PYTHON

```
def ClearBookmarks(self) -> void
```

C++17

```
void ClearBookmarks()
```

C++11

```
void ClearBookmarks()
```

COM

```
HRESULT ClearBookmarks()
```

Canvas::Close method

Close the canvas object, disposing of any resources.

PROTOTYPE

.NET

```
void Close()
```

JAVA

```
void Close() throws IGRException;
```

PYTHON

```
def Close(self) -> void
```

C++17

```
void Close()
```

C++11

```
void Close()
```

COM

```
HRESULT Close()
```

ADDITIONAL INFORMATION

This method should be called when finished working on the canvas to release its resources. The method will be internally called when the instance itself is released. Calling this method on closed canvases has no effect.

Resources can also be managed with the `using IDisposable` pattern in C#, or the `try-with-resources` pattern in Java.

Canvas::DrawImage method

DrawImage renders an image from a buffer onto the Canvas.

OVERLOADS

`DrawImage(int, int, byte[], string)`

DrawImage renders an image from a buffer onto the Canvas.

DRAWIMAGE(INT, INT, BYTE[], STRING)

Prototype

.NET

```
void DrawImage(int x, int y, byte[] imagedata, string mimetype)
```

JAVA

```
void DrawImage(int x, int y, byte[] imagedata, string mimetype) throws IGRException;
```

PYTHON

```
def DrawImage(self, x: int, y: int, imagedata: byte[], mimetype: string) -> void
```

C++17

```
void DrawImage(int x, int y, const void* imagedata, size_t imagedataSize, const std::wstring& mimetype)
```

C++11

```
void DrawImage(int x, int y, byte[] imagedata, std::wstring mimetype)
```

COM

```
HRESULT DrawImage([in] int x, [in] int y, [in] byte[]* imagedata, [in] BSTR mimetype)
```

Parameters

`x: int` : Left-most coordinate of the image bounding box.

`y: int` : Top-most coordinate of the image bounding box.

`imagedata: byte[]` : Binary data of the image.

`mimetype: string` : Describes the format of the image data.

Canvas::DrawScaleImage method

DrawScaleImage renders an image from a buffer onto the Canvas. The image is scaled to a specified size.

PROTOTYPE

.NET

```
void DrawScaleImage(int x, int y, int width, int height, byte[] imagedata, string mimetype)
```

JAVA

```
void DrawScaleImage(int x, int y, int width, int height, byte[] imagedata, String mimetype) throws  
IGRException;
```

PYTHON

```
def DrawScaleImage(self, x: int, y: int, width: int, height: int, imagedata: byte[], mimetype: string) ->  
void
```

C++17

```
void DrawScaleImage(int x, int y, int width, int height, const void* imagedata, size_t imagedataSize, const  
std::wstring& mimetype)
```

C++11

```
void DrawScaleImage(int x, int y, int width, int height, byte[] imagedata, std::wstring mimetype)
```

COM

```
HRESULT DrawScaleImage([in] int x, [in] int y, [in] int width, [in] int height, [in] byte[]* imagedata, [in]  
BSTR mimetype)
```

PARAMETERS

x: int : Left-most coordinate of the image bounding box.

y: int : Top-most coordinate of the image bounding box.

width: int : Desired width of the rendered image in pixels.

height: int : Desired height of the rendered image in pixels.

imagedata: byte[] : Binary data of the image.

mimetype: string : Describes the format of the image data.

Canvas::Ellipse method

The Ellipse method draws the ellipse defined by a bounding rectangle on the canvas, outlined with the current pen and filled with the current brush.

PROTOTYPE

.NET

```
void Ellipse(int x, int y, int x2, int y2)
```

JAVA

```
void Ellipse(int x, int y, int x2, int y2) throws IGRException;
```

PYTHON

```
def Ellipse(self, x: int, y: int, x2: int, y2: int) -> void
```

C++17

```
void Ellipse(int x, int y, int x2, int y2)
```

C++11

```
void Ellipse(int x, int y, int x2, int y2)
```

COM

```
HRESULT Ellipse([in] int x, [in] int y, [in] int x2, [in] int y2)
```

PARAMETERS

x: int : Left-most coordinate of the bounding box.

y: int : Top-most coordinate of the bounding box.

x2: int : Right-most coordinate of the bounding box.

y2: int : Bottom-most coordinate of the bounding box.

Canvas::Handle property

Returns the internal HCANVAS handle for the canvas object.

.NET

```
int Handle { get; }
```

JAVA

```
int getHandle() throws IGRException;
```

PYTHON

```
@property  
def Handle(self) -> int
```

C++17

```
int getHandle() const;
```

COM

```
[propget] HRESULT Handle([out, retval] int *result);
```

RETURN VALUE

`int` : The HCANVAS handle that can be used by the C API.

Canvas::LineTo method

LineTo draws a line on the canvas from current pen position to the point specified by X and Y, and sets the pen position to (X, Y) coordinates.

PROTOTYPE

.NET

```
void LineTo(int x, int y)
```

JAVA

```
void LineTo(int x, int y) throws IGRException;
```

PYTHON

```
def LineTo(self, x: int, y: int) -> void
```

C++17

```
void LineTo(int x, int y)
```

C++11

```
void LineTo(int x, int y)
```

COM

```
HRESULT LineTo([in] int x, [in] int y)
```

PARAMETERS

x: int : The X coordinate for the new pen position.

y: int : The Y coordinate for the new pen position.

Canvas::MoveTo method

MoveTo changes the current drawing position to the point (X, Y).

PROTOTYPE

.NET

```
void MoveTo(int x, int y)
```

JAVA

```
void MoveTo(int x, int y) throws IGRException;
```

PYTHON

```
def MoveTo(self, x: int, y: int) -> void
```

C++17

```
void MoveTo(int x, int y)
```

C++11

```
void MoveTo(int x, int y)
```

COM

```
HRESULT MoveTo([in] int x, [in] int y)
```

PARAMETERS

x: int : The X coordinate for the new pen position.

y: int : The Y coordinate for the new pen position.

Canvas::Pie method

The Pie method draws a pie-shaped section of the ellipse on the canvas, bounded by the rectangle (X, Y) and (X2, Y2).

PROTOTYPE

.NET

```
void Pie(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4)
```

JAVA

```
void Pie(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4) throws IGRException;
```

PYTHON

```
def Pie(self, x: int, y: int, x2: int, y2: int, x3: int, y3: int, x4: int, y4: int) -> void
```

C++17

```
void Pie(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4)
```

C++11

```
void Pie(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4)
```

COM

```
HRESULT Pie([in] int x, [in] int y, [in] int x2, [in] int y2, [in] int x3, [in] int y3, [in] int x4, [in] int y4)
```

PARAMETERS

x: int : Left-most coordinate of the bounding box.

y: int : Top-most coordinate of the bounding box.

x2: int : Right-most coordinate of the bounding box.

y2: int : Bottom-most coordinate of the bounding box.

x3: int : X coordinate of the start point.

y3: int : Y coordinate of the start point.

x4: int : X coordinate of the end point.

y4: int : Y coordinate of the end point.

Canvas::Rect method

The Rect method draws a rectangle using the Brush and Pen of the canvas to fill and draw the border.

OVERLOADS

<code>Rect(int, int, int, int)</code>	The Rect method draws a rectangle using the Brush and Pen of the canvas to fill and draw the border.
<code>Rect(System.Drawing.Rectangle)</code>	The Rect method draws a rectangle using the Brush and Pen of the canvas to fill and draw the border.

RECT(INT, INT, INT, INT)

Prototype

.NET

```
void Rect(int left, int top, int right, int bottom)
```

JAVA

```
void Rect(int left, int top, int right, int bottom) throws IGRException;
```

PYTHON

```
def Rect(self, left: int, top: int, right: int, bottom: int) -> void
```

C++17

```
void Rect(int left, int top, int right, int bottom)
```

C++11

```
void Rect(int left, int top, int right, int bottom)
```

COM

```
HRESULT Rect([in] int left, [in] int top, [in] int right, [in] int bottom)
```

Parameters

- `left: int` : Left-most coordinate of the bounding box.
- `top: int` : Top-most coordinate of the bounding box.
- `right: int` : Right-most coordinate of the bounding box.

bottom: int : Bottom-most coordinate of the bounding box.

RECT(SYSTEM.DRAWING.RECTANGLE)

Prototype

.NET

```
void Rect(System.Drawing.Rectangle rect)
```

C++17

```
void Rect(Hyland.DocFilters.RectI32 rect)
```

Parameters

rect: System.Drawing.Rectangle : The rectangle to draw.

Canvas::RenderPage method

Render a page to the canvas using default options.

OVERLOADS

RenderPage(Page)	Render a page to the canvas using default options.
RenderPage(Page, string)	Render a page to the canvas using the specified options.
RenderPage(Page, string, RenderPageProperties)	Render the page to the canvas with the specified properties.

RENDERPAGE(PAGE)

Prototype

.NET

```
void RenderPage(Page page)
```

JAVA

```
void RenderPage(Page page) throws IGREException;
```

PYTHON

```
def RenderPage(self, page: Page) -> void
```

C++17

```
void RenderPage(Page page)
```

C++11

```
void RenderPage(Page page)
```

COM

```
HRESULT RenderPage([in] Page* page)
```

Parameters

page : Page : The page to be rendered

RENDERPAGE(PAGE, STRING)**Prototype****.NET**

```
void RenderPage(Page page, string options)
```

JAVA

```
void RenderPage(Page page, string options) throws IGRException;
```

PYTHON

```
def RenderPage(self, page: Page, options: string) -> void
```

C++17

```
void RenderPage(Page page, const std::wstring& options)
```

C++11

```
void RenderPage(Page page, std::wstring options)
```

COM

```
HRESULT RenderPage([in] Page* page, [in] BSTR options)
```

Parameters

page: Page : The page to be rendered

options: string : Canvas processing options to use when creating the new page.

RENDERPAGE(PAGE, STRING, RENDERPAGEPROPERTIES)**Prototype****.NET**

```
void RenderPage(Page page, string options, RenderPageProperties renderPageProperties)
```

JAVA

```
void RenderPage(Page page, string options, RenderPageProperties renderPageProperties) throws IGRException;
```

PYTHON

```
def RenderPage(self, page: Page, options: string, renderPageProperties: RenderPageProperties) -> void
```

C++17

```
void RenderPage(Page page, const std::wstring& options, RenderPageProperties renderPageProperties)
```

Parameters

page: Page : The page to render.

options: string : Canvas options to use when rendering page.

renderPageProperties: RenderPageProperties : Render properties to use when rendering the page.

SAMPLE CODE

.NET

```
1 using System.Linq;
2 using Hyland.DocumentFilters;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
8
9 foreach (var pageIter in doc.Pages.Select((page, index) => new { page, index }))
10 {
11     using (pageIter.page)
12     {
13         using var canvas = api.MakeOutputCanvas($"page-{pageIter.index+1}.png", CanvasType.PNG);
14
15         var renderPageProperties = new RenderPageProperties();
16         renderPageProperties.AddFormValue("FormKey", "NewFormValue", true);
17
18         canvas.RenderPage(pageIter.page, "", renderPageProperties);
19     }
20 }
```

C++17

```
1 #include <DocumentFiltersObjects.h>
2
3 int main() {
4     try {
5         // Create and initialize the API object
6         Hyland::DocFilters::Api api;
7         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
8
9         // Open the input file
10        Hyland::DocFilters::Extractor doc = api.OpenExtractor("filename.doc",
Hyland::DocFilters::OpenMode::Paginated);
11
12        // Create the output canvas
13        Hyland::DocFilters::Canvas canvas = api.MakeOutputCanvas("output.pdf",
Hyland::DocFilters::CanvasType::PDF);
14
15        Hyland::DocFilters::RenderPageProperties renderProps;
16        renderProps.AddFormValue(L"FormKey", L"NewFormValue", true);
17
18        // Render all pages to the output
19        for (auto page : doc.pages()) {
20            canvas.RenderPage(page, renderProps);
21        }
22    } catch (const std::exception& e) {
23        std::cerr << "Error: " << e.what() << std::endl;
24        return 1; // Indicate an error
25    }
26
27    return 0; // Successful execution
28 }
```

Canvas::RenderPages method

Render all pages of the extractor to the canvas.

PROTOTYPE

.NET

```
void RenderPages(Extractor extractor)
```

JAVA

```
void RenderPages(Extractor extractor) throws IGRException;
```

PYTHON

```
def RenderPages(self, extractor: Extractor) -> void
```

C++17

```
void RenderPages(Extractor extractor)
```

PARAMETERS

`extractor: Extractor` : The extractor containing the pages to render.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
7 using var canvas = api.MakeOutputCanvas("filename.pdf", CanvasType.PDF);
8
9 canvas.RenderPages(doc);
```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class hidef_pdf
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.doc"))
11        {
12            try (Canvas canvas = df.MakeOutputCanvas("filename.pdf",
13 isys_docfiltersConstants.IGR_DEVICE_IMAGE_PDF, ""))
14            {
15                doc.Open(isys_docfiltersConstants.IGR_FORMAT_IMAGE, "");
16
17                for (int i = 0, c = doc.GetPageCount(); i < c; ++i)
18                {
19                    try (Page page = doc.GetPage(i)) {
20                        canvas.RenderPage(page);
21                    }
22                }
23            }
24        }
25 }

```

PYTHON

```

1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.OpenExtractor("filename.doc", mode=IGR_FORMAT_IMAGE) as doc:
7     with api.MakeOutputCanvas("filename.pdf", canvasType=IGR_DEVICE_PDF) as canvas:
8         canvas.RenderPages(doc)

```

C++17

```
1  #include <DocumentFiltersObjects.h>
2
3  int main() {
4      try {
5          // Create and initialize the API object
6          Hyland::DocFilters::Api api;
7          api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
8
9          // Open the input file
10         Hyland::DocFilters::Extractor doc = api.OpenExtractor("filename.doc",
Hyland::DocFilters::OpenMode::Paginated);
11
12         // Create the output canvas
13         Hyland::DocFilters::Canvas canvas = api.MakeOutputCanvas("output.pdf",
Hyland::DocFilters::CanvasType::PDF);
14
15         // Render all pages to the output
16         canvas.RenderPages(doc);
17     } catch (const std::exception& e) {
18         std::cerr << "Error: " << e.what() << std::endl;
19         return 1; // Indicate an error
20     }
21
22     return 0; // Successful execution
23 }
```

Canvas::RoundRect method

RoundRect draws a rectangle with rounded corners on the canvas, outlined with the current pen and filled with the current brush.

PROTOTYPE

.NET

```
void RoundRect(int x, int y, int x2, int y2, int radius)
```

JAVA

```
void RoundRect(int x, int y, int x2, int y2, int radius) throws IGRException;
```

PYTHON

```
def RoundRect(self, x: int, y: int, x2: int, y2: int, radius: int) -> void
```

C++17

```
void RoundRect(int x, int y, int x2, int y2, int radius)
```

C++11

```
void RoundRect(int x, int y, int x2, int y2, int radius)
```

COM

```
HRESULT RoundRect([in] int x, [in] int y, [in] int x2, [in] int y2, [in] int radius)
```

PARAMETERS

x: int : Left-most coordinate of the bounding box.

y: int : Top-most coordinate of the bounding box.

x2: int : Right-most coordinate of the bounding box.

y2: int : Bottom-most coordinate of the bounding box.

radius: int : The radius to use for the rounded corner.

Canvas::SetBrush method

SetBrush updates the current brush on the canvas with the given color and style. Brushes are used when drawing rectangles and text.

OVERLOADS

<code>SetBrush(int, int)</code>	SetBrush updates the current brush on the canvas with the given color and style. Brushes are used when drawing rectangles and text.
<code>SetBrush(System.Drawing.Color, int)</code>	SetBrush updates the current brush on the canvas with the given color and style. Brushes are used when drawing rectangles and text.

SETBRUSH(INT, INT)

Prototype

.NET

```
void SetBrush(int color, int style)
```

JAVA

```
void SetBrush(int color, int style) throws IGRException;
```

PYTHON

```
def SetBrush(self, color: int, style: int) -> void
```

C++17

```
void SetBrush(int color, int style)
```

C++11

```
void SetBrush(int color, int style)
```

COM

```
HRESULT SetBrush([in] int color, [in] int style)
```

Parameters

`color: int` : The color expressed as a 32-bit integer.

`style: int` : The brush style used when drawing. See [Brush Styles](#).

SETBRUSH(SYSTEM.DRAWING.COLOR, INT)

Prototype

.NET

```
void SetBrush(System.Drawing.Color color, int style)
```

C++17

```
void SetBrush(Hyland.DocFilters.Color color, int style)
```

Parameters

`color: System.Drawing.Color` : The color expressed as a System.Drawing.Color

`style: int` : The brush style used when drawing. See [Brush Styles](#).

Canvas::SetFont method

SetFont specifies the font to be used when drawing text to the canvas. All subsequent calls to TextOut and MeasureText will use this font.

PROTOTYPE

.NET

```
void SetFont(string name, int size, int style)
```

JAVA

```
void SetFont(string name, int size, int style) throws IGRException;
```

PYTHON

```
def SetFont(self, name: string, size: int, style: int) -> void
```

C++17

```
void SetFont(const std::wstring& name, int size, int style)
```

C++11

```
void SetFont(std::wstring name, int size, int style)
```

COM

```
HRESULT SetFont([in] BSTR name, [in] int size, [in] int style)
```

PARAMETERS

name: string : Font Family name to use; this is the font display name such as 'Arial' or 'Courier New.'

size: int : The size to render the font.

style: int : A bitmask of style information. See [Font Styles](#) for details.

Canvas::SetOpacity method

SetOpacity sets the opacity or transparency for future drawing routines.

PROTOTYPE

.NET

```
void SetOpacity(int opacity)
```

JAVA

```
void SetOpacity(int opacity) throws IGRException;
```

PYTHON

```
def SetOpacity(self, opacity: int) -> void
```

C++17

```
void SetOpacity(int opacity)
```

C++11

```
void SetOpacity(int opacity)
```

COM

```
HRESULT SetOpacity([in] int opacity)
```

PARAMETERS

opacity: int : The opacity expressed from 0 to 255.

Canvas::SetPen method

SetPen updates the canvas pen with the specific color, width, and style.

OVERLOADS

<code>SetPen(int, int, int)</code>	SetPen updates the canvas pen with the specific color, width, and style.
<code>SetPen(System.Drawing.Color, int, int)</code>	SetPen updates the canvas pen with the specific color, width, and style.

SETPEN(INT, INT, INT)

Prototype

.NET

```
void SetPen(int color, int width, int style)
```

JAVA

```
void SetPen(int color, int width, int style) throws IGREException;
```

PYTHON

```
def SetPen(self, color: int, width: int, style: int) -> void
```

C++17

```
void SetPen(int color, int width, int style)
```

C++11

```
void SetPen(int color, int width, int style)
```

COM

```
HRESULT SetPen([in] int color, [in] int width, [in] int style)
```

Parameters

- `color: int` : The color expressed as a 32-bit integer.
- `width: int` : The width of the pen, expressed in points.
- `style: int` : The pen style used when drawing. See [Pen Styles](#).

SETPEN(SYSTEM.DRAWING.COLOR, INT, INT)

Prototype

.NET

```
void SetPen(System.Drawing.Color color, int width, int style)
```

C++17

```
void SetPen(Hyland.DocFilters.Color color, int width, int style)
```

Parameters

color: System.Drawing.Color : The color expressed as a System.Drawing.Color

width: int : The width of the pen, expressed in points.

style: int : The pen style used when drawing. See [Pen Styles](#).

Canvas::TextOut method

TextOut writes a string on the canvas, starting at (X, Y). It updates the pen position to the end of the string and uses the current font and brush.

PROTOTYPE

.NET

```
void TextOut(int x, int y, string text)
```

JAVA

```
void TextOut(int x, int y, string text) throws IGRException;
```

PYTHON

```
def TextOut(self, x: int, y: int, text: string) -> void
```

C++17

```
void TextOut(int x, int y, const std::wstring& text)
```

C++11

```
void TextOut(int x, int y, std::wstring text)
```

COM

```
HRESULT TextOut([in] int x, [in] int y, [in] BSTR text)
```

PARAMETERS

x: int : Left-most coordinate of the bounding box.

y: int : Top-most coordinate of the bounding box.

text: string : The text to output to the canvas.

Canvas::TextRect method

Writes a string inside a clipping rectangle, using the current brush and font.

PROTOTYPE

.NET

```
void TextRect(int x, int y, int x2, int y2, string text, int flags)
```

JAVA

```
void TextRect(int x, int y, int x2, int y2, String text, int flags) throws IGRException;
```

PYTHON

```
def TextRect(self, x: int, y: int, x2: int, y2: int, text: string, flags: int) -> void
```

C++17

```
void TextRect(int x, int y, int x2, int y2, const std::wstring& text, int flags)
```

C++11

```
void TextRect(int x, int y, int x2, int y2, std::wstring text, int flags)
```

COM

```
HRESULT TextRect([in] int x, [in] int y, [in] int x2, [in] int y2, [in] BSTR text, [in] int flags)
```

PARAMETERS

x: int : Left-most coordinate of the bounding box.

y: int : Top-most coordinate of the bounding box.

x2: int : Right-most coordinate of the bounding box.

y2: int : Bottom-most coordinate of the bounding box.

text: string : The text to output to the canvas.

flags: int : Reserved for future use.

Canvas::TextWidth method

Returns the width in pixels, of a string if rendered with the current font.

PROTOTYPE

.NET

```
int TextWidth(string text)
```

JAVA

```
int TextWidth(string text) throws IGREException;
```

PYTHON

```
def TextWidth(self, text: string) -> int
```

C++17

```
int TextWidth(const std::wstring& text)
```

C++11

```
int TextWidth(std::wstring text)
```

COM

```
HRESULT TextWidth([in] BSTR text, [out, retval] int *result)
```

PARAMETERS

text: string : A string containing the text to be measured.

RETURN VALUE

int : Integer expressing the width or height.

Canvas::TextHeight method

Returns the height in pixels, of a string if rendered with the current font.

PROTOTYPE

.NET

```
int TextHeight(string text)
```

JAVA

```
int TextHeight(string text) throws IGRException;
```

PYTHON

```
def TextHeight(self, text: string) -> int
```

C++17

```
int TextHeight(const std::wstring& text)
```

C++11

```
int TextHeight(std::wstring text)
```

COM

```
HRESULT TextHeight([in] BSTR text, [out, retval] int *result)
```

PARAMETERS

text: string : A string containing the text to be measured.

RETURN VALUE

int : Integer expressing the width or height.

3.3.46 CompareDocumentSettings

CompareDocumentSettings interface

The CompareDocumentSettings controls the compare settings for a document.

CompareDocumentSettings::FirstPage property	Indicates the index of the first page to include in the document comparison.
CompareDocumentSettings::PageCount property	Indicates the number of pages to include in the document comparison.
CompareDocumentSettings::Margins property	Indicates the margins from top, left, bottom and right for the document.

CompareDocumentSettings::FirstPage property

Indicates the index of the first page to include in the document comparison.

.NET

```
int FirstPage { get; set; }
```

JAVA

```
int getFirstPage() throws IGRException;  
void setFirstPage(int value) throws IGRException;
```

PYTHON

```
@property  
def FirstPage(self) -> int  
  
@FirstPage.setter  
def FirstPage(self, value: int)
```

C++17

```
int getFirstPage() const;  
void setFirstPage(int value);
```

RETURN VALUE

`int` : The first page to compare.

SEE ALSO

- [Extractor::Compare](#)

CompareDocumentSettings::PageCount property

Indicates the number of pages to include in the document comparison.

.NET

```
int PageCount { get; set; }
```

JAVA

```
int getPageCount() throws IGRException;  
void setPageCount(int value) throws IGRException;
```

PYTHON

```
@property  
def PageCount(self) -> int  
  
@PageCount.setter  
def PageCount(self, value: int)
```

C++17

```
int getPageCount() const;  
void setPageCount(int value);
```

RETURN VALUE

`int` : The number of pages to compare.

SEE ALSO

- [Extractor::Compare](#)

CompareDocumentSettings::Margins property

Indicates the margins from top, left, bottom and right for the document.

.NET

```
System.Drawing.Rectangle Margins { get; set; }
```

JAVA

```
IGR_FRect getMargins() throws IGRException;  
void setMargins(IGR_FRect value) throws IGRException;
```

PYTHON

```
@property  
def Margins(self) -> IGR_FRect  
  
@Margins.setter  
def Margins(self, value: IGR_FRect)
```

C++17

```
Hyland.DocFilters.RectF getMargins() const;  
void setMargins(Hyland.DocFilters.RectF value);
```

RETURN VALUE

`IGR_FRect` : The margins for the document.

SEE ALSO

- [Extractor::Compare](#)

3.3.47 CompareDocumentSource

CompareDocumentSource interface

The `CompareDocumentSource` interface is used when comparing two documents, allowing you to set an extractor and provide optional settings.

<code>CompareDocumentSource::Extractor</code> property	Set <code>Extractor</code> property to point to an open <code>Extractor</code> instance.

INHERITED FROM COMPAREDOCUMENTSETTINGS

<code>CompareDocumentSettings::CompareDocumentSettings</code> interface	The <code>CompareDocumentSettings</code> controls the compare settings for a document.
<code>CompareDocumentSettings::FirstPage</code> property	Indicates the index of the first page to include in the document comparison.
<code>CompareDocumentSettings::PageCount</code> property	Indicates the number of pages to include in the document comparison.
<code>CompareDocumentSettings::Margins</code> property	Indicates the margins from top, left, bottom and right for the document.

CompareDocumentSource::Extractor property

Set `Extractor` property to point to an open `Extractor` instance.

.NET

```
Extractor Extractor { get; set; }
```

JAVA

```
Extractor getExtractor() throws IGREException;  
void setExtractor(Extractor value) throws IGREException;
```

PYTHON

```
@property  
def Extractor(self) -> Extractor  
  
@Extractor.setter  
def Extractor(self, value: Extractor)
```

C++17

```
Extractor getExtractor() const;  
void setExtractor(Extractor value);
```

RETURN VALUE

`Extractor` : The extractor to use as the revised document

SEE ALSO

- [Extractor::Compare](#)

3.3.48 CompareResultDifference

CompareResultDifference interface

The CompareResultDifference represents a single diff in a document or page comparison.

CompareResultDifference::Details property	Contains the words that are part of this diff.
CompareResultDifference::OriginalPageIndex property	Indicates the page index into the original document where the difference occurs.
CompareResultDifference::RevisedPageIndex property	Indicates the page index into the revised document where the difference occurs.
CompareResultDifference::Source property	Indicates the source of the difference.
CompareResultDifference::Type property	Indicates the source of the difference.

CompareResultDifference::Details property

Contains the words that are part of this diff.

.NET

```
IEnumerable<DifferenceDetail> Details { get; }
```

JAVA

```
VectDiffDifferences getDetails() throws IGRException;
```

PYTHON

```
@property  
def Details(self) -> list
```

C++17

```
CompareResultDifference::details_t getDetails() const;
```

RETURN VALUE

`IEnumerable<DifferenceDetail>` : A collection of found differences

SEE ALSO

- [Extractor::Compare](#)

CompareResultDifference::OriginalPageIndex property

Indicates the page index into the original document where the difference occurs.

.NET

```
int OriginalPageIndex { get; }
```

JAVA

```
int getOriginalPageIndex() throws IGRException;
```

PYTHON

```
@property  
def OriginalPageIndex(self) -> int
```

C++17

```
int getOriginalPageIndex() const;
```

RETURN VALUE

`int` : The page index in the original document.

SEE ALSO

- [Extractor::Compare](#)

CompareResultDifference::RevisedPageIndex property

Indicates the page index into the revised document where the difference occurs.

.NET

```
int RevisedPageIndex { get; }
```

JAVA

```
int getRevisedPageIndex() throws IGREException;
```

PYTHON

```
@property  
def RevisedPageIndex(self) -> int
```

C++17

```
int getRevisedPageIndex() const;
```

RETURN VALUE

`int` : The page index in the revised document.

SEE ALSO

- [Extractor::Compare](#)

CompareResultDifference::Source property

Indicates the source of the difference.

.NET

```
DifferenceSource Source { get; }
```

JAVA

```
DifferenceSource getSource() throws IGRException;
```

PYTHON

```
@property
def Source(self) -> DifferenceSource
```

C++17

```
DifferenceSource getSource() const;
```

RETURN VALUE

DifferenceSource : Indicates the source of the difference

Name	Constant
Original	ISYS11dfConstants.IGR_COMPARE_DOCUMENTS_DIFFERENCE_SOURCE_ORIGINAL
Revised	ISYS11dfConstants.IGR_COMPARE_DOCUMENTS_DIFFERENCE_SOURCE_REVISIED
Both	ISYS11dfConstants.IGR_COMPARE_DOCUMENTS_DIFFERENCE_SOURCE_BOTH

SEE ALSO

- [Extractor::Compare](#)

CompareResultDifference::Type property

Indicates the source of the difference.

.NET

```
DifferenceType Type { get; }
```

JAVA

```
DifferenceType getType() throws IGRException;
```

PYTHON

```
@property
def Type(self) -> DifferenceType
```

C++17

```
DifferenceType getType() const;
```

RETURN VALUE

DifferenceType : Indicates the type of diff found.

Name	Constant
Equal	IGR_COMPARE_DOCUMENTS_DIFFERENCE_EQUAL
Insert	IGR_COMPARE_DOCUMENTS_DIFFERENCE_INSERT
Delete	IGR_COMPARE_DOCUMENTS_DIFFERENCE_DELETE
Formatting	IGR_COMPARE_DOCUMENTS_DIFFERENCE_FORMATTING
NextBatch	IGR_COMPARE_DOCUMENTS_DIFFERENCE_NEXT_BATCH

SEE ALSO

- [Extractor::Compare](#)

3.3.49 CompareResultDifferenceDetail

CompareResultDifferenceDetail interface

The CompareResultDifferenceDetail interface contains information about the words that make up a CompareResultDifference , including the text, page index and bounding box.

CompareResultDifferenceDetail::Bounds property	Contains the bounding box of the word.
CompareResultDifferenceDetail::PageIndex property	Contains the page index.
CompareResultDifferenceDetail::Text property	Contains the text of the word.

CompareResultDifferenceDetail::Bounds property

Contains the bounding box of the word.

.NET

```
System.Drawing.RectangleF Bounds { get; }
```

JAVA

```
IGR_FRect getBounds() throws IGRException;
```

PYTHON

```
@property  
def Bounds(self) -> IGR_FRect
```

C++17

```
Hyland.DocFilters.RectF getBounds() const;
```

RETURN VALUE

`IGR_FRect` : The bounding box of the diff.

SEE ALSO

- [Extractor::Compare](#)

CompareResultDifferenceDetail::PageIndex property

Contains the page index.

.NET

```
int PageIndex { get; }
```

JAVA

```
int getPageIndex() throws IGREException;
```

PYTHON

```
@property  
def PageIndex(self) -> int
```

C++17

```
int getPageIndex() const;
```

RETURN VALUE

`int` : The page index.

SEE ALSO

- [Extractor::Compare](#)

CompareResultDifferenceDetail::Text property

Contains the text of the word.

.NET

```
string Text { get; }
```

JAVA

```
string getText() throws IGREException;
```

PYTHON

```
@property  
def Text(self) -> string
```

C++17

```
std::wstring getText() const;
```

RETURN VALUE

`string` : The text of the word.

SEE ALSO

- [Extractor::Compare](#)

3.3.50 CompareResults

CompareResults interface

The CompareResults interface is the returned object from a call to [Extractor::Compare](#) or [Page::Compare](#) and is used to enumerate the differences between the two entities.

CompareResults::Close Method	Close the instances of a page or document comparison.
CompareResults::GetNext Method	Returns the next <code>Difference</code> object for a document or page comparison.

CompareResults::Close Method

Close the instances of a page or document comparison.

PROTOTYPE

.NET

```
void Close()
```

JAVA

```
void Close() throws IGRException;
```

PYTHON

```
def Close(self) -> void
```

C++17

```
void Close()
```

ADDITIONAL INFORMATION

Resources can also be managed with the `using IDisposable` pattern in C#, or the `try-with-resources` pattern in Java.

SEE ALSO

- [CompareResultDifference](#)
- [Extractor::Compare](#)

CompareResults::GetNext Method

Returns the next `Difference` object for a document or page comparison.

PROTOTYPE

.NET

```
CompareResultDifference GetNext()
```

JAVA

```
CompareResultDifference GetNext() throws IGRException;
```

PYTHON

```
def GetNext(self) -> CompareResultDifference
```

C++17

```
CompareResultDifference GetNext()
```

RETURN VALUE

`CompareResultDifference` : Returns a `CompareResultDifference` interface that contains the details of the next difference.

SEE ALSO

- `CompareResultDifference`
- `Extractor::Compare`

3.3.51 CompareSettings

CompareSettings interface

The CompareSettings instance allows you to set the comparison type and flags when comparing documents or pages.

CompareSettings::CompareType property	Indicates the flags that control the comparison logic.
CompareSettings::Flags property	Indicates the flags that control the comparison logic.

CompareSettings::CompareType property

Indicates the flags that control the comparison logic.

.NET

```
CompareType CompareType { get; set; }
```

JAVA

```
CompareType getCompareType() throws IGRException;  
void setCompareType(CompareType value) throws IGRException;
```

PYTHON

```
@property  
def CompareType(self) -> CompareType  
  
@CompareType.setter  
def CompareType(self, value: CompareType)
```

C++17

```
CompareType getCompareType() const;  
void setCompareType(CompareType value);
```

RETURN VALUE

CompareType : Indicates the type of comparison to perform.

SEE ALSO

- [CompareSettings](#)
- [Extractor::Compare](#)
- [IGR_COMPARE_DOCUMENTS_COMPARE_TYPE](#)

CompareSettings::Flags property

Indicates the flags that control the comparison logic.

.NET

```
int Flags { get; set; }
```

JAVA

```
int getFlags() throws IGRException;  
void setFlags(int value) throws IGRException;
```

PYTHON

```
@property  
def Flags(self) -> int  
  
@Flags.setter  
def Flags(self, value: int)
```

C++17

```
int getFlags() const;  
void setFlags(int value);
```

RETURN VALUE

`int` : Provide the flags that control the comparison logic.

SEE ALSO

- [CompareSettings](#)
- [Extractor::Compare](#)
- [IGR_COMPARE_DOCUMENTS_FLAGS_TYPE](#)

3.3.52 DocumentFilters

DocumentFilters interface

DocumentFilters (formerly IFileReaders) is the primary factory object in the Document Filters Object Library. You will need to create and initialize an instance of this object to start using the API. It is recommended to define the object in the application scope and create and initialize it only once.

The samples for each method are provided in JScript and assume a global instance of the DocumentFilters factory object that is already created and initialized.

DocumentFilters::GetAvailableOptions method	Returns the list of available configuration options.
DocumentFilters::GetExtractor method	The GetExtractor method obtains an Extractor object to process a document.
DocumentFilters::GetSupportedFormats method	Returns the list of supported file types.
DocumentFilters::Initialize method	The Initialize method initializes and authorizes the Document Filters API. It is the first method that your application must call.
DocumentFilters::MakeOutputCanvas method	Creates a new canvas that is used for rendering page content. The output data will be written to the file specified in Filename.
DocumentFilters::OpenExtractor method	The OpenExtractor is a utility method that obtains an Extractor object and immediately call's it's Open method to process a document.

DocumentFilters::GetAvailableOptions method

Returns the list of available configuration options.

PROTOTYPE

.NET

```
IEnumerable<IGROption> GetAvailableOptions()
```

JAVA

```
VectIGROption GetAvailableOptions() throws IGRException;
```

PYTHON

```
def GetAvailableOptions(self) -> IGROption[]
```

C++17

```
std::vector<Option> getAvailableOptions()
```

RETURN VALUE

`IGROption[]` : List of available configuration options

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 // Call the GetAvailableOptions method
8 var availableOptions = api.GetAvailableOptions();
9
10 // Iterate over the available options and print their details
11 foreach (Option option in availableOptions) {
12     Console.WriteLine("Option Display Name: " + option.DisplayName);
13     Console.WriteLine("Option Description: " + option.Description);
14     Console.WriteLine("Option Default Value: " + option.DefaultValue);
15     Console.WriteLine("Option Type: " + option.Type);
16     Console.WriteLine("Option Flags: " + option.Flags);
17     Console.WriteLine("Option Possible Values: " + string.Join(", ", option.PossibleValues));
18     Console.WriteLine();
19 }
```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         DocumentFilters df = new DocumentFilters();
6         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
7
8         // Call the GetAvailableOptions method
9         VectIGROption availableOptions = df.GetAvailableOptions();
10
11        // Iterate over the available options and print their details
12        for (int i = 0; i < availableOptions.size(); i++) {
13            IGROption option = availableOptions.get(i);
14            System.out.println("Option Display Name: " + option.GetDisplayName());
15            System.out.println("Option Description: " + option.GetDescription());
16            System.out.println("Option Default Value: " + option.GetDefaultValue());
17            System.out.println("Option Type: " + option.GetType());
18            System.out.println("Option Flags: " + option.GetFlags());
19            System.out.println("Option Possible Values: " + option.GetPossibleValuesStr());
20            System.out.println();
21        }
22    }
23 }

```

PYTHON

```

1 from DocumentFilters import *
2
3 df = DocumentFilters()
4 df.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 # Call the getAvailableOptions method
7 available_options = df.GetAvailableOptions()
8
9 # Iterate over the available options and print their details
10 for option in available_options:
11     print("Option Display Name:", option.DisplayName)
12     print("Option Description:", option.Description)
13     print("Option Default Value:", option.DefaultValue)
14     print("Option Type:", option.Type)
15     print("Option Flags:", option.Flags)
16     print("Option Possible Values:", option.PossibleValues)
17     print()

```

C++17

```
1  #include <DocumentFiltersObjects.h>
2
3  int main(int argc, char** argv) {
4      Hyland::DocFilters::DocumentFilters df;
5      df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7      // Call the getAvailableOptions method
8      const std::vector<Hyland::DocFilters::Option>& availableOptions = df.getAvailableOptions();
9
10     // Iterate over the available options and print their details
11     for (const auto& option : availableOptions) {
12         std::wcout << L"Option Display Name: " << option.getDisplayName() << std::endl;
13         std::wcout << L"Option Description: " << option.getDescription() << std::endl;
14         std::wcout << L"Option Default Value: " << option.getDefaultValue() << std::endl;
15         std::wcout << L"Option Type: " << option.getType() << std::endl;
16         std::wcout << L"Option Flags: " << option.getFlags() << std::endl;
17
18         // Get possible values
19         const std::vector<std::wstring>& possibleValues = option.getPossibleValues();
20         std::wcout << L"Option Possible Values: ";
21         for (const auto& value : possibleValues) {
22             std::wcout << value << L", ";
23         }
24         std::wcout << std::endl;
25     }
26
27     return 0;
28 }
```

DocumentFilters::GetExtractor method

The GetExtractor method obtains an [Extractor](#) object to process a document. An [Extractor](#) allows you to:

- Identify a document’s type.
- Extract its text.
- Extract its metadata.
- Enumerate and extract sub-documents.
- Convert a document to HTML.
- Convert a document to an image or series of images.

OVERLOADS

GetExtractor(string)	Get an Extractor object from a filename.
GetExtractor(stream)	Get an Extractor object from a stream.
GetExtractor(byte[])	Get an Extractor object from a byte array.

GETEXTRACTOR(STRING)

Prototype

.NET

```
Extractor GetExtractor(string filename)
```

JAVA

```
Extractor GetExtractor(string filename) throws IGRException;
```

PYTHON

```
def GetExtractor(self, filename: string) -> Extractor
```

C++17

```
Extractor GetExtractor(const std::wstring& filename)
```

Parameters

`filename: string` : Path to the document to be opened.

Return value

`Extractor` :

GETEXTRACTOR(STREAM)**Prototype****.NET**

```
Extractor GetExtractor(System.IO.Stream stream)
```

JAVA

```
Extractor GetExtractor(IGRStream stream) throws IGRException;
```

PYTHON

```
def GetExtractor(self, stream: Stream) -> Extractor
```

C++17

```
Extractor GetExtractor(Stream& stream);  
Extractor GetExtractor(IGR_Stream* stream);  
Extractor GetExtractor(std::istream& stream);  
Extractor GetExtractor(std::istream* stream, bool own_stream);  
Extractor GetExtractor(Stream* stream, bool own_stream);
```

Parameters

stream: stream : Stream object pointing to the binary document content, the stream type is language dependent.

Return value

Extractor :

GETEXTRACTOR(BYTE[])**Prototype****.NET**

```
Extractor GetExtractor(byte[] bytes)
```

JAVA

```
Extractor GetExtractor(byte[] bytes) throws IGRException;
```

PYTHON

```
def GetExtractor(self, bytes: byte[]) -> Extractor
```

C++17

```
Extractor GetExtractor(const void* bytes, size_t bytesSize)
```

Parameters

bytes: `byte[]` : A memory block containing binary document content.

Return value

Extractor :

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc"); // (2) Create an instance of the Extractor for a file
8
9 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta); // (3) Open the document for reading in text-mode
10
11 while (!doc.EndOfStream) // (4) Read the document in 4KB chunks and write it to the console
12 {
13     var text = doc.GetText(4096);
14     Console.Out.WriteLine(text);
15 }
16
17 doc.Close();
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.doc")) {
11            doc.Open(isys_docfilters.IGR_BODY_AND_META);
12
13            while (!doc.getEOF()) {
14                string text = doc.GetText(4096);
15                System.out.println(text);
16            }
17        }
18    }
19 }
```

PYTHON

```

1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.doc") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     while not doc.getEOF():
10         output.write(doc.GetText(MaxCharsPerGetText, stripControlCodes=True))

```

C++17

```

1 #include <iostream>
2 #include <string>
3 #include "DocumentFiltersObjects.h"
4
5 int main() {
6     try {
7         // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8         Hyland::DocFilters::Api api;
9
10        // Initialize the DocumentFilters object with license and path
11        std::string license = "YOUR_LICENSE_KEY_HERE";
12        std::string path = ".";
13        api.Initialize(license, path);
14
15        // Get an extractor for the specified file
16        Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.doc");
17
18        // Open the document with BodyAndMeta flag
19        doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
20
21        // Read and print the text content
22        while (!doc.getEOF()) {
23            std::wstring text = doc.GetText(4096);
24            std::wcout << text << std::endl;
25        }
26
27        // Close the document
28        doc.Close();
29
30    } catch (const std::exception& ex) {
31        std::cerr << "Error: " << ex.what() << std::endl;
32    }
33    return 0;
34 }

```

SEE ALSO

- [DocumentFilters interface](#)
- [Extractor interface](#)

DocumentFilters::GetSupportedFormats method

Returns the list of supported file types.

PROTOTYPE

.NET

```
IEnumerable<IGRFormat> GetSupportedFormats()
```

JAVA

```
VectIGRFormat GetSupportedFormats() throws IGRException;
```

PYTHON

```
def GetSupportedFormats(self) -> IGRFormat[]
```

C++17

```
std::vector<Format> getSupportedFormats()
```

RETURN VALUE

[IGRFormat\[\]](#) : List of supported file types.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var df = new Hyland.DocumentFilters.Api();
5 df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 // Call the GetSupportedFormats method
8 var supportedFormats = df.GetSupportedFormats();
9
10 // Iterate over the supported formats and print their details
11 foreach (Format format in supportedFormats) {
12     Console.WriteLine("Format Display Name: " + format.DisplayName);
13     Console.WriteLine("Format Short Name: " + format.ShortName);
14     Console.WriteLine("Format Type: " + format.FileTypeCategory);
15     Console.WriteLine();
16 }
```


JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         DocumentFilters df = new DocumentFilters();
6         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
7
8         // Call the GetSupportedFormats method
9         VectIGRFormat supportedFormats = df.GetSupportedFormats();
10
11         // Iterate over the supported formats and print their details
12         for (int i = 0; i < supportedFormats.size(); i++) {
13             IGRFormat format = supportedFormats.get(i);
14             System.out.println("Format Display Name: " + format.GetDisplayName());
15             System.out.println("Format Short Name: " + format.GetShortName());
16             System.out.println("Format Config Name: " + format.GetConfigName());
17             System.out.println("Format MIME Type: " + format.GetMimeType());
18             System.out.println("Format ID: " + format.GetId());
19             System.out.println("Format File Type Category: " + format.GetFileTypeCategory());
20             System.out.println();
21         }
22     }
23 }
```

PYTHON

```
1 from DocumentFilters import *
2
3 df = DocumentFilters()
4 df.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 # Call the getSupportedFormats method
7 supported_formats = df.getSupportedFormats()
8
9 # Iterate over the supported formats and print their details
10 for format in supported_formats:
11     print("Format Display Name:", format.DisplayName)
12     print("Format Short Name:", format.ShortName)
13     print("Format Config Name:", format.ConfigName)
14     print("Format MIME Type:", format.MimeType)
15     print("Format ID:", format.id)
16     print()
```

C++17

```
1 #include <DocumentFiltersObjects.h>
2
3 int main(int argc, char **argv) {
4     Hyland::DocFilters::DocumentFilters df;
5     df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7     // Call the getSupportedFormats method
8     const std::vector<Hyland::DocFilters::Format> &supportedFormats = df.getSupportedFormats();
9
10    // Iterate over the supported formats and print their details
11    for (const auto &format : supportedFormats)
12    {
13        std::wcout << L"Format Display Name: " << format.getDisplayName() << std::endl;
14        std::wcout << L"Format Short Name: " << format.getShortName() << std::endl;
15        std::wcout << L"Format Config Name: " << format.getConfigName() << std::endl;
16        std::wcout << L"Format MIME Type: " << format.getMimeType() << std::endl;
17        std::wcout << L"Format ID: " << format.id() << std::endl;
18        // Access additional properties if needed
19        std::wcout << std::endl;
20    }
21
22    return 0;
23 }
```

DocumentFilters::Initialize method

The Initialize method initializes and authorizes the Document Filters API. It is the first method that your application must call.

PROTOTYPE

.NET

```
void Initialize(string license, string install_dir)
```

JAVA

```
void Initialize(string license, string install_dir) throws IGRException;
```

PYTHON

```
def Initialize(self, license: string, install_dir: string) -> void
```

C++17

```
void Initialize(const std::wstring& license, const std::wstring& install_dir)
```

PARAMETERS

`license: string` : Document Filters License Code.

`install_dir: string` : Path to Document Filters resources, such as configuration files and fonts.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 try
5 {
6     // Create a DocumentFilters object
7     DocumentFilters api = new DocumentFilters();
8
9     // Initialize the DocumentFilters object with license and path
10    string license = "YOUR_LICENSE_KEY_HERE";
11    string resource_path = "";
12    api.Initialize(license, resource_path);
13
14    // Your code using the DocumentFilters object goes here
15 }
16 catch (Exception ex)
17 {
18     Console.WriteLine("Error: " + ex.Message);
19 }
```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class Example {
4     public static void main(String[] args) {
5         try {
6             // Create a DocumentFilters object
7             DocumentFilters df = new DocumentFilters();
8
9             // Initialize the DocumentFilters object with license and path
10            String license = "YOUR_LICENSE_KEY_HERE";
11            String resource_path = "";
12            df.Initialize(license, path);
13
14            // Your code using the DocumentFilters object goes here
15
16        } catch (IGRException ex) {
17            System.out.println("Error: " + ex.getMessage());
18        }
19    }
20 }

```

PYTHON

```

1 from DocumentFilters import DocumentFilters
2
3 # Create a DocumentFilters object
4 df = DocumentFilters()
5
6 # Initialize the DocumentFilters object with license and path
7 license = "YOUR_LICENSE_KEY_HERE"
8 resource_path = "path/to/your/resources"
9 df.Initialize(license, resource_path)

```

C++17

```

1 #include <iostream>
2 #include <string>
3 #include "DocumentFiltersObjects.h"
4
5 int main() {
6     try {
7         // Create a DocumentFilters object
8         Hyland::DocFilters::DocumentFilters df;
9
10        // Initialize the DocumentFilters object with license and path
11        std::wstring license = L"YOUR_LICENSE_KEY_HERE";
12        std::wstring resource_path = L"";
13        df.Initialize(license, resource_path);
14
15        // Your code using the DocumentFilters object goes here
16
17    } catch (const std::exception& ex) {
18        std::cerr << "Error: " << ex.what() << std::endl;
19    }
20    return 0;
21 }
22 C#

```

ADDITIONAL INFORMATION

.NET Native Library path - ISYSdf11.dll will be loaded by P/Invoke through the [DllImport] attribute and must be either in same folder as the currently executing Assembly or found by [Default Probing rules for Unmanaged \(native\) libraries](#)

Java Native Library path - ISYS11df.(dll/so/dylib) will be loaded by a call to `System.loadLibrary("ISYS11df")`
See [System.LoadLibrary](#) for more information

SEE ALSO

- [DocumentFilters interface](#)

DocumentFilters::MakeOutputCanvas method

Creates a new canvas that is used for rendering page content. The output data will be written to the file specified in Filename.

OVERLOADS

MakeOutputCanvas(string, int, string)	Create a new canvas writing to a file.
MakeOutputCanvas(stream, int, string)	Create a new canvas writing to a stream.

MAKEOUTPUTCANVAS(STRING, INT, STRING)

Prototype

.NET

```
Canvas MakeOutputCanvas(string filename, int canvasType, string options)
```

JAVA

```
Canvas MakeOutputCanvas(string filename, int canvasType, string options) throws IGRException;
```

PYTHON

```
def MakeOutputCanvas(self, filename: string, canvasType: int, options: string) -> Canvas
```

C++17

```
Canvas MakeOutputCanvas(const std::wstring& filename, int canvasType, const std::wstring& options = std::wstring(L""))
```

C++11

```
Canvas* MakeOutputCanvas(std::wstring filename, int canvasType, std::wstring options)
```

COM

```
HRESULT MakeOutputCanvas([in] BSTR filename, [in] int canvasType, [in] BSTR options, [out, retval] Canvas* *result)
```

Parameters

filename: string : Filename where canvas is to be created

canvasType: int : Indicates the type of output device to create; see [Canvas Types](#) for details.

options: string : Semicolon separated list of name value pair options; see [Constants and Codes](#) for details.

Return value

Canvas : The newly created canvas object.

MAKEOUTPUTCANVAS(STREAM, INT, STRING)**Prototype****.NET**

```
Canvas MakeOutputCanvas(System.IO.Stream stream, int canvasType, string options)
```

JAVA

```
Canvas MakeOutputCanvas(IGRStream stream, int canvasType, string options) throws IGRException;
```

PYTHON

```
def MakeOutputCanvas(self, stream: Stream, canvasType: int, options: string) -> Canvas
```

C++17

```
Canvas MakeOutputCanvas(Stream& stream, int canvasType, const std::wstring& options = std::wstring(L""));
Canvas MakeOutputCanvas(std::iostream& stream, CanvasType type, const std::wstring& options);
Canvas MakeOutputCanvas(std::iostream* stream, bool own_stream, CanvasType type, const std::wstring&
options);
Canvas MakeOutputCanvas(Stream* stream, bool own_stream, bool own_stream, CanvasType type, const
std::wstring& options);
```

C++11

```
Canvas* MakeOutputCanvas(Stream* stream, int canvasType, std::wstring options)
```

COM

```
HRESULT MakeOutputCanvasOnStream([in] IStream* stream, [in] int canvasType, [in] BSTR options, [out, retval]
Canvas* *result)
```

Parameters

stream: stream : Stream where canvas is to be created

canvasType: int : Indicates the type of output device to create; see [Canvas Types](#) for details.

options: string : Semicolon separated list of name value pair options; see [Constants and Codes](#) for details.

Return value

Canvas : The newly created canvas object.

SAMPLE CODE

.NET

```

1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
7 using var canvas = api.MakeOutputCanvas("filename.pdf", CanvasType.PDF);
8
9 canvas.RenderPages(doc);

```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class hidef_pdf
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.doc"))
11        {
12            try (Canvas canvas = df.MakeOutputCanvas("filename.pdf",
13 isys_docfiltersConstants.IGR_DEVICE_IMAGE_PDF, ""))
14            {
15                doc.Open(isys_docfiltersConstants.IGR_FORMAT_IMAGE, "");
16
17                for (int i = 0, c = doc.GetPageCount(); i < c; ++i)
18                {
19                    try (Page page = doc.GetPage(i)) {
20                        canvas.RenderPage(page);
21                    }
22                }
23            }
24        }
25 }

```

PYTHON

```

1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.OpenExtractor("filename.doc", mode=IGR_FORMAT_IMAGE) as doc:
7     with api.MakeOutputCanvas("filename.pdf", canvasType=IGR_DEVICE_PDF) as canvas:
8         canvas.RenderPages(doc)

```


C++17

```
1 #include <DocumentFiltersObjects.h>
2
3 int main() {
4     try {
5         // Create and initialize the API object
6         Hyland::DocFilters::Api api;
7         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
8
9         // Open the input file
10        Hyland::DocFilters::Extractor doc = api.OpenExtractor("filename.doc",
Hyland::DocFilters::OpenMode::Paginated);
11
12        // Create the output canvas
13        Hyland::DocFilters::Canvas canvas = api.MakeOutputCanvas("output.pdf",
Hyland::DocFilters::CanvasType::PDF);
14
15        // Render all pages to the output
16        canvas.RenderPages(doc);
17    } catch (const std::exception& e) {
18        std::cerr << "Error: " << e.what() << std::endl;
19        return 1; // Indicate an error
20    }
21
22    return 0; // Successful execution
23 }
```

ADDITIONAL INFORMATION

For output formats that support multiple pages, you may choose to write multiple input documents to a single output document.

SEE ALSO

- [DocumentFilters interface](#)
- [Canvas interface](#)

DocumentFilters::OpenExtractor method

The OpenExtractor is a utility method that obtains an [Extractor](#) object and immediately call's it's Open method to process a document. An [Extractor](#) allows you to:

- Identify a document's type.
- Extract its text.
- Extract its metadata.
- Enumerate and extract sub-documents.
- Convert a document to HTML.
- Convert a document to an image or series of images.

OVERLOADS

OpenExtractor (string, OpenMode, string, Func<OpenCallback, int>)	Get and open an extractor from a file.
OpenExtractor (stream, OpenMode, string, Func<OpenCallback, int>)	Get and open an extractor from a stream.
OpenExtractor (byte[], OpenMode, string, Func<OpenCallback, int>)	Get and open an extractor from an array of bytes.

OPENEXTRACTOR(STRING, OPENMODE, STRING, FUNC<OPENCALLBACK, INT>)

Prototype

.NET

```
Extractor OpenExtractor(string filename, OpenMode openMode, string options, Func<OpenCallback, int> callback)
```

JAVA

```
Extractor OpenExtractor(string filename, OpenMode openMode, string options, Func<OpenCallback, int> callback) throws IGRException;
```

PYTHON

```
def OpenExtractor(self, filename: string, openMode: OpenMode, options: string, callback: Func<OpenCallback, int>) -> Extractor
```

C++17

```
Extractor OpenExtractor(const std::wstring& filename, OpenMode openMode, const std::wstring& options, open_callback_t callback)
```

Parameters

`filename: string` : Path to the document to be opened.

`openMode: OpenMode` : Indicates the processing mode for the document, can be one of:

- Text
- Paginated
- ClassicHTML

`options: string` : See [Open Options](#) on page.

`callback: Func<OpenCallback, int>` : Optional IGR_Open_Ex callback function.

Return value

[Extractor](#) : New extractor object for the provided file.

OPENEXTRACTOR(STREAM, OPENMODE, STRING, FUNC<OPENCALLBACK, INT>)

Prototype**.NET**

```
Extractor OpenExtractor(System.IO.Stream stream, OpenMode openMode, string options, Func<OpenCallback, int>
callback)
```

JAVA

```
Extractor OpenExtractor(IGRStream stream, OpenMode openMode, string options, Func<OpenCallback, int>
callback) throws IGRException;
```

PYTHON

```
def OpenExtractor(self, stream: Stream, openMode: OpenMode, options: string, callback: Func<OpenCallback,
int>) -> Extractor
```

C++17

```
Extractor OpenExtractor(Stream& stream, OpenMode openMode, const std::wstring& options, open_callback_t
callback);
Extractor OpenExtractor(IGR_Stream* stream, OpenMode mode, int open_flags = 0, const std::wstring& options,
const open_callback_t& callback);
Extractor OpenExtractor(std::istream& stream, OpenMode mode, int open_flags = 0, const std::wstring&
options, const open_callback_t& callback);
Extractor OpenExtractor(std::istream* stream, bool own_stream, OpenMode mode, int open_flags = 0, const
std::wstring& options, const open_callback_t& callback);
```

Parameters

`stream: stream` : Stream object pointing to the binary document content.

`openMode: OpenMode` : Indicates the processing mode for the document, can be one of: - Text - Paginated - ClassicHTML

`options: string` : See [Open Options](#) on page.

`callback: Func<OpenCallback, int>` : Optional IGR_Open_Ex callback function.

Return value

`Extractor` : New extractor object for the provided file.

OPENEXTRACTOR(BYTE[], OPENMODE, STRING, FUNC<OPENCALLBACK, INT>)

Prototype

.NET

```
Extractor OpenExtractor(byte[] bytes, OpenMode openMode, string options, Func<OpenCallback, int> callback)
```

JAVA

```
Extractor OpenExtractor(byte[] bytes, OpenMode openMode, string options, Func<OpenCallback, int> callback)
throws IGRException;
```

PYTHON

```
def OpenExtractor(self, bytes: byte[], openMode: OpenMode, options: string, callback: Func<OpenCallback,
int>) -> Extractor
```

C++17

```
Extractor OpenExtractor(const void* bytes, size_t bytesSize, OpenMode openMode, const std::wstring& options,
open_callback_t callback)
```

Parameters

`bytes: byte[]` : A memory block pointing to the binary document content.

`openMode: OpenMode` : Indicates the processing mode for the document, can be one of: - Text - Paginated - ClassicHTML

`options: string` : See [Open Options](#) on page.

`callback: Func<OpenCallback, int>` : Optional IGR_Open_Ex callback function.

Return value

`Extractor` : New extractor object for the provided file.

SAMPLE CODE

.NET

```

1 using Hyland.DocumentFilters;
2 using System;
3
4 try
5 {
6     // Create a DocumentFilters object
7     DocumentFilters df = new DocumentFilters();
8
9     // Initialize the DocumentFilters object with license and path
10    string license = "YOUR_LICENSE_KEY_HERE";
11    string path = "path/to/resources";
12    df.Initialize(license, path);
13
14    // Open a document using OpenExtractor with filename (string) and OpenType flag
15    using var extractor = df.OpenExtractor("sample.pdf", OpenMode.Text, OpenType.BodyAndMeta, "", null);
16
17    // Your code to process the document using extractor goes here
18
19 }
20 catch (Exception ex)
21 {
22     Console.WriteLine("Error: " + ex.Message);
23 }

```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class OpenExtractorExample {
4     public static void main(String[] args) {
5         try {
6             // Create a DocumentFilters object
7             DocumentFilters df = new DocumentFilters();
8
9             // Initialize the DocumentFilters object (assuming a similar Initialize method exists in Java)
10            String license = "YOUR_LICENSE_KEY_HERE";
11            String path = "path/to/resources";
12            df.Initialize(license, path); // This might need adjustment based on the Java API
13
14            // Open a document using OpenExtractor with filename (string)
15            Extractor extractor = df.OpenExtractor("sample.pdf", OpenMode.Text, OpenType.BodyAndMeta, "",
16            null);
17
18            // Your code to process the document using extractor goes here
19
20            // Close the extractor
21            extractor.Close();
22
23        } catch (IGRException ex) {
24            System.err.println("Error: " + ex.getMessage());
25        }
26    }
27 }

```

PYTHON

```

1 from DocumentFilters import DocumentFilters, OpenMode
2
3 try:
4     # Create an Api object (which is an alias for DocumentFilters)
5     api = DocumentFilters()
6
7     # Initialize the DocumentFilters object with license and path
8     license = "YOUR_LICENSE_KEY_HERE"
9     resource_path = "."
10    api.Initialize(license, resource_path)
11
12    # Open a document using OpenExtractor with filename (string)
13    extractor = api.OpenExtractor(source="sample.pdf", mode=OpenMode.Text, options="", callback=None)
14
15    # Your code to process the document using extractor goes here
16
17 except DocumentFilters.IGREException as ex:
18     print("Error:", ex)

```

C++17

```

1 #include <iostream>
2 #include <string>
3 #include "DocumentFiltersObjects.h"
4
5 int main() {
6     try {
7         // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8         Hyland::DocFilters::Api api;
9
10        // Initialize the DocumentFilters object with license and path
11        std::string license = "YOUR_LICENSE_KEY_HERE";
12        std::string path = ".";
13        api.Initialize(license, path);
14
15        // Open a document using OpenExtractor with filename (string)
16        Hyland::DocFilters::Extractor extractor = api.OpenExtractor("sample.pdf",
17        Hyland::DocFilters::OpenMode::Text, 0, L"", nullptr);
18
19        // Your code to process the document using extractor goes here
20
21        // Close the extractor
22        extractor.Close();
23
24    } catch (const std::exception& ex) {
25        std::cerr << "Error: " << ex.what() << std::endl;
26    }
27    return 0;
28 }

```

ADDITIONAL INFORMATION

OpenMode indicates extended processing options, can be one of:

- BodyOnly
- MetaOnly

- `BodyAndMeta`
- `FormatText`
- `FormatHTML`
- `FormatXML`
- `FormatHTMLIfYouCan`
- `FormatImage`
- `FormatHTMLNoFallback`

SEE ALSO

- [DocumentFilters interface](#)
- [Extractor interface](#)

3.3.53 Extractor

Extractor interface

The Extractor interface allows you to extract the content of a document and/or enumerate its sub-documents, such as email attachments and ZIP archives.

To obtain this interface, call the [DocumentFilters.GetExtractor](#) method. The Extractor interface contains the following methods and properties.

Extractor::ApproveExternalResourceCallback property	Set a callback hook called when an embedded images is to be downloaded.
Extractor::Close method	The Close method releases the document resources referenced by this Extractor object.
Extractor::Compare method	The Compare method allows you to compare two documents returning the differences.
Extractor::CopyTo method	The CopyTo method extracts the binary content of the sub-document to a file.
Extractor::EOF property	The EOF property is only valid for documents where the SupportsText property is TRUE. The EOF property will be set to TRUE when no more text can be extracted from the document with calls to GetText . If the document needs to be re-read, call Close and Open first.
Extractor::FileType property	The FileType property is the document format code, as listed in Document Format Codes chart on page . The function is overloaded to be able to return the format name as a string.
Extractor::GetFirstImage method	The GetFirstImage method obtains a SubFile object representing the first embedded image of the current document when converting using classic HTML.
Extractor::GetFirstPage method	The GetFirstPage method returns the first page object of an opened document. The document must be opened in image mode (IGR_FORMAT_IMAGE).
Extractor::GetFirstSubFile method	The GetFirstSubFile method obtains a SubFile object representing the first sub-document of the current document.

Extractor::GetHashMD5 method	The getHashMD5 methods obtain a string representing the calculated hash of the current document for unique identification.
Extractor::GetHashSHA1 method	The getHashSHA1 methods obtain a string representing the calculated hash of the current document for unique identification.
Extractor::GetNextImage method	The GetNextImage method obtains a SubFile object representing the next embedded image of the current document when converting using classic HTML.
Extractor::GetNextPage method	The GetNextPage method returns the next page object of an opened document. The document must be opened in image mode (IGR_FORMAT_IMAGE).
Extractor::GetNextSubFile method	The GetNextSubFile method obtains a SubFile object representing the next sub-document of the current document.
Extractor::GetPage method	The GetPage method returns the page at the given index, where the page index is 0-based. An exception is raised if the index is invalid.
Extractor::GetPageCount method	Returns the number of pages in the current document, the document must be opened in image mode for the page count to be populated.
Extractor::GetResourceStreamCallback property	Set a callback hook called when an rendering an embedded image.
Extractor::GetRootBookmark method	The GetRootBookmark method returns a Bookmark node representing the top-most node of the bookmark hierarchy. The root bookmark only has Children data, it has no title or destination properties.
Extractor::GetSubFile method	The GetSubFile method obtains a SubFile object representing the nominated sub-file of the current document.
Extractor::GetText method	The GetText method extracts the next portion of text content from the document.
Extractor::HeartbeatCallback property	Set a callback hook called when an heartbeat is sent.
Extractor::Images property	

	The Images method property provides an enumerable collection of SubFile objects representing the embedded image of the current document when converting using classic HTML.
Extractor::Localize property	Utility function that allows for localization of metadata without providing a callback. Any localization options must be set before an <code>.Open</code> call.
Extractor::LocalizeCallback property	Set a callback hook called when a string is to be localized.
Extractor::LogLevelCallback property	Set a callback hook called a log-level is requested for a module.
Extractor::LogMessageCallback property	Set a callback hook called a log-message is sent.
Extractor::MimeType property	Returns the MimeType of the file.
Extractor::OcrImageCallback property	Set a callback hook called when an OCRing an image.
Extractor::Open method	The Open method opens a document for processing.
Extractor::PageCount property	Returns the number of pages in the current document, the document must be opened in image mode for the page count to be populated.
Extractor::Pages property	The Pages property provides an enumerable collection of pages for an opened document. The document must be opened in image mode (IGR_FORMAT_IMAGE).
Extractor::PasswordCallback property	Set a callback hook called when a password is required.
Extractor::SaveTo method	The SaveTo method extracts the entire text content of the document in a single call. The text may be saved to a file with the given name or via an instance of an IStream (COM) object.
Extractor::SubFiles property	Returns an enumerable set of SubFiles.
Extractor::getFileType method	The FileType method allows for extended information to be returned about the file type.
Extractor::getSupportsHTML method	getSupportsHTML method is TRUE if document can be converted to classic HTML.

Extractor::getSupportsSubFiles property	getSupportsSubFiles property is TRUE if the document is a compound or archive document, potentially with sub-documents.
Extractor::getSupportsText method	getSupportsText method return TRUE if text content can be extracted from the document. This property must be TRUE to be able to call to the Extractor::SaveTo and Extractor::GetText methods.

Extractor::ApproveExternalResourceCallback property

Set a callback hook called when an embedded images is to be downloaded.

.NET

```
Func<string, bool> ApproveExternalResourceCallback { get; set; }
```

JAVA

```
void setApproveExternalResourceCallback(IApproveExternalResourceCallback value) throws IGRException;
```

PYTHON

```
@property
def ApproveExternalResourceCallback(self) -> Callable[[OpenCallback.ActionApproveExternalResource], bool]

@ApproveExternalResourceCallback.setter
def ApproveExternalResourceCallback(self, value: Callable[[OpenCallback.ActionApproveExternalResource],
bool])
```

C++17

```
void setApproveExternalResourceCallback(std::function<bool(const std::wstring&)> value);
```

RETURN VALUE

Function

Extractor::Close method

The Close method releases the document resources referenced by this [Extractor](#) object.

PROTOTYPE

.NET

```
void Close()
```

JAVA

```
void Close() throws IGRException;
```

PYTHON

```
def Close(self) -> void
```

C++17

```
void Close()
```

C++11

```
void Close()
```

COM

```
HRESULT Close()
```

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc"); // (2) Create an instance of the Extractor for a file
8
9 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta); // (3) Open the document for reading in text-mode
10
11 while (!doc.EndOfStream) // (4) Read the document in 4KB chunks and write it to the console
12 {
13     var text = doc.GetText(4096);
14     Console.Out.WriteLine(text);
15 }
16
17 doc.Close();
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.doc")) {
11            doc.Open(IGR_BODY_AND_META);
12
13            while (!doc.getEOF()) {
14                string text = doc.GetText(4096);
15                System.out.println(text);
16            }
17        }
18    }
19 }
```

PYTHON

```
1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.doc") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     while not doc.getEOF():
10         output.write(doc.GetText(MaxCharsPerGetText, stripControlCodes=True))
```

C++17

```

1  #include <iostream>
2  #include <string>
3  #include "DocumentFiltersObjects.h"
4
5  int main() {
6      try {
7          // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8          Hyland::DocFilters::Api api;
9
10         // Initialize the DocumentFilters object with license and path
11         std::string license = "YOUR_LICENSE_KEY_HERE";
12         std::string path = ".";
13         api.Initialize(license, path);
14
15         // Get an extractor for the specified file
16         Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.doc");
17
18         // Open the document with BodyAndMeta flag
19         doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
20
21         // Read and print the text content
22         while (!doc.getEOF()) {
23             std::wstring text = doc.getText(4096);
24             std::wcout << text << std::endl;
25         }
26
27         // Close the document
28         doc.Close();
29
30     } catch (const std::exception& ex) {
31         std::cerr << "Error: " << ex.what() << std::endl;
32     }
33     return 0;
34 }

```

ADDITIONAL INFORMATION

Call this method when finished working with the document to release its resources. The method will be internally called when the instance itself is released. Calling this method on closed documents has no effect.

SEE ALSO

- [Extractor Interface](#)
- [Open method](#)

Extractor::Compare method

The Compare method allows you to compare two documents returning the differences.

OVERLOADS

<code>Compare(Extractor, CompareSettings)</code>	Compare this document to another document.
<code>Compare(CompareDocumentSettings, Extractor, CompareDocumentSettings, CompareSettings)</code>	Compare this document to another document.
<code>Compare(CompareDocumentSettings, CompareDocumentSource, CompareSettings)</code>	Compare this document to another document.
<code>Compare(Extractor, System.Drawing.RectangleF, CompareSettings)</code>	Compare this document to another document.

COMPARE(EXTRACTOR, COMPARESETTINGS)

Prototype

.NET

```
CompareResults Compare(Extractor otherDocument, CompareSettings settings)
```

JAVA

```
CompareResults Compare(Extractor otherDocument, CompareSettings settings) throws IGRException;
```

PYTHON

```
def Compare(self, otherDocument: Extractor, settings: CompareSettings) -> CompareResults
```

C++17

```
CompareResults Compare(Extractor otherDocument, CompareSettings settings)
```

Parameters

- `otherDocument` : `Extractor` : Provide the other document to compare.
- `settings` : `CompareSettings` : The settings for the comparison.

Return value

`CompareResults` : CompareResults containing the results of the comparison.

COMPARE(COMPAREDOCUMENTSETTINGS, EXTRACTOR, COMPAREDOCUMENTSETTINGS, COMPARESETTINGS)**Prototype****.NET**

```
CompareResults Compare(CompareDocumentSettings thisDocSettings, Extractor otherDocument,
    CompareDocumentSettings otherDocSettings, CompareSettings settings)
```

JAVA

```
CompareResults Compare(CompareDocumentSettings thisDocSettings, Extractor otherDocument,
    CompareDocumentSettings otherDocSettings, CompareSettings settings) throws IGRException;
```

PYTHON

```
def Compare(self, thisDocSettings: CompareDocumentSettings, otherDocument: Extractor, otherDocSettings:
    CompareDocumentSettings, settings: CompareSettings) -> CompareResults
```

C++17

```
CompareResults Compare(CompareDocumentSettings thisDocSettings, Extractor otherDocument,
    CompareDocumentSettings otherDocSettings, CompareSettings settings)
```

Parameters

thisDocSettings : [CompareDocumentSettings](#) : The settings that apply to this left/original document in the comparison.

otherDocument : [Extractor](#) : Provide the other document to compare.

otherDocSettings : [CompareDocumentSettings](#) : The settings that apply to this right/revised document in the comparison.

settings : [CompareSettings](#) : The settings for the comparison.

Return value

[CompareResults](#) : CompareResults containing the results of the comparison.

COMPARE(COMPAREDOCUMENTSETTINGS, COMPAREDOCUMENTSOURCE, COMPARESETTINGS)**Prototype****.NET**

```
CompareResults Compare(CompareDocumentSettings thisDocSettings, CompareDocumentSource otherDocument,
    CompareSettings settings)
```

JAVA

```
CompareResults Compare(CompareDocumentSettings thisDocSettings, CompareDocumentSource otherDocument,
CompareSettings settings) throws IGRException;
```

PYTHON

```
def Compare(self, thisDocSettings: CompareDocumentSettings, otherDocument: CompareDocumentSource, settings:
CompareSettings) -> CompareResults
```

C++17

```
CompareResults Compare(CompareDocumentSettings thisDocSettings, CompareDocumentSource otherDocument,
CompareSettings settings)
```

Parameters

`thisDocSettings`: `CompareDocumentSettings` : The settings that apply to this left/original document in the comparison.

`otherDocument`: `CompareDocumentSource` : Provide the other document to compare.

`settings`: `CompareSettings` : The settings for the comparison.

Return value

`CompareResults` : CompareResults containing the results of the comparison.

COMPARE(EXTRACTOR, SYSTEM.DRAWING.RECTANGLEF, COMPARESETTINGS)

Prototype

.NET

```
CompareResults Compare(Extractor otherDocument, System.Drawing.RectangleF margins, CompareSettings settings)
```

C++17

```
CompareResults Compare(Extractor otherDocument, Hyland.DocFilters.RectF margins, CompareSettings settings)
```

Parameters

`otherDocument`: `Extractor` : Provide the other document to compare.

`margins`: `System.Drawing.RectangleF` : The top, left, right and bottom margins.

`settings`: `CompareSettings` : The settings for the comparison.

Return value

`CompareResults` : CompareResults containing the results of the comparison.

SAMPLE CODE

.NET

```

1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using (var doc1 = api.OpenExtractor("original.docx", OpenMode.Paginated))
7 using (var doc2 = api.OpenExtractor("revision.docx", OpenMode.Paginated))
8 using (var compare = doc1.Compare(doc2))
9 {
10     while (compare.MoveNext())
11     {
12         var diff = compare.Current;
13         // work with diff...
14     }
15 }

```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class DocumentComparison {
4
5     public static void main(String[] args) {
6         try {
7             // Create an Api object, which is an alias for DocumentFilters
8             DocumentFilters api = new DocumentFilters();
9
10            // Initialize the DocumentFilters object
11            api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
12
13            // Open the documents using OpenExtractor in Paginated mode
14            Extractor doc1 = api.OpenExtractor("original.docx", OpenMode.Paginated, OpenType.Auto, "");
15            Extractor doc2 = api.OpenExtractor("revision.docx", OpenMode.Paginated, OpenType.Auto, "");
16
17            // Compare the documents
18            CompareResults compare = doc1.Compare(doc2, new CompareSettings());
19
20            // Iterate through the differences
21            while (compare.MoveNext()) {
22                CompareResultDifference diff = compare.Current();
23
24                // Work with the diff object
25                // ... Your code to process the differences goes here ...
26
27                // Example: Print the type and text of the difference
28                System.out.println("Difference Type: " + diff.GetType());
29                System.out.println("Difference Text: " + diff.GetText());
30            }
31
32            // Close the documents (not explicitly shown in the C# sample, but good practice)
33            doc1.Close();
34            doc2.Close();
35
36        } catch (IGRException ex) {
37            System.err.println("Error: " + ex.getMessage());
38        }
39    }
40 }

```

PYTHON

```
1 from DocumentFilters import DocumentFilters, OpenMode, CompareSettings
2
3 try:
4     # Create an Api object (an alias for DocumentFilters)
5     api = DocumentFilters.Api()
6
7     # Initialize the DocumentFilters object
8     api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
9
10    # Open the documents using OpenExtractor in Paginated mode
11    doc1 = api.OpenExtractor(filename="original.docx", mode=OpenMode.Paginated)
12    doc2 = api.OpenExtractor(filename="revision.docx", mode=OpenMode.Paginated)
13
14    # Compare the documents
15    compare = doc1.Compare(doc2, CompareSettings())
16
17    # Iterate through the differences
18    while compare.MoveNext():
19        diff = compare.Current
20
21        # Work with the diff object
22        # ... Your code to process the differences goes here ...
23
24        # Example: Print the type and text of the difference
25        print(f"Difference Type: {diff.GetType()}")
26        print(f"Difference Text: {diff.GetText()}")
27
28    # Close the documents (not explicitly shown in the C# sample, but good practice)
29    doc1.Close()
30    doc2.Close()
31
32 except DocumentFilters.IGRException as ex:
33     print("Error:", ex)
```

C++17

```

1  #include <iostream>
2  #include <string>
3  #include "DocumentFiltersObjects.h"
4
5  int main() {
6      try {
7          // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8          Hyland::DocFilters::Api api;
9
10         // Initialize the DocumentFilters object
11         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
12
13         // Open the documents using OpenExtractor in Paginated mode
14         Hyland::DocFilters::Extractor doc1 = api.OpenExtractor("original.docx",
Hyland::DocFilters::OpenMode::Paginated);
15         Hyland::DocFilters::Extractor doc2 = api.OpenExtractor("revision.docx",
Hyland::DocFilters::OpenMode::Paginated);
16
17         // Compare the documents
18         Hyland::DocFilters::CompareResults compare = doc1.Compare(doc2);
19
20         // Iterate through the differences
21         while (compare.MoveNext()) {
22             Hyland::DocFilters::CompareResultDifference diff = compare.getCurrent();
23
24             // Work with the diff object
25             // ... Your code to process the differences goes here ...
26
27             // Example: Print the type and text of the difference
28             std::wcout << L"Difference Type: " << (int)diff.getType() << std::endl;
29             std::wcout << L"Difference Text: " << diff.getText() << std::endl;
30         }
31
32         // Close the documents (not explicitly shown in the C# sample, but good practice)
33         doc1.Close();
34         doc2.Close();
35
36     } catch (const std::exception& ex) {
37         std::cerr << "Error: " << ex.what() << std::endl;
38         return 1; // Indicate an error occurred
39     }
40
41     return 0;
42 }

```

SEE ALSO

- [CompareDocumentSettings interface](#)
- [Extractor interface](#)

Extractor::CopyTo method

The CopyTo method extracts the binary content of the sub-document to a file.

OVERLOADS

CopyTo(string)	Copy the the binary content of the sub-document to a file.
CopyTo(stream)	Copy the the binary content of the sub-document to a stream.

COPYTO(STRING)

Prototype

.NET

```
void CopyTo(string filename)
```

JAVA

```
void CopyTo(string filename) throws IGRException;
```

PYTHON

```
def CopyTo(self, filename: string) -> void
```

C++17

```
void CopyTo(const std::wstring& filename)
```

Parameters

filename: string : Path to a file where the binary content of the sub-document will be written.

COPYTO(STREAM)

Prototype

.NET

```
void CopyTo(System.IO.Stream stream)
```

JAVA

```
void CopyTo(IGRStream stream) throws IGRException;
```

PYTHON

```
def CopyTo(self, stream: Stream) -> void
```

C++17

```
void CopyTo(Stream& stream);  
void CopyTo(std::ostream& Stream);
```

Parameters

stream: stream : Stream where the binary content of the sub-document will be written.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;  
2  
3 var api = new Hyland.DocumentFilters.Api();  
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");  
5  
6 using var extractor = api.GetExtractor("archive.zip");  
7  
8 foreach (var subFile in extractor.SubFiles)  
9 {  
10     using (subFile)  
11         subFile.CopyTo("subfile_" + subFile.Name);  
12 }
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2 import java.io.FileOutputStream;
3 import java.io.IOException;
4
5 public class ExtractSubfiles {
6     public static void main(String[] args) {
7         try {
8             // Initialize the API
9             DocumentFilters docfilters = new DocumentFilters();
10            docfilters.Initialize("YOUR_LICENSE_KEY_HERE", ".");
11
12            // Open the archive file
13            Extractor extractor = docfilters.GetExtractor("archive.zip");
14            extractor.Open(isys_docfiltersConstants.IGR_BODY_AND_META, "");
15
16            // Create the output directory if it doesn't exist
17            String outputDir = "extracted_files";
18            new java.io.File(outputDir).mkdirs();
19
20            // Enumerate subfiles and copy them to disk
21            SubFilesEnumerator subFilesEnum = extractor.getSubFiles();
22            while (subFilesEnum.MoveNext()) {
23                SubFile subFile = subFilesEnum.Current();
24
25                // Note: Your application should sanitize the filename to prevent path traversal attacks
26                String outputFilename = "subfile_" + subFile.getName();
27                String outputPath = outputDir + "/" + outputFilename;
28
29                try (FileOutputStream outputStream = new FileOutputStream(outputPath)) {
30                    subFile.CopyTo(outputStream);
31                }
32
33                System.out.println("Extracted: " + outputFilename);
34            }
35            subFilesEnum.Close();
36
37        } catch (IGRException e) {
38            System.err.println("Error: " + e.getMessage());
39        } catch (IOException e) {
40            System.err.println("Error writing file: " + e.getMessage());
41        }
42    }
43 }
```


PYTHON

```
1 import DocumentFilters as df
2 import os
3
4 try:
5     # Initialize the API
6     docfilters = df.Api()
7     docfilters.Initialize("YOUR_LICENSE_KEY_HERE", ".")
8
9     # Open the archive file
10    extractor = docfilters.GetExtractor(filename="archive.zip")
11    extractor.Open(mode=df.OpenMode.BodyAndMeta)
12
13    # Create the output directory if it doesn't exist
14    output_dir = "extracted_files"
15    os.makedirs(output_dir, exist_ok=True)
16
17    # Enumerate subfiles and copy them to disk
18    for subfile in extractor.SubFiles:
19        # Note: Your application should sanitize the filename to prevent path traversal attacks
20        output_filename = f"subfile_{subfile.Name}"
21        output_path = os.path.join(output_dir, output_filename)
22
23        with open(output_path, 'wb') as f:
24            subfile.CopyTo(f)
25
26        print(f"Extracted: {output_filename}")
27
28 except df.IGRException as e:
29    print(f"Error: {e}")
```

C++17

```

1  #include "DocumentFiltersObjects.h"
2  #include <iostream>
3  #include <filesystem>
4
5  using namespace Hyland::DocFilters;
6
7  int main() {
8      try {
9          // Initialize the API
10         DocumentFilters docfilters("YOUR_LICENSE_KEY_HERE", ".");
11
12         // Open the archive file
13         Extractor extractor = docfilters.GetExtractor("archive.zip");
14         extractor.Open(OpenMode::Text, IGR_BODY_AND_META);
15
16         // Enumerate subfiles and copy them to disk
17         for (auto subFile : extractor.subfiles()) {
18             // Construct output filename
19             // Note: Your application should sanitize the filename to prevent path traversal attacks
20             std::string outputFilename = "subfile_" + w_to_u8(subFile.getName());
21
22             // Create the output directory if it doesn't exist
23             std::filesystem::create_directory("extracted_files");
24
25             // Construct the full output path
26             std::string outputPath = "extracted_files/" + outputFilename;
27
28             // Copy the subfile to disk
29             subFile.CopyTo(u8_to_w(outputPath));
30
31             std::cout << "Extracted: " << outputFilename << std::endl;
32         }
33     } catch (const std::exception& e) {
34         std::cerr << "Error: " << e.what() << std::endl;
35     }
36     return 0;
37 }

```

SEE ALSO

- [Extractor Interface](#)

Extractor::EOF property

The EOF property is only valid for documents where the [SupportsText](#) property is TRUE. The [EOF](#) property will be set to TRUE when no more text can be extracted from the document with calls to [GetText](#). If the document needs to be re-read, call [Close](#) and [Open](#) first.

.NET

```
bool EndOfStream { get; }
```

JAVA

```
boolean getEOF() throws IGREException;
```

PYTHON

```
@property
def EOF(self) -> bool
```

C++17

```
bool getEOF() const;
```

COM

```
[propget] HRESULT EOF([out, retval] VARIANT_BOOL *result);
```

RETURN VALUE

bool : Indicates if more text can be read from the file.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc"); // (2) Create an instance of the Extractor for a file
8
9 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta); // (3) Open the document for reading in text-mode
10
11 while (!doc.EndOfStream) // (4) Read the document in 4KB chunks and write it to the console
12 {
13     var text = doc.GetText(4096);
14     Console.Out.WriteLine(text);
15 }
16
17 doc.Close();
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.doc")) {
11            doc.Open(IGR_BODY_AND_META);
12
13            while (!doc.getEOF()) {
14                string text = doc.GetText(4096);
15                System.out.println(text);
16            }
17        }
18    }
19 }
```

PYTHON

```
1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.doc") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     while not doc.getEOF():
10         output.write(doc.GetText(MaxCharsPerGetText, stripControlCodes=True))
```

C++17

```

1  #include <iostream>
2  #include <string>
3  #include "DocumentFiltersObjects.h"
4
5  int main() {
6      try {
7          // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8          Hyland::DocFilters::Api api;
9
10         // Initialize the DocumentFilters object with license and path
11         std::string license = "YOUR_LICENSE_KEY_HERE";
12         std::string path = ".";
13         api.Initialize(license, path);
14
15         // Get an extractor for the specified file
16         Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.doc");
17
18         // Open the document with BodyAndMeta flag
19         doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
20
21         // Read and print the text content
22         while (!doc.getEOF()) {
23             std::wstring text = doc.getText(4096);
24             std::wcout << text << std::endl;
25         }
26
27         // Close the document
28         doc.Close();
29
30     } catch (const std::exception& ex) {
31         std::cerr << "Error: " << ex.what() << std::endl;
32     }
33     return 0;
34 }

```

ADDITIONAL INFORMATION

Accessing this property will open the document. Call the Close method when finished.

SEE ALSO

- [Extractor Interface](#)
- [Extractor::GetText method](#)

Extractor::FileType property

The FileType property is the document format code, as listed in [Document Format Codes](#) chart on page . The function is overloaded to be able to return the format name as a string.

.NET

```
int FileType { get; }
```

JAVA

```
int getFileType() throws IGRException;
```

PYTHON

```
@property  
def FileType(self) -> int
```

C++17

```
int getFileType() const;
```

RETURN VALUE

`int` : Integer containing the format code

SEE ALSO

- [Extractor Interface](#)
- [Document Format Codes](#)

Extractor::GetFirstImage method

The GetFirstImage method obtains a [SubFile](#) object representing the first embedded image of the current document when converting using classic HTML.

PROTOTYPE

.NET

```
SubFile GetFirstImage()
```

JAVA

```
SubFile GetFirstImage() throws IGRException;
```

PYTHON

```
def GetFirstImage(self) -> SubFile
```

C++17

```
SubFile GetFirstImage()
```

C++11

```
SubFile* GetFirstImage()
```

COM

```
HRESULT GetFirstImage([out, retval] SubFile* *result)
```

RETURN VALUE

[SubFile](#) : Returns a [SubFile](#) object for the first image or NULL if the document does not contain sub-documents.

SAMPLE CODE

.NET

```

1  using Hyland.DocumentFilters;
2  using System;
3
4  class ExtractTextAndImages
5  {
6      static void Main(string[] args)
7      {
8          try
9          {
10             // Initialize the API
11             DocumentFilters docFilters = new DocumentFilters();
12             docFilters.Initialize("YOUR_LICENSE_KEY_HERE", "."); // Replace with your license key and
resource path
13
14             // Open the HTML file in ClassicHTML mode
15             Extractor extractor = docFilters.GetExtractor("input.docx"); // Replace with your HTML file path
16             extractor.Open(OpenMode.ClassicHTML, OpenType.BodyAndMeta, "");
17
18             // Extract the text from the HTML
19             string extractedText = extractor.GetText(0xffff);
20             Console.WriteLine("Extracted Text:\n" + extractedText);
21
22             // Enumerate and process the images using GetFirstImage and GetNextImage
23             Console.WriteLine("\nImages:");
24             SubFile image = extractor.GetFirstImage();
25             while (image != null)
26             {
27                 Console.WriteLine("- ID: " + image.ID);
28                 Console.WriteLine("  Name: " + image.Name);
29                 Console.WriteLine("  Size: " + image.FileSize + " bytes");
30
31                 // Additional image processing can be done here
32
33                 image = extractor.GetNextImage();
34             }
35
36             // Close the extractor
37             extractor.Close(true);
38         }
39         catch (IGRException e)
40         {
41             Console.WriteLine("Error: " + e.Message);
42         }
43     }
44 }

```


JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class ExtractTextAndImages {
4     public static void main(String[] args) {
5         try {
6             // Initialize DocumentFilters API
7             DocumentFilters docFilters = new DocumentFilters();
8             docFilters.Initialize("YOUR_LICENSE_KEY_HERE", "."); // Replace with your actual license key
9             and resource path. This is not from the given sources.
10
11             // Open the HTML file in ClassicHTML mode
12             Extractor extractor = docFilters.GetExtractor("input.docx"); // Replace "input.html" with your
13             HTML file's path. This is not from the given sources.
14             extractor.Open(OpenMode.ClassicHTML, OpenType.BodyAndMeta, ""); // Open the extractor in
15             ClassicHTML mode. This is not from the given sources.
16
17             // Extract text from the HTML
18             String extractedText = extractor.GetText(0xffff);
19             System.out.println("Extracted Text:\n" + extractedText);
20
21             // Enumerate and process the images
22             System.out.println("\nImages:");
23             SubFile image = extractor.GetFirstImage();
24             while (image != null) { // This condition for continuing the loop is not from the given sources.
25                 System.out.println("- ID: " + image.GetID());
26                 System.out.println("  Name: " + image.GetName());
27                 System.out.println("  Size: " + image.GetSize() + " bytes");
28
29                 // Additional image processing can be done here. This is not from the given sources.
30
31                 image = extractor.GetNextImage();
32             }
33
34             // Close the extractor
35             extractor.Close();
36         } catch (IGRException e) {
37             System.err.println("Error: " + e.getMessage());
38         }
39     }
40 }
```

PYTHON

```
1 import DocumentFilters
2
3 # Initialize the API
4 doc_filters = DocumentFilters.DocumentFilters()
5 doc_filters.Initialize("YOUR_LICENSE_KEY_HERE", ".") # Replace with your license key and resource path
6
7 # Open the HTML file in ClassicHTML mode
8 extractor = doc_filters.GetExtractor(filename="input.docx") # Replace with your HTML file path
9 extractor.Open(mode=DocumentFilters.OpenMode.ClassicHTML, options="")
10
11 # Extract the text from the HTML
12 extracted_text = extractor.GetText(0xffff)
13 print("Extracted Text:\n" + extracted_text)
14
15 # Enumerate and process the images
16 print("\nImages:")
17 image = extractor.GetFirstImage()
18 while image:
19     print(f"- ID: {image.ID}")
20     print(f"  Name: {image.Name}")
21     print(f"  Size: {image.FileSize} bytes")
22
23     # Additional image processing can be done here
24
25     image = extractor.GetNextImage()
26
27 # Close the extractor
28 extractor.Close(True)
```

C++17

```

1  #include <iostream>
2  #include <fstream>
3  #include "DocumentFiltersObjects.h"
4
5  using namespace Hyland::DocFilters;
6
7  int main() {
8      try {
9          // Initialize DocumentFilters API
10         DocumentFilters docFilters;
11         docFilters.Initialize(L"YOUR_LICENSE_KEY_HERE", L"."); // Replace with your license and path
12
13         // Open HTML file in ClassicHTML mode
14         Extractor extractor = docFilters.OpenExtractor("input.docx", OpenMode::ClassicHtml, 0, L "");
15         // Replace with your HTML file path
16
17         // Extract the text from the HTML
18         std::wstring extractedText = extractor.getText(0xffff, false);
19         std::wcout << L"Extracted Text:\n" << extractedText << std::endl;
20
21         // Enumerate images using GetFirstImage and GetNextImage
22         std::wcout << L"\nImages:\n";
23         for (auto&& image : extractor.images()) {
24             std::wcout << L"  ID: " << image.id() << std::endl;
25             std::wcout << L"  Name: " << image.getName() << std::endl;
26             std::wcout << L"  Size: " << image.getSize() << L" bytes" << std::endl;
27         }
28
29         // Close the extractor
30         extractor.Close();
31     }
32     catch (DocumentFilters::Error& e) {
33         std::wcerr << L"Error: " << e.what() << std::endl;
34         return 1; // Indicate an error occurred
35     }
36     return 0;
37 }

```

ADDITIONAL INFORMATION

This method will implicitly open the document. Call the [Close](#) method when finished.

C# and Python support enumerating and accessing images through the Images property.

SEE ALSO

- [Extractor Interface](#)
- [Extractor::GetNextImage method](#)

Extractor::GetFirstPage method

The `GetFirstPage` method returns the first page object of an opened document. The document must be opened in image mode (`IGR_FORMAT_IMAGE`). It is also possible to iterate over pages with [GetPageCount](#) and [GetPage](#) methods.

PROTOTYPE

.NET

```
Page GetFirstPage()
```

JAVA

```
Page GetFirstPage() throws IGRException;
```

PYTHON

```
def GetFirstPage(self) -> Page
```

C++17

```
Page GetFirstPage()
```

C++11

```
Page* GetFirstPage()
```

COM

```
HRESULT GetFirstPage([out, retval] Page* *result)
```

RETURN VALUE

[Page](#) : Returns a [Page](#) object representing the first page of the document. If there are no pages available or the document was not opened in high-definition (HD) image mode, the method returns `NULL`.

SAMPLE CODE

.NET

```

1 using System.Linq;
2 using Hyland.DocumentFilters;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
8
9 foreach (var pageIter in doc.Pages.Select((page, index) => new { page, index }))
10 {
11     using (pageIter.page)
12     {
13         using var canvas = api.MakeOutputCanvas($"page-{pageIter.index+1}.png", CanvasType.PNG);
14         canvas.RenderPage(pageIter.page);
15     }
16 }

```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         DocumentFilters df = new DocumentFilters();
6         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
7
8         try (Extractor doc = df.GetExtractor("filename.doc")) {
9             doc.Open(igrs_constants.IGR_FORMAT_IMAGE, "");
10
11             for (int i = 0, c = doc.getPageCount(); i < c; ++i) {
12                 try (Page page = doc.getPage(i)) {
13                     try (Canvas canvas = df.MakeOutputCanvas(String.format("page-%d.png", i+1),
14 isys_docfilters_constants.IGR_DEVICE_IMAGE_PNG, "")) {
15                         canvas.RenderPage(page);
16                     }
17                 }
18             }
19         }
20 }

```

PYTHON

```

1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.OpenExtractor("filename.doc", mode=IGR_FORMAT_IMAGE) as doc:
7     for page in doc.Pages:
8         with page:
9             with api.MakeOutputCanvas(f"page-{page.PageIndex+1}.png", canvasType=IGR_DEVICE_IMAGE_PNG) as
10 canvas:
11             canvas.RenderPages(doc)

```

C++17

```

1  #include <DocumentFiltersObjects.h>
2  #include <sstream>
3
4  int main() {
5      try {
6          // Create and initialize the API object
7          Hyland::DocFilters::Api api;
8          api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10         // Open the input file
11         Hyland::DocFilters::Extractor doc = api.OpenExtractor("filename.doc",
Hyland::DocFilters::OpenMode::Paginated);
12
13         // Render all pages to the output
14         for (auto page : doc.pages()) {
15             // Create a unique output filename
16             std::stringstream outputFilename;
17             outputFilename << "output_" << (page.getIndex() + 1) << ".png";
18
19             // Create the output canvas and render the page
20             Hyland::DocFilters::Canvas canvas = api.MakeOutputCanvas(outputFilename.str(),
Hyland::DocFilters::CanvasType::PNG);
21             canvas.RenderPage(page);
22         }
23     } catch (const std::exception& e) {
24         std::cerr << "Error: " << e.what() << std::endl;
25         return 1; // Indicate an error
26     }
27
28     return 0; // Successful execution
29 }

```

ADDITIONAL INFORMATION

Call the Close method when finished working with the page to release its resources. A page will be internally freed when the instance itself is released, however, this can be at indeterminate times in some garbage collected languages such as .NET and Java.

SEE ALSO

- [Extractor Interface](#)
- [Extractor::GetNextPage method](#)

Extractor::GetFirstSubFile method

The GetFirstSubFile method obtains a [SubFile](#) object representing the first sub-document of the current document.

PROTOTYPE

.NET

```
SubFile GetFirstSubFile()
```

JAVA

```
SubFile GetFirstSubFile() throws IGRException;
```

PYTHON

```
def GetFirstSubFile(self) -> SubFile
```

C++17

```
SubFile GetFirstSubFile()
```

C++11

```
SubFile* GetFirstSubFile()
```

COM

```
HRESULT GetFirstSubFile([out, retval] SubFile* *result)
```

RETURN VALUE

[SubFile](#) : Returns a [SubFile](#) object for the first subfile or NULL if the document does not contain sub-documents.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 // Get the first subfile
11 SubFile subfile = doc.GetFirstSubFile();
12
13 while (subfile != null)
14 {
15     // act on subfile
16     Console.Out.WriteLine("Name: " + subfile.Name);
17     Console.Out.WriteLine("ID: " + subfile.ID);
18     Console.Out.WriteLine("Date: " + subfile.FileDate);
19     Console.Out.WriteLine("Size: " + subfile.FileSize);
20
21     // Get the next subfile
22     subfile = doc.GetNextSubFile();
23 }
```


JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class Main {
4     public static void main(String[] args) {
5         try {
6             // Initialize Document Filters API
7             DocumentFilters api = new DocumentFilters();
8             api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10            // Get the extractor
11            Extractor doc = api.GetExtractor("filename.doc");
12            doc.Open(OpenType.BodyAndMeta);
13
14            // Get the first SubFile
15            SubFile subfile = doc.GetFirstSubFile();
16
17            // Check if a subfile was found
18            if (subfile != null) {
19                do {
20                    // Act on the subfile
21                    System.out.println("Name: " + subfile.getName());
22                    System.out.println("ID: " + subfile.getID());
23                    System.out.println("Date: " + subfile.getFileDate());
24                    System.out.println("Size: " + subfile.getFileSize());
25
26                    // Get the next subfile
27                    subfile = doc.GetNextSubFile();
28                } while (subfile != null);
29            } else {
30                System.out.println("No subfiles found.");
31            }
32
33            // Close the extractor
34            doc.Close();
35        } catch (IGRException e) {
36            System.err.println("Error: " + e.getMessage());
37        }
38    }
39 }
```

PYTHON

```
1 import DocumentFilters
2
3 # Initialize the API. Replace with your license and resource path.
4 # Not from the sources
5 api = DocumentFilters.Api()
6 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
7
8 # Get the extractor
9 doc = api.GetExtractor(filename="filename.doc")
10 doc.Open(DocumentFilters.OpenType.BodyAndMeta)
11
12 # Get the first subfile
13 subfile = doc.GetFirstSubFile()
14
15 # Check if a subfile was found
16 if subfile: # If a subfile exists
17     while subfile:
18         try:
19             # Process the subfile
20             print("Name:", subfile.GetName())
21             print("ID:", subfile.GetID())
22             print("Date:", subfile.GetFileDate())
23             print("Size:", subfile.GetFileSize())
24
25             # Get the next subfile
26             subfile = doc.GetNextSubFile()
27         except DocumentFilters.IGREException as e:
28             print("Error:", e.message)
29     else:
30         print("No subfiles found.")
```

ADDITIONAL INFORMATION

This method will implicitly open the document. Call the [Close](#) method when finished.

C#, Java, Python and C++17 support enumerating and accessing subfiles through the [SubFiles](#) property.

SEE ALSO

- [Extractor Interface](#)
- [Extractor::GetNextSubFile](#) method

Extractor::GetHashMD5 method

The getHashMD5 methods obtain a string representing the calculated hash of the current document for unique identification.

PROTOTYPE

.NET

```
string getHashMD5()
```

JAVA

```
string getHashMD5() throws IGRException;
```

PYTHON

```
def getHashMD5(self) -> string
```

C++17

```
std::wstring getHashMD5()
```

C++11

```
std::wstring getHashMD5()
```

COM

```
HRESULT getHashMD5([out, retval] BSTR *result)
```

RETURN VALUE

string : The hash string for the input (binary) document.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.GetExtractor("filename.doc");
7 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
8
9 var md5 = doc.getHashMD5();
10 var sha1 = doc.getHashSHA1();
```

Extractor::GetHashSHA1 method

The getHashSHA1 methods obtain a string representing the calculated hash of the current document for unique identification.

PROTOTYPE

.NET

```
string getHashSHA1()
```

JAVA

```
string getHashSHA1() throws IGRException;
```

PYTHON

```
def getHashSHA1(self) -> string
```

C++17

```
std::wstring getHashSHA1()
```

C++11

```
std::wstring getHashSHA1()
```

COM

```
HRESULT getHashSHA1([out, retval] BSTR *result)
```

RETURN VALUE

string : The hash string for the input (binary) document.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.GetExtractor("filename.doc");
7 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
8
9 var md5 = doc.getHashMD5();
10 var sha1 = doc.getHashSHA1();
```

Extractor::GetNextImage method

The GetNextImage method obtains a [SubFile](#) object representing the next embedded image of the current document when converting using classic HTML.

PROTOTYPE

.NET

```
SubFile GetNextImage()
```

JAVA

```
SubFile GetNextImage() throws IGRException;
```

PYTHON

```
def GetNextImage(self) -> SubFile
```

C++11

```
SubFile* GetNextImage()
```

COM

```
HRESULT GetNextImage([out, retval] SubFile* *result)
```

RETURN VALUE

[SubFile](#) : Returns a [SubFile](#) object for the first image or NULL if the document does not contain sub-documents.

SAMPLE CODE

.NET

```

1  using Hyland.DocumentFilters;
2  using System;
3
4  class ExtractTextAndImages
5  {
6      static void Main(string[] args)
7      {
8          try
9          {
10             // Initialize the API
11             DocumentFilters docFilters = new DocumentFilters();
12             docFilters.Initialize("YOUR_LICENSE_KEY_HERE", "."); // Replace with your license key and
resource path
13
14             // Open the HTML file in ClassicHTML mode
15             Extractor extractor = docFilters.GetExtractor("input.docx"); // Replace with your HTML file path
16             extractor.Open(OpenMode.ClassicHTML, OpenType.BodyAndMeta, "");
17
18             // Extract the text from the HTML
19             string extractedText = extractor.GetText(0xffff);
20             Console.WriteLine("Extracted Text:\n" + extractedText);
21
22             // Enumerate and process the images using GetFirstImage and GetNextImage
23             Console.WriteLine("\nImages:");
24             SubFile image = extractor.GetFirstImage();
25             while (image != null)
26             {
27                 Console.WriteLine("- ID: " + image.ID);
28                 Console.WriteLine("  Name: " + image.Name);
29                 Console.WriteLine("  Size: " + image.FileSize + " bytes");
30
31                 // Additional image processing can be done here
32
33                 image = extractor.GetNextImage();
34             }
35
36             // Close the extractor
37             extractor.Close(true);
38         }
39         catch (IGRException e)
40         {
41             Console.WriteLine("Error: " + e.Message);
42         }
43     }
44 }

```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class ExtractTextAndImages {
4     public static void main(String[] args) {
5         try {
6             // Initialize DocumentFilters API
7             DocumentFilters docFilters = new DocumentFilters();
8             docFilters.Initialize("YOUR_LICENSE_KEY_HERE", "."); // Replace with your actual license key
and resource path. This is not from the given sources.
9
10            // Open the HTML file in ClassicHTML mode
11            Extractor extractor = docFilters.GetExtractor("input.docx"); // Replace "input.html" with your
HTML file's path. This is not from the given sources.
12            extractor.Open(OpenMode.ClassicHTML, OpenType.BodyAndMeta, ""); // Open the extractor in
ClassicHTML mode. This is not from the given sources.
13
14            // Extract text from the HTML
15            String extractedText = extractor.GetText(0xffff);
16            System.out.println("Extracted Text:\n" + extractedText);
17
18            // Enumerate and process the images
19            System.out.println("\nImages:");
20            SubFile image = extractor.GetFirstImage();
21            while (image != null) { // This condition for continuing the loop is not from the given sources.
22                System.out.println("- ID: " + image.GetID());
23                System.out.println("  Name: " + image.GetName());
24                System.out.println("  Size: " + image.GetSize() + " bytes");
25
26                // Additional image processing can be done here. This is not from the given sources.
27
28                image = extractor.GetNextImage();
29            }
30
31            // Close the extractor
32            extractor.Close();
33        } catch (IGRException e) {
34            System.err.println("Error: " + e.getMessage());
35        }
36    }
37 }
```

PYTHON

```
1 import DocumentFilters
2
3 # Initialize the API
4 doc_filters = DocumentFilters.DocumentFilters()
5 doc_filters.Initialize("YOUR_LICENSE_KEY_HERE", ".") # Replace with your license key and resource path
6
7 # Open the HTML file in ClassicHTML mode
8 extractor = doc_filters.GetExtractor(filename="input.docx") # Replace with your HTML file path
9 extractor.Open(mode=DocumentFilters.OpenMode.ClassicHTML, options="")
10
11 # Extract the text from the HTML
12 extracted_text = extractor.GetText(0xffff)
13 print("Extracted Text:\n" + extracted_text)
14
15 # Enumerate and process the images
16 print("\nImages:")
17 image = extractor.GetFirstImage()
18 while image:
19     print(f"- ID: {image.ID}")
20     print(f"  Name: {image.Name}")
21     print(f"  Size: {image.FileSize} bytes")
22
23     # Additional image processing can be done here
24
25     image = extractor.GetNextImage()
26
27 # Close the extractor
28 extractor.Close(True)
```


C++17

```

1  #include <iostream>
2  #include <fstream>
3  #include "DocumentFiltersObjects.h"
4
5  using namespace Hyland::DocFilters;
6
7  int main() {
8      try {
9          // Initialize DocumentFilters API
10         DocumentFilters docFilters;
11         docFilters.Initialize(L"YOUR_LICENSE_KEY_HERE", L"."); // Replace with your license and path
12
13         // Open HTML file in ClassicHTML mode
14         Extractor extractor = docFilters.OpenExtractor("input.docx", OpenMode::ClassicHtml, 0, L "");
15         // Replace with your HTML file path
16
17         // Extract the text from the HTML
18         std::wstring extractedText = extractor.getText(0xffff, false);
19         std::wcout << L"Extracted Text:\n" << extractedText << std::endl;
20
21         // Enumerate images using GetFirstImage and GetNextImage
22         std::wcout << L"\nImages:\n";
23         for (auto& image : extractor.images()) {
24             std::wcout << L"  ID: " << image.id() << std::endl;
25             std::wcout << L"  Name: " << image.getName() << std::endl;
26             std::wcout << L"  Size: " << image.getSize() << L" bytes" << std::endl;
27         }
28
29         // Close the extractor
30         extractor.Close();
31     }
32     catch (DocumentFilters::Error& e) {
33         std::wcerr << L"Error: " << e.what() << std::endl;
34         return 1; // Indicate an error occurred
35     }
36     return 0;
37 }

```

ADDITIONAL INFORMATION

This method will implicitly open the document. Call the [Close](#) method when finished.

C# and Python support enumerating and accessing images through the Images property.

SEE ALSO

- [Extractor Interface](#)
- [Extractor::GetFirstImage method](#)

Extractor::GetNextPage method

The `GetNextPage` method returns the next page object of an opened document. The document must be opened in image mode (`IGR_FORMAT_IMAGE`). It is also possible to iterate over pages with [GetPageCount](#) and [GetPage](#) methods.

PROTOTYPE

.NET

```
Page GetNextPage()
```

JAVA

```
Page GetNextPage() throws IGRException;
```

PYTHON

```
def GetNextPage(self) -> Page
```

C++17

```
Page GetNextPage()
```

C++11

```
Page* GetNextPage()
```

COM

```
HRESULT GetNextPage([out, retval] Page* *result)
```

RETURN VALUE

[Page](#) : Returns a [Page](#) object representing the next page of the document. If there are no pages available or the document was not opened in high-definition (HD) image mode, the method returns `NULL`.

SAMPLE CODE

.NET

```

1 using System.Linq;
2 using Hyland.DocumentFilters;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
8
9 foreach (var pageIter in doc.Pages.Select((page, index) => new { page, index }))
10 {
11     using (pageIter.page)
12     {
13         using var canvas = api.MakeOutputCanvas($"page-{pageIter.index+1}.png", CanvasType.PNG);
14         canvas.RenderPage(pageIter.page);
15     }
16 }

```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         DocumentFilters df = new DocumentFilters();
6         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
7
8         try (Extractor doc = df.GetExtractor("filename.doc")) {
9             doc.Open(isys_docfiltersConstants.IGR_FORMAT_IMAGE, "");
10
11             for (int i = 0, c = doc.GetPageCount(); i < c; ++i) {
12                 try (Page page = doc.GetPage(i)) {
13                     try (Canvas canvas = df.MakeOutputCanvas(String.format("page-%d.png", i+1),
14 isys_docfiltersConstants.IGR_DEVICE_IMAGE_PNG, "")) {
15                         canvas.RenderPage(page);
16                     }
17                 }
18             }
19         }
20 }

```

PYTHON

```

1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.OpenExtractor("filename.doc", mode=IGR_FORMAT_IMAGE) as doc:
7     for page in doc.Pages:
8         with page:
9             with api.MakeOutputCanvas(f"page-{page.PageIndex+1}.png", canvasType=IGR_DEVICE_IMAGE_PNG) as
10 canvas:
11             canvas.RenderPages(doc)

```

C++17

```

1  #include <DocumentFiltersObjects.h>
2  #include <sstream>
3
4  int main() {
5      try {
6          // Create and initialize the API object
7          Hyland::DocFilters::Api api;
8          api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10         // Open the input file
11         Hyland::DocFilters::Extractor doc = api.OpenExtractor("filename.doc",
Hyland::DocFilters::OpenMode::Paginated);
12
13         // Render all pages to the output
14         for (auto page : doc.pages()) {
15             // Create a unique output filename
16             std::stringstream outputFilename;
17             outputFilename << "output_" << (page.getIndex() + 1) << ".png";
18
19             // Create the output canvas and render the page
20             Hyland::DocFilters::Canvas canvas = api.MakeOutputCanvas(outputFilename.str(),
Hyland::DocFilters::CanvasType::PNG);
21             canvas.RenderPage(page);
22         }
23     } catch (const std::exception& e) {
24         std::cerr << "Error: " << e.what() << std::endl;
25         return 1; // Indicate an error
26     }
27
28     return 0; // Successful execution
29 }

```

ADDITIONAL INFORMATION

Call the Close method when finished working with the page to release its resources. A page will be internally freed when the instance itself is released, however, this can be at indeterminate times in some garbage collected languages such as .NET and Java.

SEE ALSO

- [Extractor Interface](#)
- [Extractor::GetFirstPage](#) method

Extractor::GetNextSubFile method

The GetNextSubFile method obtains a SubFile object representing the next sub-document of the current document.

PROTOTYPE

.NET

```
SubFile GetNextSubFile()
```

JAVA

```
SubFile GetNextSubFile() throws IGREException;
```

PYTHON

```
def GetNextSubFile(self) -> SubFile
```

C++17

```
SubFile GetNextSubFile()
```

C++11

```
SubFile* GetNextSubFile()
```

COM

```
HRESULT GetNextSubFile([out, retval] SubFile* *result)
```

RETURN VALUE

SubFile : Returns a **SubFile** object for the next subfile or NULL if the document does not contain sub-documents.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 // Get the first subfile
11 SubFile subfile = doc.GetFirstSubFile();
12
13 while (subfile != null)
14 {
15     // act on subfile
16     Console.Out.WriteLine("Name: " + subfile.Name);
17     Console.Out.WriteLine("ID: " + subfile.ID);
18     Console.Out.WriteLine("Date: " + subfile.FileDate);
19     Console.Out.WriteLine("Size: " + subfile.FileSize);
20
21     // Get the next subfile
22     subfile = doc.GetNextSubFile();
23 }
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class Main {
4     public static void main(String[] args) {
5         try {
6             // Initialize Document Filters API
7             DocumentFilters api = new DocumentFilters();
8             api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10            // Get the extractor
11            Extractor doc = api.GetExtractor("filename.doc");
12            doc.Open(OpenType.BodyAndMeta);
13
14            // Get the first SubFile
15            SubFile subfile = doc.GetFirstSubFile();
16
17            // Check if a subfile was found
18            if (subfile != null) {
19                do {
20                    // Act on the subfile
21                    System.out.println("Name: " + subfile.getName());
22                    System.out.println("ID: " + subfile.getID());
23                    System.out.println("Date: " + subfile.getFileDate());
24                    System.out.println("Size: " + subfile.getFileSize());
25
26                    // Get the next subfile
27                    subfile = doc.GetNextSubFile();
28                } while (subfile != null);
29            } else {
30                System.out.println("No subfiles found.");
31            }
32
33            // Close the extractor
34            doc.Close();
35        } catch (IGRException e) {
36            System.err.println("Error: " + e.getMessage());
37        }
38    }
39 }
```

PYTHON

```
1 import DocumentFilters
2
3 # Initialize the API. Replace with your license and resource path.
4 # Not from the sources
5 api = DocumentFilters.Api()
6 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
7
8 # Get the extractor
9 doc = api.GetExtractor(filename="filename.doc")
10 doc.Open(DocumentFilters.OpenType.BodyAndMeta)
11
12 # Get the first subfile
13 subfile = doc.GetFirstSubFile()
14
15 # Check if a subfile was found
16 if subfile: # If a subfile exists
17     while subfile:
18         try:
19             # Process the subfile
20             print("Name:", subfile.GetName())
21             print("ID:", subfile.GetID())
22             print("Date:", subfile.GetFileDate())
23             print("Size:", subfile.GetFileSize())
24
25             # Get the next subfile
26             subfile = doc.GetNextSubFile()
27         except DocumentFilters.IGREException as e:
28             print("Error:", e.message)
29     else:
30         print("No subfiles found.")
```

ADDITIONAL INFORMATION

This method will implicitly open the document. Call the [Close](#) method when finished.

C#, Java, Python and C++17 support enumerating and accessing subfiles through the [SubFiles](#) property.

SEE ALSO

- [Extractor Interface](#)
- [Extractor::GetFirstSubFile](#) method

Extractor::GetPage method

The GetPage method returns the page at the given index, where the page index is 0-based. An exception is raised if the index is invalid.

PROTOTYPE

.NET

```
Page GetPage(int pageIndex)
```

JAVA

```
Page GetPage(int pageIndex) throws IGRException;
```

PYTHON

```
def GetPage(self, pageIndex: int) -> Page
```

C++17

```
Page GetPage(int pageIndex)
```

C++11

```
Page* GetPage(int pageIndex)
```

COM

```
HRESULT GetPage([in] int pageIndex, [out, retval] Page* *result)
```

PARAMETERS

`pageIndex: int` : The index of the page to return.

RETURN VALUE

`Page` : Returns a `Page` object for the requested page.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
7
8 for (int i = 0; i < doc.PageCount; i++)
9 {
10     using var page = doc.GetPage(i);
11     using var canvas = api.MakeOutputCanvas($"page-{i+1}.png", CanvasType.PNG);
12     canvas.RenderPage(page);
13 }
```

ADDITIONAL INFORMATION

Call the Close method when you have finished working with the page to release its resources. A page will be internally freed when the instance itself is released, however, this can be at indeterminate times in some garbage collected languages such as .NET and Java.

SEE ALSO

- [Extractor Interface](#)
- [Page interface](#)

Extractor::GetPageCount method

Returns the number of pages in the current document, the document must be opened in image mode for the page count to be populated.

PROTOTYPE

.NET

```
int GetPageCount()
```

JAVA

```
int GetPageCount() throws IGRException;
```

PYTHON

```
def GetPageCount(self) -> int
```

C++17

```
int GetPageCount()
```

RETURN VALUE

int : The number of pages in the document

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
7
8 for (int i = 0; i < doc.PageCount; i++)
9 {
10     using var page = doc.GetPage(i);
11     using var canvas = api.MakeOutputCanvas($"page-{i+1}.png", CanvasType.PNG);
12     canvas.RenderPage(page);
13 }
```

SEE ALSO

- [Extractor Interface](#)
- [Page interface](#)

Extractor::GetResourceStreamCallback property

Set a callback hook called when an rendering an embedded image.

.NET

```
Func<string, Stream> GetResourceCallback { get; set; }
```

JAVA

```
void setGetResourceStreamCallback(IGetExternalResourceCallback value) throws IGRException;
```

PYTHON

```
@property
def GetResourceStreamCallback(self) -> Callable[[str], IGRStream]

@GetResourceStreamCallback.setter
def GetResourceStreamCallback(self, value: Callable[[str], IGRStream])
```

C++17

```
void setGetResourceStreamCallback(approve_external_resource_callback_t value);
```

RETURN VALUE

Function

Extractor::GetRootBookmark method

The GetRootBookmark method returns a [Bookmark](#) node representing the top-most node of the bookmark hierarchy. The root bookmark only has Children data, it has no title or destination properties.

PROTOTYPE

.NET

```
Bookmark GetRootBookmark()
```

JAVA

```
Bookmark GetRootBookmark() throws IGREException;
```

PYTHON

```
def GetRootBookmark(self) -> Bookmark
```

C++17

```
Bookmark GetRootBookmark()
```

RETURN VALUE

[Bookmark](#) : [Bookmark](#) representing the root of the document.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var extractor = api.GetExtractor("word.docx");
7 extractor.Open(OpenMode.Paginated);
8
9 var bookmark = extractor.RootBookmark;
10 foreach (var child in bookmark.Children)
11 {
12     //
13 }
```

ADDITIONAL INFORMATION

Call the Close method when you have finished working with the page to release its resources. A page will be internally freed when the instance itself is released, however, this can be at indeterminate times in some garbage collected languages such as .NET and Java.

SEE ALSO

- [Extractor Interface](#)
- [Page interface](#)
- [Bookmark interface](#)

Extractor::GetSubFile method

The GetSubFile method obtains a [SubFile](#) object representing the nominated sub-file of the current document.

PROTOTYPE

.NET

```
SubFile GetSubFile(string id)
```

JAVA

```
SubFile GetSubFile(string id) throws IGRException;
```

PYTHON

```
def GetSubFile(self, id: string) -> SubFile
```

C++17

```
SubFile GetSubFile(const std::wstring& id)
```

C++11

```
SubFile* GetSubFile(std::wstring id)
```

COM

```
HRESULT GetSubFile([in] BSTR id, [out, retval] SubFile* *result)
```

PARAMETERS

id: string : An ID that was previously returned when enumerating sub files with [GetFirstSubFile](#) and [GetNextSubFile](#).

Note The sub file ID is not necessarily the same as its name.

RETURN VALUE

SubFile : A SubFile object for the nominated sub-document, or NULL if the document is not found.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.GetExtractor("archive.zip");
7 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
8
9 using var subfile = doc.GetSubFile("subfile.txt");
```

ADDITIONAL INFORMATION

The identifier string (ID) should be treated as an opaque identifier, and can only be reliably obtained by first iterating over an archive document and retrieving it via a call to [SubFile::getID\(\)](#). Some archive formats may yield identifiers that look like file system paths, but several do not. The value of the ID for a given sub-file may also vary between releases of the DocumentFilters API. Therefore, never try to formulate an ID. Always use IDs returned from [SubFile::getID\(\)](#).

This method will implicitly open the document. Call the [Close](#) method when finished.

Calling this method will not affect the “next” document that will be returned by [GetNextSubFile](#). Use the [GetFirstSubFile](#) / [GetNextSubFile](#) methods in an interleaved manner if required.

SEE ALSO

- [Extractor Interface](#)
- [SubFile::ID](#)

Extractor::GetText method

The GetText method extracts the next portion of text content from the document.

PROTOTYPE

.NET

```
string GetText(int maxLength)
```

JAVA

```
string GetText(int maxLength) throws IGREException;
```

PYTHON

```
def GetText(self, maxLength: int) -> string
```

C++17

```
std::wstring GetText(int maxLength)
```

PARAMETERS

`maxLength: int` : Maximum number of characters to be returned.

RETURN VALUE

`string` : Returns a string of up to MaxLength characters from the document.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc"); // (2) Create an instance of the Extractor for a file
8
9 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta); // (3) Open the document for reading in text-mode
10
11 while (!doc.EndOfStream) // (4) Read the document in 4KB chunks and write it to the console
12 {
13     var text = doc.GetText(4096);
14     Console.Out.WriteLine(text);
15 }
16
17 doc.Close();
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.doc")) {
11            doc.Open(IGR_BODY_AND_META);
12
13            while (!doc.getEOF()) {
14                string text = doc.GetText(4096);
15                System.out.println(text);
16            }
17        }
18    }
19 }
```

PYTHON

```
1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.doc") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     while not doc.getEOF():
10         output.write(doc.GetText(MaxCharsPerGetText, stripControlCodes=True))
```

C++17

```

1  #include <iostream>
2  #include <string>
3  #include "DocumentFiltersObjects.h"
4
5  int main() {
6      try {
7          // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8          Hyland::DocFilters::Api api;
9
10         // Initialize the DocumentFilters object with license and path
11         std::string license = "YOUR_LICENSE_KEY_HERE";
12         std::string path = ".";
13         api.Initialize(license, path);
14
15         // Get an extractor for the specified file
16         Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.doc");
17
18         // Open the document with BodyAndMeta flag
19         doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
20
21         // Read and print the text content
22         while (!doc.getEOF()) {
23             std::wstring text = doc.getText(4096);
24             std::wcout << text << std::endl;
25         }
26
27         // Close the document
28         doc.Close();
29
30     } catch (const std::exception& ex) {
31         std::cerr << "Error: " << ex.what() << std::endl;
32     }
33     return 0;
34 }

```

ADDITIONAL INFORMATION

This method will implicitly open the document. The [Close](#) method should be called when finished. The text returned may contain markup characters that the calling application will need to process.

After the document is opened, each call to `GetText` will return the next portion of text until the end of the document is reached. To retrieve the whole text of the document, the application should call this method in a loop and check the value of the [EOF](#) property as shown above.

Note You can request any size of string from `GetText`, but Java and .NET will return 65535 bytes at most.

SEE ALSO

- [Extractor Interface](#)
- [Extractor::EOF](#) property
- [Extractor::SupportsText](#) property

Extractor::HeartbeatCallback property

Set a callback hook called when an heartbeat is sent.

.NET

```
Func<bool> HeartbeatCallback { get; set; }
```

JAVA

```
void setHeartbeatCallback(IHeartbeatCallback value) throws IGRException;
```

PYTHON

```
@property
def HeartbeatCallback(self) -> Callable[[OpenCallback.HeartbeatAction], bool]

@HeartbeatCallback.setter
def HeartbeatCallback(self, value: Callable[[OpenCallback.HeartbeatAction], bool])
```

C++17

```
void setHeartbeatCallback(std::function<int()> value);
```

RETURN VALUE

Function

Extractor::Images property

The Images method property provides an enumerable collection of [SubFile](#) objects representing the embedded image of the current document when converting using classic HTML.

.NET

```
IEnumerable<SubFile> Images { get; }
```

PYTHON

```
@property  
def Images(self) -> Enumerator<SubFile>
```

C++17

```
subfiles_t images() const;
```

RETURN VALUE

`SubFile[]` : Returns an enumerable set of images.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 try {
5     // Initialize DocumentFilters API
6     DocumentFilters docFilters = new DocumentFilters();
7     docFilters.Initialize("YOUR_LICENSE_KEY_HERE", "."); // Replace with your actual license key and
resource path. This is not from the given sources.
8
9     // Open the HTML file in ClassicHTML mode
10    Extractor extractor = docFilters.OpenExtractor("input.docx", OpenMode.ClassicHTML,
OpenType.BodyAndMeta, "");
11
12    // Extract the text
13    string extractedText = extractor.GetText(0xffff);
14    Console.WriteLine("Extracted Text:\n" + extractedText);
15
16    // Enumerate and process images using the 'images' enumerable
17    Console.WriteLine("\nImages:");
18    foreach (SubFile image in extractor.Images) {
19        Console.WriteLine("- ID: " + image.ID);
20        Console.WriteLine("  Name: " + image.Name);
21        Console.WriteLine("  Size: " + image.FileSize + " bytes");
22
23        // Additional image processing can be done here. This is not from the given sources.
24    }
25
26    // Close the extractor
27    extractor.Close(true);
28 } catch (IGRException e) {
29     Console.WriteLine("Error: " + e.Message);
30 }
```

PYTHON

```
1 import DocumentFilters
2
3 def extract_text_and_images(filename="input.docx"): # This is not from the given sources.
4     try:
5         # Initialize DocumentFilters API. Replace placeholders with your actual values. This is not from
the given sources.
6         doc_filters = DocumentFilters.DocumentFilters()
7         doc_filters.Initialize("YOUR_LICENSE_KEY_HERE", ".")
8
9         # Open the DOCX file in ClassicHTML mode. This is not from the given sources.
10        extractor = doc_filters.OpenExtractor(filename, DocumentFilters.OpenMode.ClassicHTML,
DocumentFilters.OpenType.BodyAndMeta, "")
11
12        # Extract the text.
13        # Note: The use of 0xffff for maxLength is based on the previous C# example,
14        # but its relevance in the hypothetical Python API is unknown.
15        extracted_text = extractor.GetText(0xffff)
16        print("Extracted Text:\n" + extracted_text)
17
18        # Enumerate and process images using the 'images' enumerable.
19        # This assumes a similar structure to the C# API.
20        print("\nImages:")
21        for image in extractor.Images:
22            print(f"  ID: {image.ID}")
23            print(f"  Name: {image.Name}")
24            print(f"  Size: {image.FileSize} bytes")
25
26        # Additional image processing can be done here. This is not from the given sources.
27
28        # Close the extractor.
29        extractor.Close(True)
30    except DocumentFilters.IGREException as e:
31        print(f"Error: {e}")
```

C++17

```

1  #include <DocumentFiltersObjects.h>
2  #include <iostream>
3
4  int main() {
5      try {
6          // Initialize DocumentFilters API
7          DocumentFilters docFilters;
8          docFilters.Initialize("YOUR_LICENSE_KEY_HERE", "."); // Replace with your actual license key and
resource path. This is not from the given sources.
9
10         // Open the DOCX file in ClassicHTML mode.
11         Extractor extractor = docFilters.OpenExtractor(L"input.docx", OpenMode::ClassicHTML); // The
OpenMode::ClassicHTML enumeration member is not found in the given sources.
12
13         // Extract the text
14         std::wstring extractedText = extractor.getText(0xffff); // The use of 0xffff as the max_length
argument is based on your previous requests but not found in the C++ source code examples.
15         std::wcout << L"Extracted Text:\n" << extractedText << std::endl;
16
17         // Enumerate and process images using the 'images' enumerable.
18         std::wcout << L"\nImages:" << std::endl;
19         for (const auto& image : extractor.images()) {
20             std::wcout << L"- ID: " << image.getId() << std::endl;
21             std::wcout << L"  Name: " << image.getName() << std::endl;
22             std::wcout << L"  Size: " << image.getSize() << L" bytes" << std::endl;
23
24             // Additional image processing can be done here. This is not from the given sources.
25         }
26
27         // Close the extractor
28         extractor.Close();
29     } catch (const DocumentFilters::Error& e) {
30         std::cerr << "Error: " << e.what() << std::endl;
31     }
32     return 0;
33 }

```


Extractor::Localize property

Utility function that allows for localization of metadata without providing a callback. Any localization options must be set before an `.Open` call.

.NET

```
Dictionary<string, string> Localize { get; }
```

JAVA

```
LocalizedDictionary getLocalize() throws IGREException;
```

PYTHON

```
@property
def Localize(self) -> dict
```

RETURN VALUE

`Dictionary<string, string>` : Dictionary of localized words

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using (var doc = api.GetExtractor("outlook.msg"))
7 {
8     doc.Localize["Subject"] = "Objet";
9     doc.Localize["To"] = "À";
10    doc.Localize["From"] = "De";
11    doc.Localize["Bcc"] = "Cci";
12    doc.Localize["Sent"] = "Envoyé";
13    doc.Open(OpenMode.Paginated);
14
15    using (var canvas = api.MakeOutputCanvas("output.pdf", CanvasType.PDF))
16    {
17        canvas.RenderPages(doc);
18    }
19 }
```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("outlook.msg"))
11        {
12            doc.GetLocalize().set("Subject", "Objet");
13            doc.GetLocalize().set("To", "À");
14            doc.GetLocalize().set("From", "De");
15            doc.GetLocalize().set("Bcc", "Cci");
16            doc.GetLocalize().set("Sent", "Envoyé");
17            doc.Open(isisys_docfiltersConstants.IGR_FORMAT_IMAGE);
18
19            try (Canvas canvas = df.MakeOutputCanvas("filename.pdf",
isisys_docfiltersConstants.IGR_DEVICE_PDF, ""))
20            {
21                for (int i = 0, c = doc.GetPageCount(); i < c; ++i)
22                {
23                    try (Page page = doc.GetPage(i)) {
24                        canvas.RenderPage(page);
25                    }
26                }
27            }
28        }
29    }
30 }

```

PYTHON

```

1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("outlook.msg") as doc:
7     doc.Localize["Subject"] = "Objet"
8     doc.Localize["To"] = "À"
9     doc.Localize["From"] = "De"
10    doc.Localize["Bcc"] = "Cci"
11    doc.Localize["Sent"] = "Envoyé"
12    doc.Open(IGR_FORMAT_IMAGE)
13
14    with api.MakeOutputCanvas("filename.pdf", canvasType=IGR_DEVICE_PDF) as canvas:
15        canvas.RenderPages(doc)

```

Extractor::LocalizeCallback property

Set a callback hook called when a string is to be localized.

.NET

```
Func<uint, string, string> LocalizeCallback { get; set; }
```

JAVA

```
void setLocalizeCallback(ILocalizeCallback value) throws IGRException;
```

PYTHON

```
@property
def LocalizeCallback(self) -> Callable[[OpenCallback.LocalizeAction], str]

@LocalizeCallback.setter
def LocalizeCallback(self, value: Callable[[OpenCallback.LocalizeAction], str])
```

C++17

```
void setLocalizeCallback(std::function<std::wstring(IGR_ULONG, const std::wstring&)> value);
```

RETURN VALUE

Function

Extractor::LogLevelCallback property

Set a callback hook called a log-level is requested for a module.

.NET

```
Func<string, int> LogLevelCallback { get; set; }
```

JAVA

```
void setLogLevelCallback(ILogLevelCallback value) throws IGRException;
```

PYTHON

```
@property
def LogLevelCallback(self) -> Callable[[OpenCallback.ActionLogLevel], str]

@LogLevelCallback.setter
def LogLevelCallback(self, value: Callable[[OpenCallback.ActionLogLevel], str])
```

C++17

```
void setLogLevelCallback(std::function<IGR_LOG_LEVEL_TYPE(const std::string&)> value);
```

RETURN VALUE

Function

Extractor::LogMessageCallback property

Set a callback hook called a log-message is sent.

.NET

```
Action<int, string, string> LogMessageCallback { get; set; }
```

JAVA

```
void setLogMessageCallback(ILogMessageCallback value) throws IGRException;
```

PYTHON

```
@property
def LogMessageCallback(self) -> Callable[[OpenCallback.ActionLogMessage], void]

@LogMessageCallback.setter
def LogMessageCallback(self, value: Callable[[OpenCallback.ActionLogMessage], void])
```

C++17

```
void setLogMessageCallback(std::function<void(IGR_LOG_LEVEL_TYPE, const std::string&, const std::string&)>
value);
```

RETURN VALUE

Function

Extractor::MimeType property

Returns the MimeType of the file.

.NET

```
string MimeType { get; }
```

JAVA

```
string getMimeType() throws IGREException;
```

PYTHON

```
@property  
def MimeType(self) -> string
```

C++17

```
std::wstring getMimeType() const;
```

RETURN VALUE

`string` : The MimeType of the file.

SEE ALSO

- [Document Format Codes](#)

Extractor::OcrImageCallback property

Set a callback hook called when an OCRing an image.

.NET

```
Func<OpenCallbackActionOcrImage, bool> OcrImageCallback { get; set; }
```

JAVA

```
void setOcrImageCallback(IOcrImageCallback value) throws IGRException;
```

PYTHON

```
@property
def OcrImageCallback(self) -> Callable[[OpenCallback.ActionOcrImage], bool]

@OcrImageCallback.setter
def OcrImageCallback(self, value: Callable[[OpenCallback.ActionOcrImage], bool])
```

C++17

```
void setOcrImageCallback(std::function<bool(OcrImage&)> value);
```

RETURN VALUE

Function

Extractor::Open method

The Open method opens a document for processing.

OVERLOADS

Open(int)	Open the file with the given flags.
Open(int, string)	Open the file with the given flags and options set.
Open(int, string, Func<OpenCallback, int>)	Open the file with the given flags and options set.

OPEN(INT)

Prototype

.NET

```
void Open(int flags)
```

JAVA

```
void Open(int flags) throws IGREException;
```

PYTHON

```
def Open(self, flags: int) -> void
```

C++17

```
void Open(int flags)
```

C++11

```
void Open(int flags)
```

COM

```
HRESULT Open([in] int flags)
```

Parameters

flags: int : Specify the open-mode flags; see [Open Document Flags](#) for details.

OPEN(INT, STRING)**Prototype****.NET**

```
void Open(int flags, string options)
```

JAVA

```
void Open(int flags, string options) throws IGRException;
```

PYTHON

```
def Open(self, flags: int, options: string) -> void
```

C++17

```
void Open(int flags, const std::wstring& options)
```

C++11

```
void Open(int flags, std::wstring options)
```

COM

```
HRESULT Open([in] int flags, [in] BSTR options)
```

Parameters

flags: int : Specify the open-mode flags; see [Open Document Flags](#) for details.

options: string : Semicolon separated list of name value pair options; see [Constants and Codes](#) for details.

OPEN(INT, STRING, FUNC<OPENCALLBACK, INT>)**Prototype****.NET**

```
void Open(int flags, string options, Func<OpenCallback, int> callback)
```

JAVA

```
void Open(int flags, string options, OpenCallback callback) throws IGRException;
```

PYTHON

```
def Open(self, flags: int, options: string, callback: Callable[[DocumentFilters.OpenCallback], int]) -> void
```

C++17

```
void Open(int flags, const std::wstring& options, open_callback_t callback)
```

Parameters

flags: int : Specify the open-mode flags; see [Open Document Flags](#) for details.

options: string : Semicolon separated list of name value pair options; see [Constants and Codes](#) for details.

callback: Func<OpenCallback, int> : Optional IGR_Open_Ex callback function.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc"); // (2) Create an instance of the Extractor for a file
8
9 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta); // (3) Open the document for reading in text-mode
10
11 while (!doc.EndOfStream) // (4) Read the document in 4KB chunks and write it to the console
12 {
13     var text = doc.GetText(4096);
14     Console.Out.WriteLine(text);
15 }
16
17 doc.Close();
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.doc")) {
11            doc.Open(isys_docfilters.IGR_BODY_AND_META);
12
13            while (!doc.getEOF()) {
14                string text = doc.GetText(4096);
15                System.out.println(text);
16            }
17        }
18    }
19 }
```

PYTHON

```

1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.doc") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     while not doc.getEOF():
10         output.write(doc.GetText(MaxCharsPerGetText, stripControlCodes=True))

```

C++17

```

1 #include <iostream>
2 #include <string>
3 #include "DocumentFiltersObjects.h"
4
5 int main() {
6     try {
7         // Create a DocumentFilters object (Api is an alias for DocumentFilters)
8         Hyland::DocFilters::Api api;
9
10        // Initialize the DocumentFilters object with license and path
11        std::string license = "YOUR_LICENSE_KEY_HERE";
12        std::string path = ".";
13        api.Initialize(license, path);
14
15        // Get an extractor for the specified file
16        Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.doc");
17
18        // Open the document with BodyAndMeta flag
19        doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
20
21        // Read and print the text content
22        while (!doc.getEOF()) {
23            std::wstring text = doc.GetText(4096);
24            std::wcout << text << std::endl;
25        }
26
27        // Close the document
28        doc.Close();
29
30    } catch (const std::exception& ex) {
31        std::cerr << "Error: " << ex.what() << std::endl;
32    }
33    return 0;
34 }

```

ADDITIONAL INFORMATION

Most members of this interface operate on an open document. If you have not called the Open method, it will be called internally.

Below is the list of members that will implicitly open the document when accessed:

- [SaveTo method](#)
- [GetText method](#)

- [EOF property](#)
- [GetFirstSubFile method](#)
- [GetNextSubFile method](#)

SEE ALSO

- [Extractor Interface](#)
- [Close method](#)

Extractor::PageCount property

Returns the number of pages in the current document, the document must be opened in image mode for the page count to be populated.

.NET

```
int PageCount { get; }
```

JAVA

```
int getPageCount() throws IGRException;
```

PYTHON

```
@property  
def PageCount(self) -> int
```

C++17

```
int getPageCount() const;
```

RETURN VALUE

int : The number of pages in the document

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;  
2  
3 var api = new Hyland.DocumentFilters.Api();  
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");  
5  
6 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);  
7  
8 for (int i = 0; i < doc.PageCount; i++)  
9 {  
10     using var page = doc.GetPage(i);  
11     using var canvas = api.MakeOutputCanvas($"page-{i+1}.png", CanvasType.PNG);  
12     canvas.RenderPage(page);  
13 }
```

SEE ALSO

- [Extractor Interface](#)
- [Page interface](#)

Extractor::Pages property

The Pages property provides an enumerable collection of pages for an opened document. The document must be opened in image mode (IGR_FORMAT_IMAGE).

.NET

```
IEnumerable<Page> Pages { get; }
```

PYTHON

```
@property  
def Pages(self) -> Enumerator<Page>
```

C++17

```
pages_t pages() const;
```

RETURN VALUE

`Pages[ ]` : Returns an enumerable set of pages.

Extractor::PasswordCallback property

Set a callback hook called when a password is required.

.NET

```
Func<string, string> PasswordCallback { get; set; }
```

JAVA

```
void setPasswordCallback(IPasswordCallback value) throws IGREException;
```

PYTHON

```
@property
def PasswordCallback(self) -> Callable[[str], str]

@PasswordCallback.setter
def PasswordCallback(self, value: Callable[[str], str])
```

C++17

```
void setPasswordCallback(std::function<std::wstring(const std::wstring&)> value);
```

RETURN VALUE

Function

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api(); // (1) Create an instance of the API and initialize it
5 var licenseKey = Environment.GetEnvironmentVariable("DOCFILTERS_LICENSE_KEY") ?? "YOUR_LICENSE_KEY_HERE";
6 api.Initialize(licenseKey, ".");
7
8 using var container = api.GetExtractor("sample.7z"); // (2) Create an instance of the extractor
9 container.PasswordCallback = (string id) => "password1234"; // (3) Set the password callback
10 container.Open(OpenMode.Text);
11
12 foreach (var item in container.SubFiles)
13 {
14     Console.WriteLine($"{item.FileDate:yyyy-MM-dd HH:mm} {item.FileSize,10} {item.Name}");
15 }
```

JAVA

```

1 package password_callback;
2
3 import com.perceptive.documentfilters.*;
4
5 public class App {
6
7     public static void main(String[] args) throws Exception {
8         String licenseKey = System.getenv("DOCFILTERS_LICENSE_KEY");
9         if (licenseKey == null || licenseKey.isEmpty())
10             licenseKey = "YOUR_LICENSE_KEY_HERE";
11
12         DocumentFilters df = new DocumentFilters(); // (1) Create an instance of the API and initialize it
13         df.Initialize(licenseKey, ".");
14
15         // (2) Create an instance of the extractor
16         try (Extractor container = df.GetExtractor("sample.7z")) {
17             // (3) Set the password callback
18             container.setPasswordCallback(id -> "password1234");
19             container.Open(isys_docfilters.IGR_BODY_AND_META, "");
20
21             // (4) Walk through the files in the container
22             for (SubFile subFile : container.getSubFiles()) {
23                 System.out.printf("%1$tY-%1$tm-%1$td %1$tH:%1$tM %2$10d %3$s%n",
24                     subFile.getFileTime(), subFile.getFileSize(), subFile.getName());
25             }
26         }
27     }
28 }

```

PYTHON

```

1 import DocumentFilters
2 import os
3
4 # (1) Create an instance of the API and initialize it
5 api = DocumentFilters.Api()
6 license_key = os.getenv("DOCFILTERS_LICENSE_KEY", "YOUR_LICENSE_KEY_HERE")
7 api.Initialize(license_key, ".")
8
9 # (2) Create an instance of the extractor
10 with api.GetExtractor("sample.7z") as container:
11     # (3) Set the password callback
12     container.PasswordCallback = lambda id: "password1234"
13     container.Open(DocumentFilters.IGR_BODY_AND_META)
14
15     # Iterate through the subfiles and print their details
16     for item in container.SubFiles:
17         print(f"{item.FileDate:%Y-%m-%d %H:%M} {item.FileSize:10} {item.Name}")

```


C++17

```

1  #include "DocumentFiltersObjects.h"
2  #include <iostream>
3  #include <string>
4  #include <iomanip>
5
6  int main() {
7      try {
8          // Create a DocumentFilters object (Api is an alias for DocumentFilters)
9          Hyland::DocFilters::Api api;
10
11         // Initialize the DocumentFilters object with license and path
12         std::string license = std::getenv("DOCFILTERS_LICENSE_KEY") ?
std::getenv("DOCFILTERS_LICENSE_KEY") : "YOUR_LICENSE_KEY_HERE";
13         std::string path = ".";
14         api.Initialize(license, path);
15
16         // Get an extractor for the specified file
17         Hyland::DocFilters::Extractor container = api.GetExtractor(L"sample.7z");
18         container.SetPasswordCallback([](const std::wstring& id) {
19             // Provide the password for the encrypted file
20             return L"password1234";
21         });
22
23         // Open the document with BodyAndMeta flag
24         container.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
25
26         // Read and print the text content
27         for (const auto& item : container.subfiles()) {
28             auto tm = item.getFileDate().ToTm(false).value();
29             std::cout << std::put_time(&tm, "%Y-%m-%d %H:%M")
30                 << " " << std::setw(10) << item.GetSize()
31                 << " " << Hyland::DocFilters::w_to_u8(item.getName()) << std::endl;
32         }
33
34         // Close the document
35         container.Close();
36
37     } catch (const std::exception& ex) {
38         std::cerr << "Error: " << ex.what() << std::endl;
39     }
40     return 0;
41 }

```

Extractor::SaveTo method

The SaveTo method extracts the entire text content of the document in a single call. The text may be saved to a file with the given name or via an instance of an IStream (COM) object.

OVERLOADS

SaveTo(string)	Save the text of the document to a file.
SaveTo(Stream)	Save the text of the document to a stream.
SaveTo(string, System.Text.Encoding)	Save the text of the document to a file with the given encoding.
SaveTo(Stream, System.Text.Encoding)	Save the text of the document to a file with the given encoding.

SAVETO(STRING)

Prototype

.NET

```
void SaveTo(string filename)
```

JAVA

```
void SaveTo(string filename) throws IGRException;
```

PYTHON

```
def SaveTo(self, filename: string) -> void
```

C++17

```
void SaveTo(const std::wstring& filename)
```

C++11

```
void SaveTo(std::wstring filename)
```

COM

```
HRESULT SaveTo([in] BSTR filename)
```

Parameters

filename: string : The filename where the text is to be written

SAVETO(STREAM)**Prototype****.NET**

```
void SaveTo(Stream stream)
```

JAVA

```
void SaveTo(Stream stream) throws IOException;
```

PYTHON

```
def SaveTo(self, stream: Stream) -> void
```

C++17

```
void SaveTo(std::ostream stream)
```

C++11

```
void SaveTo(Stream stream)
```

COM

```
HRESULT SaveTo([in] Stream* stream)
```

Parameters

`stream: Stream` : The stream where the text is to be written

SAVETO(STRING, SYSTEM.TEXT.ENCODING)**Prototype****.NET**

```
void SaveTo(string filename, System.Text.Encoding encoding)
```

C++17

```
void SaveTo(const std::wstring& filename, int encoding)
```

Parameters

`filename: string` : The filename where the text is to be written

`encoding: System.Text.Encoding` : The encoding to use when saving the text.

SAVETO(STREAM, SYSTEM.TEXT.ENCODING)**Prototype****.NET**

```
void SaveTo(Stream stream, System.Text.Encoding encoding)
```

PYTHON

```
def SaveTo(self, stream: Stream, encoding: string) -> void
```

C++17

```
void SaveTo(std::ostream stream, int encoding)
```

Parameters

stream: Stream : The stream where the text is to be written

encoding: System.Text.Encoding : The encoding to use when saving the text.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3 using System.Text;
4
5 try {
6     // Initialize the API.
7     DocumentFilters api = new DocumentFilters();
8     api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10    // Obtain an Extractor for the input file.
11    Extractor doc = api.GetExtractor("input.pdf");
12
13    // Open the document in Text mode.
14    doc.Open(OpenMode.Text);
15
16    // Attempt to save the document, specifying UTF-8 encoding
17    // and stripping characters.
18    doc.SaveTo("output.txt", Encoding.UTF8);
19
20    Console.WriteLine("Document saved successfully."); // Not from the sources
21
22    // Close the document.
23    doc.Close(); // [4]
24 } catch (IGRException e) {
25     Console.WriteLine("Error saving document: " + e.Message); // [6] Not from the sources
26 }
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class SaveToExample {
4     public static void main(String[] args) {
5         try {
6             // Initialize the API.
7             DocumentFilters api = new DocumentFilters();
8             api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10            // Obtain an Extractor for the input file.
11            Extractor doc = api.GetExtractor("input.pdf");
12
13            // Open the document in Text mode.
14            doc.Open(OpenMode.Text);
15
16            // Save the document content to a file, specifying UTF-8 encoding.
17            doc.SaveTo("output.txt");
18
19            System.out.println("Document saved successfully.");
20
21            // Close the extractor.
22            doc.Close();
23        } catch (IGRException e) {
24            System.err.println("Error saving document: " + e.getMessage());
25        }
26    }
27 }
```

PYTHON

```
1 import DocumentFilters
2
3 # Initialize the API.
4 api = DocumentFilters.Api()
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
6
7 # Obtain an Extractor for the input file.
8 doc = api.GetExtractor(filename="input.pdf")
9
10 # Open the document in Text mode.
11 doc.Open(DocumentFilters.OpenMode.Text)
12
13 # Attempt to save the document, specifying UTF-8 encoding and
14 try:
15     doc.SaveTo(filename="output.txt", encoding='utf-8', stripCharacters=True)
16     print("Document saved successfully.")
17
18 except DocumentFilters.IGRException as e:
19     print("Error saving document:", e.message)
20
21 doc.Close()
```

C++17

```
1 #include <DocumentFiltersObjects.h>
2 #include <iostream>
3
4 int main() {
5     try {
6         // Initialize the API.
7         DocumentFilters api;
8         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        // Obtain an Extractor for the input file.
11        Extractor doc = api.GetExtractor("input.pdf");
12
13        // Open the document in Text mode.
14        doc.Open(OpenMode::Text);
15
16        // Save the document, specifying UTF-8 encoding.
17        doc.SaveTo("output.txt", 65001);
18
19        std::cout << "Document saved successfully." << std::endl;
20
21        // Close the document.
22        doc.Close();
23
24    } catch (DocumentFilters::IGRException& e) {
25        std::cerr << "Error saving document: " << e.what() << std::endl;
26        return 1;
27    }
28    return 0;
29 }
```

ADDITIONAL INFORMATION

The text stream may contain markup characters that the application will need to process. This method will implicitly open the document. Call the [Close](#) method when finished.

Extractor::SubFiles property

Returns an enumerable set of SubFiles.

.NET

```
IEnumerable<SubFile> SubFiles { get; }
```

JAVA

```
java.lang.Iterable<SubFile> getSubFiles() throws IGRException;
```

PYTHON

```
@property
def SubFiles(self) -> Enumerator<SubFile>
```

C++17

```
subfiles_t subfiles() const;
```

RETURN VALUE

`SubFile[]` : Returns an enumerable set of SubFiles.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.zip");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 foreach (var subfile in doc.SubFiles)
11 {
12     using (subfile)
13     {
14         // act on subfile
15         Console.Out.WriteLine("Name: " + subfile.Name);
16         Console.Out.WriteLine("ID: " + subfile.ID);
17         Console.Out.WriteLine("Date: " + subfile.FileDate);
18         Console.Out.WriteLine("Size: " + subfile.FileSize);
19     }
20 }
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.zip")) {
11            doc.Open(isys_docfilters.IGR_BODY_AND_META);
12
13            for (SubFile subfile = doc.GetFirstSubFile(); subfile != null; subfile = doc.GetNextSubFile())
14            {
15                try (subfile) {
16                    // act on subfile
17                }
18            }
19        }
20    }
21 }
```

PYTHON

```
1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.zip") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     for subfile in doc.SubFiles:
10         with subfile:
11             # act on subfile
```


C++17

```

1  #include <DocumentFiltersObjects.h>
2  #include <iostream>
3
4  using namespace Hyland::DocFilters;
5
6  int main() {
7      try {
8          // Initialize the Document Filters API
9          Hyland::DocFilters::DocumentFilters api;
10         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
11
12         // Get the extractor
13         Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.zip");
14
15         // Open the document
16         doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
17
18         // Access subfiles using the subfiles() iterator
19         for (auto subfile : doc.subfiles()) {
20             try {
21                 // Process the subfile
22                 std::wcout << L"Name: " << subfile.getName() << std::endl;
23                 std::wcout << L"ID: " << subfile.getId() << std::endl;
24                 std::wcout << L"Date: " << u8_to_w(subfile.getFileDate().ToIsoString()) << std::endl;
25                 std::wcout << L"Size: " << subfile.getSize() << std::endl;
26             } catch (const std::exception& e) {
27                 std::cerr << "Error: " << e.what() << std::endl;
28             }
29         }
30
31         // Close the extractor
32         doc.Close();
33     } catch (const std::exception& e) {
34         std::cerr << "Error: " << e.what() << std::endl;
35     }
36
37     return 0;
38 }

```

Extractor::getFileType method

The FileType method allows for extended information to be returned about the file type.

OVERLOADS

<code>getFileType(int)</code>	The FileType method allows for extended information to be returned about the file type
<code>getFileType(IGRFormatWhat)</code>	The FileType method allows for extended information to be returned about the file type

GETFILETYPE(INT)

Prototype

.NET

```
string getFileType(int what)
```

JAVA

```
string getFileType(int what) throws IGRException;
```

PYTHON

```
def getFileType(self, what: int) -> string
```

C++17

```
std::wstring getFileType(int what)
```

C++11

```
std::wstring getFileType(int what)
```

COM

```
HRESULT GetFileTypeName([in] int what, [out, retval] BSTR *result)
```

Parameters

`what: int` : Indicates the string information that is requested, can be one of the following:

- 0: IGR_FORMAT_LONG_NAME
- 1: IGR_FORMAT_SHORT_NAME
- 2: IGR_FORMAT_CONFIG_NAME

- 3: IGR_FORMAT_CLASS_NAME
- 4: IGR_FORMAT_LEGACY
- 5: IGR_FORMAT_MIMETYPE
- 6: IGR_FORMAT_FILETYPE_CATEGORY

Return value

`string` : String value contained the requested data

GETFILETYPE(IGRFORMATWHAT)**Prototype****.NET**

```
string getFileType(IGRFormatWhat what)
```

JAVA

```
string getFileType(IGRFormatWhat what) throws IGRException;
```

PYTHON

```
def getFileType(self, what: IGRFormatWhat) -> string
```

C++17

```
std::wstring getFileType(IGRFormatWhat what)
```

C++11

```
std::wstring getFileType(IGRFormatWhat what)
```

COM

```
HRESULT GetFileTypeName([in] IGRFormatWhat* what, [out, retval] BSTR *result)
```

Parameters

`what` : `IGRFormatWhat` : Indicates the string information that is requested, can be one of the following:

- 0: IGR_FORMAT_LONG_NAME
- 1: IGR_FORMAT_SHORT_NAME
- 2: IGR_FORMAT_CONFIG_NAME
- 3: IGR_FORMAT_CLASS_NAME
- 4: IGR_FORMAT_LEGACY

- 5: IGR_FORMAT_MIMETYPE
- 6: IGR_FORMAT_FILETYPE_CATEGORY

Return value

`string` : String value contained the requested data

SEE ALSO

- [Extractor Interface](#)
- [Document Format Codes](#)

Extractor::getSupportsHTML method

getSupportsHTML method is TRUE if document can be converted to classic HTML.

PROTOTYPE

.NET

```
bool getSupportsHTML()
```

JAVA

```
boolean getSupportsHTML() throws IGRException;
```

PYTHON

```
def getSupportsHTML(self) -> bool
```

C++17

```
bool getSupportsHTML()
```

RETURN VALUE

`bool` : Indicates if the file supports HTML conversion.

SAMPLE CODE

.NET

```
1 using System;
2 using Hyland.DocumentFilters;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 var caps = doc.Capabilities;
11
12 if (doc.getSupportsText())
13     Console.WriteLine("File supports text extraction");
14
15 if (doc.getSupportsSubFiles())
16     Console.WriteLine("File supports subfile enumeration");
17
18 if (doc.getSupportsHTML())
19     Console.WriteLine("File supports HTML conversion");
```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class Main {
4     public static void main(String[] args) {
5         try {
6             // Initialize the API.
7             DocumentFilters api = new DocumentFilters();
8             api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10            // Obtain an Extractor for the input file.
11            Extractor doc = api.GetExtractor("filename.doc");
12
13            // Open the document, specifying flags.
14            doc.Open(isys_docfiltersConstants.IGR_BODY_AND_META);
15
16            // Check for specific capabilities.
17            if (doc.getSupportsText())
18                System.out.println("File supports text extraction");
19
20            if (doc.getSupportsSubFiles())
21                System.out.println("File supports subfile enumeration");
22
23            if (doc.getSupportsHTML())
24                System.out.println("File supports HTML conversion");
25
26            // Close the document.
27            doc.Close();
28        } catch (IGRException e) {
29            System.err.println("Error processing document: " + e.getMessage());
30        }
31    }
32 }

```

PYTHON

```

1 import DocumentFilters
2
3 api = DocumentFilters.Api()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 doc = api.GetExtractor(filename="filename.doc")
7 doc.Open(DocumentFilters.OpenType.BodyAndMeta)
8
9 caps = doc.Capabilities
10
11 if doc.getSupportsText():
12     print("File supports text extraction")
13
14 if doc.getSupportsSubFiles():
15     print("File supports subfile enumeration")
16
17 if doc.getSupportsHTML():
18     print("File supports HTML conversion")

```

C++17

```

1 #include "DocumentFiltersObjects.h" // This includes necessary headers from the source.
2 #include <iostream> // Not from sources.
3
4 int main() {
5     try {
6         // Initialize the API.
7         DocumentFilters api;
8         api.Initialize(L"YOUR_LICENSE_KEY_HERE", L".");
9
10        // Obtain an Extractor for the input file.
11        Extractor doc = api.GetExtractor("filename.doc");
12
13        // Open the document, specifying flags.
14        doc.Open(OpenMode::Paginated, IGR_BODY_AND_META);
15
16        // Check for specific capabilities.
17        if (doc.getSupportsText())
18            std::cout << "File supports text extraction" << std::endl;
19
20        if (doc.getSupportsSubFiles())
21            std::cout << "File supports subfile enumeration" << std::endl;
22
23        if (doc.getSupportsHtml())
24            std::cout << "File supports HTML conversion" << std::endl;
25
26        // Close the document.
27        doc.Close();
28    } catch (const DocumentFilters::Error& e) {
29        std::cerr << "Error processing document: " << e.what() << std::endl;
30    }
31
32    return 0;
33 }

```

SEE ALSO

- [Extractor Interface](#)
- [Extractor::GetText method](#)
- [Extractor::getSupportsText method](#)

Extractor::getSupportsSubFiles property

getSupportsSubFiles property is TRUE if the document is a compound or archive document, potentially with sub-documents.

PROTOTYPE

.NET

```
bool getSupportsSubFiles()
```

JAVA

```
boolean getSupportsSubFiles() throws IGREException;
```

PYTHON

```
def getSupportsSubFiles(self) -> bool
```

C++17

```
bool getSupportsSubFiles()
```

RETURN VALUE

bool : Indicates if the file supports subfiles conversion.

SAMPLE CODE

.NET

```
1 using System;
2 using Hyland.DocumentFilters;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 var caps = doc.Capabilities;
11
12 if (doc.getSupportsText())
13     Console.WriteLine("File supports text extraction");
14
15 if (doc.getSupportsSubFiles())
16     Console.WriteLine("File supports subfile enumeration");
17
18 if (doc.getSupportsHTML())
19     Console.WriteLine("File supports HTML conversion");
```


JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class Main {
4     public static void main(String[] args) {
5         try {
6             // Initialize the API.
7             DocumentFilters api = new DocumentFilters();
8             api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10            // Obtain an Extractor for the input file.
11            Extractor doc = api.GetExtractor("filename.doc");
12
13            // Open the document, specifying flags.
14            doc.Open(isys_docfiltersConstants.IGR_BODY_AND_META);
15
16            // Check for specific capabilities.
17            if (doc.getSupportsText())
18                System.out.println("File supports text extraction");
19
20            if (doc.getSupportsSubFiles())
21                System.out.println("File supports subfile enumeration");
22
23            if (doc.getSupportsHTML())
24                System.out.println("File supports HTML conversion");
25
26            // Close the document.
27            doc.Close();
28        } catch (IGRException e) {
29            System.err.println("Error processing document: " + e.getMessage());
30        }
31    }
32 }

```

PYTHON

```

1 import DocumentFilters
2
3 api = DocumentFilters.Api()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 doc = api.GetExtractor(filename="filename.doc")
7 doc.Open(DocumentFilters.OpenType.BodyAndMeta)
8
9 caps = doc.Capabilities
10
11 if doc.getSupportsText():
12     print("File supports text extraction")
13
14 if doc.getSupportsSubFiles():
15     print("File supports subfile enumeration")
16
17 if doc.getSupportsHTML():
18     print("File supports HTML conversion")

```

C++17

```

1  #include "DocumentFiltersObjects.h" // This includes necessary headers from the source.
2  #include <iostream> // Not from sources.
3
4  int main() {
5      try {
6          // Initialize the API.
7          DocumentFilters api;
8          api.Initialize(L"YOUR_LICENSE_KEY_HERE", L".");
9
10         // Obtain an Extractor for the input file.
11         Extractor doc = api.GetExtractor("filename.doc");
12
13         // Open the document, specifying flags.
14         doc.Open(OpenMode::Paginated, IGR_BODY_AND_META);
15
16         // Check for specific capabilities.
17         if (doc.getSupportsText())
18             std::cout << "File supports text extraction" << std::endl;
19
20         if (doc.getSupportsSubFiles())
21             std::cout << "File supports subfile enumeration" << std::endl;
22
23         if (doc.getSupportsHtml())
24             std::cout << "File supports HTML conversion" << std::endl;
25
26         // Close the document.
27         doc.Close();
28     } catch (const DocumentFilters::Error& e) {
29         std::cerr << "Error processing document: " << e.what() << std::endl;
30     }
31
32     return 0;
33 }

```

SEE ALSO

- [Extractor Interface](#)
- [Extractor::GetText method](#)
- [Extractor::getSupportsText method](#)

Extractor::getSupportsText method

getSupportsText method return TRUE if text content can be extracted from the document. This property must be TRUE to be able to call to the [Extractor::SaveTo](#) and [Extractor::GetText](#) methods.

PROTOTYPE

.NET

```
bool getSupportsText()
```

JAVA

```
boolean getSupportsText() throws IGREException;
```

PYTHON

```
def getSupportsText(self) -> bool
```

C++17

```
bool getSupportsText()
```

RETURN VALUE

bool : Indicates if the file supports subfiles conversion.

SAMPLE CODE

.NET

```
1 using System;
2 using Hyland.DocumentFilters;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 var caps = doc.Capabilities;
11
12 if (doc.getSupportsText())
13     Console.WriteLine("File supports text extraction");
14
15 if (doc.getSupportsSubFiles())
16     Console.WriteLine("File supports subfile enumeration");
17
18 if (doc.getSupportsHTML())
19     Console.WriteLine("File supports HTML conversion");
```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class Main {
4     public static void main(String[] args) {
5         try {
6             // Initialize the API.
7             DocumentFilters api = new DocumentFilters();
8             api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10            // Obtain an Extractor for the input file.
11            Extractor doc = api.GetExtractor("filename.doc");
12
13            // Open the document, specifying flags.
14            doc.Open(isys_docfiltersConstants.IGR_BODY_AND_META);
15
16            // Check for specific capabilities.
17            if (doc.getSupportsText())
18                System.out.println("File supports text extraction");
19
20            if (doc.getSupportsSubFiles())
21                System.out.println("File supports subfile enumeration");
22
23            if (doc.getSupportsHTML())
24                System.out.println("File supports HTML conversion");
25
26            // Close the document.
27            doc.Close();
28        } catch (IGRException e) {
29            System.err.println("Error processing document: " + e.getMessage());
30        }
31    }
32 }

```

PYTHON

```

1 import DocumentFilters
2
3 api = DocumentFilters.Api()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 doc = api.GetExtractor(filename="filename.doc")
7 doc.Open(DocumentFilters.OpenType.BodyAndMeta)
8
9 caps = doc.Capabilities
10
11 if doc.getSupportsText():
12     print("File supports text extraction")
13
14 if doc.getSupportsSubFiles():
15     print("File supports subfile enumeration")
16
17 if doc.getSupportsHTML():
18     print("File supports HTML conversion")

```

C++17

```

1  #include "DocumentFiltersObjects.h" // This includes necessary headers from the source.
2  #include <iostream> // Not from sources.
3
4  int main() {
5      try {
6          // Initialize the API.
7          DocumentFilters api;
8          api.Initialize(L"YOUR_LICENSE_KEY_HERE", L".");
9
10         // Obtain an Extractor for the input file.
11         Extractor doc = api.GetExtractor("filename.doc");
12
13         // Open the document, specifying flags.
14         doc.Open(OpenMode::Paginated, IGR_BODY_AND_META);
15
16         // Check for specific capabilities.
17         if (doc.getSupportsText())
18             std::cout << "File supports text extraction" << std::endl;
19
20         if (doc.getSupportsSubFiles())
21             std::cout << "File supports subfile enumeration" << std::endl;
22
23         if (doc.getSupportsHtml())
24             std::cout << "File supports HTML conversion" << std::endl;
25
26         // Close the document.
27         doc.Close();
28     } catch (const DocumentFilters::Error& e) {
29         std::cerr << "Error processing document: " << e.what() << std::endl;
30     }
31
32     return 0;
33 }

```

SEE ALSO

- [Extractor Interface](#)
- [Extractor::GetText method](#)
- [Extractor::getSupportsText method](#)

3.3.54 FormElement

FormElement interface

The FormElement interface represents a single form control for a given page. FormElements are currently supported for static PDF documents.

FormElement::Action property	The Action property contains the behavior when a user clicks the form element.
FormElement::ActionDest property	The ActionDest property contains the destination when a user interacts with certain form controls.
FormElement::Alignment property	The Alignment property contains the text alignment that a viewer should use when presenting a single or multiline edit control.
FormElement::Caption property	The Caption property returns the optional text associated with a check box or radio button.
FormElement::ElementType property	The ElementType property contains the control type that a viewer should use to collect input from a user.
FormElement::Flags property	The Flags property contains a bit set of flags applicable to the control.
FormElement::FontName property	The FontName property contains the font that a viewer should use when presenting a single or multiline edit control.
FormElement::FontSize property	The FontSize property contains the font size that a viewer should use when presenting a single or multiline edit control.
FormElement::GetOption method	The GetOption method returns a FormElementOption object that represents a single selectable option in a list box or dropdown box.
FormElement::Height property	The Height property return the dimensions on the page where the form element should be positioned. The dimension information is based on the DPI used when loading the page.
FormElement::Name property	The Name property returns the qualified name for the form element.
FormElement::OptionCount property	The OptionCount property contains the number of items that are available in list boxes and dropdown boxes.

FormElement::Rotation property	The Rotation property contains the rotation in degrees of the form element. Only rotation in 90 degree increments will be returned.
FormElement::Selected property	The Selected property contains the current selected item in a single-select list box, or drop down box.
FormElement::Value property	The Value property returns the stored value for the form element.
FormElement::Width property	The Width property return the dimensions on the page where the form element should be positioned. The dimension information is based on the DPI used when loading the page.
FormElement::X property	The X property return the position on the page where the user should be scrolled to. The position information is based on the DPI used when loading the page.
FormElement::Y property	The Y property return the position on the page where the user should be scrolled to. The position information is based on the DPI used when loading the page.

FormElement::Action property

The Action property contains the behavior when a user clicks the form element.

.NET

```
string Action { get; }
```

JAVA

```
string getAction() throws IGRException;
```

PYTHON

```
@property  
def Action(self) -> string
```

C++17

```
std::wstring getAction() const;
```

RETURN VALUE

`string` : Returns the behavior when a user clicks the form element.

SEE ALSO

- [FormElement interface](#)

FormElement::ActionDest property

The ActionDest property contains the destination when a user interacts with certain form controls.

.NET

```
string ActionDest { get; }
```

JAVA

```
string getActionDest() throws IGRException;
```

PYTHON

```
@property  
def ActionDest(self) -> string
```

C++17

```
std::wstring getActionDest() const;
```

RETURN VALUE

`string` : Returns the destination when a user interacts with certain form controls.

SEE ALSO

- [FormElement interface](#)

FormElement::Alignment property

The Alignment property contains the text alignment that a viewer should use when presenting a single or multiline edit control.

.NET

```
int Alignment { get; }
```

JAVA

```
int getAlignment() throws IGRException;
```

PYTHON

```
@property
def Alignment(self) -> int
```

C++17

```
int getAlignment() const;
```

RETURN VALUE

int : Returns the text alignment that a viewer should use when presenting a single or multiline edit control, one of:

Name	Value
IGR_TEXT_ALIGN_LEFT	0x00
IGR_TEXT_ALIGN_RIGHT	0x01
IGR_TEXT_ALIGN_CENTER	0x02
IGR_TEXT_ALIGN_TOP	0x10
IGR_TEXT_ALIGN_MIDDLE	0x11
IGR_TEXT_ALIGN_BOTTOM	0x12

SEE ALSO

- [FormElement interface](#)

FormElement::Caption property

The Caption property returns the optional text associated with a check box or radio button.

.NET

```
string Caption { get; }
```

JAVA

```
string getCaption() throws IGRException;
```

PYTHON

```
@property  
def Caption(self) -> string
```

C++17

```
std::wstring getCaption() const;
```

RETURN VALUE

`string` : Returns the optional text associated with a check box or radio button.

SEE ALSO

- [FormElement interface](#)

FormElement::ElementType property

The ElementType property contains the control type that a viewer should use to collect input from a user.

.NET

```
int ElementType { get; }
```

JAVA

```
int getElementType() throws IGRException;
```

PYTHON

```
@property
def ElementType(self) -> int
```

C++17

```
int getElementType() const;
```

RETURN VALUE

`int` : Returns the control type that a viewer should use to collect input from a user, one of:

Name	Value
IGR_PAGE_FORM_ELEMENT_TYPE_BUTTON	0x0
IGR_PAGE_FORM_ELEMENT_TYPE_RADIOBUTTON	0x1
IGR_PAGE_FORM_ELEMENT_TYPE_CHECKBOX	0x2
IGR_PAGE_FORM_ELEMENT_TYPE_FILESELECT	0x3
IGR_PAGE_FORM_ELEMENT_TYPE_MULTILINE_TEXT	0x4
IGR_PAGE_FORM_ELEMENT_TYPE_SINGLELINE_TEXT	0x5
IGR_PAGE_FORM_ELEMENT_TYPE_COMBOBOX	0x6
IGR_PAGE_FORM_ELEMENT_TYPE_LISTBOX	0x7
IGR_PAGE_FORM_ELEMENT_TYPE_SIGNATURE	0x8
IGR_PAGE_FORM_ELEMENT_TYPE_PASSWORD	0x10

SEE ALSO

- [FormElement interface](#)

FormElement::Flags property

The Flags property contains a bit set of flags applicable to the control.

.NET

```
int ElementType { get; }
```

JAVA

```
int getElementType() throws IGRException;
```

PYTHON

```
@property
def ElementType(self) -> int
```

C++17

```
int getElementType() const;
```

RETURN VALUE

`int` : Returns a bit set of flags applicable to the control, one or more of:

Name	Value
IGR_PAGE_FORM_ELEMENT_FLAG_READONLY	0x1
IGR_PAGE_FORM_ELEMENT_FLAG_REQUIRED	0x2
IGR_PAGE_FORM_ELEMENT_FLAG_COMB	0x4
IGR_PAGE_FORM_ELEMENT_FLAG_CHECKED	0x8

SEE ALSO

- [FormElement interface](#)

FormElement::FontName property

The FontName property contains the font that a viewer should use when presenting a single or multiline edit control.

.NET

```
string FontName { get; }
```

JAVA

```
string getFontName() throws IGRException;
```

PYTHON

```
@property  
def FontName(self) -> string
```

C++17

```
std::wstring getFontName() const;
```

RETURN VALUE

`string` : Returns the font that a viewer should use when presenting a single or multiline edit control.

SEE ALSO

- [FormElement interface](#)

FormElement::FontSize property

The FontSize property contains the font size that a viewer should use when presenting a single or multiline edit control.

.NET

```
int FontSize { get; }
```

JAVA

```
int getFontSize() throws IGRException;
```

PYTHON

```
@property  
def FontSize(self) -> int
```

C++17

```
int getFontSize() const;
```

RETURN VALUE

`int` : Returns the font size that a viewer should use when presenting a single or multiline edit control.

SEE ALSO

- [FormElement interface](#)

FormElement::GetOption method

The GetOption method returns a [FormElementOption](#) object that represents a single selectable option in a list box or dropdown box.

PROTOTYPE

.NET

```
FormElementOption GetOption(int index)
```

JAVA

```
FormElementOption GetOption(int index) throws IGRException;
```

PYTHON

```
def GetOption(self, index: int) -> FormElementOption
```

C++17

```
FormElementOption GetOption(int index)
```

PARAMETERS

`index: int` : The index of the option.

RETURN VALUE

[FormElementOption](#) : Returns a [FormElementOption](#) object that represents a single selectable option in a list box or dropdown box.

SEE ALSO

- [FormElement interface](#)

FormElement::Height property

The Height property return the dimensions on the page where the form element should be positioned. The dimension information is based on the DPI used when loading the page.

.NET

```
int Height { get; }
```

JAVA

```
int getHeight() throws IGRException;
```

PYTHON

```
@property  
def Height(self) -> int
```

C++17

```
int getHeight() const;
```

RETURN VALUE

int : Integer containing the dimension in pixels.

SEE ALSO

- [FormElement interface](#)

FormElement::Name property

The Name property returns the qualified name for the form element.

.NET

```
string Name { get; }
```

JAVA

```
string getName() throws IGRException;
```

PYTHON

```
@property  
def Name(self) -> string
```

C++17

```
std::wstring getName() const;
```

RETURN VALUE

`string` : Returns the qualified name for the form element.

SEE ALSO

- [FormElement interface](#)

FormElement::OptionCount property

The OptionCount property contains the number of items that are available in list boxes and dropdown boxes.

.NET

```
int OptionCount { get; }
```

JAVA

```
int getOptionCount() throws IGRException;
```

PYTHON

```
@property  
def OptionCount(self) -> int
```

C++17

```
int getOptionCount() const;
```

RETURN VALUE

`int` : Returns the number of items that are available in list boxes and dropdown boxes.

SEE ALSO

- [FormElement interface](#)

FormElement::Rotation property

The Rotation property contains the rotation in degrees of the form element. Only rotation in 90 degree increments will be returned.

.NET

```
int Rotation { get; }
```

JAVA

```
int getRotation() throws IGRException;
```

PYTHON

```
@property  
def Rotation(self) -> int
```

C++17

```
int getRotation() const;
```

RETURN VALUE

int : Returns the rotation in degrees of the form element. Only rotation in 90 degree increments will be returned.

SEE ALSO

- [FormElement interface](#)

FormElement::Selected property

The Selected property contains the current selected item in a single-select list box, or drop down box.

.NET

```
int Selected { get; }
```

JAVA

```
int getSelected() throws IGRException;
```

PYTHON

```
@property  
def Selected(self) -> int
```

C++17

```
int getSelected() const;
```

RETURN VALUE

`int` : Returns the current selected item in a single-select list box, or drop down box.

SEE ALSO

- [FormElement interface](#)

FormElement::Value property

The Value property returns the stored value for the form element.

.NET

```
string Value { get; }
```

JAVA

```
string getValue() throws IGRException;
```

PYTHON

```
@property  
def Value(self) -> string
```

C++17

```
std::wstring getValue() const;
```

RETURN VALUE

`string` : Returns the stored value for the form element.

SEE ALSO

- [FormElement interface](#)

FormElement::Width property

The Width property return the dimensions on the page where the form element should be positioned. The dimension information is based on the DPI used when loading the page.

.NET

```
int Width { get; }
```

JAVA

```
int getWidth() throws IGRException;
```

PYTHON

```
@property  
def Width(self) -> int
```

C++17

```
int getWidth() const;
```

RETURN VALUE

`int` : Integer containing the dimension in pixels.

SEE ALSO

- [FormElement interface](#)

FormElement::X property

The X property return the position on the page where the user should be scrolled to. The position information is based on the DPI used when loading the page.

.NET

```
int X { get; }
```

JAVA

```
int getX() throws IGRException;
```

PYTHON

```
@property  
def X(self) -> int
```

C++17

```
int getX() const;
```

RETURN VALUE

int : Integer containing the coordinate in pixels.

SEE ALSO

- [FormElement interface](#)

FormElement::Y property

The Y property return the position on the page where the user should be scrolled to. The position information is based on the DPI used when loading the page.

.NET

```
int Y { get; }
```

JAVA

```
int getY() throws IGRException;
```

PYTHON

```
@property  
def Y(self) -> int
```

C++17

```
int getY() const;
```

RETURN VALUE

`int` : Integer containing the coordinate in pixels.

SEE ALSO

- [FormElement interface](#)

3.3.55 FormElementOption

FormElementOption interface

The FormElement interface represents a single form control for a given page. FormElements are currently support for static PDF documents.

FormElementOption::Name property	The Name property contains the display name for the option.
FormElementOption::Selected property	The Selected property contains whether the option is the selected value in the dropdown or list box.
FormElementOption::Value property	The Value property contains the value for the option.

FormElementOption::Name property

The Name property contains the display name for the option.

.NET

```
string Name { get; }
```

JAVA

```
string getName() throws IGRException;
```

PYTHON

```
@property  
def Name(self) -> string
```

C++17

```
std::wstring getName() const;
```

RETURN VALUE

`string` : Returns the display name for the option.

SEE ALSO

- [FormElementOption interface](#)

FormElementOption::Selected property

The Selected property contains whether the option is the selected value in the dropdown or list box.

.NET

```
bool Selected { get; }
```

JAVA

```
boolean getSelected() throws IGRException;
```

PYTHON

```
@property  
def Selected(self) -> bool
```

C++17

```
bool getSelected() const;
```

RETURN VALUE

bool : Returns if the option is the selected value in the dropdown or list box.

SEE ALSO

- [FormElementOption interface](#)

FormElementOption::Value property

The Value property contains the value for the option.

.NET

```
string Value { get; }
```

JAVA

```
string getValue() throws IGRException;
```

PYTHON

```
@property  
def Value(self) -> string
```

C++17

```
std::wstring getValue() const;
```

RETURN VALUE

`string` : Returns the value for the option.

SEE ALSO

- [FormElementOption interface](#)

3.3.56 FormKeyValue

FormKeyValue interface

The FormKeyValue interface is used to provided updated values for Forms.

This interface can be created directly.

FormKeyValue::Name property	The Name property contains the display name for the form to be updated.
FormKeyValue::Selected property	The Selected property contains the new boolean value representing the selected state for checkbox and radio button forms.
FormKeyValue::Value property	The Value property contains the new value for the form. If the form type is single select list box or a combo box this value must be set to one of the available choices. If the form type is a multi-select list box this value may be zero or more of the available choices separated delimited by a space character.

FormKeyValue::Name property

The Name property contains the display name for the form to be updated.

.NET

```
string Name { get; }
```

JAVA

```
string getName() throws IGRException;
```

PYTHON

```
@property  
def Name(self) -> string
```

C++17

```
std::wstring getName() const;
```

RETURN VALUE

`string` : Returns the display name for the form.

SEE ALSO

- [FormKeyValue interface](#)

FormKeyValue::Selected property

The Selected property contains the new boolean value representing the selected state for checkbox and radio button forms.

.NET

```
string Selected { get; }
```

JAVA

```
string getSelected() throws IGRException;
```

PYTHON

```
@property  
def Selected(self) -> string
```

C++17

```
std::wstring getSelected() const;
```

RETURN VALUE

`string` : Returns the form selected state.

SEE ALSO

- [FormKeyValue interface](#)

FormKeyValue::Value property

The Value property contains the new value for the form. If the form type is single select list box or a combo box this value must be set to one of the available choices. If the form type is a multi-select list box this value may be zero or more of the available choices separated delimited by a space character.

.NET

```
string Value { get; }
```

JAVA

```
string getValue() throws IGRException;
```

PYTHON

```
@property  
def Value(self) -> string
```

C++17

```
std::wstring getValue() const;
```

RETURN VALUE

`string` : Returns the value for the form

SEE ALSO

- [FormKeyValue interface](#)

3.3.57 Hyperlink

Hyperlink interface

The Hyperlink interface represents a single hyperlink on a given page.

To obtain this interface, call the [Page.GetNextHyperlink](#) method.

Hyperlink::DestBottom property	The DestBottom gets the bottom position of the hyperlink destination in pixels.
Hyperlink::DestFit property	The DestFit property stores the destination fit type. See PDF Reference for additional details.
Hyperlink::DestLeft property	The DestLeft gets the left position of the hyperlink destination in pixels.
Hyperlink::DestRight property	The DestRight gets the right position of the hyperlink destination in pixels.
Hyperlink::DestTop property	The DestTop gets the top position of the hyperlink destination in pixels.
Hyperlink::DestZoom property	The DestTop gets the top position of the hyperlink destination in pixels.
Hyperlink::Flags property	The Flags property stores the flags describing which fields are valid. This property is used in combination with the DestFit property. See PDF reference for more information.
Hyperlink::Height property	The Height property stores the height of the hyperlink element in pixels.
Hyperlink::HyperlinkType property	The HyperlinkType property stores the actions that should be performed when a user clicks on the link.
Hyperlink::PageNumber property	The PageNumber property stores the page number to navigate to if the HyperlinkType contains IGR_HYPERLINK_ACTION_GOTO.
Hyperlink::Ref property	The Ref property stores the reference within a document to navigate to when the user clicks on the link. This value is only valid if the hyperlink type contains the value IGR_HYPERLINK_ACTION_GOTO.
Hyperlink::Uri property	

	The Uri property stores the URI to navigate to when the user clicks on the link. This value is only valid if the hyperlink type contains the value IGR_HYPERLINK_ACTION_URI.
Hyperlink::Width property	The Width property stores the width of the hyperlink element in pixels.
Hyperlink::X property	The X property stores the horizontal position of the hyperlink element in pixels.
Hyperlink::Y property	The Y property stores the vertical position of the hyperlink element in pixels.

Hyperlink::DestBottom property

The DestBottom gets the bottom position of the hyperlink destination in pixels.

.NET

```
int DestBottom { get; }
```

JAVA

```
int getDestBottom() throws IGRException;
```

PYTHON

```
@property  
def DestBottom(self) -> int
```

C++17

```
int getDestBottom() const;
```

COM

```
[propget] HRESULT DestBottom([out, retval] int *result);
```

RETURN VALUE

int : The bottom position of the hyperlink destination in pixels.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::DestFit property

The DestFit property stores the destination fit type. See PDF Reference for additional details.

.NET

```
int DestFit { get; }
```

JAVA

```
int getDestFit() throws IGRException;
```

PYTHON

```
@property
def DestFit(self) -> int
```

C++17

```
int getDestFit() const;
```

COM

```
[propget] HRESULT DestFit([out, retval] int *result);
```

RETURN VALUE

int : The destination fit type. One of:

Name	Value
IGR_HYPERLINK_FIT_XYZ	0
IGR_HYPERLINK_FIT_FIT	1
IGR_HYPERLINK_FIT_FITH	2
IGR_HYPERLINK_FIT_FITV	3
IGR_HYPERLINK_FIT_FITR	4
IGR_HYPERLINK_FIT_FITB	5
IGR_HYPERLINK_FIT_FITBH	6
IGR_HYPERLINK_FIT_FITBV	7

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::DestLeft property

The DestLeft gets the left position of the hyperlink destination in pixels.

.NET

```
int DestLeft { get; }
```

JAVA

```
int getDestLeft() throws IGRException;
```

PYTHON

```
@property  
def DestLeft(self) -> int
```

C++17

```
int getDestLeft() const;
```

COM

```
[propget] HRESULT DestLeft([out, retval] int *result);
```

RETURN VALUE

int : The left position of the hyperlink destination in pixels.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::DestRight property

The DestRight gets the right position of the hyperlink destination in pixels.

.NET

```
int DestRight { get; }
```

JAVA

```
int getDestRight() throws IGREException;
```

PYTHON

```
@property  
def DestRight(self) -> int
```

C++17

```
int getDestRight() const;
```

COM

```
[propget] HRESULT DestRight([out, retval] int *result);
```

RETURN VALUE

`int` : The right position of the hyperlink destination in pixels.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::DestTop property

The DestTop gets the top position of the hyperlink destination in pixels.

.NET

```
int DestTop { get; }
```

JAVA

```
int getDestTop() throws IGRException;
```

PYTHON

```
@property  
def DestTop(self) -> int
```

C++17

```
int getDestTop() const;
```

COM

```
[propget] HRESULT DestTop([out, retval] int *result);
```

RETURN VALUE

`int` : The top position of the hyperlink destination in pixels.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::DestZoom property

The DestTop gets the top position of the hyperlink destination in pixels.

.NET

```
int Zoom { get; }
```

JAVA

```
int getZoom() throws IGREException;
```

PYTHON

```
@property  
def Zoom(self) -> int
```

C++17

```
int getZoom() const;
```

COM

```
[propget] HRESULT Zoom([out, retval] int *result);
```

RETURN VALUE

`int` : The zoom property.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::Flags property

The Flags property stores the flags describing which fields are valid. This property is used in combination with the DestFit property. See PDF reference for more information.

.NET

```
int Flags { get; }
```

JAVA

```
int getFlags() throws IGRException;
```

PYTHON

```
@property
def Flags(self) -> int
```

C++17

```
int getFlags() const;
```

COM

```
[propget] HRESULT Flags([out, retval] int *result);
```

RETURN VALUE

`int` : Zero or more flags describing which destination properties have a value.

Name	Value
IGR_HYPERLINK_FLAGS_CHANGE_LEFT	0x1
IGR_HYPERLINK_FLAGS_CHANGE_TOP	0x2
IGR_HYPERLINK_FLAGS_CHANGE_ZOOM	0x4

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::Height property

The Height property stores the height of the hyperlink element in pixels.

.NET

```
int Height { get; }
```

JAVA

```
int getHeight() throws IGRException;
```

PYTHON

```
@property  
def Height(self) -> int
```

C++17

```
int getHeight() const;
```

COM

```
[propget] HRESULT Height([out, retval] int *result);
```

RETURN VALUE

`int` : The height of the hyperlink element in pixels.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::HyperlinkType property

The HyperlinkType property stores the actions that should be performed when a user clicks on the link.

.NET

```
int HyperlinkType { get; }
```

JAVA

```
int getHyperlinkType() throws IGRException;
```

PYTHON

```
@property
def HyperlinkType(self) -> int
```

C++17

```
int getHyperlinkType() const;
```

COM

```
[propget] HRESULT HyperlinkType([out, retval] int *result);
```

RETURN VALUE

`int` : Integer value containing the IGR_HYPERLINK_ACTION_TYPE enumerated value, one of:

Name	Value
IGR_HYPERLINK_ACTION_GOTO	0x1
IGR_HYPERLINK_ACTION_URI	0x2

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::PageNumber property

The PageNumber property stores the page number to navigate to if the HyperlinkType contains IGR_HYPERLINK_ACTION_GOTO.

.NET

```
int PageNumber { get; }
```

JAVA

```
int getPageNumber() throws IGRException;
```

PYTHON

```
@property  
def PageNumber(self) -> int
```

C++17

```
int getPageNumber() const;
```

COM

```
[propget] HRESULT PageNumber([out, retval] int *result);
```

RETURN VALUE

`int` : The page number to navigate to.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::Ref property

The Ref property stores the reference within a document to navigate to when the user clicks on the link. This value is only valid if the hyperlink type contains the value IGR_HYPERLINK_ACTION_GOTO.

.NET

```
string Ref { get; }
```

JAVA

```
string getRef() throws IGRException;
```

PYTHON

```
@property  
def Ref(self) -> string
```

C++17

```
std::wstring getRef() const;
```

COM

```
[propget] HRESULT Ref([out, retval] BSTR *result);
```

RETURN VALUE

`string` : The reference within a document to navigate to when the user clicks on the link.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::Uri property

The Uri property stores the URI to navigate to when the user clicks on the link. This value is only valid if the hyperlink type contains the value IGR_HYPERLINK_ACTION_URI.

.NET

```
string Uri { get; }
```

JAVA

```
string getUri() throws IGRException;
```

PYTHON

```
@property  
def Uri(self) -> string
```

C++17

```
std::wstring getUri() const;
```

COM

```
[propget] HRESULT Uri([out, retval] BSTR *result);
```

RETURN VALUE

`string` : The URI to navigate to when the user clicks on the link.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::Width property

The Width property stores the width of the hyperlink element in pixels.

.NET

```
int Width { get; }
```

JAVA

```
int getWidth() throws IGREException;
```

PYTHON

```
@property  
def Width(self) -> int
```

C++17

```
int getWidth() const;
```

COM

```
[propget] HRESULT Width([out, retval] int *result);
```

RETURN VALUE

`int` : The width of the hyperlink element in pixels.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::X property

The X property stores the horizontal position of the hyperlink element in pixels.

.NET

```
int X { get; }
```

JAVA

```
int getX() throws IGRException;
```

PYTHON

```
@property  
def X(self) -> int
```

C++17

```
int getX() const;
```

COM

```
[propget] HRESULT X([out, retval] int *result);
```

RETURN VALUE

`int` : The horizontal position of the hyperlink element in pixels.

SEE ALSO

- [Hyperlink Interface](#)

Hyperlink::Y property

The Y property stores the vertical position of the hyperlink element in pixels.

.NET

```
int Y { get; }
```

JAVA

```
int getY() throws IGRException;
```

PYTHON

```
@property  
def Y(self) -> int
```

C++17

```
int getY() const;
```

COM

```
[propget] HRESULT Y([out, retval] int *result);
```

RETURN VALUE

`int` : The vertical position of the hyperlink element in pixels.

SEE ALSO

- [Hyperlink Interface](#)

3.3.58 IGRFormat

IGRFormat interface

Represents a file-type as returned by `GetSupportedFormats` .

IGRFormat::ConfigName property	Gets the configuration name of the format.
IGRFormat::DisplayName property	Gets the display name of the format.
IGRFormat::FileTypeCategory property	Gets the file type category of the format.
IGRFormat::Id property	Gets the ID of the format.
IGRFormat::MimeType property	Gets the MIME type of the format.
IGRFormat::ShortName property	Gets the short name of the format.

SEE ALSO

- [DocumentFilters::GetSupportedFormats](#)

IGRFormat::ConfigName property

Gets the configuration name of the format.

.NET

```
string ConfigName { get; }
```

JAVA

```
string getConfigName() throws IGRException;
```

PYTHON

```
@property  
def ConfigName(self) -> string
```

C++17

```
std::wstring getConfigName() const;
```

RETURN VALUE

`string` : The configuration name for the format.

IGRFormat::DisplayName property

Gets the display name of the format.

.NET

```
string DisplayName { get; }
```

JAVA

```
string getDisplayName() throws IGRException;
```

PYTHON

```
@property  
def DisplayName(self) -> string
```

C++17

```
std::wstring getDisplayName() const;
```

RETURN VALUE

`string` : The display name for the format.

IGRFormat::FileTypeCategory property

Gets the file type category of the format.

.NET

```
int FileTypeCategory { get; }
```

JAVA

```
int getFileTypeCategory() throws IGRException;
```

PYTHON

```
@property  
def FileTypeCategory(self) -> int
```

C++17

```
int getFileTypeCategory() const;
```

RETURN VALUE

`int` : The FileTypeCategory for the format. See [IGR_FILETYPE_CATEGORY](#) for details.

IGRFormat::Id property

Gets the ID of the format.

.NET

```
int Id { get; }
```

JAVA

```
int getId() throws IGRException;
```

PYTHON

```
@property  
def Id(self) -> int
```

C++17

```
int getId() const;
```

RETURN VALUE

`int` : The ID for the format.

IGRFormat::MimeType property

Gets the MIME type of the format.

.NET

```
string MimeType { get; }
```

JAVA

```
string getMimeType() throws IGRException;
```

PYTHON

```
@property  
def MimeType(self) -> string
```

C++17

```
std::wstring getMimeType() const;
```

RETURN VALUE

`string` : The MIME type for the format.

IGRFormat::ShortName property

Gets the short name of the format.

.NET

```
string ShortName { get; }
```

JAVA

```
string getShortName() throws IGRException;
```

PYTHON

```
@property  
def ShortName(self) -> string
```

C++17

```
std::wstring getShortName() const;
```

RETURN VALUE

`string` : The short name for the format.

3.3.59 IGROption

IGROption interface

Represents a configuration option as returned by `GetAvailableOptions`.

<code>IGROption::DefaultValue</code> property	Gets the default value of the option.
<code>IGROption::Description</code> property	Gets the description of the option.
<code>IGROption::DisplayName</code> property	Gets the display name of the option.
<code>IGROption::Flags</code> property	Gets the flags of the option.
<code>IGROption::PossibleValues</code> property	Gets the possible values of the option.
<code>IGROption::Type</code> property	Gets the type of the option.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 // Call the GetAvailableOptions method
8 var availableOptions = api.GetAvailableOptions();
9
10 // Iterate over the available options and print their details
11 foreach (Option option in availableOptions) {
12     Console.WriteLine("Option Display Name: " + option.DisplayName);
13     Console.WriteLine("Option Description: " + option.Description);
14     Console.WriteLine("Option Default Value: " + option.DefaultValue);
15     Console.WriteLine("Option Type: " + option.Type);
16     Console.WriteLine("Option Flags: " + option.Flags);
17     Console.WriteLine("Option Possible Values: " + string.Join(", ", option.PossibleValues));
18     Console.WriteLine();
19 }
```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         DocumentFilters df = new DocumentFilters();
6         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
7
8         // Call the GetAvailableOptions method
9         VectIGROption availableOptions = df.GetAvailableOptions();
10
11        // Iterate over the available options and print their details
12        for (int i = 0; i < availableOptions.size(); i++) {
13            IGROption option = availableOptions.get(i);
14            System.out.println("Option Display Name: " + option.GetDisplayName());
15            System.out.println("Option Description: " + option.GetDescription());
16            System.out.println("Option Default Value: " + option.GetDefaultValue());
17            System.out.println("Option Type: " + option.GetType());
18            System.out.println("Option Flags: " + option.GetFlags());
19            System.out.println("Option Possible Values: " + option.GetPossibleValuesStr());
20            System.out.println();
21        }
22    }
23 }

```

PYTHON

```

1 from DocumentFilters import *
2
3 df = DocumentFilters()
4 df.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 # Call the getAvailableOptions method
7 available_options = df.GetAvailableOptions()
8
9 # Iterate over the available options and print their details
10 for option in available_options:
11     print("Option Display Name:", option.DisplayName)
12     print("Option Description:", option.Description)
13     print("Option Default Value:", option.DefaultValue)
14     print("Option Type:", option.Type)
15     print("Option Flags:", option.Flags)
16     print("Option Possible Values:", option.PossibleValues)
17     print()

```

C++17

```

1  #include <DocumentFiltersObjects.h>
2
3  int main(int argc, char** argv) {
4      Hyland::DocFilters::DocumentFilters df;
5      df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7      // Call the getAvailableOptions method
8      const std::vector<Hyland::DocFilters::Option>& availableOptions = df.getAvailableOptions();
9
10     // Iterate over the available options and print their details
11     for (const auto& option : availableOptions) {
12         std::wcout << L"Option Display Name: " << option.getDisplayName() << std::endl;
13         std::wcout << L"Option Description: " << option.getDescription() << std::endl;
14         std::wcout << L"Option Default Value: " << option.getDefaultValue() << std::endl;
15         std::wcout << L"Option Type: " << option.getType() << std::endl;
16         std::wcout << L"Option Flags: " << option.getFlags() << std::endl;
17
18         // Get possible values
19         const std::vector<std::wstring>& possibleValues = option.getPossibleValues();
20         std::wcout << L"Option Possible Values: ";
21         for (const auto& value : possibleValues) {
22             std::wcout << value << L", ";
23         }
24         std::wcout << std::endl;
25     }
26
27     return 0;
28 }

```

SEE ALSO

- [DocumentFilters::GetAvailableOptions](#)

IGROption::DefaultValue property

Gets the default value of the option.

.NET

```
string DefaultValue { get; }
```

JAVA

```
string getDefaultValue() throws IGRException;
```

PYTHON

```
@property  
def DefaultValue(self) -> string
```

C++17

```
std::wstring getDefaultValue() const;
```

RETURN VALUE

`string` : The default value for the option.

IGROption::Description property

Gets the description of the option.

.NET

```
string Description { get; }
```

JAVA

```
string getDescription() throws IGRException;
```

PYTHON

```
@property  
def Description(self) -> string
```

C++17

```
std::wstring getDescription() const;
```

RETURN VALUE

`string` : The description for the option.

IGROption::DisplayName property

Gets the display name of the option.

.NET

```
string DisplayName { get; }
```

JAVA

```
string getDisplayName() throws IGRException;
```

PYTHON

```
@property  
def DisplayName(self) -> string
```

C++17

```
std::wstring getDisplayName() const;
```

RETURN VALUE

`string` : The display name for the option.

IGROption::Flags property

Gets the flags of the option.

.NET

```
int Flags { get; }
```

JAVA

```
int getFlags() throws IGRException;
```

PYTHON

```
@property  
def Flags(self) -> int
```

C++17

```
int getFlags() const;
```

RETURN VALUE

`int` : The flags for the option.

IGROption::PossibleValues property

Gets the possible values of the option.

.NET

```
IEnumerable<string> PossibleValues { get; }
```

JAVA

```
string[] getPossibleValues() throws IGRException;
```

PYTHON

```
@property  
def PossibleValues(self) -> string[]
```

C++17

```
std::vector<std::wstring> getPossibleValues() const;
```

RETURN VALUE

`string[]` : The possible values for the option.

IGROption::Type property

Gets the type of the option.

.NET

```
string Type { get; }
```

JAVA

```
string getType() throws IGRException;
```

PYTHON

```
@property  
def Type(self) -> string
```

C++17

```
std::wstring getType() const;
```

RETURN VALUE

`string` : The type for the option.

3.3.60 Page

Page interface

The Page interface represents a single page in an image laid-out document. The page allows access to the words on a page, as well as the ability to render it onto a canvas such as TIFF, PNG, or PDF.

To obtain this interface, call the `Extractor::GetPage`.

Page::Annotations property	Gets an enumerable collections of annotations on the page.
Page::Close method	The Close method releases any resources associated with the page.
Page::Compare method	The Compare method allows you to compare two pages returning the differences.
Page::FormElements property	Returns an enumerable collections of form elements on the page.
Page::GetAnnotation method	The GetAnnotation method is used to retrieve the annotation on the page at the specified index.
Page::GetAnnotationCount method	The GetAnnotationCount method returns the number of annotations that are on a page. The annotations can be enumerated using the GetFirstAnnotation and GetNextAnnotation methods, or accessed via index using the GetAnnotation method.
Page::GetAttribute method	IGR_Get_Page_Attribute returns style or properties of an open page; see <page> under Structured XML for a full list of options.
Page::GetFirstAnnotation method	The GetFirstAnnotation method is used to enumerate the annotations that are on the page.
Page::GetFirstFormElement method	The GetFirstFormElement method is used to enumerate the form controls that are on the page.
Page::GetFirstHyperlink method	The GetFirstHyperlink method is used to enumerate the hyperlinks that are on the page.
Page::GetFirstImage method	The GetFirstImage method is used to enumerate the embedded images that are on the page. This method is useful if the page images are to be extracted and stored in separate files. These methods are not needed if the page is to be rendered into an image output canvas such as PNG, TIFF, or PDF.

Page::GetFirstWord method	The GetFirstWord method is used to enumerate all the words on the current page. GetFirstWord resets the enumeration back to the beginning. The method will return a reference to a Word object, or NULL if there are no more words.
Page::GetNextAnnotation method	The GetNextAnnotation method is used to enumerate the annotations that are on the page.
Page::GetNextFormElement method	The GetNextFormElement method is used to enumerate the form controls that are on the page.
Page::GetNextHyperlink method	The GetNextHyperlink method is used to enumerate the hyperlinks that are on the page.
Page::GetNextImage method	The GetNextImage method is used to enumerate the embedded images that are on the page. This method is useful if the page images are to be extracted and stored in separate files. These methods are not needed if the page is to be rendered into an image output canvas such as PNG, TIFF, or PDF.
Page::GetNextWord method	The GetNextWord method is used to enumerate all the words on the current page. The method will return a reference to a Word object, or NULL if there are no more words.
Page::GetRootPageElement method	Returns the root page element for this page.
Page::Handle property	Returns the pages IGR_HPAGE handle that can be used with the C API.
Page::Height property	The height property return the dimensions of a page in pixels.
Page::Hyperlinks property	Returns an enumerable collections of hyperlinks on the page.
Page::Images property	Returns an enumerable collections of image elements on the page.
Page::Redact method	Redact removes a range of words and blacks out the location for the specified range from the page.
Page::Text property	The Text property returns all the text contained on the page.
Page::Width property	The width property return the dimensions of a page in pixels.
Page::WordCount property	The WordCount property returns the number of “Word”s that are on a page. The words can be enumerated using the GetFirstWord and GetNextWord methods.

Page::Words property	Returns an enumerable collections of form elements on the page.

Page::Annotations property

Gets an enumerable collections of annotations on the page.

.NET

```
IEnumerable<Annotation> Annotations { get; }
```

PYTHON

```
@property  
def Annotations(self) -> list
```

C++17

```
Hyland::DocFilters::Page::annotations_t getAnnotations() const;
```

RETURN VALUE

`IEnumerable<Annotation>` : Returns an enumerable set of annotations

Page::Close method

The Close method releases any resources associated with the page.

PROTOTYPE

.NET

```
void Close()
```

JAVA

```
void Close() throws IGRException;
```

PYTHON

```
def Close(self) -> void
```

C++17

```
void Close()
```

ADDITIONAL INFORMATION

This method should be called when finished working with a page to release its resources. The method will be internally called when the instance itself is released. Calling this method on closed pages has no effect.

Resources can also be managed with the `using IDisposable` pattern in C#, or the `try-with-resources` pattern in Java.

SEE ALSO

- [Page interface](#)
- [Extractor::GetPage method](#)
- [Extractor::GetFirstPage method](#)

Page::Compare method

The Compare method allows you to compare two pages returning the differences.

OVERLOADS

<code>Compare(Page, CompareSettings)</code>	The Compare method allows you to compare two pages returning the differences.
<code>Compare(Page, System.Drawing.RectangleF, CompareSettings)</code>	The Compare method allows you to compare two pages, applying margins, returning the differences.
<code>Compare(Page, System.Drawing.RectangleF, System.Drawing.RectangleF, CompareSettings)</code>	The Compare method allows you to compare two pages, applying margins, returning the differences.

COMPARE(PAGE, COMPARESETTINGS)

Prototype

.NET

```
CompareResults Compare(Page otherPage, CompareSettings settings)
```

JAVA

```
CompareResults Compare(Page otherPage, CompareSettings settings) throws IGREException;
```

PYTHON

```
def Compare(self, otherPage: Page, settings: CompareSettings) -> CompareResults
```

C++17

```
CompareResults Compare(Page otherPage, CompareSettings settings)
```

Parameters

- `otherPage` : `Page` : Provide the other `Page` to compare.
- `settings` : `CompareSettings` : Provide the settings that control the compare logic.

Return value

`CompareResults` : A new instance of a `CompareResults` interface.

COMPARE(PAGE, SYSTEM.DRAWING.RECTANGLEF, COMPARESETTINGS)**Prototype****.NET**

```
CompareResults Compare(Page otherPage, System.Drawing.RectangleF margins, CompareSettings settings)
```

C++17

```
CompareResults Compare(Page otherPage, System.Drawing.RectangleF margins, CompareSettings settings)
```

Parameters

otherPage : [Page](#) : Provide the other [Page](#) to compare.

margins : [System.Drawing.RectangleF](#) : The margins to apply to both pages.

settings : [CompareSettings](#) : Provide the settings that control the compare logic.

Return value

[CompareResults](#) : A new instance of a [CompareResults](#) interface.

COMPARE(PAGE, SYSTEM.DRAWING.RECTANGLEF, SYSTEM.DRAWING.RECTANGLEF, COMPARESETTINGS)**Prototype****.NET**

```
CompareResults Compare(Page otherPage, System.Drawing.RectangleF leftMargins, System.Drawing.RectangleF rightMargins, CompareSettings settings)
```

C++17

```
CompareResults Compare(Page otherPage, System.Drawing.RectangleF leftMargins, System.Drawing.RectangleF rightMargins, CompareSettings settings)
```

Parameters

otherPage : [Page](#) : Provide the other [Page](#) to compare.

leftMargins : [System.Drawing.RectangleF](#) : The margins to apply to the left/original page.

rightMargins : [System.Drawing.RectangleF](#) : The margins to apply to the right/revised page.

settings : [CompareSettings](#) : Provide the settings that control the compare logic.

Return value

[CompareResults](#) : A new instance of a [CompareResults](#) interface.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using (var doc1 = api.OpenExtractor("original.docx", OpenMode.Paginated))
7 using (var doc2 = api.OpenExtractor("revision.docx", OpenMode.Paginated))
8 using (var page1 = doc1.GetPage(0))
9 using (var page2 = doc2.GetPage(0))
10 using (var compare = page1.Compare(page2))
11 {
12     while (compare.MoveNext())
13     {
14         var diff = compare.Current;
15         // work with diff...
16     }
17 }
```

C++17

```

1  #include <DocumentFilters.h>
2  #include <string.h>
3  #include <string>
4
5  #define UCS2(src) reinterpret_cast<const IGR_UCS2 *>(std::u16string(src).c_str())
6
7  int main(int argc, char **argv)
8  {
9      Instance_Status_Block isb = {0};
10     Error_Control_Block ecb = {0};
11     IGR_SHORT df;
12     IGR_LONG caps, type, pageCount = 0;
13     IGR_HDOC doc1_handle = 0, doc2_handle = 0;
14     IGR_HPAGE doc1_page1_handle = 0, doc2_page1_handle = 0;
15     IGR_HTEXTCOMPARE compare = 0;
16
17     strncpy(isb.Licensee_ID1, "YOUR_LICENSE_KEY_HERE", sizeof(isb.Licensee_ID1) - 1);
18     Init_Instance(0, ".", &isb, &df, &ecb);
19
20     if ((res = IGR_Open_File_Ex(UCS2(u"original.doc"), IGR_FORMAT_IMAGE, UCS2(u""), &caps, &type,
&doc1_handle, &ecb)) == IGR_OK
21         && (res = IGR_Open_File_Ex(UCS2(u"revised.doc"), IGR_FORMAT_IMAGE, UCS2(u""), &caps, &type,
&doc2_handle, &ecb)) == IGR_OK
22         && (res = IGR_Open_Page(doc1_handle, 0, &doc1_page1_handle, &ecb)) == IGR_OK
23         && (res = IGR_Open_Page(doc2_handle, 0, &doc2_page1_handle, &ecb)) == IGR_OK)
24     {
25         IGR_Text_Compare_Settings settings = { sizeof(IGR_Text_Compare_Settings) };
26         IGR_Compare_Documents_Difference diff = { sizeof(IGR_Compare_Documents_Difference) };
27
28         if ((ret = IGR_Text_Compare_Pages(doc1_page1_handle, nullptr, doc2_page1_handle, nullptr,
&settings, &compare, &ecb)) == IGR_OK)
29         {
30             while (IGR_Text_Compare_Next(compare, &diff, &ecb) == IGR_OK)
31             {
32                 // ... work with diff object
33
34                 IGR_Text_Compare_Difference_Dispose(&diff, &ecb);
35             }
36
37             IGR_Text_Compare_Close(compare, &ecb);
38         }
39     }
40
41     if (doc1_page1_handle)
42         IGR_Close_Page(doc1_page1_handle, &ecb);
43     if (doc2_page1_handle)
44         IGR_Close_Page(doc2_page1_handle, &ecb);
45     if (doc1_handle)
46         IGR_Close_File(doc1_handle, &ecb);
47     if (doc2_handle)
48         IGR_Close_File(doc2_handle, &ecb);
49     return 0;
50 }

```

SEE ALSO

- [CompareDocumentSettings interface](#)

- [Page interface](#)

Page::FormElements property

Returns an enumerable collections of form elements on the page.

.NET

```
IEnumerable<FormElement> FormElements { get; }
```

PYTHON

```
@property  
def FormElements(self) -> list
```

C++17

```
std::vector<FormElement> formElements() const;
```

RETURN VALUE

`IEnumerable<FormElement>` : An enumerable collection of form element objects.

SEE ALSO

- [FormElement interface](#)

Page::GetAnnotation method

The GetAnnotation method is used to retrieve the annotation on the page at the specified index.

PROTOTYPE

.NET

```
Annotation GetAnnotation(int index)
```

JAVA

```
Annotation GetAnnotation(int index) throws IGRException;
```

PYTHON

```
def GetAnnotation(self, index: int) -> Annotation
```

C++17

```
Annotation GetAnnotation(int index)
```

PARAMETERS

`index: int` : The index of the annotation to return.

RETURN VALUE

`Annotation` : Returns the annotation on the current page at the specified index, or NULL if the index is out of range.

SEE ALSO

- [Page interface](#)
- [Annotation interface](#)

Page::GetAnnotationCount method

The GetAnnotationCount method returns the number of annotations that are on a page. The annotations can be enumerated using the GetFirstAnnotation and GetNextAnnotation methods, or accessed via index using the GetAnnotation method.

PROTOTYPE

.NET

```
int GetAnnotationCount()
```

JAVA

```
int GetAnnotationCount() throws IGREException;
```

PYTHON

```
def GetAnnotationCount(self) -> int
```

C++17

```
int GetAnnotationCount()
```

RETURN VALUE

int : Integer containing the number of annotations on the page.

SEE ALSO

- [Page interface](#)
- [Annotation interface](#)

Page::GetAttribute method

[IGR_Get_Page_Attribute](#) returns style or properties of an open page; see <page> under Structured XML for a full list of options.

PROTOTYPE

.NET

```
string GetAttribute(string name)
```

JAVA

```
string GetAttribute(string name) throws IGRException;
```

PYTHON

```
def GetAttribute(self, name: string) -> string
```

C++17

```
std::wstring getAttribute(const std::wstring& name)
```

PARAMETERS

name: string : Name of attribute to return.

RETURN VALUE

string : String of the requested attribute.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.GetExtractor("filename.doc");
7 doc.Open(OpenMode.Paginated);
8
9 foreach (var page in doc.Pages)
10 {
11     using (page)
12     {
13         var dpi = page.GetAttribute("SourceDpiX");
14     }
15 }
```

ADDITIONAL INFORMATION

In addition to the <page> styles and properties, you can use the following attribute names.

Name	Description
FONTs	A semicolon delimited list of fonts present on the current page.
FONT_MAPPINGS	A semicolon delimited list of "OriginalName=MappedName" fonts on the current page.
DOCUMENT_FONTs	A semicolon delimited list of fonts processed in the document, up to the point of the call.
DOCUMENT_FONT_MAPPINGS	A semicolon delimited list of "OriginalName=MappedName" fonts processed in the document, up to the point of the call.
SOURCEWIDTH	Width of source document if it is a raster image.
SOURCEHEIGHT	Height of source document if it is a raster image.
OPTIONS	Document options currently in effect.

Note To retrieve all fonts for a document, use DOCUMENT_FONTs or DOCUMENT_FONT_MAPPINGS after processing the last page of the document. These attributes are more efficient than using FONTs or FONT_MAPPINGS per page.

Page::GetFirstAnnotation method

The GetFirstAnnotation method is used to enumerate the annotations that are on the page.

PROTOTYPE

.NET

```
Annotation GetFirstAnnotation()
```

JAVA

```
Annotation GetFirstAnnotation() throws IGREException;
```

PYTHON

```
def GetFirstAnnotation(self) -> Annotation
```

RETURN VALUE

[Annotation](#) : Returns the first annotation on the current page, or NULL if no annotations are present.

SEE ALSO

- [Page interface](#)
- [Annotation interface](#)
- [Annotation::GetNextAnnotation](#)

Page::GetFirstFormElement method

The GetFirstFormElement method is used to enumerate the form controls that are on the page.

PROTOTYPE

.NET

```
FormElement GetFirstFormElement()
```

JAVA

```
FormElement GetFirstFormElement() throws IGREException;
```

PYTHON

```
def GetFirstFormElement(self) -> FormElement
```

COM

```
HRESULT GetFirstFormElement([out, retval] FormElement* *result)
```

RETURN VALUE

[FormElement](#) : Returns the first form element on the current page, or NULL if no forms are present.

SEE ALSO

- [Page interface](#)
- [FormElement interface](#)
- [Page::GetNextFormElement](#)

Page::GetFirstHyperlink method

The GetFirstHyperlink method is used to enumerate the hyperlinks that are on the page.

PROTOTYPE

.NET

```
Hyperlink GetFirstHyperlink()
```

JAVA

```
Hyperlink GetFirstHyperlink() throws IGRException;
```

PYTHON

```
def GetFirstHyperlink(self) -> Hyperlink
```

C++17

```
Hyperlink GetFirstHyperlink()
```

COM

```
HRESULT GetFirstHyperlink([out, retval] Hyperlink* *result)
```

RETURN VALUE

[Hyperlink](#) : Returns the first hyperlink on the current page, or NULL if no hyperlinks are present.

SEE ALSO

- [Page interface](#)
- [Hyperlink interface](#)
- [Page::GetNextHyperlink](#)

Page::GetFirstImage method

The GetFirstImage method is used to enumerate the embedded images that are on the page. This method is useful if the page images are to be extracted and stored in separate files. These methods are not needed if the page is to be rendered into an image output canvas such as PNG, TIFF, or PDF.

PROTOTYPE

.NET

```
SubFile GetFirstImage()
```

JAVA

```
SubFile GetFirstImage() throws IGRException;
```

PYTHON

```
def GetFirstImage(self) -> SubFile
```

C++17

```
SubFile GetFirstImage()
```

COM

```
HRESULT GetFirstImage([out, retval] SubFile* *result)
```

RETURN VALUE

[SubFile](#) : Reference to a SubFile object, SubFile::Close must be called when finished with the object.

SEE ALSO

- [Page interface](#)
- [SubFile interface](#)
- [Page::GetNextImage](#)

Page::GetFirstWord method

The GetFirstWord method is used to enumerate all the words on the current page. GetFirstWord resets the enumeration back to the beginning. The method will return a reference to a Word object, or NULL if there are no more words.

PROTOTYPE

.NET

```
Word GetFirstWord()
```

JAVA

```
Word GetFirstWord() throws IGREException;
```

PYTHON

```
def GetFirstWord(self) -> Word
```

C++17

```
Word GetFirstWord()
```

C++11

```
Word* GetFirstWord()
```

COM

```
HRESULT GetFirstWord([out, retval] Word* *result)
```

RETURN VALUE

Word : Reference to a Word object.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.Paginated);
9
10 foreach (var page in doc.Pages)
11 {
12     using (page)
13     {
14         // Iterate over all words on the page
15         for (Word word = page.GetFirstWord(); word != null; word = page.GetNextWord())
16         {
17             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
18         }
19
20         // Alternatively, use the iterator
21         foreach (Word word in page.Words)
22         {
23             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
24         }
25     }
26 }
27 }
```

SEE ALSO

- [Page interface](#)
- [Word interface](#)
- [Page::GetNextWord](#)

Page::GetNextAnnotation method

The GetNextAnnotation method is used to enumerate the annotations that are on the page.

PROTOTYPE

.NET

```
Annotation GetNextAnnotation()
```

JAVA

```
Annotation GetNextAnnotation() throws IGRException;
```

PYTHON

```
def GetNextAnnotation(self) -> Annotation
```

RETURN VALUE

[Annotation](#) : Returns the next annotation on the current page, or NULL if no annotations are present.

SEE ALSO

- [Page interface](#)
- [Annotation interface](#)
- [Annotation::GetFirstAnnotation](#)

Page::GetNextFormElement method

The GetNextFormElement method is used to enumerate the form controls that are on the page.

PROTOTYPE

.NET

```
FormElement GetNextFormElement()
```

JAVA

```
FormElement GetNextFormElement() throws IGREException;
```

PYTHON

```
def GetNextFormElement(self) -> FormElement
```

COM

```
HRESULT GetNextFormElement([out, retval] FormElement* *result)
```

RETURN VALUE

FormElement : Returns the next form element on the current page, or NULL if no more forms elements are available.

SEE ALSO

- [Page interface](#)
- [FormElement interface](#)
- [Page::GetFirstFormElement](#)

Page::GetNextHyperlink method

The GetNextHyperlink method is used to enumerate the hyperlinks that are on the page.

PROTOTYPE

.NET

```
Hyperlink GetNextHyperlink()
```

JAVA

```
Hyperlink GetNextHyperlink() throws IGREException;
```

PYTHON

```
def GetNextHyperlink(self) -> Hyperlink
```

C++17

```
Hyperlink GetNextHyperlink()
```

COM

```
HRESULT GetNextHyperlink([out, retval] Hyperlink* *result)
```

RETURN VALUE

[Hyperlink](#) : Returns the next hyperlink on the current page, or NULL if no more hyperlinks are available.

SEE ALSO

- [Page interface](#)
- [Hyperlink interface](#)
- [Page::GetFirstHyperlink](#)

Page::GetNextImage method

The GetNextImage method is used to enumerate the embedded images that are on the page. This method is useful if the page images are to be extracted and stored in separate files. These methods are not needed if the page is to be rendered into an image output canvas such as PNG, TIFF, or PDF.

PROTOTYPE

.NET

```
SubFile GetNextImage()
```

JAVA

```
SubFile GetNextImage() throws IGREException;
```

PYTHON

```
def GetNextImage(self) -> SubFile
```

COM

```
HRESULT GetNextImage([out, retval] SubFile* *result)
```

RETURN VALUE

SubFile : Reference to a SubFile object, SubFile::Close must be called when finished with the object.

SEE ALSO

- [Page interface](#)
- [SubFile interface](#)
- [Page::GetFirstImage](#)

Page::GetNextWord method

The GetNextWord method is used to enumerate all the words on the current page. The method will return a reference to a Word object, or NULL if there are no more words.

PROTOTYPE

.NET

```
Word GetNextWord()
```

JAVA

```
Word GetNextWord() throws IGREException;
```

PYTHON

```
def GetNextWord(self) -> Word
```

C++17

```
Word GetNextWord()
```

C++11

```
Word* GetNextWord()
```

COM

```
HRESULT GetNextWord([out, retval] Word* *result)
```

RETURN VALUE

Word : Reference to a Word object.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.Paginated);
9
10 foreach (var page in doc.Pages)
11 {
12     using (page)
13     {
14         // Iterate over all words on the page
15         for (Word word = page.GetFirstWord(); word != null; word = page.GetNextWord())
16         {
17             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
18         }
19
20         // Alternatively, use the iterator
21         foreach (Word word in page.Words)
22         {
23             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
24         }
25     }
26 }
27 }
```

SEE ALSO

- [Page interface](#)
- [Word interface](#)
- [Page::GetFirstWord](#)

Page::GetRootPageElement method

Returns the root page element for this page.

PROTOTYPE

.NET

```
PageElement GetRootPageElement()
```

JAVA

```
PageElement GetRootPageElement() throws IGREException;
```

PYTHON

```
def GetRootPageElement(self) -> PageElement
```

C++17

```
PageElement GetRootPageElement()
```

RETURN VALUE

[PageElement](#) : PageElement representing the root of the page.

SEE ALSO

- [PageElement](#)

Page::Handle property

Returns the pages IGR_HPAGE handle that can be used with the C API.

.NET

```
int Handle { get; }
```

JAVA

```
int getHandle() throws IGRException;
```

PYTHON

```
@property  
def Handle(self) -> int
```

C++17

```
int getHandle() const;
```

RETURN VALUE

`int` : The IGR_HPAGE handle for the page.

Page::Height property

The height property return the dimensions of a page in pixels.

.NET

```
int Height { get; }
```

JAVA

```
int getHeight() throws IGREException;
```

PYTHON

```
@property  
def Height(self) -> int
```

C++17

```
int getHeight() const;
```

COM

```
[propget] HRESULT Height([out, retval] int *result);
```

RETURN VALUE

int : Integer containing the height of the page in pixels.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;  
2 using System;  
3  
4 var api = new Hyland.DocumentFilters.Api();  
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");  
6  
7 using var doc = api.GetExtractor("filename.doc");  
8 doc.Open(OpenMode.Paginated);  
9  
10 foreach (var page in doc.Pages)  
11 {  
12     using (page)  
13     {  
14         Console.WriteLine($"Width: {page.Width}");  
15         Console.WriteLine($"Height: {page.Height}");  
16         Console.WriteLine($"Text: {page.Text}");  
17     }  
18 }
```

ADDITIONAL INFORMATION

The dimensions are calculated based on the stored page width of the source document, or the default page width for text documents.

The calculated dimensions of a page can be controlled by setting options, such as DPI, when loading the document.

SEE ALSO

- [Page interface](#)
- [Page::Width](#)

Page::Hyperlinks property

Returns an enumerable collections of hyperlinks on the page.

.NET

```
IEnumerable<Hyperlink> Hyperlinks { get; }
```

PYTHON

```
@property  
def Hyperlinks(self) -> list
```

C++17

```
IEnumerable<Hyperlink> getHyperlinks() const;
```

RETURN VALUE

`IEnumerable<Hyperlink>` : An enumerable collection of hyperlink objects.

SEE ALSO

- [Hyperlink interface](#)

Page::Images property

Returns an enumerable collections of image elements on the page.

.NET

```
IEnumerable<SubFile> Images { get; }
```

PYTHON

```
@property  
def Images(self) -> list
```

C++17

```
IEnumerable<SubFile> getImages() const;
```

RETURN VALUE

`IEnumerable<SubFile>` : An enumerable collection of image element objects.

SEE ALSO

- [SubFile interface](#)

Page::Redact method

Redact removes a range of words and blacks out the location for the specified range from the page.

OVERLOADS

<code>Redact(Word, Word)</code>	Redacts the words from the given range.
<code>Redact(int, int)</code>	Redacts the words from the given range.

REDACT(WORD, WORD)

Prototype

.NET

```
void Redact(Word firstWord, Word lastWord)
```

JAVA

```
void Redact(Word firstWord, Word lastWord) throws IGREException;
```

PYTHON

```
def Redact(self, firstWord: Word, lastWord: Word) -> void
```

C++17

```
void Redact(Word firstWord, Word lastWord)
```

C++11

```
void Redact(Word firstWord, Word lastWord)
```

COM

```
HRESULT Redact([in] Word* firstWord, [in] Word* lastWord)
```

Parameters

`firstWord: Word` : The first word to redact.

`lastWord: Word` : The first word to redact.

REDACT(INT,INT)**Prototype****.NET**

```
void Redact(int firstWord, int lastWord)
```

JAVA

```
void Redact(int firstWord, int lastWord) throws IGRException;
```

PYTHON

```
def Redact(self, firstWord: int, lastWord: int) -> void
```

C++17

```
void Redact(int firstWord, int lastWord)
```

C++11

```
void Redact(int firstWord, int lastWord)
```

COM

```
HRESULT Redact([in] int firstWord, [in] int lastWord)
```

Parameters

`firstWord: int` : The first word to redact.

`lastWord: int` : The first word to redact.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2
3 var api = new Hyland.DocumentFilters.Api();
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
5
6 using var doc = api.GetExtractor("filename.doc");
7 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
8
9 using var page = doc.GetPage(0);
10 if (page.WordCount > 15)
11     page.Redact(0, 14);
12
13 using var canvas = api.MakeOutputCanvas("output.pdf", CanvasType.PDF, "");
14 canvas.RenderPage(page);
```

ADDITIONAL INFORMATION

It should be assumed that redacted content will not persist between closing and re-opening a page. To create a redacted Image, PDF, or HTML file, first open a Page, perform the redaction, and render the Page to a Canvas before closing it.

The API allows for redacting single words or a range of words. When redacting a range, whitespace between the words will also be redacted.

SEE ALSO

- [Page interface](#)

Page::Text property

The Text property returns all the text contained on the page.

.NET

```
string Text { get; }
```

JAVA

```
string getText() throws IGREException;
```

PYTHON

```
@property
def Text(self) -> string
```

C++17

```
std::wstring getText() const;
```

COM

```
[propget] HRESULT Text([out, retval] BSTR *result);
```

RETURN VALUE

string : Unicode String containing the text of the page.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.Paginated);
9
10 foreach (var page in doc.Pages)
11 {
12     using (page)
13     {
14         Console.WriteLine($"Width: {page.Width}");
15         Console.WriteLine($"Height: {page.Height}");
16         Console.WriteLine($"Text: {page.Text}");
17     }
18 }
```

SEE ALSO

- [Page interface](#)

Page::Width property

The width property return the dimensions of a page in pixels.

.NET

```
int Width { get; }
```

JAVA

```
int getWidth() throws IGREException;
```

PYTHON

```
@property  
def Width(self) -> int
```

C++17

```
int getWidth() const;
```

COM

```
[propget] HRESULT Width([out, retval] int *result);
```

RETURN VALUE

int : Integer containing the width of the page in pixels.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;  
2 using System;  
3  
4 var api = new Hyland.DocumentFilters.Api();  
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");  
6  
7 using var doc = api.GetExtractor("filename.doc");  
8 doc.Open(OpenMode.Paginated);  
9  
10 foreach (var page in doc.Pages)  
11 {  
12     using (page)  
13     {  
14         Console.WriteLine($"Width: {page.Width}");  
15         Console.WriteLine($"Height: {page.Height}");  
16         Console.WriteLine($"Text: {page.Text}");  
17     }  
18 }
```

ADDITIONAL INFORMATION

The dimensions are calculated based on the stored page width of the source document, or the default page width for text documents.

The calculated dimensions of a page can be controlled by setting options, such as DPI, when loading the document.

SEE ALSO

- [Page interface](#)
- [Page::Height](#)

Page::WordCount property

The WordCount property returns the number of “Word”s that are on a page. The words can be enumerated using the GetFirstWord and GetNextWord methods.

.NET

```
int WordCount { get; }
```

JAVA

```
int getWordCount() throws IGREException;
```

PYTHON

```
@property  
def WordCount(self) -> int
```

C++17

```
int getWordCount() const;
```

COM

```
[propget] HRESULT WordCount([out, retval] int *result);
```

RETURN VALUE

`int` : Integer containing the number of words on the page.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.Paginated);
9
10 foreach (var page in doc.Pages)
11 {
12     using (page)
13     {
14         // Iterate over all words on the page
15         for (Word word = page.GetFirstWord(); word != null; word = page.GetNextWord())
16         {
17             Console.Out.WriteLine($"{word.X}, {word.Y} @ {word.CharacterOffset}: {word.Text}");
18         }
19
20         // Alternatively, use the iterator
21         foreach (Word word in page.Words)
22         {
23             Console.Out.WriteLine($"{word.X}, {word.Y} @ {word.CharacterOffset}: {word.Text}");
24         }
25     }
26 }
27 }
```

SEE ALSO

- [Page interface](#)
- [Page::GetFirstWord](#)
- [Page::GetNextWord](#)

Page::Words property

Returns an enumerable collections of form elements on the page.

.NET

```
IEnumerable<Word> Words { get; }
```

PYTHON

```
@property  
def Words(self) -> list
```

C++17

```
IEnumerable<Word> getWords() const;
```

RETURN VALUE

`IEnumerable<Word>` : An enumerable collection of word element objects.

SEE ALSO

- [Word interface](#)

3.3.61 PageElement

PageElement interface

Represents a page element in a document.

PageElement::Bounds property	Gets the bounds of the page element.
PageElement::Depth property	Gets the depth of the page element.
PageElement::Flags property	Gets the flags of the page element.
PageElement::GetFirstChild method	Gets the first child of the page element.
PageElement::GetNextSibling method	Gets the next sibling of the page element.
PageElement::GetStyle method	Get the named style from the page element.
PageElement::Styles property	Gets the styles of the page element.
PageElement::Text property	Gets the text of the page element.
PageElement::Type property	Gets the type of the page element.

PageElement::Bounds property

Gets the bounds of the page element.

.NET

```
System.Drawing.Rectangle Bounds { get; }
```

JAVA

```
IGR_FRect getBounds() throws IGRException;
```

PYTHON

```
@property  
def Bounds(self) -> IGR_FRect
```

C++17

```
IGR_FRect getBounds() const;
```

RETURN VALUE

`IGR_FRect` : The bounding box of the page element

PageElement::Depth property

Gets the depth of the page element.

.NET

```
int Depth { get; }
```

JAVA

```
int getDepth() throws IGRException;
```

PYTHON

```
@property  
def Depth(self) -> int
```

C++17

```
int getDepth() const;
```

RETURN VALUE

`int` : The depth of the element.

PageElement::Flags property

Gets the flags of the page element.

.NET

```
int Flags { get; }
```

JAVA

```
int getFlags() throws IGRException;
```

PYTHON

```
@property  
def Flags(self) -> int
```

C++17

```
int getFlags() const;
```

RETURN VALUE

`int` : The flags of the element.

SEE ALSO

- [IGR_PAGE_FORM_ELEMENT_FLAG_TYPE](#)

PageElement::GetFirstChild method

Gets the first child of the page element.

PROTOTYPE

.NET

```
PageElement GetFirstChild()
```

JAVA

```
PageElement GetFirstChild() throws IGREException;
```

PYTHON

```
def GetFirstChild(self) -> PageElement
```

C++17

```
PageElement GetFirstChild()
```

RETURN VALUE

[PageElement](#) : PageElement of the first child, or null if no children present.

PageElement::GetNextSibling method

Gets the next sibling of the page element.

PROTOTYPE

.NET

```
PageElement GetNextSibling()
```

JAVA

```
PageElement GetNextSibling() throws IGRException;
```

PYTHON

```
def GetNextSibling(self) -> PageElement
```

C++17

```
PageElement GetNextSibling()
```

RETURN VALUE

[PageElement](#) : PageElement of the next sibling child, or null if last item.

PageElement::GetStyle method

Get the named style from the page element.

PROTOTYPE

.NET

```
string GetStyle()
```

JAVA

```
string GetStyle() throws IGRException;
```

PYTHON

```
def GetStyle(self) -> string
```

C++17

```
std::wstring GetStyle()
```

RETURN VALUE

`string` : String containing the name, or null if not available.

PageElement::Styles property

Gets the styles of the page element.

.NET

```
IDictionary<string, string> Styles { get; }
```

JAVA

```
LocalizedDictionary getStyles() throws IGRException;
```

PYTHON

```
@property  
def Styles(self) -> LocalizedDictionary
```

C++17

```
LocalizedDictionary getStyles() const;
```

RETURN VALUE

`LocalizedDictionary` : The styles of the page elements.

PageElement::Text property

Gets the text of the page element.

.NET

```
string Text { get; }
```

JAVA

```
string getText() throws IGRException;
```

PYTHON

```
@property  
def Text(self) -> string
```

C++17

```
std::wstring getText() const;
```

RETURN VALUE

`string` : The text of the page element

PageElement::Type property

Gets the type of the page element.

.NET

```
Hyland.DocumentFilters.PageElementType Type { get; }
```

JAVA

```
int getType() throws IGRException;
```

PYTHON

```
@property  
def Type(self) -> int
```

C++17

```
int getType() const;
```

RETURN VALUE

`int` : The page element type.

SEE ALSO

- [IGR_PAGE_FORM_ELEMENT_TYPE](#)

3.3.62 RenderPageProperties

RenderPageProperties interface

The RenderPageProperties interface provides additional information for rendering a page to a canvas.

This interface can be created directly.

RenderPageProperties::AddFormValue method	AddFormValue adds a new FormKeyValue to the RenderPageProperties
RenderPageProperties::AddRedaction method	Add a redaction to the list of redactions.
RenderPageProperties::GetFirstFormValue method	GetFirstFormValue returns the first form value or NULL if there are no form values.
RenderPageProperties::GetFormValue method	GetFormValue returns the form value at the specified index
RenderPageProperties::GetFormValueCount method	GetFormValueCount returns the number of form values stored in the RenderPageProperties.
RenderPageProperties::GetNextFormValue method	GetNextFormValue returns the first form value or NULL if there are no form values.

SAMPLE CODE

C/C++

```
var RenderPageProperties = new RenderPageProperties();

RenderPageProperties.AddFormValue("FormKey", "NewFormValue", true);

canvas.RenderPage(page, "", RenderPageProperties)
```

RenderPageProperties::AddFormValue method

AddFormValue adds a new FormKeyValue to the RenderPageProperties

PROTOTYPE

.NET

```
void AddFormValue(string name, string value, bool None)
```

JAVA

```
void AddFormValue(string name, string value, boolean None) throws IGRException;
```

PYTHON

```
def AddFormValue(self, name: string, value: string, None: bool) -> void
```

C++17

```
void AddFormValue(const std::wstring& name, const std::wstring& value, bool None)
```

PARAMETERS

`name: string` : Name of form element.

`value: string` : Value to set.

`None: bool` : Indicates if a checkbox or radio button is selected.

SAMPLE CODE

.NET

```

1 using System.Linq;
2 using Hyland.DocumentFilters;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.OpenExtractor("filename.doc", OpenMode.Paginated);
8
9 foreach (var pageIter in doc.Pages.Select((page, index) => new { page, index }))
10 {
11     using (pageIter.page)
12     {
13         using var canvas = api.MakeOutputCanvas($"page-{pageIter.index+1}.png", CanvasType.PNG);
14
15         var renderPageProperties = new RenderPageProperties();
16         renderPageProperties.AddFormValue("FormKey", "NewFormValue", true);
17
18         canvas.RenderPage(pageIter.page, "", renderPageProperties);
19     }
20 }

```

C++17

```

1 #include <DocumentFiltersObjects.h>
2
3 int main() {
4     try {
5         // Create and initialize the API object
6         Hyland::DocFilters::Api api;
7         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
8
9         // Open the input file
10        Hyland::DocFilters::Extractor doc = api.OpenExtractor("filename.doc",
Hyland::DocFilters::OpenMode::Paginated);
11
12        // Create the output canvas
13        Hyland::DocFilters::Canvas canvas = api.MakeOutputCanvas("output.pdf",
Hyland::DocFilters::CanvasType::PDF);
14
15        Hyland::DocFilters::RenderPageProperties renderProps;
16        renderProps.AddFormValue(L"FormKey", L"NewFormValue", true);
17
18        // Render all pages to the output
19        for (auto page : doc.pages()) {
20            canvas.RenderPage(page, renderProps);
21        }
22    } catch (const std::exception& e) {
23        std::cerr << "Error: " << e.what() << std::endl;
24        return 1; // Indicate an error
25    }
26
27    return 0; // Successful execution
28 }

```

SEE ALSO

- [RenderPageProperties](#)

RenderPageProperties::AddRedaction method

Add a redaction to the list of redactions.

OVERLOADS

<code>AddRedaction(System.Drawing.Rectangle, System.Drawing.Color)</code>	Add a redaction to the list of redactions.
<code>AddRedaction(int, int, int, int, string)</code>	Add a redaction to the list of redactions.

ADDREDACTION(SYSTEM.DRAWING.RECTANGLE, SYSTEM.DRAWING.COLOR)

Prototype

.NET

```
void AddRedaction(System.Drawing.Rectangle rect, System.Drawing.Color color)
```

C++17

```
void AddRedaction(System.Drawing.Rectangle rect, Hyland.DocFilters.Color color)
```

Parameters

- `rect: System.Drawing.Rectangle` : The bounding rectangle of the redaction.
- `color: System.Drawing.Color` : The color of the redaction.

ADDREDACTION(INT, INT, INT, INT, STRING)

Prototype

.NET

```
void AddRedaction(int left, int top, int right, int bottom, string color)
```

C++17

```
void AddRedaction(int left, int top, int right, int bottom, const std::wstring& color)
```

Parameters

- `left: int` : The left bounds of the rectangle to redact.
- `top: int` : The top bounds of the rectangle to redact.
- `right: int` : The right bounds of the rectangle to redact.

`bottom: int` : The bottom bounds of the rectangle to redact.

`color: string` : The color of the redaction.

RenderPageProperties::GetFirstFormValue method

GetFirstFormValue returns the first form value or NULL if there are no form values.

PROTOTYPE

.NET

```
FormKeyValue GetFirstFormValue()
```

JAVA

```
FormKeyValue GetFirstFormValue() throws IGREException;
```

PYTHON

```
def GetFirstFormValue(self) -> FormKeyValue
```

C++17

```
FormKeyValue GetFirstFormValue()
```

RETURN VALUE

[FormKeyValue](#) : The first form value or NULL if there are no form values.

SEE ALSO

- [RenderPageProperties](#)
- [FormKeyValue](#) interface

RenderPageProperties::GetFormValue method

GetFormValue returns the form value at the specified index

PROTOTYPE

.NET

```
FormKeyValue GetFormValue(int index)
```

JAVA

```
FormKeyValue GetFormValue(int index) throws IGRException;
```

PYTHON

```
def GetFormValue(self, index: int) -> FormKeyValue
```

C++17

```
FormKeyValue GetFormValue(int index)
```

PARAMETERS

`index: int` : Index of item.

RETURN VALUE

`FormKeyValue` : The form value at the specified index or NULL if the index is out of range.

SEE ALSO

- [RenderPageProperties](#)

RenderPageProperties::GetFormValueCount method

GetFormValueCount returns the number of form values stored in the RenderPageProperties.

PROTOTYPE

.NET

```
int GetFormValueCount()
```

JAVA

```
int GetFormValueCount() throws IGREException;
```

PYTHON

```
def GetFormValueCount(self) -> int
```

C++17

```
int GetFormValueCount()
```

RETURN VALUE

`int` : The count of form values.

SEE ALSO

- [RenderPageProperties](#)

RenderPageProperties::GetNextFormValue method

GetNextFormValue returns the first form value or NULL if there are no form values.

PROTOTYPE

.NET

```
FormKeyValue GetNextFormValue()
```

JAVA

```
FormKeyValue GetNextFormValue() throws IGREException;
```

PYTHON

```
def GetNextFormValue(self) -> FormKeyValue
```

C++17

```
FormKeyValue GetNextFormValue()
```

RETURN VALUE

[FormKeyValue](#) : The next form value or NULL if there are no more form values.

SEE ALSO

- [RenderPageProperties](#)
- [FormKeyValue](#) interface

3.3.63 SubFile

SubFile interface

The SubFile interface is a descendant of [Extractor](#), allowing work with sub-documents, extracted from a parent document, by calling the parent's [Extractor::GetFirstSubFile](#) and [Extractor::GetNextSubFile](#) methods.

Open the sub-document associated with an instance of SubFile, in the same way as described for Extractor, allowing processing of sub-documents to any depth. This means that text can be extracted and/or sub-documents contained in this SubFile maybe enumerated.

SubFile::Comment property	The Comment property returns any stored comment for the subfile.
SubFile::FileDate property	The FileDate property contains last-modified date and time of the sub-document as a double-precision number (DATE). If the date information is not available, the value is 0.
SubFile::FileSize property	The FileSize property contains the size, in bytes, of the sub-document as a 64-bit number (INT64). If the size information is not available, the value is 0.
SubFile::FileTime property	The FileTime property contains last-modified date and time of the sub-document in the date time format supported by the language.
SubFile::Flags property	Returns stored flags for the subfile.
SubFile::ID property	The ID property contains the unique ID of the sub-document.
SubFile::IsEncrypted property	The IsEncrypted property indicates if the subfile requires a password to extract.
SubFile::Name property	The Name property displays name of the sub-document, if available.

SubFile::Comment property

The Comment property returns any stored comment for the subfile.

.NET

```
string Comment { get; }
```

JAVA

```
string getComment() throws IGRException;
```

PYTHON

```
@property  
def Comment(self) -> string
```

C++17

```
std::wstring getComment() const;
```

RETURN VALUE

`string` : Returns the stored comment for the subfile.

SubFile::FileDate property

The FileDate property contains last-modified date and time of the sub-document as a double-precision number (DATE).

If the date information is not available, the value is 0.

.NET

```
IGRTime FileDate { get; }
```

JAVA

```
IGRTime getFileDate() throws IGRException;
```

PYTHON

```
@property  
def FileDate(self) -> IGRTime
```

C++17

```
IGRTime getFileDate() const;
```

COM

```
[propget] HRESULT FileDate([out, retval] IGRTime* *result);
```

RETURN VALUE

IGRTime : The date of the file.

SAMPLE CODE

.NET

```

1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.zip");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 foreach (var subfile in doc.SubFiles)
11 {
12     using (subfile)
13     {
14         // act on subfile
15         Console.Out.WriteLine("Name: " + subfile.Name);
16         Console.Out.WriteLine("ID: " + subfile.ID);
17         Console.Out.WriteLine("Date: " + subfile.FileDate);
18         Console.Out.WriteLine("Size: " + subfile.FileSize);
19     }
20 }

```

JAVA

```

1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.zip")) {
11            doc.Open(IGR_BODY_AND_META);
12
13            for (SubFile subfile = doc.GetFirstSubFile(); subfile != null; subfile = doc.getNextSubFile())
14            {
15                try (subfile) {
16                    // act on subfile
17                }
18            }
19        }
20    }
21 }

```

PYTHON

```

1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.zip") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     for subfile in doc.SubFiles:
10         with subfile:
11             # act on subfile

```

C++17

```

1  #include <DocumentFiltersObjects.h>
2  #include <iostream>
3
4  using namespace Hyland::DocFilters;
5
6  int main() {
7      try {
8          // Initialize the Document Filters API
9          Hyland::DocFilters::DocumentFilters api;
10         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
11
12         // Get the extractor
13         Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.zip");
14
15         // Open the document
16         doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
17
18         // Access subfiles using the subfiles() iterator
19         for (auto subfile : doc.subfiles()) {
20             try {
21                 // Process the subfile
22                 std::wcout << L"Name: " << subfile.getName() << std::endl;
23                 std::wcout << L"ID: " << subfile.getId() << std::endl;
24                 std::wcout << L>Date: " << u8_to_w(subfile.getFileDate().ToIsoString()) << std::endl;
25                 std::wcout << L"Size: " << subfile.getSize() << std::endl;
26             } catch (const std::exception& e) {
27                 std::cerr << "Error: " << e.what() << std::endl;
28             }
29         }
30
31         // Close the extractor
32         doc.Close();
33     } catch (const std::exception& e) {
34         std::cerr << "Error: " << e.what() << std::endl;
35     }
36
37     return 0;
38 }

```

ADDITIONAL INFORMATION

The integral part of the FileDate value is the number of days that have passed since 12/30/1899 and the fractional part represents the percentage of a 24-hour day that has elapsed.

SEE ALSO

- [SubFile Interface](#)
- [FileTime](#)

SubFile::FileSize property

The FileSize property contains the size, in bytes, of the sub-document as a 64-bit number (INT64).

If the size information is not available, the value is 0.

.NET

```
long FileSize { get; }
```

JAVA

```
long getFileSize() throws IGREException;
```

PYTHON

```
@property
def FileSize(self) -> long
```

C++17

```
long getFileSize() const;
```

RETURN VALUE

long : The size of the file in bytes

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.zip");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 foreach (var subfile in doc.SubFiles)
11 {
12     using (subfile)
13     {
14         // act on subfile
15         Console.Out.WriteLine("Name: " + subfile.Name);
16         Console.Out.WriteLine("ID: " + subfile.ID);
17         Console.Out.WriteLine("Date: " + subfile.FileDate);
18         Console.Out.WriteLine("Size: " + subfile.FileSize);
19     }
20 }
```


JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.zip")) {
11            doc.Open(isys_docfilters.IGR_BODY_AND_META);
12
13            for (SubFile subfile = doc.GetFirstSubFile(); subfile != null; subfile = doc.GetNextSubFile())
14            {
15                try (subfile) {
16                    // act on subfile
17                }
18            }
19        }
20    }
21 }
```

PYTHON

```
1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.zip") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     for subfile in doc.SubFiles:
10         with subfile:
11             # act on subfile
```

C++17

```

1  #include <DocumentFiltersObjects.h>
2  #include <iostream>
3
4  using namespace Hyland::DocFilters;
5
6  int main() {
7      try {
8          // Initialize the Document Filters API
9          Hyland::DocFilters::DocumentFilters api;
10         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
11
12         // Get the extractor
13         Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.zip");
14
15         // Open the document
16         doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
17
18         // Access subfiles using the subfiles() iterator
19         for (auto subfile : doc.subfiles()) {
20             try {
21                 // Process the subfile
22                 std::wcout << L"Name: " << subfile.getName() << std::endl;
23                 std::wcout << L"ID: " << subfile.getId() << std::endl;
24                 std::wcout << L>Date: " << u8_to_w(subfile.getFileDate().ToIsoString()) << std::endl;
25                 std::wcout << L"Size: " << subfile.getSize() << std::endl;
26             } catch (const std::exception& e) {
27                 std::cerr << "Error: " << e.what() << std::endl;
28             }
29         }
30
31         // Close the extractor
32         doc.Close();
33     } catch (const std::exception& e) {
34         std::cerr << "Error: " << e.what() << std::endl;
35     }
36
37     return 0;
38 }

```

SEE ALSO

- [SubFile Interface](#)

SubFile::FileTime property

The FileTime property contains last-modified date and time of the sub-document in the date time format supported by the language.

.NET

```
System.DateTime FileTime { get; }
```

JAVA

```
java.util.Date getFileTime() throws IGRException;
```

PYTHON

```
@property
def FileTime(self) -> datetime
```

C++17

```
Hyland.DocFilters.DateTime getFileTime() const;
```

RETURN VALUE

datetime : The last modified date of the file.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.zip");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 foreach (var subfile in doc.SubFiles)
11 {
12     using (subfile)
13     {
14         // act on subfile
15         Console.Out.WriteLine("Name: " + subfile.Name);
16         Console.Out.WriteLine("ID: " + subfile.ID);
17         Console.Out.WriteLine("Date: " + subfile.FileDate);
18         Console.Out.WriteLine("Size: " + subfile.FileSize);
19     }
20 }
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.zip")) {
11            doc.Open(isys_docfilters.IGR_BODY_AND_META);
12
13            for (SubFile subfile = doc.GetFirstSubFile(); subfile != null; subfile = doc.GetNextSubFile())
14            {
15                try (subfile) {
16                    // act on subfile
17                }
18            }
19        }
20    }
21 }
```

PYTHON

```
1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.zip") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     for subfile in doc.SubFiles:
10         with subfile:
11             # act on subfile
```

C++17

```

1  #include <DocumentFiltersObjects.h>
2  #include <iostream>
3
4  using namespace Hyland::DocFilters;
5
6  int main() {
7      try {
8          // Initialize the Document Filters API
9          Hyland::DocFilters::DocumentFilters api;
10         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
11
12         // Get the extractor
13         Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.zip");
14
15         // Open the document
16         doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
17
18         // Access subfiles using the subfiles() iterator
19         for (auto subfile : doc.subfiles()) {
20             try {
21                 // Process the subfile
22                 std::wcout << L"Name: " << subfile.getName() << std::endl;
23                 std::wcout << L"ID: " << subfile.getId() << std::endl;
24                 std::wcout << L>Date: " << u8_to_w(subfile.getFileDate().ToIsoString()) << std::endl;
25                 std::wcout << L"Size: " << subfile.getSize() << std::endl;
26             } catch (const std::exception& e) {
27                 std::cerr << "Error: " << e.what() << std::endl;
28             }
29         }
30
31         // Close the extractor
32         doc.Close();
33     } catch (const std::exception& e) {
34         std::cerr << "Error: " << e.what() << std::endl;
35     }
36
37     return 0;
38 }

```

SEE ALSO

- [FileDate](#)

"

SubFile::Flags property

Returns stored flags for the subfile.

.NET

```
int Flags { get; }
```

JAVA

```
int getFlags() throws IGRException;
```

PYTHON

```
@property  
def Flags(self) -> int
```

C++17

```
int getFlags() const;
```

RETURN VALUE

`int` : The flags for the sub file. See [IGR_SUBFILE_INFO_FLAG_TYPE](#) for details.

SEE ALSO

- [IGR_SUBFILE_INFO_FLAG_TYPE](#)

SubFile::ID property

The ID property contains the unique ID of the sub-document.

.NET

```
string ID { get; }
```

JAVA

```
string getID() throws IGRException;
```

PYTHON

```
@property
def ID(self) -> string
```

C++17

```
std::wstring getID() const;
```

COM

```
[propget] HRESULT ID([out, retval] BSTR *result);
```

RETURN VALUE

string : The id of the subfile.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.zip");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 foreach (var subfile in doc.SubFiles)
11 {
12     using (subfile)
13     {
14         // act on subfile
15         Console.Out.WriteLine("Name: " + subfile.Name);
16         Console.Out.WriteLine("ID: " + subfile.ID);
17         Console.Out.WriteLine("Date: " + subfile.FileDate);
18         Console.Out.WriteLine("Size: " + subfile.FileSize);
19     }
20 }
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.zip")) {
11            doc.Open(isys_docfilters.IGR_BODY_AND_META);
12
13            for (SubFile subfile = doc.GetFirstSubFile(); subfile != null; subfile = doc.GetNextSubFile())
14            {
15                try (subfile) {
16                    // act on subfile
17                }
18            }
19        }
20    }
21 }
```

PYTHON

```
1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.zip") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     for subfile in doc.SubFiles:
10         with subfile:
11             # act on subfile
```


C++17

```

1  #include <DocumentFiltersObjects.h>
2  #include <iostream>
3
4  using namespace Hyland::DocFilters;
5
6  int main() {
7      try {
8          // Initialize the Document Filters API
9          Hyland::DocFilters::DocumentFilters api;
10         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
11
12         // Get the extractor
13         Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.zip");
14
15         // Open the document
16         doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
17
18         // Access subfiles using the subfiles() iterator
19         for (auto subfile : doc.subfiles()) {
20             try {
21                 // Process the subfile
22                 std::wcout << L"Name: " << subfile.getName() << std::endl;
23                 std::wcout << L"ID: " << subfile.getId() << std::endl;
24                 std::wcout << L>Date: " << u8_to_w(subfile.getFileDate().ToIsoString()) << std::endl;
25                 std::wcout << L"Size: " << subfile.getSize() << std::endl;
26             } catch (const std::exception& e) {
27                 std::cerr << "Error: " << e.what() << std::endl;
28             }
29         }
30
31         // Close the extractor
32         doc.Close();
33     } catch (const std::exception& e) {
34         std::cerr << "Error: " << e.what() << std::endl;
35     }
36
37     return 0;
38 }

```

SEE ALSO

- [SubFile Interface](#)

SubFile::IsEncrypted property

The IsEncrypted property indicates if the subfile requires a password to extract.

.NET

```
bool IsEncrypted { get; }
```

JAVA

```
boolean getIsEncrypted() throws IGRException;
```

PYTHON

```
@property  
def IsEncrypted(self) -> bool
```

C++17

```
bool getIsEncrypted() const;
```

RETURN VALUE

bool : Returns if a password is required to extract the file.

SubFile::Name property

The Name property displays name of the sub-document, if available.

.NET

```
string Name { get; }
```

JAVA

```
string getName() throws IGRException;
```

PYTHON

```
@property
def Name(self) -> string
```

C++17

```
std::wstring getName() const;
```

COM

```
[propget] HRESULT Name([out, retval] BSTR *result);
```

RETURN VALUE

string : The name of the subfile.

SAMPLE CODE

.NET

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.zip");
8 doc.Open(Hyland.DocumentFilters.OpenType.BodyAndMeta);
9
10 foreach (var subfile in doc.SubFiles)
11 {
12     using (subfile)
13     {
14         // act on subfile
15         Console.Out.WriteLine("Name: " + subfile.Name);
16         Console.Out.WriteLine("ID: " + subfile.ID);
17         Console.Out.WriteLine("Date: " + subfile.FileDate);
18         Console.Out.WriteLine("Size: " + subfile.FileSize);
19     }
20 }
```

JAVA

```
1 import com.perceptive.documentfilters.*;
2
3 public class App
4 {
5     public static void main(String[] args) throws Exception
6     {
7         DocumentFilters df = new DocumentFilters();
8         df.Initialize("YOUR_LICENSE_KEY_HERE", ".");
9
10        try (Extractor doc = df.GetExtractor("filename.zip")) {
11            doc.Open(isys_docfilters.IGR_BODY_AND_META);
12
13            for (SubFile subfile = doc.GetFirstSubFile(); subfile != null; subfile = doc.GetNextSubFile())
14            {
15                try (subfile) {
16                    // act on subfile
17                }
18            }
19        }
20    }
21 }
```

PYTHON

```
1 from DocumentFilters import *
2
3 api = DocumentFilters()
4 api.Initialize("YOUR_LICENSE_KEY_HERE", ".")
5
6 with api.GetExtractor("filename.zip") as doc:
7     doc.Open(IGR_BODY_AND_META, "")
8
9     for subfile in doc.SubFiles:
10         with subfile:
11             # act on subfile
```

C++17

```

1  #include <DocumentFiltersObjects.h>
2  #include <iostream>
3
4  using namespace Hyland::DocFilters;
5
6  int main() {
7      try {
8          // Initialize the Document Filters API
9          Hyland::DocFilters::DocumentFilters api;
10         api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
11
12         // Get the extractor
13         Hyland::DocFilters::Extractor doc = api.GetExtractor("filename.zip");
14
15         // Open the document
16         doc.Open(Hyland::DocFilters::OpenMode::Text, IGR_BODY_AND_META);
17
18         // Access subfiles using the subfiles() iterator
19         for (auto subfile : doc.subfiles()) {
20             try {
21                 // Process the subfile
22                 std::wcout << L"Name: " << subfile.getName() << std::endl;
23                 std::wcout << L"ID: " << subfile.getId() << std::endl;
24                 std::wcout << L>Date: " << u8_to_w(subfile.getFileDate().ToIsoString()) << std::endl;
25                 std::wcout << L"Size: " << subfile.getSize() << std::endl;
26             } catch (const std::exception& e) {
27                 std::cerr << "Error: " << e.what() << std::endl;
28             }
29         }
30
31         // Close the extractor
32         doc.Close();
33     } catch (const std::exception& e) {
34         std::cerr << "Error: " << e.what() << std::endl;
35     }
36
37     return 0;
38 }

```

SEE ALSO

- [SubFile Interface](#)

3.3.64 Word

Word interface

The Word interface allows extraction of words and their bounding boxes when in paginated image mode.

To obtain this interface, call the Page.[GetFirstWord](#), Page.[GetNextWord](#) methods, or Page.[Words](#) property.

Word::CharacterOffset property	The CharacterOffset property contains the character offset of the word into the text on the current page.
Word::Text property	The Text property returns a Unicode string for the text of this word.
Word::Width property	The Width property return the dimensions of the word in pixels. The dimension information is based on the DPI used when loading the page.
Word::Height property	The Height property return the dimensions of the word in pixels. The dimension information is based on the DPI used when loading the page.
Word::WordIndex property	Return the index of the word on the current page.
Word::X property	The X property return the position of the word in pixels. The position information is based on the DPI used when loading the page.
Word::Y property	The Y property return the position of the word in pixels. The position information is based on the DPI used when loading the page.

Word::CharacterOffset property

The CharacterOffset property contains the character offset of the word into the text on the current page.

.NET

```
int CharacterOffset { get; }
```

JAVA

```
int getCharacterOffset() throws IGREException;
```

PYTHON

```
@property  
def CharacterOffset(self) -> int
```

C++17

```
int getCharacterOffset() const;
```

COM

```
[propget] HRESULT CharacterOffset([out, retval] int *result);
```

RETURN VALUE

`int` : The offset into the text of the current page.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.Paginated);
9
10 foreach (var page in doc.Pages)
11 {
12     using (page)
13     {
14         // Iterate over all words on the page
15         for (Word word = page.GetFirstWord(); word != null; word = page.GetNextWord())
16         {
17             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
18         }
19
20         // Alternatively, use the iterator
21         foreach (Word word in page.Words)
22         {
23             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
24         }
25     }
26 }
27 }
```

ADDITIONAL INFORMATION

The value returned is the offset into the current page. To calculate the offset into the document, the size of the text of the previous pages must be accumulated.

SEE ALSO

- [Word interface](#)

Word::Text property

The Text property returns a Unicode string for the text of this word.

.NET

```
string Text { get; }
```

JAVA

```
string getText() throws IGRException;
```

PYTHON

```
@property  
def Text(self) -> string
```

C++17

```
std::wstring getText() const;
```

COM

```
[propget] HRESULT Text([out, retval] BSTR *result);
```

RETURN VALUE

string : Unicode string containing the text of the word.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.Paginated);
9
10 foreach (var page in doc.Pages)
11 {
12     using (page)
13     {
14         // Iterate over all words on the page
15         for (Word word = page.GetFirstWord(); word != null; word = page.GetNextWord())
16         {
17             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
18         }
19
20         // Alternatively, use the iterator
21         foreach (Word word in page.Words)
22         {
23             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
24         }
25     }
26 }
27 }
```

SEE ALSO

- [Word interface](#)

Word::Width property

The Width property return the dimensions of the word in pixels. The dimension information is based on the DPI used when loading the page.

.NET

```
int Width { get; }
```

JAVA

```
int getWidth() throws IGRException;
```

PYTHON

```
@property  
def Width(self) -> int
```

C++17

```
int getWidth() const;
```

COM

```
[propget] HRESULT Width([out, retval] int *result);
```

RETURN VALUE

`int` : Integer containing the dimension of the word in pixels.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.Paginated);
9
10 foreach (var page in doc.Pages)
11 {
12     using (page)
13     {
14         // Iterate over all words on the page
15         for (Word word = page.GetFirstWord(); word != null; word = page.GetNextWord())
16         {
17             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
18         }
19
20         // Alternatively, use the iterator
21         foreach (Word word in page.Words)
22         {
23             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
24         }
25     }
26 }
27 }
```

SEE ALSO

- [Word interface](#)

Word::Height property

The Height property return the dimensions of the word in pixels. The dimension information is based on the DPI used when loading the page.

.NET

```
int Height { get; }
```

JAVA

```
int getHeight() throws IGRException;
```

PYTHON

```
@property  
def Height(self) -> int
```

C++17

```
int getHeight() const;
```

COM

```
[propget] HRESULT Height([out, retval] int *result);
```

RETURN VALUE

`int` : Integer containing the dimension of the word in pixels.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.Paginated);
9
10 foreach (var page in doc.Pages)
11 {
12     using (page)
13     {
14         // Iterate over all words on the page
15         for (Word word = page.GetFirstWord(); word != null; word = page.GetNextWord())
16         {
17             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
18         }
19
20         // Alternatively, use the iterator
21         foreach (Word word in page.Words)
22         {
23             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
24         }
25     }
26 }
27 }
```

SEE ALSO

- [Word interface](#)

Word::WordIndex property

Return the index of the word on the current page.

.NET

```
int WordIndex { get; }
```

JAVA

```
int getWordIndex() throws IGREException;
```

PYTHON

```
@property  
def WordIndex(self) -> int
```

C++17

```
int getWordIndex() const;
```

COM

```
[propget] HRESULT WordIndex([out, retval] int *result);
```

RETURN VALUE

`int` : The index of the word on the current page.

SEE ALSO

- [Word interface](#)

Word::X property

The X property return the position of the word in pixels. The position information is based on the DPI used when loading the page.

.NET

```
int X { get; }
```

JAVA

```
int getX() throws IGREException;
```

PYTHON

```
@property  
def X(self) -> int
```

C++17

```
int getX() const;
```

COM

```
[propget] HRESULT X([out, retval] int *result);
```

RETURN VALUE

int : Integer containing the coordinate of the word in pixels.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.Paginated);
9
10 foreach (var page in doc.Pages)
11 {
12     using (page)
13     {
14         // Iterate over all words on the page
15         for (Word word = page.GetFirstWord(); word != null; word = page.GetNextWord())
16         {
17             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
18         }
19
20         // Alternatively, use the iterator
21         foreach (Word word in page.Words)
22         {
23             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
24         }
25     }
26 }
27 }
```

SEE ALSO

- [Word interface](#)

Word::Y property

The Y property return the position of the word in pixels. The position information is based on the DPI used when loading the page.

.NET

```
int T { get; }
```

JAVA

```
int getT() throws IGRException;
```

PYTHON

```
@property  
def T(self) -> int
```

C++17

```
int getT() const;
```

COM

```
[propget] HRESULT T([out, retval] int *result);
```

RETURN VALUE

`int` : Integer containing the coordinate of the word in pixels.

SAMPLE CODE**.NET**

```
1 using Hyland.DocumentFilters;
2 using System;
3
4 var api = new Hyland.DocumentFilters.Api();
5 api.Initialize("YOUR_LICENSE_KEY_HERE", ".");
6
7 using var doc = api.GetExtractor("filename.doc");
8 doc.Open(OpenMode.Paginated);
9
10 foreach (var page in doc.Pages)
11 {
12     using (page)
13     {
14         // Iterate over all words on the page
15         for (Word word = page.GetFirstWord(); word != null; word = page.GetNextWord())
16         {
17             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
18         }
19
20         // Alternatively, use the iterator
21         foreach (Word word in page.Words)
22         {
23             Console.Out.WriteLine($"[{word.X}, {word.Y} @ {word.CharacterOffset}]: {word.Text}");
24         }
25     }
26 }
27 }
```

SEE ALSO

- [Word interface](#)

3.4 Structured XML

3.4.1 Overview

Structured XML is a hierarchical Document Object Model (DOM) that represents the paginated view of a document. Most nodes have two distinct sections:

- Where: Pixel Coordinates relative to the Page.
- Why: Style Information used to calculate the coordinates.

Pixel coordinates are stored in a node with the attributes **left**, **top**, **width**, and **height**. All coordinates are stored relative to the **page**.

Style information is stored in a single **style** attribute. Its content is a semicolon (;) delimited list of name: value pairs. A style value can be one of the following data types:

Type	Description
String	Value is output “as is”, no encoding required.
Number	Value is output “as is”, no encoding required.
Boolean	0 = false, 1 = true.
Rectangle	Rect(left, top, right, bottom)
Margins	Rect(left, top, right, bottom)
Borders	(Style Width Color)
Color	HTML format (#RRGGBB)

3.4.2 <doc> element

The <doc> node is the root-most element of Structured XML. There can only be one <doc> node per document. It does not contain any attributes.

Attributes

None

Styles

None

Children

Node	Description
<head>	Contains the metadata about the document; a <doc> can contain only one <head> element.
<page>	Contains the data for a page; a <doc> can contain multiple <page> elements.

Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<doc>
  <head>
    <meta name="Author" value="Troy Jennings" />
    <meta name="Last Author" value="Ben Truscott" />
    <meta name="Creation Date" value="Sun, 4 Jun 2013 8:15:00 AM" />
    <meta name="Last Saved Date" value="Sun, 4 Jun 2013 8:15:00 AM" />
    <meta name="Revision Number" value="2" />
    <meta name="Page Count" value="50" />
    <meta name="Word Count" value="4633" />
    <meta name="Character Count" value="26412" />
  </head>
  <page styles="..." left="0" top="0" width="816" height="1056">
    ...
  </page>
</doc>
```

3.4.3 <head> element

The `<head>` node contains information about the document as a whole, such as metadata.

Attributes

None

Styles

None

Children

Node	Description
<code><meta></code>	There is a single <code><meta></code> tag for each name/value pair of metadata in the document.

Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<doc>
  <head>
    <meta name="Author" value="Troy Jennings" />
    <meta name="Last Author" value="Ben Truscott" />
    <meta name="Creation Date" value="Sun, 4 Jun 2013 8:15:00 AM" />
    <meta name="Last Saved Date" value="Sun, 4 Jun 2013 8:15:00 AM" />
    <meta name="Revision Number" value="2" />
    <meta name="Page Count" value="50" />
    <meta name="Word Count" value="4633" />
    <meta name="Character Count" value="26412" />
  </head>
  <page styles="..." left="0" top="0" width="816" height="1056">
    ...
  </page>
</doc>
```

3.4.4 <meta> element

The `<meta>` node contains a single name/value pair of metadata from the source document. The node does not have child elements.

Attributes

Attribute	Description
name	The name of the metadata field, for example: Title, Author or Page Count.
value	The value of the metadata field.

Styles

None

Children

None

Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<doc>
  <head>
    <meta name="Author" value="Troy Jennings" />
    <meta name="Last Author" value="Ben Truscott" />
    <meta name="Creation Date" value="Sun, 4 Jun 2013 8:15:00 AM" />
    <meta name="Last Saved Date" value="Sun, 4 Jun 2013 8:15:00 AM" />
    <meta name="Revision Number" value="2" />
    <meta name="Page Count" value="50" />
    <meta name="Word Count" value="4633" />
    <meta name="Character Count" value="26412" />
  </head>
  <page styles="..." left="0" top="0" width="816" height="1056">
    ...
  </page>
</doc>
```

3.4.5 <page> element

The `<page>` node represents a single page in the source document containing all the elements required to render that page.

Attributes

Attribute	Description
left	The left offset in pixels in pixels. This is typically 0.
top	The top offset in pixels in pixels. This is typically 0.
width	The width of the page in pixels.
height	The height of the page in pixels.
styles	Contains a semicolon delimited list of name:value pair style information values.

Styles

Style	Description
pagewidth	The width of the page in pixels.
pageheight	The height of the page in pixels.
headerFromTop	Indicates the space, in points, that the header is placed from the top of the page.
footerFromBottom	Indicates the space, in points, that the footer is placed from the bottom of the page.
headerToBodySpacing	Indicates the space, in points, that content is placed from the bottom of the header.
footerFromBottomSpacing	Indicates the space, in points, that the content is placed from the top of the footer.
endSectionBreakType	Indicates the type of break to use at the end of the current section, can be one of the following:0 Continuous1 NewColumn2 NewPage3 EvenPage4 OddPage

Style	Description
pageNumFormat	Indicates the format that fielded page numbers use, can be one of the following:0 decimal1 upperRoman2 lowerRoman3 upperLetter4 lowerLetter5 ordinal6 cardinalText7 ordinalText8 hex
pageNumStart	Indicates the first number to use when numbering pages.
pagemargins	Specifies the top,left,right,bottom margin for the page in points.
areColumnsEvenlySpaced	Indicates that columns should be evenly sized.
pageNumOffset	Indicates the first page number for the section.
clipPage	Indicates that content outside of the page margins should be clipped.
clipRect	Indicates the top,left,right,bottom clipping rectangle for the page in points.
borderLeft	Indicates the border style, width and color for the left side of the page.
borderRight	Indicates the border style, width and color for the right side of the page.
borderTop	Indicates the border style, width and color for the top side of the page.
borderBottom	Indicates the border style, width and color for the bottom side of the page.
borderOffsetText	Indicates if the left and right position of the text should be incremented by the width of the borders.
sourceDpiX	Indicates the horizontal DPI from the source document, when present. Value is normally set for TIFF files.
sourceDpiY	Indicates the vertical DPI from the source document, when present. Value is normally set for TIFF files.
sourceOrientation	Indicates the page orientation from the source document, value is normally set for TIFF files, can be one of: TopLeft, TopRight, BottomLeft, BottomRight, LeftTop, RightTop, RightBottom, LeftBottom.
outputDpiX	Indicates the output DPI used to create the generated output.
outputDpiY	Indicates the output DPI used to create the generated output.
outputOrientation	Indicates the page orientation used to create the generated output.

Children

Node	Description
<header>	A page can have zero or one <header> .
<footer>	A page can have zero or one <footer> .
<section>	A page can have zero or more <section> s.
<float>	A page can have zero or more <float> ing objects.

Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<doc>
  <head>
    <meta name="Last Author" value="Ben Truscott" />
    <meta name="Page Count" value="50" />
    <meta name="Word Count" value="4633" />
    <meta name="Character Count" value="26412" />
  </head>

  <page styles="pagewidth: 816; pageheight: 1056; headerFromTop: 48; footerFromBottom: 9.33333; headerToBodySpacin
    ...
  </page>
</doc>
```

3.4.6 <header>/<footer> elements

The `<header>` element represents the header section of a page, generated from the header/footer section of a source document. There can be one `<header>` and one `<footer>` per page.

Attributes

Attribute	Description
left	The left offset of the header in pixels.
top	The top offset of the header in pixels.
width	The width of the header in pixels.
height	The height of the header in pixels.
styles	Contains a semicolon delimited list of name:value pair style information values.

Styles

Style	Description
sectionType	Indicates the type of header/footer that was used to generate the content; can be one of the following:0 Even1 Odd2 First
header	Indicates if the node is a header or footer; this can also be achieved by looking at the node name.

Children

Node	Description
<code><p></code>	
<code><float></code>	
<code><table></code>	

Example

```
<page styles="..." left="0" top="0" width="816" height="1056">
  <header styles="sectionType: Odd; header: 1; " left="96" top="48" width="624" height="81">
    <p styles="..." left="96" top="48" width="624" height="19" paragraphId="0">
      ...
    </p>
  </header>
</page>
```

3.4.7 <section> element

The `<section>` element contains a logical grouping of text. A page can have multiple sections depending on the source document type.

Attributes

Attribute	Description
styles	Contains a semicolon delimited list of name:value pair style information values.

Styles

Same as `<page>` element; see page .

Children

Node	Description
<code><column></code>	A section can have zero or more <code><column></code> elements.

Example

```
<page styles="..." left="0" top="0" width="816" height="1056">
  <header styles="..." left="96" top="48" width="624" height="81">
    <p styles="..." left="96" top="48" width="624" height="19" paragraphId="0">
      <line left="96" top="48" width="624" height="19">
        ...
      </line>
    </p>
  </header>
  <section styles="pagewidth: 816; pageheight: 1056; headerFromTop: 48; footerFromBottom: 9.33333;
headerToBodySpacing: 0; footerFromBodySpacing: 0; endSectionBreakType: 0; pageNumFormat: decimal; pageNumChapter
  <column styles="..." left="96" top="144" width="624" height="635">
    <p styles="..." left="76" top="160" width="663" height="24" paragraphId="6">
      ...
    </p>
  </column>
</section>
</page>
```

3.4.8 <column> element

The <column> element contains the data for a single column within a section. There can be one or more columns per section.

Attributes

Attribute	Description
width	The width of the column in pixels.
height	The height of the column in pixels.
left	The left offset of the column in pixels.
top	The top offset of the column in pixels.
styles	Contains a semicolon delimited list of name:value pair style information values.

Styles

Style	Description
iCol	Index of the column when multiple columns exist within the section.
width	Width to use for the column, -255 indicates that it is autosized.
space	Spacing to use between columns, -255 indicates default space.
drawSeparator	Indicates whether a vertical line should be drawn between columns.

Children

Node	Description
<p>	A column can contain zero or more <p> elements.
<float>	A column can contain zero or more <float> elements.

Example

```
<page styles="..." left="0" top="0" width="816" height="1056">
  <header styles="..." left="96" top="48" width="624" height="81">
    <p styles="..." left="96" top="48" width="624" height="19" paragraphId="0">
      <line left="96" top="48" width="624" height="19">
        ...
      </line>
    </p>
  </header>
  <section styles="...">
    <column styles="iCol: 1; width: -255; spacing: 255;" left="96" top="144" width="624" height="635">
      <p styles="..." left="76" top="160" width="663" height="24" paragraphId="6">
        </p>
      </column>
    </section>
  </page>
```

3.4.9 <p> element

The <p> element contains the data for a single paragraph of text. There can be zero or more paragraphs per section.

Attributes

Attribute	Description
width	The width of the paragraph in pixels.
height	The height of the paragraph in pixels.
left	The left offset of the paragraph in pixels.
top	The top offset of the paragraph in pixels.
styles	Contains a semicolon delimited list of name:value pair style information values.

Styles

Style	Description
numberingId	Populated by MS Word for calculating leveled numbered lists.
numberingLevel	Populated by MS Word for calculating leveled numbered lists.
alignment	Text alignment, can be one of the following:- Left- Center- Right- Both- Distributed- Medium Kashida- Numbered Tab- High Kashida- Low Kashida- Thai Distributed
indenthanging	Contains the hanging indent value in points.
indentfirstline	Contains the first line indent value in points.
indentleft	Contains the left indent value in points.
indentright	Contains the right indent value in points.
spacebefore	Contains the space to use before the paragraph in points.
spaceafter	Contains the space to use after the paragraph in points.
linespacing	Contains the line spacing to use for the paragraph expressed as a multiplier.

Style	Description
marginleft	Contains the left margin in points.
hyperlinked	Indicates if the paragraph is a hyperlink.
defaultFontSize	The default font style to use, in points, when a run does not contain a font height.
backgroundColor	The background color that should be shaded behind the paragraph text.
borderLeft	Contains details (width, color, style) for the left border.
borderRight	Contains details (width, color, style) for the right border.
borderTop	Contains details (width, color, style) for the top border.
borderBottom	Contains details (width, color, style) for the bottom border.
borderInternalVert	Contains details (width, color, style) for the internal vertical borders.
borderInternalHorz	Contains details (width, color, style) for the internal horizontal borders.
tabProps	Contains the tab stops to use for this paragraph.
hasBarTabs	Indicates that the paragraph uses bar tabs.
keepLinesTogether	Indicates the paragraph has the “Keep Lines Together” feature enabled.
keepWithNextParagraph	Indicates the paragraph has the “Keep With Next Paragraph” feature enabled.
minWidowOrphanLines	Indicates the minimum number of lines to leave on a page when splitting paragraphs.
keepOnPage	Indicates that the paragraph must not be moved to a new page.
pageBreakBefore	Indicates that a page break should occur before the paragraph.
styleType	Indicates the type of paragraph, can be one of the following:- Normal- Heading 1- Heading 2- Heading 3- Heading 4- Heading 5- Heading 6- Heading 7- Heading 8- Heading 9- Caption- Title- Sub Title- Strong
lineheight	Indicates the minimum height that each line must take.
lineheightSize	Indicates how the lineheight is to be applied, can be one of the following:- AtLeast- Exact- None

Style	Description
styleId	Style id as a string.
nowrap	Indicates that the paragraph should not wrap the text to new lines.
spacebeforeAuto	Indicates that auto calculation of space before is enabled.
spaceafterAuto	Indicates that auto calculation of space after is enabled.
contextualSpacing	Indicates that contextual spacing is to be used, paragraphs that have the same style will merge the spaceBefore and spaceAfter values.
bulleted	Indicates the paragraph is part of a bulleted list.
placeholder	Indicates the type of placeholder, can be one of the following: 0 SlideNumber 1 Footer 2 Date

Children

Node	Description
<line>	A paragraph can contain zero or more <line> elements.
<table>	A paragraph can contain zero or one <table> element.

3.4.10 <line> element

The `<line>` element contains the content for a single line of text in the paginated output. The `<line>` may differ from what the source document defines as a line.

Attributes

Attribute	Description
width	The width of the line in pixels.
height	The height of the line in pixels.
left	The left offset of the line in pixels.
top	The top offset of the line in pixels.

Styles

None

Children

Node	Description
<code><run></code>	A line can contain zero or more <code><run></code> elements.
<code><g></code>	A line can contain zero or more <code><g></code> elements.

3.4.11 <run> element

The `<run>` element groups words that are sequential and have the same style. Styles stored at the `<run>` level should be applied to the words contained within.

Attributes

Attribute	Description
width	The width of the run in pixels.
height	The height of the run in pixels.
left	The left offset of the run in pixels.
top	The top offset of the run in pixels.
styles	Contains a semicolon delimited list of name:value pair style information values.

Styles

Attribute	Description
fontColor	Contains the color to use for the run, in HTML format.
fontSize	Contains the size of the font in points.
fontName	Contains the font name to use for this run.
highlightColor	Contains the background highlight color to use, if any, in HTML format.
underlineColor	Contains the underline color to use in HTML format.
bold	Indicates the text should be rendered as bold.
italics	Indicates the text should be rendered as italics.
underline	Indicates the text should be rendered underlined.
strikeout	Indicates the text should be rendered striked out.
allCaps	Indicates the text should be rendered in all-capitals.
smallCaps	Indicates the text should be rendered in small-capitals.

Attribute	Description
markDel	Indicates the text is marked for deletion (track-changes).
Vanish	Indicates the text is marked as hidden.
superscript	Indicates the text should be rendered in superscript.
subscript	Indicates the text should be rendered in subscript.
scriptSizeScaler	Indicates the scaler to use for either super or subscript.
charScaling	Contains a character scaling multiplier.
charSpacing	Contains a character spacing scaling multiplier.
charVertOffsetPos	Contains a vertical offset to be applied to word position.
charKerning	Indicates a kerning value.
hyperlink	Contains a hyperlink that should be applied to the words.
dynamicField	Indicates that the content of the run should be replaced with a dynamic field value, can be one of the following:- None- PageNumber- NumberOfPages- SectionNumber- SectionNumberOfPages
dynamicFieldFormat	Indicates the format to use when replacing with a dynamic field, can be one of the following:- decimal- upperRoman- lowerRoman- upperLetter- lowerLetter- ordinal- cardinalText- ordinalText- hex
fontType	Indicates the base font type, used when looking for font substitutions, can be one of the following:- Unknown- TypeSans- TypeSerif- TypeMono
direction	Indicates the direction of the text, can be one of the following:- LTR- RTF

Children

Node	Description
<word>	A run can contain zero or more <word> elements.
<g>	A run can contain zero or more <g> elements.

Example

```
<run styles="fontColor: #01ffff; fontSize: 10; fontName: HelveticaNeueLT Pro 55 Roman; bold: 0; italics: 0; underline; bInternal: 0; ; ">
  <word left="468" top="565" width="33" height="16">Open</word>
  <word left="505" top="565" width="62" height="16">Document</word>
  <word left="569" top="565" width="33" height="16">Flags</word>
</run>
```

3.4.12 <word> element

The `<word>` element represents a single word on the page. A `<word>` contains the pixel coordinate on the page. Style information for the word is stored on its parent `<run>`.

Attributes

Attribute	Description
left	The left offset of the word in pixels.
top	The top offset of the word in pixels.
width	The width of the word in pixels.
height	The height of the word in pixels.

Styles

None

Children

None

3.4.13 <float> element

A `<float>` element represents an object that exists outside of the normal flow of paragraphs and text, that are typically elements that text flows around, or are rendered above or below the text.

Attributes

Attribute	Description
left	The left offset of the float in pixels.
top	The top offset of the float in pixels.
width	The width of the float in pixels.
height	The height of the float in pixels.

Styles

Style	Description
width	Indicates the original width of the image, in points.
height	Indicates the original height of the image, in points.
left	Indicates the original position of the image, in points.
top	Indicates the original position of the image, in points.
distList	Indicates the distance from the text, in points.
distTop	Indicates the distance from the text, in points.
distRight	Indicates the distance from the text, in points.
distBottom	Indicates the distance from the text, in points.
degRotated	Indicates any rotation that is to be applied, in degrees.
wrapType	Indicates how text should wrap around the object, can be one of the following:- Inline- Square- Tight- Through- TopAndBottom- Behind- Infront
wrapSubType	Indicates extra wrapping information when wrapType = tight, can be one of the following:- BothSides- LeftSide- RightSide- LargestSize- TopAndBottom

Style	Description
anchorLeft	Indicates the section that the left coordinate is anchored to, can be one of the following:- ColumnPara- Margin- Page- InsideMargin- OutsideMargin
anchorTop	Indicates the section that the top coordinate is anchored to, can be one of the following:- ColumnPara- Margin- Page- InsideMargin- OutsideMargin
borderLeft	Contains details (width, color, style) for the left border.
borderRight	Contains details (width, color, style) for the right border.
borderTop	Contains details (width, color, style) for the top border.
borderBottom	Contains details (width, color, style) for the bottom border.
zIndex	Contains the z-index for the element.
alignHorz	Contains the alignment for the horizontal axis, can be one of the following:- Left- Middle- Right
alignVert	Contains the alignment for the vertical axis, can be one of the following:- Top- Middle- Bottom
autoWidth	Indicates the element should be auto-sized based on its content.
borderInternalVert	Contains details (width, color, style) for the internal vertical borders.
borderInternalHorz	Contains details (width, color, style) for the internal horizontal borders.
horizontalRule	Contains details for the horizontal rule.

Children

Node	Description
	A float can have zero or more elements.
<p>	A float can have zero or more <p> elements.

3.4.14 element

An `<img>` element contains encoded image data that is rendered on the page. The body of the node is base-64 encoded binary data of the image.

Attributes

Attribute	Description
left	The left offset of the header in pixels.
top	The top offset of the header in pixels.
width	The width of the header in pixels.
height	The height of the header in pixels.
mimetype	The encoding of the base-64 encoded image, image/png for example.

Styles

None

Children

None

3.4.15 <table> element

The `<table>` element represents a table on the page. Each table can have one or more rows.

Attributes

Attribute	Description
left	The left offset of the header in pixels.
top	The top offset of the header in pixels.
width	The width of the header in pixels.
height	The height of the header in pixels.
styles	Contains a semicolon delimited list of name:value pair style information values.

Styles

Node	Description
width	Indicates the width of the table, the unit of measure is defined by widthType.
widthType	Indicates the how width is applied, can be one of the following:- Auto- Percentage- Point
indent	Indicates the indentation of the table, in points.
repeatHeader	Contains how may rows should be repeated when a table is split.
cellSpacing	Contains the cell spacing, in points.
rightToLeft	Indicates that the table is right to left.
type	Indicates the type of table, can be one of the follow:- Normal (word processing style) - text will wrap in cells.- Spreadsheet – text does not wrap, it will span over empty cells.

Children

Node	Description
<tr>	A table can contain one or more Table Row <tr> elements.

3.4.16 <tr> element

The `<tr>` element represents a row of a table. Each table can have one or more rows, and each `<tr>` can have one or more cells.

Attributes

Attribute	Description
left	The left offset of the header in pixels.
top	The top offset of the header in pixels.
width	The width of the header in pixels.
height	The height of the header in pixels.
styles	Contains a semicolon delimited list of name:value pair style information values.

Styles

Node	Description
height	Contains the height of the cell, unit of measure is stored in heightRule.
heightRule	Contains the rule to use when applying height, can be one of the following:- AtLeast- Exact- None
percentScale	Contains the percentage of width to take, expresses as an integer.
percentWidth	Contains the percentage of width to take, expresses as a double.
fixedWidth	If present, indicates that the cell should be this width in pixels.
widthType	Indicates the how width is applied, can be one of the following:- Auto- Percentage- Point
borderLeft	Contains details (width, color, style) for the left border.
borderRight	Contains details (width, color, style) for the right border.
borderTop	Contains details (width, color, style) for the top border.
borderBottom	Contains details (width, color, style) for the bottom border.

Node	Description
allowSplit	Indicates whether the row scan be split across pages.
keepWithNext	Indicates that the rows are to be kept together.
borderInternalVert	Contains details (width, color, style) for the internal vertical borders.
borderInternalHorz	Contains details (width, color, style) for the internal horizontal borders.

Children

Node	Description
<td>	A row can contain one or more Table Cell <td> elements.

3.4.17 <td> element

The `<td>` element represents a cell of a table. Each row can have one or more cells.

Attributes

Attribute	Description
left	The left offset of the header in pixels.
top	The top offset of the header in pixels.
width	The width of the header in pixels.
height	The height of the header in pixels.
styles	Contains a semicolon delimited list of name:value pair style information values.

Styles

Node	Description
verticalAlignment	Contains the vertical alignment to apply to text within the cell, can be one of the following:- Top- Center- Bottom
width	Contains the width of the cell, unit of measure is stored in widthType.
widthType	Indicates the how width is applied, can be one of the following:- Auto- Percentage- Point
colSpan	Contains how many cells this cell should span across.
rowSpan	Contains how may rows this cell should span across.
backgroundColor	Contains the background color for the cell, in HTML format.
borderLeft	Contains details (width, color, style) for the left border.
borderRight	Contains details (width, color, style) for the right border.
borderTop	Contains details (width, color, style) for the top border.
borderBottom	Contains details (width, color, style) for the bottom border.

Node	Description
wrap	Indicates where word wrap is supported on the cell.
margins	Contains the margins, in points, for the cell.
borderInternalVert	Contains details (width, color, style) for the internal vertical borders.
borderInternalHorz	Contains details (width, color, style) for the internal horizontal borders.

Children

Node	Description
<p>	A cell can contain zero or more <p> elements.

4. Tech Info

4.1 Supported platforms

Document Filters supports the following platforms:

- AIX POWER-64
- FreeBSD 10+ Intel-32
- FreeBSD 10+ Intel-64
- Linux ARM v7-32
- Linux ARM v8-64 (AArch64)
- Linux ARM v8-64 (Alpine)
- Linux Intel-32
- Linux Intel-64
- Linux Intel-64 (Alpine)
- Linux PPC32BE
- Linux PPC64BE
- Linux PPC64LE
- MacOS M1 (ARM64)
- MacOS X Intel-32
- MacOS X Intel-64
- Solaris Sparc 32
- Solaris Sparc 64
- Windows Intel-32
- Windows Intel-64

For a detailed list of all supported platforms, [Platform Requirement](#) lists the required and optional parameters.

4.2 Platform Requirements

4.2.1 Windows

Document Filters has been certified for the following:

Operating Systems

- Windows 7 (x86, x64)
- Windows 8 (x86, x64)
- Windows 10 (x86, x64)
- Windows 11 (x86, x64)
- Windows Server 2008 (x86, x64)
- Windows Server 2012 (x86, x64)
- Windows Server 2019 (x86, x64)
- Windows Server 2022 (x86, x64)

Refer to <https://github.com/Hyland/DocumentFilters> for detailed information on supported programming language bindings, installation instructions, and any additional runtime requirements.

4.2.2 Linux

Document Filters has been certified for the following:

Operating Systems

- Debian-based (Debian 10+, Ubuntu 16+) (x86, x64)
- RedHat-based (RHEL 7+, CentOS 7+, Fedora 12+) (x86, x64, PPC32BE, PPC64BE, PPC64LE)
- SUSE-based (SUSE LE 12+, openSUSE 12+) (x86, x64, PPC64LE)

Dependencies (all ship with system + GCC)

Module	Default Path
libdl.so.2	/lib/libdl.so.2
libpthread.so.0	/lib/libpthread.so.0

Module	Default Path
libstdc++.so.6	/usr/lib/libstdc++.so.6
libm.so.6	/lib/libm.so.6
libgcc_s.so.1	/lib/libgcc_s.so.1
libc.so.6	/lib/libc.so.6

Refer to <https://github.com/Hyland/DocumentFilters> for detailed information on supported programming language bindings, installation instructions, and any additional runtime requirements.

4.2.3 Alpine Linux

Document Filters has been certified for the following:

Operating Systems

- Alpine Linux 3.10 (AMD64, AARCH64)
- Alpine Linux 3.11 (AMD64, AARCH64)
- Alpine Linux 3.12 (AMD64, AARCH64)
- Alpine Linux 3.13 (AMD64, AARCH64)
- Alpine Linux 3.14 (AMD64, AARCH64)
- Alpine Linux 3.15 (AMD64, AARCH64)
- Alpine Linux 3.16 (AMD64, AARCH64)
- Alpine Linux 3.17 (AMD64, AARCH64)
- Alpine Linux 3.18 (AMD64, AARCH64)

Dependencies (all ship with system + GCC + libstdc++)

Module	Default Path	Architecture
libstdc++.so.6	/usr/lib/libstdc++.so.6	
libgcc_s.so.1	/lib/libgcc_s.so.1	
ld-musl-x86-64.so	/lib/ld-musl-x86-64.so	AMD64 only
libc.musl-x86_64.so.1	/lib/ld-musl-x86_64.so.1	AMD64 only

Module	Default Path	Architecture
ld-musl-aarch64.so	/lib/ld-musl-aarch64.so	AARCH64 only
libc.musl-aarch64.so.1	/lib/ld-musl-aarch64.so.1	AARCH64 only

Refer to <https://github.com/Hyland/DocumentFilters> for detailed information on supported programming language bindings, installation instructions, and any additional runtime requirements.

4.2.4 FreeBSD

Document Filters has been certified for the following:

Operating Systems

- FreeBSD 13.x (x86, x64)
- FreeBSD 14.x (x86, x64)

Dependencies (all ship with system + GCC)

Module	Default Path
libdl.so.2	/lib/libdl.so.2
libpthread.so.0	/lib/libpthread.so.0
libstdc++.so.6	/usr/lib/libstdc++.so.6 (Minimum Version: 4.1.2)
libm.so.6	/lib/libm.so.6
libgcc_s.so.1	/lib/libgcc_s.so.1 (Minimum Version: 4.1.2)
libc.so.6	/lib/libc.so.6 (Minimum Version: 2.5)

Refer to <https://github.com/Hyland/DocumentFilters> for detailed information on supported programming language bindings, installation instructions, and any additional runtime requirements.

4.2.5 Solaris

Document Filters has been certified for the following:

Operating Systems

- Solaris 10 (SPARC32, SPARC64)
- Solaris 11 (SPARC32, SPARC64)

Dependencies (all ship with system, no external or 3rd party)

Module	Default Path
libc.so.1	/lib/64/libc.so.1
libCrunG3.so.1	/usr/lib/64/libCrunG3.so.1
libdl.so.1	/lib/64/libdl.so.1
libgcc_s.so.1	/opt/developerstudio12.6/lib/compilers/CC-gcc/lib/sparcv9/libgcc_s.so.1
libm.so.2	/lib/64/libm.so.2
libpthread.so.1	/lib/64/libpthread.so.1
librt.so.1	/lib/64/librt.so.1
libstatomic.so.1	/opt/developerstudio12.6/lib/compilers/atomic/sparcv9/libstatomic.so.1
libstdc++.so.6	/opt/developerstudio12.6/lib/compilers/CC-gcc/lib/sparcv9/libstdc++.so.6

C++ Runtime

Document Filters requires the C++14 runtime that ships with Solaris Developer Studio 12.6 released in 2017. The RPATH is configured to look for dependencies in the following order:

- \$ORIGIN
- /opt/developerstudio12.6/lib/compilers/CC-gcc/lib
- /opt/developerstudio12.6/lib/compilers/atomic
- /lib

This allows you to ship the C++ runtime with your application as an alternate to installing the Solaris Developer Studio runtime.

Refer to <https://github.com/Hyland/DocumentFilters> for detailed information on supported programming language bindings, installation instructions, and any additional runtime requirements.

4.2.6 macOS

Document Filters has been certified for the following:

Operating Systems

- macOS 11 - Big Sur (x86, x64)
- macOS 12 - Monterey (x86, x64, ARM64)
- macOS 13 - Ventura (x86, x64, ARM64)
- macOS 14 - Sonoma (x86, x64, ARM64)

Dependencies (all ship with system, no external or 3rd party)

Module	Default Path
libc++.1.dylib	/usr/lib/libc++.1.dylib
libSystem.B.dylib	/usr/lib/libSystem.B.dylib

Refer to <https://github.com/Hyland/DocumentFilters> for detailed information on supported programming language bindings, installation instructions, and any additional runtime requirements.

4.2.7 IBM AIX

Document Filters has been certified for the following:

Operating Systems

- AIX 6.1 + (POWER64)

Dependencies (all ship with system, no external or 3rd party)

Module	Default Path
libc.a	/usr/lib/libc.a
libpthreads.a	/usr/lib/libpthreads.a
libpthread.a	/usr/lib/libpthread.a
libcrypt.a	/usr/lib/libcrypt.a

Refer to <https://github.com/Hyland/DocumentFilters> for detailed information on supported programming language bindings, installation instructions, and any additional runtime requirements.

4.3 Platform Differences

Feature	Windows	Linux Intel	Linux PPC	Linux ARM v7	Linux ARM v8	MacOS Intel	MacOS AARCH64	AIX
Apple DWG Subfiles	✗	✗	✗	✗	✗	✓	✗	✗
AutoCAD 2018+ Identification	✓	✓	✓	✓	✓	✓	✓	✓
AutoCAD 2018+ Metadata	✓	✓	✗	✓	✓	✓	✓	✗
AutoCAD 2018+ Text	✓	✓	✗	✓	✓	✓	✓	✗
AutoCAD 2018+ HiDef	✓	✓	✗	✓	✓	✓	✓	✗
PDF Redactions ¹	✓	✓	✗	✗	✓	✓	✓	✗
PDF PDFium Text (alt) ²	✓	✓	✗	✗	✓	✓	✓	✗
PDF PDFium HiDef (alt) ²	✓	✓	✗	✗	✓	✓	✓	✗

1. On unsupported platforms, the PDF pages will be converted to images prior to redacting. ←

2. On unsupported platforms, the Default PDF processor will be used. ←←

4.4 Supported File Formats

4.4.1 Archive

Name	Versions	Extensions	ID	Meta	Text	Hi
7-Zip		7Z	86			
7-Zip MultiPart		7Z	86			
7-Zip Self Extracting		EXE	259			
ACE	1.x, 2.x	ACE	143			
Apple Disk Image ²		DMG	95			
ARJ	2.x	ARJ	101			
Bzip2	1.x	BZ2, TBZ2	97			
ESTSoft ALZip		ALZ	159			
ESTSoft EGG		EGG	160			
GNU Zip	0.1, 1.x	GZ	87			
ISO Disk Image (ISO 9660)		ISO	93			
ISO Disk Image (UDF)	1.x, 2.x	ISO	93			
Java Archive		JAR	158			
LZH	1.x, 2.x, 3.x, 5.x	LZH	100			
Microsoft Cabinet		CAB	94			
Microsoft Office Binder		OBD	102			

Name	Versions	Extensions	ID	Meta	Text	Hi
RedHat Package Manager		RPM	103			
Roshal Archive ³	1.x-5.x	RAR	88		✓	
Roshal Archive (Multi-part) ³	1.x-5.x	RAR	88		✓	
Roshal Archive (Solid) ³	1.x-5.x	RAR	88		✓	
Roshal Archive Self Extracting		EXE	258			
Stuftlt ¹	1.x-2009	SIT	138			
Stuftlt Self Extracting Archive ¹		SEA, EXE	145			
Stuftlt X ¹		SITX	140			
UNIX AR Archive ¹		A	179			
UNIX Compress Archive		Z	168			
UNIX cpio		CPIO	98			
UNIX Tar		TAR	99			
XZ		XZ	182			
Zip ³	2.x-6.x	ZIP	59		✓	
	2.x-6.x	ZIP	59		✓	

Name	Versions	Extensions	ID	Meta	Text	Hi
Zip (Multi-part) ³						
Zip Self Extracting .exe		EXE	257		✓	
WinZip Zipx ³		ZIPX	59		✓	
PKWARE ZIP ³	2.x-6.x	ZIP	59		✓	

4.4.2 CAD

Name	Versions	Extensions	ID	Meta	Text	Hi
AutoCAD Drawing	2018-2024	DWG	238	✓	✓	
AutoCAD Drawing	2.x - 2017	DWG	55	✓	✓	
AutoCAD Drawing Exchange Format	R10 - 2010	DXF	151	✓	✓	
Dassault Systemes CATIA CAD ¹		CATIA	197			
Dassault Systemes SolidWorks		SLDASM, SLDDRW, SLDPRT	203	✓	✓	
Intergraph- Microstation CAD	8.x	DGN	112	✓	✓	
JT Open CAD ¹		JT	204			
MathCAD ¹		MCD, XMCD	148			
		X_T	205	✓	✓	

Name	Versions	Extensions	ID	Meta	Text	
Parasolid Model Part						
Pro/ENGINEER Assembly ¹		ASM	206			
Pro/ENGINEER Drawing ¹		DRW	207			
Pro/ENGINEER Drawing Form ¹		FRM	208			
Pro/ENGINEER Model Part ¹		PRT	202			
STEP 3D CAD		STP	200	✓	✓	
Stereolithography CAD (Binary) ¹		STL	198			
Stereolithography CAD (Text)		STL	199	✓	✓	
Wavefront OBJ		OBJ	209	✓	✓	

4.4.3 Database

Name	Versions	Extensions	ID	Meta	Text	HiDe
dBASE file	3,4	DBF	23		✓	
dBASE III file	3.x	DB, DB3	23		✓	
dBASE IV file	4.x	DBF	23		✓	
ISYS Index ¹	1.x - 11.x	IXA, IXB, IXC	163			
Microsoft Access	97-2021, 365	MDB, ACCDB	119		✓	
	3.x-7.x	DB	137		✓	

Name	Versions	Extensions	ID	Meta	Text	HiDe
Paradox Database File						

4.4.4 Email and Messaging

Name	Versions	Extensions	ID	Meta	Text	HiDe
Apple Mail		EMLX	64	✓	✓	
Domino DXL message		DXL	223	✓	✓	
Email Message		EML, EMLX	64	✓	✓	
MIME HTML Archive		MHT	65	✓	✓	
Encoded mail message (Multipart Alternative)		EML	64	✓	✓	
Encoded mail message (Multipart Digest)		EML	64	✓	✓	
Encoded mail message (Multipart Mixed)		EML	64	✓	✓	
Encoded mail message		EML	64	✓	✓	

Name (Multipart Newsgroup)	Versions	Extensions	ID	Meta	Text	
Encoded mail message (Multipart Signed)		EML	64	✓	✓	
Encoded mail message (TNEF)		WINMAIL.DAT	155	✓	✓	
Eudora	1-7, OSE	MBX	64	✓	✓	
HP TRIM email rendition		VMBX	64	✓	✓	
Microsoft Outlook Message	97-2019, 365	MSG	54	✓	✓	
Microsoft Outlook Personal Storage	97-2019, 365	PST	78			
Microsoft Outlook Offline Storage	97-2019, 365	OST	78			
Microsoft Outlook for Mac	2011-2019, 365	OLK14	231	✓	✓	
Microsoft Outlook for Mac	2016-2019, 365	OLK15msgsource	231	✓	✓	

Name	Versions	Extensions	ID	Meta	Text	
Microsoft Outlook for Mac Archive	2011-2019, 365	OLM	230			
Microsoft Outlook Forms Template	97-2019, 365	OFT	54	✓	✓	
Microsoft Outlook MSO object	97-2019, 365	MSO	54			
Sendmail "mbox"		MBOX	79			
Thunderbird		MBOX	79			

4.4.5 Multimedia

Name	Versions	Extensions	ID	Meta	Text	HiDe
3GP		3GP	108	✓		
Adobe Flash	1.x-CS6	SWF	66	✓		
Adobe Flash Video ¹	1.x-5.x	FLV	107			
Audio Video Interleave (AVI)	1.x-2.x	AVI	125	✓		
DVD Information File ¹	1.x-3.x	IFO, BUP	146			
DVD Video Object	1.x-3.x	VOB	109	✓		
	1.x-2.x	MXF	176			

Name	Versions	Extensions	ID	Meta	Text	HiDe
Media Exchange Format ¹						
Microsoft Windows Movie Maker ¹		MSWMM	147			
MPEG Video	1, 2	MPG	113	✓		
MPEG-1 Audio Layer 3	ID3v1, ID3v2	MP3	60	✓		
MPEG-2 Audio Layer 3	ID3v1, ID3v2	MP3	60	✓		
MPEG-4 Video		MP4	114	✓		
Musical Instrument Digital Interface (MIDI) ¹		MID, MIDI, SMF	127			
OGG FLAC Audio	1.x	FLAC	117	✓		
OGG Vorbis Audio	1.x	OGG	116	✓		
QuickTime ¹	1.x-10.x	MOV	126			
Real Media	1.x-10.x	RM	106	✓		
Waveform Audio File	1.x, AIFF-C	AIFF	115	✓		

Name Format (AIFF)	Versions	Extensions	ID	Meta	Text	HiDe
Waveform Audio File Format (WAVE)	1.x	WAV	124	✓		
Windows Media Audio	WMT 4.0, WMA 2, 7, 8, 9	WMA	82	✓		
Windows Media Video	WMV 7, 9	WMV	82	✓		

4.4.6 Other

Name	Versions	Extensions	ID	Meta	Text	HiDe
Apple Double File ¹	2.x		247			
Apple Executable ¹		BIN	144			
Apple iBook		IBOOK	228	✓	✓	
Apple PLIST Binary File		PLIST	226	✓	✓	
BIN HEX Encoded ¹		HBX, HEX, HQX	135			
BitTorrent Metafile ¹		TORRENT	149			
EOT Font ¹	1.x, 2.x		242			
EPUB Book		EPUB	227	✓	✓	
FujiXerox Docuworks ¹			240			
		CLASS	158			

Name	Versions	Extensions	ID	Meta	Text	Hidden
Java Class ¹						
Linux Executable and Linkable Format ¹		ELF	134			
Log File		LOG	85	✓	✓	
Microsoft Mso Object			215			
Microsoft OneNote	2007-2016, 365	ONE	80	✓	✓	
Microsoft Project (MPP)	98-2021, 365	MPP	90	✓	✓	✓
Microsoft Project (MPX)	2007, 2010, 2016	MPX	90	✓	✓	✓
Microsoft Windows Clipboard		CLIP, CLP	225	✓		✓
Microsoft Windows DLL ¹		DLL	133			
Microsoft Windows Executable ¹		EXE, COM, SYS	133			
Microsoft Windows Installer		MSI	142			
Microsoft Windows Shortcut ¹		LNK	150			
MO:DCA Files ¹			244			

Name	Versions	Extensions	ID	Meta	Text	HID
Open Access II (OAI)			22		✓	
Printer Command Language ¹	5e, 6	PCL	251			
QuarkExpress ¹			141			
SQLite Database ¹			239			
Symbian Executable ¹			165			
SysInternals ProcMon Logs ¹		PML	248			
Uniplex			37		✓	
UUEncode		UUE	169			
vCalendar		VCS	164	✓	✓	✓
vCard	2.1	VCF	68	✓	✓	
Visual Studio SUO file ¹		SUO	246			
Windows Resource File ¹		RES	249			
Windows Thumbs.db ¹		DB	245			
XXEncode		XXE	222			

4.4.7 Presentation

Name	Versions	Extensions	ID	Meta	Text	HiDe
Apple iWork Keynote	5.x, 6.x	KEY	211	✓	✓	
Apple Keynote	7.x -14.x	KEY	211	✓	✓	✓
Corel Presentation		SHW	196	✓		
Hancom Office HanShow	2010-2014	SHOW	76	✓	✓	✓
LibreOffice Impress	1.x-7.x	ODP	76	✓	✓	✓
Microsoft PowerPoint for Mac	98-2019, 365	PPT, PPTX	47	✓	✓	✓
Microsoft PowerPoint 1	3.0-4.0	PPT	47			
Microsoft PowerPoint	95-2021, 365	PPT, PPTX	47	✓	✓	✓
OpenOffice Impress	1.x-7.x	ODP	76	✓	✓	✓
StarOffice Impress 3	8, 9	SXI, SDI, SDP	76	✓	✓	✓

4.4.8 Raster Image

Name	Versions	Extensions	ID	Meta	Text	HiD
Adobe Photoshop 4	8.x-2024	PSD	123	✓	✓	✓
Adobe Photoshop Large	8.x-2024	PSB	260	✓		

Name	Versions	Extensions	ID	Meta	Text	HID
Document Format						
Brooktrout Fax Image ⁴			213	✓	✓	✓
CALS Raster ⁴	Type 1	CAL	191	✓	✓	✓
Canon CR2 Raw ⁴		CR2	262	✓	✓	✓
Canon CR3 Raw ⁴		CR3	263	✓	✓	✓
Canon CRW Raw ⁴		CRW	253	✓		
ZSoft Paintbush MultiPage (DCX) ⁴	1.x-3.x	DCX	187	✓	✓	✓
DICOM Medical Image ⁴	1.x-3.x	DCM	156	✓	✓	✓
Encapsulated PostScript	1.x-3.x	EPS	189	✓		✓
Encapsulated PostScript with Preview	1.x-3.x	EPS	189	✓		✓
FlashPix ⁴	1.x	FPX	220	✓	✓	✓
Flexible Image Transport System ⁴	1.x-4.x	FITS	166			
GEM Raster ⁴		IMG	192	✓	✓	✓

Name	Versions	Extensions	ID	Meta	Text	HID
Graphics Interchange Format (GIF) 4	87a, 89a, Animated	GFA, GIF, GIFF	121	✓	✓	✓
IMNET Group 4 Image 4		IMT	216	✓	✓	✓
Interchange File Format 4		IFF	194	✓	✓	✓
JEDMICS C4 4		C4	214	✓	✓	✓
JPEG 4		JPEG, JPG, JPE, JIF	73	✓	✓	✓
JPEG 2000 4		J2P, J2C, JPF	190	✓	✓	✓
JPEG XR 4		JXR, HDP	73	✓	✓	✓
JPEG XT 4		JXT	73	✓	✓	✓
MacPaint 1BPP Image 4		MAC	219	✓	✓	✓
Microsoft Document Imaging		MDI	81	✓	✓	
Microsoft Paint MSP 1BPP Image 4		MSP	218	✓	✓	✓
Microsoft Windows Bitmap 4		BMP	118	✓	✓	✓

Name	Versions	Extensions	ID	Meta	Text	HID
OS/2 Bitmap ⁴		BMP	118	✓	✓	✓
NCR Image ⁴		NCR	224	✓	✓	✓
PaintShop Pro Image ⁴	3.x-2023	PSP	237	✓	✓	✓
ZSoft Paintbush (PCX) ⁴	1.x-3.x	PCX	120	✓	✓	✓
PhotoCD ⁴		PCD	243	✓	✓	✓
Portable Bitmap PBM ⁴		PBM	181	✓	✓	✓
Portable Graymap PGM ⁴		PGM	181	✓	✓	✓
Portable Network Graphic (PNG) ⁴	1.0-1.5	PNG	122	✓	✓	✓
Portable Pixmap PPM ⁴		PPM	181	✓	✓	✓
JPEG (Progressive) ⁴		JPEG, JPG	73	✓	✓	✓
SGI Image ⁴		SGI	241	✓	✓	✓
Sun Raster Image ⁴		RAS	250	✓	✓	✓
Tagged Image	3.0-6.0	TIF, TIFF	72	✓	✓	✓

Name	Versions	Extensions	ID	Meta	Text	HiDe
Format File (TIFF) ⁴						
Targa ⁴	1.x-2.x	TGA	188	✓	✓	✓
WebP ⁴		WEBP	177	✓	✓	✓
Windows Cursor		CUR	234	✓		✓
Windows Icon		ICO	178	✓		✓
Wireless Bitmap Image ⁴		WBMP	193	✓	✓	✓
X-Windows pixmap ⁴		XPM	235	✓	✓	✓
X-Windows xbitmap ⁴		XBM	236	✓	✓	✓
X-Windows dump ⁴		XWD	254	✓	✓	✓

4.4.9 Spreadsheets

Name	Versions	Extensions	ID	Meta	Text	HiDe
Apple iWork Numbers	9	NUMBERS	212	✓	✓	✓
Apple iWork Numbers	3, 4	NUMBERS	212	✓	✓	✓
Apple Numbers	4x - 11.x	NUMBERS	212	✓	✓	✓
Comma Seperated Values		CSV	2	✓	✓	

Name	Versions	Extensions	ID	Meta	Text	HiDe
Framework Spreadsheet	III	FW3	18	✓	✓	
Hancom Office HanCell	2010 - 2014	CELL	167	✓	✓	✓
IBM Lotus Symphony Spreadsheet	1.x, 3.x	SXS, SX, ODS	76	✓	✓	✓
LibreOffice Calc	1.x-7.x	ODS	76	✓	✓	✓
LibreOffice Spreadsheet	3, 4, 5	ODS	76	✓	✓	✓
Lotus 1-2-3	1.0-6.0	WK, WKS, WK3, WK4	32		✓	
Lotus 1-2-3	97, Millennium	XLS	32		✓	
Microsoft Excel for Mac	98-2019, 365	XLS, XLSX, XML	52	✓	✓	✓
Microsoft Excel for Windows	2.x-95	XLS	52	✓	✓	
Microsoft Excel	97-2019, 365	XLS, XLSX, XML	52	✓	✓	✓
Microsoft Excel (XLSB)	2007-2016	XLSB	52	✓	✓	✓
Microsoft Works for DOS	1.x-3.x	WPS	48	✓	✓	
	2.x-9.x	WPS	34	✓	✓	

Name	Versions	Extensions	ID	Meta	Text	HiDe
Microsoft Works for Windows						
OpenOffice Calc	1.x-7.x	ODS	76	✓	✓	✓
Quattro Pro	1.x-8.x	QPW	229	✓	✓	
StarOffice Calc	8, 9	SXC, SXS, ODS	76	✓	✓	✓

4.4.10 Text and Markup

Name	Versions	Extensions	ID	Meta	Text	HiDe
ANSI Text	7-bit, 8-bit	TXT	2		✓	
ASCII Text	7-bit, 8-bit	TXT	2		✓	
HTML (Codes Revealed)		HTM, HTML	50	✓	✓	✓
HTML (Metadata Only)		HTM, HTML	67		✓	
HTML (Text Only)		HTM, HTML	49	✓	✓	✓
IBM DCA/FFT	1.x-4.x	TXT, DCA	19		✓	
IBM DCA/RTF	1.x-4.x	RFT, TXT, DCA	7		✓	
IMNET COLD		IMT	217		✓	✓
Microsoft HTML Help	1.x	CHM	58			
		ONE	80	✓	✓	

Name	Versions	Extensions	ID	Meta	Text	HiDe
Microsoft OneNote	2003-2019, 365					
Microsoft OneNote TOC ¹	2003-2019	ONETOC	154			
Multi-Byte Text		TXT	171		✓	
Rich Text Format	1.x-3.x	RTF	20	✓	✓	✓
SGML Text		SGML	40	✓	✓	
Source			8		✓	
Transcript			39		✓	
Unicode UCS2-BE			75		✓	
Unicode UCS2-LE			75		✓	
Unicode UCS4-BE			172		✓	
Unicode UCS4-LE			172		✓	
Unicode UTF16-BE			75		✓	
Unicode UTF16-LE			75		✓	
Unicode UTF8			74		✓	
Unicode UTF32-BE			172		✓	
			172		✓	

Name	Versions	Extensions	ID	Meta	Text	HiDe
Unicode UTF32-LE						
XML	XML Document File	XML	77	✓	✓	
XML	XML Record View	XML	77	✓	✓	

4.4.11 Vector Image

Name	Versions	Extensions	ID	Meta	Text	HiDe
Adobe Illustrator	1.x-2022	AI	132			✓
Adobe InDesign	1.x-2022	INDD	131	✓	✓	
AutoDesk Design Web Format ¹	1.x-7.x	DWF	210			
Computer Graphics Metafile	1.x-7.x	CGM	221	✓	✓	✓
Corel Draw Image ¹		CDR	139			
Initial Graphics Exchange Specification	1.x-10.x	IGS	201	✓	✓	
Macintosh PICT Image	1.x-4.x	PICT	184	✓	✓	✓
Microsoft Visio ¹	3.x	VSD	70			

Name	Versions	Extensions	ID	Meta	Text	HiDe
Microsoft Visio (VSD)	4.0-2019, 365	VSD	70	✓	✓	✓
Microsoft Visio (VSDX)	2013-2019, 365	VSDX	70	✓	✓	✓
Microsoft XPS	1.x	XPS	83	✓	✓	✓
Open XML Paper Specification (OXPS)	1.x	OXPS	83	✓	✓	✓
Scalable Vector Graphic	1.x, 2.x	SVG	185	✓	✓	✓
Scitex CT	1.x-3.x	CT	252	✓	✓	✓
StarView Metafile	1.x-4.x	SVM	180	✓	✓	✓
Windows Enhanced Meta File	1.x-2.x	EMF	129	✓	✓	✓
Windows Meta File	1.x-3.x	WMF	128	✓	✓	✓
Word Perfect Graphics	1	WPG	233	✓	✓	
Word Perfect Graphics	2	WPG	233	✓	✓	✓

4.4.12 Word Processor and Office

Name	Versions	Extensions	ID	Meta	Text	
Adobe PDF	1.x-2.x	PDF	51	✓	✓	
Adobe PDF Package	1.7-2.x	PDF	51	✓	✓	
Adobe PDF Portfolio	1.7-2.x	PDF	51	✓	✓	
Adobe PDF XFA Forms	1.6-2.x	PDF	51	✓	✓	
Adobe PostScript	1.x-3.x	PS	130			
Ami Pro for Windows		AMI, SAM	30		✓	
Apple iWork Pages	4.x	PAGES	195	✓	✓	
Apple Pages	5.x - 11.x	PAGES	195	✓	✓	
Framework WP	3.x	FW3	18		✓	
Hangul	3.x	HWP	56	✓	✓	
Hangul	96 - 2014	HWP	56	✓	✓	
Hangul	2014-2020	HWPX	265	✓	✓	
IBM DCA/FFT		RFT, FFT	19		✓	
IBM DisplayWrite 4	4	RFT, DCA, DW4, DOC	15		✓	
IBM DisplayWrite 5	5	RFT, DCA, .DW5, .DOC	26		✓	
	1.x, 3.x	ODT	76	✓	✓	

Name	Versions	Extensions	ID	Meta	Text	
IBM Lotus Symphony Document						
JungUm		GUL	170	✓		
JustSystems Ichitaro 4 ¹	4.x	JTD, JBW, JTT	92			
JustSystems Ichitaro	5.x-15.x	JTD, JBW, JTT	92	✓	✓	
LibreOffice Document	1.x-7.x	ODT	76	✓	✓	
LibreOffice Writer	1.1 - 4.x	ODT	76	✓	✓	
Lotus Manuscript	1.0, 2.x	MANU, MNU,MAN	14	✓	✓	
Lotus Notes ¹		NSF	63			
Lotus WordPro ¹		LWP	57			
Mass 11	1.x-7.x	M11	29		✓	
Microsoft Publisher	98-2019, 365	PUB	104	✓		
Microsoft Word for DOS	1.x-6.x	DOC	5	✓	✓	
Microsoft Word for Mac	1.x-6.x	DOC, DOCX	186	✓	✓	
Microsoft Word for Mac	98-2019,365	DOC, DOCX, XML	186	✓	✓	

Name	Versions	Extensions	ID	Meta	Text	
Microsoft Word for Windows	1.x-4.x	DOC	25	✓	✓	
Microsoft Word for Windows	6.x, 95-2019, 365	DOC, DOCX, XML	25	✓	✓	
MultiMate	1.x-7.x	DOX	4		✓	
MultiMate Advantage	1.x-5.x	DOX	4		✓	
OpenOffice Writer	1.x-7.x	ODT	76	✓	✓	
Professional Write for DOS	1.x-5.x	PW, PW1, PW2	9		✓	
Professional Write Plus for Windows	1.x-4.x	PW	9		✓	
Q&A Write	3.x-95	QA, QA3	24		✓	
QuarkXpress ¹		QXx, QCx	141			
QuickBooks Backup ¹		QBB	105			
QuickBooks for Windows ¹		QBW	111			
StarOffice Writer	8, 9	SXW, SDW	76	✓	✓	
TrueType Font ¹		TTF	110			
Wang IWP	1.x-3.x	IWP	16		✓	

Name	Versions	Extensions	ID	Meta	Text	
Wang WP Plus		IWP	27		✓	
Windows Write		WRI	46	✓	✓	
WordPerfect for DOS	3.x-6.x	WPD	10	✓	✓	
WordPerfect for Macintosh	1	WPD	261	✓	✓	
WordPerfect for Macintosh	1.x-3.5	WPD	6	✓	✓	
WordPerfect for Windows	5.1-12.0, X3-X9, 2020-2023	WPD, WP5	6	✓	✓	
Wordstar for DOS	3.x-7.x	WS2, DOC	3	✓	✓	
Wordstar for Windows	1.x, 2.x	WS, WSx	12		✓	
Wordstar 2000	1.x-3.x	WSD	13		✓	
XYwrite	4	XY	11		✓	

-
1. ID only ←
 2. MacOS only ←
 3. Archive comments extracted as text ←
 4. Requires OCR for text extraction ←

4.5 Document format codes

Additional codes are defined as Document Filters is enhanced to support new file formats, however existing codes never change.

ID	Description	MIME Type	File Type Category
2	Text (ASCII)	text/plain	IGR_FILETYPE_TEXT
3	Wordstar 3 or 4	application/vnd.wordstar	IGR_FILETYPE_WORD_PROCESSOR
4	Multimate	application/vnd.multimate	IGR_FILETYPE_WORD_PROCESSOR
5	Microsoft Word	application/msword	IGR_FILETYPE_WORD_PROCESSOR
6	WordPerfect 5 - X4	application/vnd.wordperfect	IGR_FILETYPE_WORD_PROCESSOR
7	IBM DCA/RFT	application/dca-rft	IGR_FILETYPE_WORD_PROCESSOR
8	Source Code	text/plain	IGR_FILETYPE_TEXT
9	Professional Write	application/vnd.prowrite	IGR_FILETYPE_WORD_PROCESSOR
10	WordPerfect 4.2	application/vnd.wordperfect	IGR_FILETYPE_WORD_PROCESSOR
11	XYWrite	application/vnd.xy-write	IGR_FILETYPE_WORD_PROCESSOR
12	Wordstar 5	application/ vnd.wordstar;version=5	IGR_FILETYPE_WORD_PROCESSOR
13	Wordstar 2000	application/ vnd.wordstar;version=2000	IGR_FILETYPE_WORD_PROCESSOR
14	Lotus Manuscript	application/vnd.lotus-manuscript	IGR_FILETYPE_WORD_PROCESSOR
15	DisplayWrite 4	application/ vnd.displaywrite;version=4	IGR_FILETYPE_WORD_PROCESSOR
16	Wang IWP	application/vnd.wangwp	IGR_FILETYPE_WORD_PROCESSOR
18	FrameWork 3	application/vnd.framework3	IGR_FILETYPE_WORD_PROCESSOR
19	IBM FFT	application/dca-fft	IGR_FILETYPE_WORD_PROCESSOR
20	Microsoft RTF	application/rtf	IGR_FILETYPE_WORD_PROCESSOR

ID	Description	MIME Type	File Type Category
22	Open Access II	application/openaccess2	IGR_FILETYPE_WORD_PROCESSOR
23	dBASE / FoxPro	application/x-dbf	IGR_FILETYPE_DATABASE
24	Q & A Write	application/vnd.qa3	IGR_FILETYPE_WORD_PROCESSOR
25	Microsoft Word Sub Types: [Word 2.0, Word 2003 XML Document, Word 2007, Word 2007 (DRM Protected), Word 2007 (Password Protected), Word 2007 XML, Word 6.0, Word 95, Word 97/2003, Word for Mac 5]	application/msword	IGR_FILETYPE_WORD_PROCESSOR
26	DisplayWrite 5	application/ vnd.displaywrite;version=5	IGR_FILETYPE_WORD_PROCESSOR
27	Wang WP PLUS	application/vnd.wangwpplus	IGR_FILETYPE_WORD_PROCESSOR
29	MASS-11 PC	application/vnd.mass11	IGR_FILETYPE_WORD_PROCESSOR
30	Lotus Ami Pro	application/vnd.amipro	IGR_FILETYPE_WORD_PROCESSOR
32	SpreadSheet/Other	application/octet-stream	IGR_FILETYPE_UNKNOWN

ID	Description	MIME Type	File Type Category
	Sub Types: [Microsoft Works (4), Microsoft Works (5), Lotus 1-2-3 (3), Lotus 1-2-3 (WK1), Lotus 1-2-3 (98), Lotus 1-2-3 (97), Quatro]		
33	Best Guess (of...)	application/octet-stream	IGR_FILETYPE_UNKNOWN
34	Microsoft Works	application/vnd.ms-works	IGR_FILETYPE_WORD_PROCESSOR
35	External Image Viewer	application/octet-stream	IGR_FILETYPE_GRAPHIC
37	Uniplex	application/vnd.uniplex	IGR_FILETYPE_UNKNOWN
38	External OEM DLL	application/octet-stream	IGR_FILETYPE_UNKNOWN
39	Transcript	text/plain	IGR_FILETYPE_TEXT
40	SGML Text Only	text/sgml	IGR_FILETYPE_TEXT_MARKUP
43	Compuserve E-Mail	application/octet-stream	IGR_FILETYPE_EMAIL_MESSAGE
44	ISYS Annotation	text/plain;version=isisanno	IGR_FILETYPE_TEXT
45	SQL Data Source	text/plain;version=sql	IGR_FILETYPE_TEXT
46	Windows Write	application/vnd.ms-winwrite	IGR_FILETYPE_WORD_PROCESSOR
47	Microsoft PowerPoint	application/vnd.ms-powerpoint	IGR_FILETYPE_PRESENTATION

ID	Description	MIME Type	File Type Category
	Sub Types: [PowerPoint 97/2003, PowerPoint 2007, PowerPoint 2007 (DRM Protected), PowerPoint 2007 (Password Protected)]		
49	HTML Text Only	text/html	IGR_FILETYPE_TEXT_MARKUP
50	HTML Raw Codes	text/html	IGR_FILETYPE_TEXT_MARKUP
51	Adobe Acrobat (PDF)	application/pdf	IGR_FILETYPE_PUBLISHING
52	Microsoft Excel	application/msexcel	IGR_FILETYPE_SPREADSHEET

ID	Description	MIME Type	File Type Category
	Sub Types: [Excel 2.0, Excel 2007, Excel 2007 (DRM Protected), Excel 2007 (Password Protected), Excel 3.0, Excel 3.0 (Password Protected), Excel 4.0, Excel 4.0 (Password Protected), Excel 5.0, Excel 5.0 (Password Protected), Excel 97/2003, Excel 97/2003 (Password Protected), XML Spreadsheet 2003, Microsoft Graph 97/2003, MHTML Web Archive]		
53	E-mail	application/octet-stream	IGR_FILETYPE_EMAIL_MESSAGE
54	Microsoft MSG	application/vnd.ms-outlook	IGR_FILETYPE_EMAIL_MESSAGE
55	AutoDesk AutoCAD	application/vnd.dwg	IGR_FILETYPE_CAD
56	Hangul	application/x-hwp	IGR_FILETYPE_WORD_PROCESSOR

ID	Description	MIME Type	File Type Category
58	Microsoft HTML Help (CHM)	application/vnd.ms-htmlhelp	IGR_FILETYPE_ARCHIVE
59	ZIP Archive	application/zip	IGR_FILETYPE_ARCHIVE
60	MP3	audio/mpeg	IGR_FILETYPE_MULTIMEDIA_AUDIO
63	Lotus Notes	application/vnd.lotus-notes	IGR_FILETYPE_EMAIL_CONTAINER
64	Microsoft EML	message/rfc822	IGR_FILETYPE_EMAIL_MESSAGE
65	Microsoft Web Archive (MHT)	multipart/related	IGR_FILETYPE_EMAIL_MESSAGE
66	Adobe Shockwave Flash	application/x-shockwave-flash	IGR_FILETYPE_MULTIMEDIA
67	HTML Metadata only	text/plain	IGR_FILETYPE_TEXT_MARKUP
68	vCard Electronic business card	text/x-vcard	IGR_FILETYPE_EMAIL_CONTACT
70	Microsoft Visio Sub Types: [Visio 2013, Visio VDX]	application/vnd.visio	IGR_FILETYPE_GRAPHIC_VECTOR
71	IFilter	application/octet-stream	IGR_FILETYPE_UNKNOWN
72	TIFF Image	image/tiff	IGR_FILETYPE_GRAPHIC_RASTER
73	JPEG Exif	image/jpeg	IGR_FILETYPE_GRAPHIC_RASTER
74	Text (UTF8)	text/plain; charset=utf-8	IGR_FILETYPE_TEXT
75	Text (UTF16)	text/plain; charset=utf-16	IGR_FILETYPE_TEXT
76	Open Document Format	application/ vnd.oasis.opendocument.text	IGR_FILETYPE_WORD_PROCESSOR

ID	Description	MIME Type	File Type Category
	Sub Types: [OpenDocument Drawing, OpenDocument Presentation, OpenDocument Spreadsheet, OpenDocument Text]		
77	XML document file	text/xml	IGR_FILETYPE_TEXT_MARKUP
78	Microsoft PST/OST	application/vnd.ms-outlook-pst	IGR_FILETYPE_EMAIL_CONTAINER
79	Sendmail MBOX	application/mbox	IGR_FILETYPE_EMAIL_CONTAINER
80	Microsoft OneNote	application/msonenote	IGR_FILETYPE_OFFICE_OTHER
81	Microsoft Document Image (MDI)	image/mdi	IGR_FILETYPE_GRAPHIC_RASTER
82	Windows Multimedia (WMA/ WMV)	video/x-ms-wmv	IGR_FILETYPE_MULTIMEDIA
83	Microsoft XPS	application/vnd.ms-xpsdocument	IGR_FILETYPE_PUBLISHING
84	Metadata	text/plain	IGR_FILETYPE_TEXT
85	Log file	text/log	IGR_FILETYPE_TEXT
86	7ZIP Archive	application/x-7z-compressed	IGR_FILETYPE_ARCHIVE
87	GZ Archive	application/gzip	IGR_FILETYPE_ARCHIVE
88	RAR Archive	application/x-rar-compressed	IGR_FILETYPE_ARCHIVE
89	Apple iWork	application/vnd.apple.iwork	IGR_FILETYPE_WORD_PROCESSOR
90	Microsoft Project	application/vnd.ms-project	IGR_FILETYPE_OFFICE_OTHER
92	JustSystems Ichitaro	application/vnd.ichitaro	IGR_FILETYPE_WORD_PROCESSOR

ID	Description	MIME Type	File Type Category
93	ISO Disk Image	application/x-iso9660-image	IGR_FILETYPE_ARCHIVE
94	Microsoft Cabinet Archive	application/vnd.ms-cab-compressed	IGR_FILETYPE_ARCHIVE
95	Apple Disk Image	application/octet-stream	IGR_FILETYPE_ARCHIVE
96	HFS Partition	application/octet-stream	IGR_FILETYPE_ARCHIVE
97	BZIP Archive	application/x-bzip	IGR_FILETYPE_ARCHIVE
98	CPIO UNIX Archive	application/x-cpio	IGR_FILETYPE_ARCHIVE
99	TAR Archive	application/x-gtar	IGR_FILETYPE_ARCHIVE
100	LZH Archive	application/x-lzh	IGR_FILETYPE_ARCHIVE
101	ARJ Archive	application/x-arj-compressed	IGR_FILETYPE_ARCHIVE
102	Microsoft Binder	application/x-msbinder	IGR_FILETYPE_ARCHIVE
103	RPM Package	application/x-rpm	IGR_FILETYPE_ARCHIVE
104	Microsoft Publisher	application/x-mspublisher	IGR_FILETYPE_PUBLISHING
105	QuickBooks Backup	application/vnd.quickbooks	IGR_FILETYPE_APPDATA
106	Real Media	application/rn-realmedia	IGR_FILETYPE_MULTIMEDIA
107	Adobe Flash Video	application/x-shockwave-flash	IGR_FILETYPE_MULTIMEDIA_VIDEO
108	3gp Audio	audio/3gpp	IGR_FILETYPE_MULTIMEDIA_AUDIO
109	DVD Video Object	video/x-ms-vob	IGR_FILETYPE_MULTIMEDIA
110	TrueType Font	application/x-font-ttf	IGR_FILETYPE_SYSTEM
111	QuickBooks for Windows	application/vnd.quickbooks	IGR_FILETYPE_APPDATA
112	Intergraph/ Microstation CAD (DGN)	application/vnd.dgn	IGR_FILETYPE_CAD

ID	Description	MIME Type	File Type Category
113	MPEG Video	video/mpeg	IGR_FILETYPE_MULTIMEDIA_VIDEO
114	MPEG-4 Video	video/mpeg4-generic	IGR_FILETYPE_MULTIMEDIA_VIDEO
115	AIFF	audio/x-aiff	IGR_FILETYPE_MULTIMEDIA_AUDIO
116	Ogg Vorbis Audio	audio/x-oggvorbis	IGR_FILETYPE_MULTIMEDIA_AUDIO
117	Ogg Flac Audio	audio/x-oggflac	IGR_FILETYPE_MULTIMEDIA_AUDIO
118	Microsoft Bitmap (BMP)	image/bmp	IGR_FILETYPE_GRAPHIC_RASTER
119	Microsoft Access Database (MDB)	application/x-msaccess	IGR_FILETYPE_DATABASE
120	PCX Image	image/x-pcx	IGR_FILETYPE_GRAPHIC_RASTER
121	GIF Image (GIF)	image/gif	IGR_FILETYPE_GRAPHIC_RASTER
122	PNG Image (PNG)	image/png	IGR_FILETYPE_GRAPHIC_RASTER
123	Adobe Photoshop Image (PSD)	application/photoshop	IGR_FILETYPE_GRAPHIC_RASTER
124	WAVE Audio	audio/wave	IGR_FILETYPE_MULTIMEDIA_AUDIO
125	AVI Video	video/avi	IGR_FILETYPE_MULTIMEDIA_VIDEO
126	Apple QuickTime	application/quicktime	IGR_FILETYPE_MULTIMEDIA_VIDEO
127	MIDI Audio	audio/midi	IGR_FILETYPE_MULTIMEDIA_AUDIO
128	Windows Metafile (WMF)	image/wmf	IGR_FILETYPE_GRAPHIC_VECTOR
129	Windows Enhanced Metafile (EMF)	image/emf	IGR_FILETYPE_GRAPHIC_VECTOR
130	PostScript (PS)	application/postscript	IGR_FILETYPE_GRAPHIC_VECTOR
131	Adobe InDesign	application/x-adobe-indesign	IGR_FILETYPE_PUBLISHING
132	Adobe Illustrator	application/illustrator	IGR_FILETYPE_PUBLISHING

ID	Description	MIME Type	File Type Category
133	Windows EXE/DLL (EXE,DLL)	application/x-dosexec	IGR_FILETYPE_SYSTEM
134	Linux ELF	application/x-elf	IGR_FILETYPE_SYSTEM
135	BinHex Encoded	application/binhex	IGR_FILETYPE_ARCHIVE
136	dBASE Database	application/x-dbf	IGR_FILETYPE_DATABASE
137	Paradox Database	application/vnd.paradox	IGR_FILETYPE_DATABASE
138	Stuftit Archive	application/x-stuffit	IGR_FILETYPE_ARCHIVE
139	Corel Draw	application/coreldraw	IGR_FILETYPE_PUBLISHING
140	Stuftit X	application/x-stuffitx	IGR_FILETYPE_ARCHIVE
141	Quark XPress	application/ vnd.quark.quarkxpress	IGR_FILETYPE_PUBLISHING
142	Microsoft Installer (MSI)	application/x-windows-installer	IGR_FILETYPE_SYSTEM
143	ACE Archive	application/x-ace-compressed	IGR_FILETYPE_ARCHIVE
144	Apple Bin File	application/vnd.applebin	IGR_FILETYPE_ARCHIVE
145	Stuftit Self Extracting Archive	application/x-stuffitsea	IGR_FILETYPE_ARCHIVE
146	DVD IFO (IFO)	application/vnd.ifo	IGR_FILETYPE_MULTIMEDIA
147	Microsoft Movie Maker	application/vnd.ms-moviemaker	IGR_FILETYPE_APPDATA
148	MathCAD	application/x-mathcad	IGR_FILETYPE_CAD
149	BitTorrent Metafile	application/x-bittorrent	IGR_FILETYPE_APPDATA
150	Shortcut/Link	application/x-ms-shortcut	IGR_FILETYPE_SYSTEM
151	AutoCAD DXF	image/vnd.dxf	IGR_FILETYPE_CAD
153	Outlook Express mailbox	application/x-ms-outlookexpress	IGR_FILETYPE_EMAIL_CONTAINER

ID	Description	MIME Type	File Type Category
154	Microsoft OneNote TOC	application/msonenote	IGR_FILETYPE_OFFICE_OTHER
155	Microsoft Outlook RTF	application/x-ms-winmail	IGR_FILETYPE_EMAIL_CONTAINER
156	DICOM	application/dicom	IGR_FILETYPE_GRAPHIC_RASTER
157	Text with ISYS markup	text/plain	IGR_FILETYPE_TEXT
158	Java Class (CLASS)	application/java-vm	IGR_FILETYPE_SYSTEM
159	ALZ Archive (ALZ)	application/x-alz-compressed	IGR_FILETYPE_ARCHIVE
160	EGG Archive (EGG)	application/x-egg-compressed	IGR_FILETYPE_ARCHIVE
161	NDOC File (NDOC)	application/vnd.ndoc	IGR_FILETYPE_WORD_PROCESSOR
162	ISYS Transaction File (ISTX)	application/x-isys-transaction	IGR_FILETYPE_SYSTEM
163	ISYS Index File (ISYS.IX*)	application/x-isys-index	IGR_FILETYPE_SYSTEM
164	VCalendar Electronic Calendar	text/calendar	IGR_FILETYPE_EMAIL_CALENDAR
165	Symbian OS File	application/symbian	IGR_FILETYPE_SYSTEM
166	FITS Image	image/fits	IGR_FILETYPE_GRAPHIC_RASTER
167	Hancom Office Hancell	application/vnd.hancell	IGR_FILETYPE_SPREADSHEET
168	Compress Archive (.Z)	application/x-compress	IGR_FILETYPE_ARCHIVE
169	UU Encoded File	text/x-uuencode	IGR_FILETYPE_ARCHIVE

ID	Description	MIME Type	File Type Category
170	JungUm File (.GUL)	application/jung	IGR_FILETYPE_WORD_PROCESSOR
171	Text (Multibyte)	text/plain	IGR_FILETYPE_TEXT
172	Text (UTF32)	text/plain; charset=utf-32	IGR_FILETYPE_TEXT
173	Microsoft Windows Binary	application/octet-stream	IGR_FILETYPE_SYSTEM
174	Empty File	application/octet-stream	IGR_FILETYPE_SYSTEM
175	Corrupt File	application/octet-stream	IGR_FILETYPE_SYSTEM
176	Material Exchange Format (MXF)	application/vnd.mxf	IGR_FILETYPE_MULTIMEDIA
177	WEBP Image	image/webp	IGR_FILETYPE_GRAPHIC_RASTER
178	ICON Image	image/vnd.microsoft.icon	IGR_FILETYPE_GRAPHIC_RASTER
179	Unix AR Archive	application/x-ar-compressed	IGR_FILETYPE_ARCHIVE
180	StarView Metafile (SVM)	image/starview-metafile	IGR_FILETYPE_GRAPHIC_VECTOR
181	Netpbm (PPM, PBM, PGM, PNM)	image/x-portable-bitmap	IGR_FILETYPE_GRAPHIC_RASTER
182	XZ Archive	application/x-xz	IGR_FILETYPE_ARCHIVE
183	DocPack	application/x-isis-docpack	IGR_FILETYPE_SYSTEM
184	Macintosh PICT Image	image/x-pict	IGR_FILETYPE_GRAPHIC_VECTOR
185	Scalable Vector Graphic	image/svg+xml	IGR_FILETYPE_GRAPHIC_VECTOR
186	Microsoft Word for Mac	application/msword	IGR_FILETYPE_WORD_PROCESSOR
187	PCX Image	image/x-dcx	IGR_FILETYPE_GRAPHIC_RASTER
188	Targa Image	image/x-tga	IGR_FILETYPE_GRAPHIC_RASTER

ID	Description	MIME Type	File Type Category
189	EPS Image	application/postscript	IGR_FILETYPE_GRAPHIC_VECTOR
190	JPEG2000 Image	image/jp2	IGR_FILETYPE_GRAPHIC_RASTER
191	CALS Raster Image	image/cals	IGR_FILETYPE_GRAPHIC_RASTER
192	GEM Image	image/gem	IGR_FILETYPE_GRAPHIC_RASTER
193	WBMP Image	image/wbmp	IGR_FILETYPE_GRAPHIC_RASTER
194	IFF Image	image/iff	IGR_FILETYPE_GRAPHIC_RASTER
195	Apple Pages	application/vnd.apple.pages	IGR_FILETYPE_WORD_PROCESSOR
196	Corel Presentation	application/x-corelpresentations	IGR_FILETYPE_PRESENTATION
197	CATIA CAD	application/catia	IGR_FILETYPE_CAD
198	Stereolithography CAD (Binary)	model/x.stl-binary	IGR_FILETYPE_CAD
199	Stereolithography CAD (Text)	model/x.stl-ascii	IGR_FILETYPE_CAD
200	STEP 3D CAD	application/x-step	IGR_FILETYPE_CAD
201	IGS CAD	model/iges	IGR_FILETYPE_CAD
202	Pro/ENGINEER Model	model/prt	IGR_FILETYPE_CAD
203	SolidWorks	application/sldworks	IGR_FILETYPE_CAD
204	JT CAD	application/x-jt	IGR_FILETYPE_CAD
205	Parasolid Model Part	application/x-parasolid	IGR_FILETYPE_CAD
206	Pro/ENGINEER Assembly	application/pro-e	IGR_FILETYPE_CAD
207	Pro/ENGINEER Drawing	application/pro-e	IGR_FILETYPE_CAD

ID	Description	MIME Type	File Type Category
208	Pro/ENGINEER Drawing Form	application/pro-e	IGR_FILETYPE_CAD
209	OBJ 3D Image	text/obj-cad	IGR_FILETYPE_CAD
210	Autodesk WHIP	application/x-dwf	IGR_FILETYPE_CAD
211	Apple Keynote	application/vnd.apple.keynote	IGR_FILETYPE_PRESENTATION
212	Apple Numbers	application/vnd.apple.numbers	IGR_FILETYPE_SPREADSHEET
213	Brooktrout Fax	image/brk	IGR_FILETYPE_GRAPHIC_RASTER
214	JEDMICS C4	image/c4	IGR_FILETYPE_GRAPHIC_RASTER
215	MSO Object	application/x-ms-object	IGR_FILETYPE_GRAPHIC_RASTER
216	IMNET Medical	image/vnd.imnet-c4	IGR_FILETYPE_GRAPHIC_RASTER
217	IMNET COLD	image/vnd.imnet-text	IGR_FILETYPE_GRAPHIC_VECTOR
218	MS Paint Image	application/vnd.ms-paint	IGR_FILETYPE_GRAPHIC_RASTER
219	MAC Paint Image	image/x-macpaint	IGR_FILETYPE_GRAPHIC_RASTER
220	FlashPix	image/vnd.fpx	IGR_FILETYPE_GRAPHIC_RASTER
221	Computer Graphics Metafile	image/cgm	IGR_FILETYPE_GRAPHIC_VECTOR
222	Xxencoded File	application/x-xxencode	IGR_FILETYPE_ARCHIVE
223	Domino XML	application/x-dxlf	IGR_FILETYPE_EMAIL_MESSAGE
224	NCR Image	image/ncr	IGR_FILETYPE_GRAPHIC_RASTER
225	Microsoft Windows Clipboard	application/x-msclip	IGR_FILETYPE_SYSTEM
226	Apple Property List (Binary)	application/x-plist	IGR_FILETYPE_SYSTEM
227	Electronic Publication	application/epub+zip	IGR_FILETYPE_PUBLISHING

ID	Description	MIME Type	File Type Category
228	Apple iBooks Publication	application/x-ibooks+zip	IGR_FILETYPE_PUBLISHING
229	Quattro Pro Spreadsheet	application/x-quattro-pro	IGR_FILETYPE_APPDATA
230	Outlook for Mac OLM	application/vnd.ms-outlook-pst	IGR_FILETYPE_EMAIL_CONTAINER
231	Outlook for Mac Message Source	application/vnd.ms-outlook-olk15-msgsource	IGR_FILETYPE_EMAIL_MESSAGE
233	WordPerfect Graphics	application/x-fmt/395	IGR_FILETYPE_GRAPHIC_VECTOR
234	Windows Cursor Image	image/vnd.microsoft.icon	IGR_FILETYPE_GRAPHIC_RASTER
235	X-Windows Bitmap	image/x-xbitmap	IGR_FILETYPE_GRAPHIC_RASTER
236	X-Windows Pixmap	image/x-xpixmap	IGR_FILETYPE_GRAPHIC_RASTER
237	Paintshop Pro	image/x-paintshoppro	IGR_FILETYPE_GRAPHIC_RASTER
238	AutoDesk AutoCAD 2018	application/ vnd.dwg;version=2018	IGR_FILETYPE_CAD
239	SQLite DB	application/vnd.sqlite	IGR_FILETYPE_DATABASE
240	Fuji Xerox DocuWorks	application/ vnd.fujixerox.docuworks	IGR_FILETYPE_WORD_PROCESSOR
241	SGI Image File	image/sgi	IGR_FILETYPE_GRAPHIC_RASTER
242	MS Embedded OpenType fonts	application/vnd.ms-fontobject	IGR_FILETYPE_SYSTEM
243	Kodak Photo CD	image/pcd	IGR_FILETYPE_GRAPHIC_RASTER
244	AFP MO:DCA	application/vnd.afpc.modca	IGR_FILETYPE_PUBLISHING
245	Windows Thumbnail Cache		IGR_FILETYPE_SYSTEM

ID	Description	MIME Type application/ vnd.microsoft.windows.thumbnail- cache	File Type Category
246	VS Solution User Options	application/vnd.ms-vs-user-options	IGR_FILETYPE_SYSTEM
247	AppleDouble	application/applefile	IGR_FILETYPE_SYSTEM
248	Process Monitor Log	application/vnd.ms-windows.procmon.log	IGR_FILETYPE_SYSTEM
249	Resource File	application/vnd.ms-resourcefile	IGR_FILETYPE_SYSTEM
250	Sun Raster Image (RAS)	image/x-sun-raster	IGR_FILETYPE_GRAPHIC_RASTER
251	Printer Command Language	application/vnd.hp-PCL	IGR_FILETYPE_PUBLISHING
252	Scitex CT Image(CT)	image/scitex-ct	IGR_FILETYPE_GRAPHIC_RASTER
253	Canon Raw Image	image/x-canon-crw	IGR_FILETYPE_GRAPHIC_RASTER
254	X Window System Screen Dump	image/x-xwindowdump	IGR_FILETYPE_SYSTEM
256	Password Protected Office File	application/octet-stream	IGR_FILETYPE_OFFICE_OTHER
257	ZIP Archive (SFX)	application/x-dosexec+zip	IGR_FILETYPE_ARCHIVE
258	RAR Archive (SFX)	application/x-dosexec+rar	IGR_FILETYPE_ARCHIVE
259	7ZIP Archive (SFX)	application/x-dosexec+7z	IGR_FILETYPE_ARCHIVE
260	Adobe Photoshop Large Document Format (PSB)	application/photoshop-psb	IGR_FILETYPE_GRAPHIC_RASTER
261	WordPerfect for Mac 1	application/vnd.wordperfect	IGR_FILETYPE_WORD_PROCESSOR

ID	Description	MIME Type	File Type Category
262	Canon Camera Raw Image 2	image/x-canon-cr2	IGR_FILETYPE_GRAPHIC_RASTER
263	Canon Camera Raw Image 3	image/x-canon-cr3	IGR_FILETYPE_GRAPHIC_RASTER
264	PDF XFA XML	application/x-pdfxfa	IGR_FILETYPE_PUBLISHING
265	Hancom Hangul HWPX	application/hwp+zip	IGR_FILETYPE_WORD_PROCESSOR
266	Microsoft Compound Binary File	application/octet-stream	IGR_FILETYPE_APPDATA
267	Microsoft Compound Binary File (Corrupted)	application/octet-stream	IGR_FILETYPE_APPDATA
268	ZIP Archive (Corrupted)	application/zip	IGR_FILETYPE_ARCHIVE
269	JSON	application/json	IGR_FILETYPE_TEXT_DATA
270	HL7 Container	application/hl7-v2	IGR_FILETYPE_APPDATA
271	HL7 Message	application/hl7-v2	IGR_FILETYPE_APPDATA

4.6 Processing Options

Name	Description	Applies To
ARCHIVE_IGNORE_CHECKSUM	Enable or disable checksum validation for extracted archive files. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
BMP_IGNORE_MEMORY_LIMITS	Enable or disable enforced memory limits when processing large images. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
BW_MODE	Set the value of Black-and-White Algorithm. [-1 .. 7] Type: string Default: -1	document
CAD_PAGE_HEIGHT	Set the default page height in points for CAD documents with unknown page information. Type: number:pixel Default: 0	canvas
CLIP_ENUMERATE	Enable or disable enumeration of attachments when processing Microsoft Clipboard files. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
CODEPAGE	Specify the default character set for documents with unspecified encoding. Type: number Default: 0	document
CONTAINERS_LIST_DIR	Enable or disable the rendering of container directories when rendering to hi-def. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
CSV_COLLAPSE_ROWS	Set how repeating rows may be collapsed when generating CSV output. Can be boolean value, or number. Type: string	document

Name	Description	Applies To
CSV_MAX_REPEATING_CELLS	Set the maximum number of repeating empty cells to include in CSV output. Type: number Default: 512	document
CSV_TRIM_EMPTY_CELLS	Enable or disable trimming of empty trailing cells in CSV output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
DBF_TABLE_VIEW	Set the output type when processing dBase database files. 1 flattens memo fields into a single output; 1 treats memos as sub-files. Type: number Default: 1	document
DEFAULT_LCID	Set the value of locale used for date conversion. Type: number Default: 0	document
DETECT_MACROS	Enable or disable reporting macro presence as meta data. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
DETECT_MACROS_ALWAYS_REPORT	Enable or disable reporting of macro detection as metadata. When enabled, macro detection is always reported, even if no macros are found. When disabled, macros are reported only when present. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
DICOM_SHOW_OFFSET	Enable or disable the display of byte offsets for DICOM elements in the extracted metadata output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
DICOM_SHOW_SIZES	Enable or disable the inclusion of data sizes for DICOM elements in the extracted metadata output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document

Name	Description	Applies To				
DICOM_SHOW_TAGS	Enable or disable the display of DICOM tags in the extracted metadata output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document				
DICOM_SHOW_TYPE	Enable or disable the inclusion of DICOM element types in the extracted metadata output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document				
DOCUMENT_FOOTERS	Enable or disable adding page footers to hi-def output Type: bool Default: ON Possible Values: ['ON', 'OFF']	document				
DOCUMENT_HEADERS	Enable or disable adding page headers to hi-def output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document				
EMAILMODE	Set which email headers to include in the output when processing email files (EML, MSG, PST). Type: string:enum Default: VisibleHeaders Possible Values: ['VisibleHeaders', 'AllHeaders', 'MapiHeaders']	document				
EMAIL_ATTACHMENT_LINKS	[ClassicHTML] When set, this replaces the "Attachments:" value in the HTML body of an email. For each attachment, the subfile-id will be placed where "\$\$\$" or "\$\$!" is found in the template and the result will be put in the "HREF" attribute of an "A" tag. Using \$\$\$ will put double quotes around the subfile-id. Type: string	document				
EMAIL_DATE_FORMAT	Specify the output format when processing date fields in email messages. The following values will be substituted: <table><thead><tr><th>Code</th><th>Description</th></tr></thead><tbody><tr><td>d</td><td>day</td></tr></tbody></table>	Code	Description	d	day	document
Code	Description					
d	day					

Name	Description		Applies To
	Code	Description	
	dd	day (padded)	
	ddd	day name (abbr)	
	dddd	day name	
	dddddd	short day/time	
	dddddd	long day/time	
	m	month	
	mm	month (padded)	
	mmm	month name (abbr)	
	mmmm	month name	
	yy	year (2-digit)	
	yyyy	year (4-digit)	
	h	hour	
	hh	hour (padded)	
	n	minute	
	nn	minute (padded)	
	s	second	
	ss	second (padded)	
	z	millisecond	
	zzz	millisecond (padded)	
	t	short day/time	
	tt	long day/time	
	tz	timezone	
	am/p	am or pm	

Name	Description	Applies To
	Code	Description
	a/	am or pm
	ampm	am or pm
	/	date separator
	:	time separator
	'xx'/'xx"	literal string
	Type: string Default: ddd, d mmm yyyy h:nn:ss ampm tz	
EMAIL_DATE_TIMEZONE_OFFSET	Specify a timezone offset for email date fields. Accepts LOCAL, UTC, GMT, or values in +hhmm/-hhmm format. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
EMAIL_FONT_FAMILY_OVERRIDE	Set the value of font family to change the default font used when rendering email messages. Type: string	document
EMAIL_FONT_SIZE_OVERRIDE	Changes the default font size used when rendering email messages. Type: number Default: 0	document
EMAIL_OUTPUT_HEADER	Enable or disable the injected header of email messages. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
ENCODING_DETECTION	Enable or disable automatic detection of text encoding in documents. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
ENCODING_DETECTION_BUFFER	Specify the buffer size (in bytes) used for encoding detection. Larger buffers improve accuracy but may impact performance.	document

Name	Description	Applies To
	Type: number Default: 1024	
ENCODING_ DETECTION_ CONFIDENCE	Set the confidence threshold (0-100) for automatic encoding detection. Higher values reduce false positives but may miss some encodings. Type: number Default: 0	document
ENUMERATE_ IMAGES	Enable or disable image enumeration as subfiles. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
ENUMERATE_ IMAGES_PDF_ TYPE	Set the value of image type to output when processing PDF with ENUMERATE_IMAGES on. Type: string:enum Default: AUTO Possible Values: ['JPG', 'PNG', 'PPM', 'AUTO']	document
EXCELMODE	Set the value of output mode for spreadsheet file processing. Type: string:enum Default: TEXT Possible Values: ['TEXT', 'CSV', 'TSV', 'CST']	document
EXCELMODE_ TEXT_ FALLBACK	Enable or disable string-only extraction on a document that cannot be converted to CSV. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
EXCEL_ PAGEBREAK_ BETWEEN_ SHEETS	Enable or disable page-break markers between sheets when exporting to CSV Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
EXTRACT_ EMBEDDED_ NAME	Enable or disable extraction of embedded file names when sub-enumerating attachments in MS Office files. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document

Name	Description	Applies To
FIRST_PAGE	Set the first page to process from. Type: number Default: 0	document
GIF_DECODE_ ALL_FRAMES	Enable or disable decoding of all frames of a GIF. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
GIF_ INTERLACED	Enable or disable interlaced GIF creation. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
GRAPHICS_ ANTIALIAS	Enable or disable anti-aliasing support for hi-def output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document, canvas
GRAPHICS_ AUTOHINT	Enable or disable auto-hinting on the canvas Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document, canvas
GRAPHICS_ DITHER	Enable or disable dithering for lower color output types. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document, canvas
GRAPHICS_ TEXTEMBEDDE DBITMAP	Enable or disable embedded text bitmap support in hi-def output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document, canvas
GRAPHICS_ TEXTKERNING	Enable or disable text kerning support for hi-def output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document, canvas
GRAPHICS_ TEXTLCD	Enable or disable text LCD support for hi-def output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document, canvas
	Enable or disable linear text support in hi-def output.	

Name	Description	Applies To
GRAPHICS_ TEXTLINEAR	Type: bool Default: ON Possible Values: ['ON', 'OFF']	document, canvas
GRAPHICS_ TEXTSUBPIXEL	Enable or disable sub-pixel placement of text in hi-def output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document, canvas
GRAPHIC_BMP_ DPI	Enable or disable using BMP source image DPI for full page images. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
GRAPHIC_ DESATURATE	Enable or disable converting images to grayscale. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
GRAPHIC_ DESKEW	Enable or disable deskew of slightly rotated images with scanned text. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
GRAPHIC_ DESPECKLE	Enable or disable noise reduction using a median filter. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
GRAPHIC_DPI	Set the dots per inch value for output images. Use AUTO to preserve original DPI. Type: number Default: 96	document, canvas
GRAPHIC_DPI_ MAX	Set the maximum dots per inch for auto-adjusted graphics when GRAPHIC_DPI=Auto Type: number Default: 0	document
GRAPHIC_DPI_ MIN	Set the maximum dots per inch for Auto mode graphic when GRAPHIC_DPI=Auto Type: number Default: 0	document

Name	Description	Applies To
GRAPHIC_EFFECT	Set the value of graphic effects applied to image output. Type: string:enum Possible Values: ['GRAYSCALE,', 'DESATURATE']	canvas
GRAPHIC_HAIRLINE_SUBPIXEL	Enable or disable sub-pixel rendering. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
GRAPHIC_HEIGHT	Set the height of the generated page in pixels. The width will be determined by the aspect ratio if not specified. Type: number:pixel Default: -1	document, canvas
GRAPHIC_MAX_WIDTH	Set the maximum image size for rasterized vector formats like PDF and AutoCAD. Type: number:pixel Default: 3000	document, canvas
GRAPHIC_OUTPUT_WITH_ALPHA	Enable or disable inclusion of alpha channel in hi-def output images. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
GRAPHIC_PNG_JPEG_DPI	Enable or disable using PNG and JPEG source image DPI for full page images. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
GRAPHIC_ROTATE	Set the value of page rotation angle. Type: string:enum Default: 0 Possible Values: ['0', '90', '180', '270']	document, canvas
GRAPHIC_WIDTH	Set the width of the generated page in pixels. If height is not specified, it will be calculated based on the input document's aspect ratio. Type: number:pixel Default: -1	document, canvas
HDHTML_ACCESSIBILITY	Enable or disable the inclusion of accessibility information in HTML output.	canvas

Name	Description	Applies To
	Type: bool Default: OFF Possible Values: ['ON', 'OFF']	
HDHTML_ IMAGE_ FORMAT	Set the value of image format for embedded images in HTML5 output. Type: string:enum Possible Values: ['png', 'jpg', 'jpeg', 'gif', 'auto']	canvas
HDHTML_ INCLUDE_TAB_ LEADER	Enable or disable the inclusion injected tab leaders as separate words with a data-tableader attribute. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
HDHTML_ INCLUDE_ WORD_ INDEXES	Enable or disable the inclusion word coordinates in HTML output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
HDHTML_ OUTPUT_ BOILERPLATE	Enable or disable generation of HTML boilerplate. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
HDHTML_ OUTPUT_ INJECT_ FOOTER	Set the value of HTML content to inject in the footer section of generated html output. If the value begins with an '@' symbol and points to a file on disk located in a "safe directory", the contents of that file will be injected. A safe directory is one specified by the environment variable ISYS_HDHTML_OUTPUT_INJECT_SAFE_PATHS or within the ISYS11df.ini file under the "HDHTML_OUTPUT_INJECT_SAFE_PATHS" section. For raw HTML input, ensure the content is wrapped in double quotes, with single quotes used within the HTML itself. Type: string	canvas
		canvas

Name	Description	Applies To
HDHTML_ OUTPUT_ INJECT_HEAD	Set the value of HTML content to inject in the HEAD section of generated html output. If the value begins with an '@' symbol and points to a file on disk located in a "safe directory", the contents of that file will be injected. A safe directory is one specified by the environment variable ISYS_HDHTML_OUTPUT_INJECT_SAFE_PATHS or within the ISYS11df.ini file under the "HDHTML_OUTPUT_INJECT_SAFE_PATHS" section. For raw HTML input, ensure the content is wrapped in double quotes, with single quotes used within the HTML itself. Type: string	
HDHTML_ OUTPUT_ STYLES	Enable or disable generation of style sections for HD-HTML output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
HDHTML_TITLE	Set the value of HTML title for generated HDHTML. Type: string	canvas
HDHTML_ WORD_RUNS	Enable or disable feature that merges similar styled words into a single run. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
HL7_MODE	Set the output format to use when processing HL7 files. Type: string:enum Default: JSON Possible Values: ['JSON', 'MARKDOWN']	document
HL7_SECTION_ FORMAT	Set the format to use when processing HL7 sections in Markdown output. Type: string:enum Default: TABLE Possible Values: ['LIST', 'TABLE']	document
HOCR		document

Name	Description	Applies To
	Enable or disable inclusion of HOCR attributes when producing HTML or XML files in hi-def modes. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	
HTML_CHARSET	Specify the character encoding to use when processing HTML content. Type: string	document
HTML_DECODE_ENTITIES	Enable or disable decoding of HTML entities when extracting text. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
HTML_IMAGES_SCALE_TO_FIT	Enable or disable forced downscaling of images attached to emails and messages. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
HTML_INLINE_IMAGES	Enable or disable embedding inline image handling in HTML output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
HTML_MAX_FILESIZE	Set the maximum size, in bytes, of a HTML document for error-free processing. Larger documents may result in errors when processed. Type: number Default: 20971520	document
HTML_MAX_NESTED_TABLES	Set the value of the maximum number of nested tables allowed before an error is raised. Type: number Default: 150	document
HTML_RASTER_TEXT	Enable or disable the rasterization of text from HTML files into flat images when processing. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document, canvas

Name	Description	Applies To
HTML_SCRIPT	[ClassicHTML] Used in conjunction with EMAIL_ATTACHMENT_LINKS to inject text into the output HTML to provide supporting code for the attachment link handling. Type: string	document
HTML_SHOW_DISPLAY_NONE	Enable or disable forced display of div elements with 'display:none' style in emails and messages. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
HTML_TEXT_OVERLAY	Enable or disable text overlays when creating hi-def HTML output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document, canvas
HTML_TITLE_AS_META	Enable or disable treating the HTML title as metadata during extraction. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
HTML_TITLE_SUPPRESS	Enable or disable suppression of the HTML title from extracted output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
HTTP_EXTERNAL_IMAGES	Enable or disable image downloading for HTML documents in hi-def mode. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
HTTP_PROXY	Set the value of HTTP proxy server to use for downloading images in HTML documents when external images are allowed. Type: string	document
	Set the value of bypassed proxy servers that should not use the proxy setting.	document

Name	Description	Applies To
HTTP_PROXY_BYPASS	Type: string	
ICS_MODE	Set the output type when processing ICS calendar items. Type: string:enum Possible Values: ["', 'JSON']	document
IMAGEURL	Set the value of image filename prefix. Type: string	canvas
IMAGE_PROCESSOR	Set the value of image processor to control how embedded images are processed. Type: string:enum Default: builtin Possible Values: ['builtin', 'gdi', 'gdiplus', 'gdi+']	document
IMNET_CHAR_SPACING	Set the inter-character spacing for IMNET overlays to a specific point value. Type: number:point Default: 0	document
IMNET_LINE_HEIGHT	Set the height of each line for IMNET overlays in points. Type: number:point Default: 0	document
IMNET_OFFSET_X	Set the horizontal offset for IMNET overlays in points. Type: number:point Default: 0	document
IMNET_OFFSET_Y	Set the vertical offset for IMNET overlays in points. Type: number:point Default: 0	document
IMNET_OVERLAY_DIR	Set the directory where IMNET overlays are searched for. Type: string	document
IMNET_OVERLAY_EXTS	Set the value of IMNET overlay extensions to search for, delimited by comma. Type: string	document
IMNET_OVERLAY_FILE	Set the value of overlay image. Type: string	document
ISOBMFF	Set the output type when processing ISOBMFF based file types.	document

Name	Description	Applies To
	Type: string:enum Possible Values: ["", 'JSON']	
ISYS_SOURCE	Set the original file path and filename of the source document for enhanced file type identification. The file does not need to be accessible from this location. Type: string	document
JPEG_ENCODER	Set the value of image encoder to use for JPEG files, applicable on Windows with WIC. Type: string:enum Default: NATIVE Possible Values: ['NATIVE', 'WIC']	canvas
JPEG_QUALITY	Set the quality of generated JPEG images to an integer from 1 to 100. A higher value produces a more detailed image. Type: number Default: 100	canvas
JPEG_REORIENT_PAGES	Enable or disable reorienting JPEG images to display top-down, left-to-right. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
JSON_CLEAN_CONTENT	Set the value of unordered list of cleaning functions to apply to each element's text when generating simplified JSON output. Type: string:enum Possible Values: ["", 'clean_non_ascii_chars', 'normalize_quotes', 'normalize_dashes']	canvas
JSON_FORMAT_OUTPUT	Enable or disable adding new lines and indentation to JSON output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas

Name	Description	Applies To
JSON_IMAGE_FORMAT	Set the value of image format for embedded images in JSON output. Type: string:enum Default: PNG Possible Values: ['PNG']	canvas
JSON_IMAGE_QUALITY	Set the image compression quality for embedded images in JSON output. Type: number Default: 90	canvas
JSON_INCLUDE_BOOKMARKS	Controls if bookmarks should be included in JSON output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
JSON_INCLUDE_BOUNDS	Enable or disable inclusion of element bounds information in JSON output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
JSON_INCLUDE_DOC_METADATA_PER_ELEMENT	Enable or disable the inclusion of document-level metadata for each element of simplified JSON output. Enables file name in MDAST JSON output schema with JSON_INCLUDE_BOUNDS set to on. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
JSON_INCLUDE_ELEMENT_ID	Enable or disable the inclusion of an element id when generating simplified JSON output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
JSON_INCLUDE_FORMS	Enable or disable inclusion of page form information in JSON output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
	Enable or disable including image elements in JSON output.	canvas

Name	Description	Applies To
JSON_ INCLUDE_ IMAGES	Type: bool Default: ON Possible Values: ['ON', 'OFF']	
JSON_ INCLUDE_ IMAGE_DATA	Enable or disable including image data in JSON output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
JSON_ INCLUDE_ METADATA	Enable or disable inclusion of document metadata in JSON output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
JSON_ INCLUDE_ STYLES	Enable or disable inclusion of element style information in JSON output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
JSON_ INCLUDE_ WHITESPACE	Enable or disable inclusion of whitespace words in JSON output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
JSON_ INCLUDE_ WORDS	Enable or disable inclusion of word-level information in JSON output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
JSON_OUTPUT_ SCHEMA	Set the output schema type to use when generating JSON. Type: string:enum Default: FULL Possible Values: ['FULL', 'PIPELINE', 'UNSTRUCTURED', 'MDAST']	canvas

Name	Description	Applies To
JSON_SKIP_EMPTY_PARAGRAPHS	Enable or disable inclusion of empty paragraphs in JSON output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
LAYOUT_DETECTION_BLOCK_HORIZONTAL_THRESHOLD	Set the horizontal distance threshold for detecting separate text blocks. Type: number Default: 20	document
LAYOUT_DETECTION_BLOCK_MERGE_HORIZONTAL_THRESHOLD	Set the maximum horizontal distance between adjacent blocks to merge them into a single block. Type: number Default: 45	document
LAYOUT_DETECTION_BLOCK_MERGE_VERTICAL_THRESHOLD	Set the maximum vertical distance between adjacent blocks to merge them into a single block. Type: number Default: 45	document
LAYOUT_DETECTION_BLOCK_VERTICAL_THRESHOLD	Set the vertical distance threshold for detecting separate text blocks. Type: number Default: 15	document
LAYOUT_DETECTION_COLUMN_THRESHOLD	Set the threshold for detecting column boundaries in a document layout. Type: number Default: 50	document
LAYOUT_DETECTION_LINE_GAP	Set the gap threshold for detecting separate lines. Type: number Default: 15	document

Name	Description	Applies To
LAYOUT_DETECTION_LINE_HORIZONTAL_THRESHOLD	Set the maximum horizontal distance between words in a line before considering them separate lines. Type: number Default: 15	document
LAYOUT_DETECTION_LINE_SPACING	Set the expected spacing between lines to help identify paragraph breaks. Type: number Default: 10	document
LAYOUT_DETECTION_LINE_VERTICAL_THRESHOLD	Set the vertical distance threshold for detecting separate lines of text. Type: number Default: 15	document
LAYOUT_DETECTION_PARA_SPACING	Set the minimum vertical spacing required to detect paragraph breaks. Type: number Default: 20	document
LAYOUT_LOCALE	Set the value of locale used for document processing. Type: string	document
LIMITS_PAGE_COUNT	Set the maximum number of pages to process. Any excess pages will result in an error. Type: number Default: 1000	document
LIMIT_PAGES	Set the maximum number of pages to process, ideal for generating thumbnails. Type: number Default: -1	document
LOCALTIME_CONVERSION	Enable or disable the conversion of local times to UTC when no timezone is provided. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
MARKDOWN_CLEAN_CONTENT	Set the value of unordered list of cleaning functions to apply to the text of each block, paragraph, or element when generating Markdown.	canvas

Name	Description	Applies To
	Type: string:enum Possible Values: [' ', 'clean_non_ascii_chars', 'normalize_quotes', 'normalize_dashes']	
MARKDOWN_COMPLEX_TABLE_STYLE	Set the style for rendering complex tables in Markdown format. Type: string:enum Default: HTML Possible Values: ['PIPE', 'GRID', 'HTML', 'PIPE_WITH_HTML', 'PARAGRAPHS']	canvas
MARKDOWN_FLAVOR	Set the value of Markdown flavor. Type: string:enum Default: gfm Possible Values: ['GFM', 'GPT']	canvas
MARKDOWN_HEADERS_STYLE	Set the style of Markdown headers. Type: string:enum Default: ATX Possible Values: ['ATX', 'SETEXT']	canvas
MARKDOWN_HR_BETWEEN_PAGES	Enable or disable inclusion of a horizontal rule between pages in Markdown output Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
MARKDOWN_INCLUDE_BOOKMARKS	Enable or disable inclusion of bookmarks in Markdown output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
MARKDOWN_INCLUDE_FIELDS	Enable or disable inclusion of text marked as fields in generated Markdown. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
MARKDOWN_INCLUDE_FOOTERS	Enable or disable inclusion of text marked as a page footer when generating Markdown. Type: string:enum Default: OFF Possible Values: ['ON', 'OFF', 'FIRST']	canvas

Name	Description	Applies To
MARKDOWN_INCLUDE_FORMATTING	Enable or disable inclusion of text formatting in Markdown output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
MARKDOWN_INCLUDE_FORMS	Enable or disable inclusion of expanded form elements in Markdown output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
MARKDOWN_INCLUDE_HEADERS	Enable or disable inclusion of text marked as a page headers when generating Markdown. Type: string:enum Default: OFF Possible Values: ['ON', 'OFF', 'FIRST']	canvas
MARKDOWN_INCLUDE_IMAGES	Enable or disable inclusion of image elements when generating Markdown. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
MARKDOWN_INCLUDE_LINKS	Enable or disable inclusion of links in Markdown output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
MARKDOWN_INCLUDE_LOCATIONS	Enable or disable inclusion of text location markers in Markdown output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
MARKDOWN_INCLUDE_METADATA	Enable or disable inclusion of document metadata in Markdown output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
MARKDOWN_LF_AFTER_BLOCK	Enable or disable new lines after each HTML block element when generating Markdown.	canvas

Name	Description	Applies To
	Type: bool Default: OFF Possible Values: ['ON', 'OFF']	
MARKDOWN_ METADATA_ FORMAT	Set the value of Markdown format for metadata inclusion. Type: string:enum Default: COMMENTS Possible Values: ['YAML', 'TOML', 'JSON', 'COMMENTS']	canvas
MARKDOWN_ PREFERRED_ LINE_LENGTH	Set the preferred line length for wrapping in Markdown. Type: number Default: 80	canvas
MARKDOWN_ SIMPLE_TABLE_ STYLE	Set the rendering style for simple tables in Markdown. Type: string:enum Default: PIPE Possible Values: ['PIPE', 'GRID', 'HTML', 'PIPE_WITH_HTML', 'PARAGRAPHS']	canvas
MARKDOWN_ TABLE_ DETECT_ TITLES	Enable or disable interpretation of full colspan first-rows as table titles. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
MARKDOWN_ TABLE_ PADDING	Enable or disable inclusion of padding on pipe style tables in Markdown output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
MARKUP_ OPTIONS	Set the value of track-changes markup types. Must be combined with SHOWHIDDEN. [INS,DEL] Type: string Default: DE	document
METADATA_ NAME_ HYPHENATE	Enable or disable the hyphenation of metadata names for HTML output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
	Set the string used to separate a metadata name from its value in text-mode output.	document

Name	Description	Applies To
METADATA_ VALUE_ SEPARATOR	Type: string Default: :	
METADATA_ WHEN_ ENCRYPTED	Enable or disable metadata extraction from encrypted files. When disabled, the document will fail to open with error code IGR_E_PASSWORD. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
MHT_HEADERS	Enable or disable extraction of MHTML email headers in the output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
MSACCESS_ SHOW_ SYSTEMTABLES	Enable or disable the inclusion of Microsoft Access system tables in extracted output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
MSPROJECT_ FONT	Set the value of font to use when converting MS Project. Type: string Default: Arial	document
MSPROJECT_ FONTSIZE	Set the font size used during MS Project conversion. Type: number:point Default: 8	document
MSPROJECT_ PAGE_HEIGHT	Set the value of page height in points. Type: number:point Default: 612	document
MSPROJECT_ PAGE_WIDTH	Set the value of page width in points. Type: number:point Default: 792	document
MSPROJECT_ ZOOM	Set the value of zoom to use when converting MS Project. Type: string Default: auto	document

Name	Description	Applies To
MSWORD_DEFAULT_UTF8	Enable or disable the use of UTF-8 encoding as the default for Microsoft Word documents. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
OCR	Enable or disable OCR processing of images. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
OCR_CACHE	Set the location of OCR cache storage. Type: string	document
OCR_ENGINE	Set the value of OCR engine to use. Type: string Default: TESSERACT	document
OCR_INLINE_IMAGES	Enable or disable OCR processing of images embedded within the contents of a document. Requires OCR=ON. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
OCR_LANGUAGE	Set the value of language for Optical Character Recognition (OCR) processing. Type: string Default: Eng	document
OCR_LAYOUT_DETECTION	Enable or disable layout detection when performing OCR extraction. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
OCR_MIN_WIDTH	Set the value of image minimum width for OCR processing. Type: number Default: 1000	document
OCR_REORIENT_PAGES	Enable or disable re-orientation of pages based on orientation detection for OCR. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document

Name	Description	Applies To
ODF_ USERMETA	Enable or disable adding user-defined fields as metadata. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
ODSMODE	Set the output format when processing Open Document Spreadsheet (ODS) files. Type: string:enum Possible Values: ['', 'JSON']	document
OFFICE_ COMMENT_ PAGE	Set the value of the page where comments are extracted to when using IGR_Get_Page_Annotations. Type: string:enum Default: first Possible Values: ['first', 'last']	document
OLM_DATE_ FIELD	Set the value of ordered list of XML properties containing document date. Type: string:enum Possible Values: ['', 'OPFMessageCopyModDate', 'OPFMessageCopyReceivedTime', 'OPFMessageCopySentTime']	document
PAGENUMBERS	Enable or disable the extraction of page numbers when processing presentation type documents in text-mode. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PAGE_ NUMBERS	Enable or disable this inclusion of injected page numbers in hi-def output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
PAGE_TEXT_ SYMBOLS_TO_ UNICODE	Enable or disable mapping of symbol fonts (such as Wingdings) to a similar Unicode font. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PBM_MODE	Set the format of saved PBM files to ASCII or BINARY.	canvas

Name	Description	Applies To
	Type: string:enum Default: ASCII Possible Values: ['ASCII', 'BINARY']	
PDF_ANNOTATIONS	Set the value of PDF annotations to draw, excluding specific types by prefixing with an exclamation point (!) Type: string	document
PDF_BOOKMARKS	Enable or disable whether to include bookmarks in the output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document, canvas
PDF_COMPRESSION	Set the value of compression type for generated PDF output. Type: string:enum Default: AUTO Possible Values: ['AUTO']	canvas
PDF_DEFAULTDPI	Set the DPI for bitmap scaling. Type: number Default: 96	document
PDF_DEHYPHENATE	Enable or disable removing hyphens from word endings to express contractions as single words. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PDF_ENCRYPT	Enable or disable to apply encryption to the generated PDF. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
PDF_ENCRYPT_PERMISSIONS	Set the value of PDF permissions (see PDF Reference 1.7, Table 3.20) Type: string	canvas
PDF_EXCLUSION_MARGINS	Specify exclusion margins to ignore content within certain areas of the PDF page. Can be expressed in %, pt, px, in, cm, mm. Type: string	document

Name	Description	Applies To
PDF_EXCLUSION_MARGIN_BOTTOM	Specify the bottom margin for content exclusion in PDF files. Can be expressed in %, pt, px, in, cm, mm. Type: string	document
PDF_EXCLUSION_MARGIN_LEFT	Specify the left margin for content exclusion in PDF files. Can be expressed in %, pt, px, in, cm, mm. Type: string	document
PDF_EXCLUSION_MARGIN_RIGHT	Specify the right margin for content exclusion in PDF files. Can be expressed in %, pt, px, in, cm, mm. Type: string	document
PDF_EXCLUSION_MARGIN_TOP	Specify the top margin for content exclusion in PDF files. Can be expressed in %, pt, px, in, cm, mm. Type: string	document
PDF_IMAGE_QUALITY	Set the image quality for PDF output, with higher values indicating higher quality. Type: number Default: 90	document
PDF_LAYOUT_DETECTION	Enable or disable layout detection for PDF files. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PDF_LAYOUT_DETECTION_DIAGNOSTICS_MARKUP	Enable or disable the drawing of PDF layout detection areas on output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PDF_LINEARIZE	Enable or disable PDF output linearization for Fast Web View. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
PDF_LIST_DETECTION	Enable or disable list detection in PDF files. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document

Name	Description	Applies To
PDF_ MARKDOWN_ UNWRAP_TEXT	Enable or disable unwrapping of text when processing PDF to Markdown output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PDF_MAX_ CHARS_PER_ PAGE	Set maximum number of characters processed for each page of PDF input. Type: number Default: 50000	document
PDF_MAX_ IMAGE_WIDTH	Set the maximum width of output images to this pixel value. Type: number:pixel Default: 3200	document
PDF_MAX_ PAGE_HEIGHT	Set the maximum allowed page height for PDF input, in points. Type: number Default: 14400	document
PDF_MAX_ PAGE_WIDTH	Set the maximum allowed page width for PDF input, in points. Type: number Default: 14400	document
PDF_META	Enable or disable whether metadata is included with PDF documents. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
PDF_MIN_ WORD_BREAK_ SPACE	Specify the minimum space between characters that constitutes a word-break. Type: number Default: 0.1	document
PDF_ OWNERPASSW ORD	Set the value of owner password used to decrypt PDF documents. Type: string	document
PDF_PAGE_ NUMBERS	Enable or disable including page numbers on PDF text-only output. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document

Name	Description	Applies To
PDF_ PHYSLAYOUT	Enable or disable whether to maintain the document's physical layout in the output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PDF_ PRESERVE_ ORIGINAL	Enable or disable preserving original page data when converting PDF to PDF. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
PDF_ PRESERVE_ PAGE_SIZE	Enable or disable whether to maintain the page size when converting from an image to PDF. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PDF_RASTER_ TEXT	Enable or disable rendering text from PDF files as raster images. This controls the handling of text in PDFs. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
PDF_RENDER_ MODE	Set the rendering mode for PDF files. PRINT mode is optimized for high-quality output, while SCREEN mode is optimized for on-screen viewing. Type: string:enum Default: SCREEN Possible Values: ['PRINT', 'SCREEN']	document
PDF_SET_ OWNER_ PASSWORD	Set the value of master password controlling document edits. Type: string	canvas
PDF_SET_ USER_ PASSWORD	Set the value of default view password. Type: string	canvas
PDF_TABLE_ DETECTION	Enable or disable table detection. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document

Name	Description	Applies To
PDF_TABLE_EDGE_MIN_LENGTH	Set the value of minimum edge length considered when detecting table edges in PDF files. Type: number Default: 11	document
PDF_TABLE_HORZ_STRATEGY	Set the value of horizontal table detection strategy for detecting tables in PDF files. Type: string:enum Default: LINES Possible Values: ['LINES']	document
PDF_TABLE_INTERSECTION_TOLERANCE	Set the tolerance value to determine edge combination into cells. This value helps control the precision of cell formation when detecting tables in PDF files. Type: number Default: 3	document
PDF_TABLE_INTERSECTION_X_TOLERANCE	Set the value of edge intersection tolerance along the x-axis. This determines how close edges must be to be considered intersecting when detecting tables in PDF files. Type: number Default: 3	document
PDF_TABLE_INTERSECTION_Y_TOLERANCE	Set the value of Y-axis tolerance for edge intersection when detecting tables in PDF files. Type: number Default: 3	document
PDF_TABLE_JOIN_TOLERANCE	Set the value of tolerance for joining line segments when detecting tables in PDF files. Type: number Default: 3	document
PDF_TABLE_JOIN_X_TOLERANCE	Set the value of x-axis tolerance for joining line segments, controlling how close line ends can be before being considered a single segment when detecting tables in PDF files. Type: number Default: 3	document

Name	Description	Applies To
PDF_TABLE_JOIN_Y_TOLERANCE	Set the value of y-axis tolerance for joining line segments, controlling how close line ends can be before being considered a single segment when detecting tables in PDF files. Type: number Default: 3	document
PDF_TABLE_MIN_CELL_HEIGHT	Set the minimum cell height for table detection when detecting tables in PDF files. Type: number Default: 8	document
PDF_TABLE_MIN_CELL_WIDTH	Set the minimum cell width for table detection when detecting tables in PDF files. Type: number Default: 8	document
PDF_TABLE_MIN_COLS	Set the minimum number of columns required for table detection in PDF files. Type: number Default: 2	document
PDF_TABLE_MIN_ROWS	Set the minimum number of rows required for table detection in PDF files. Type: number Default: 1	document
PDF_TABLE_SNAP_TOLERANCE	Set the value of line tolerance to control parallel line snapping for detecting tables in PDF files. Type: number Default: 3	document
PDF_TABLE_SNAP_X_TOLERANCE	Set the value of x-axis tolerance for snapping parallel lines when detecting tables in PDF files. Type: number Default: 3	document
PDF_TABLE_SNAP_Y_TOLERANCE	Set the value of y-axis tolerance for snapping parallel lines when detecting tables in PDF files. Type: number Default: 3	document
PDF_TABLE_VERT_STRATEGY	Set the value of horizontal vertical detection strategy for detecting tables in PDF files. Type: string:enum Default: LINES Possible Values: ['LINES']	document

Name	Description	Applies To
PDF_TAGGED	Enable or disable the creation of PDF with accessibility information. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
PDF_TRANSPARENT_BACKGROUND	Enable or disable transparent background rendering for the PDF. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PDF_USERPASSWORD	Set the value of password for decrypting PDF documents. Type: string	document
PDF_VERSION	Set the value of the compliance level of generated PDF files. Note: Image transparency is lost when generating PDF/A-1. Type: string:enum Default: 1.7 Possible Values: ['1.7', 'A1', 'A1A', 'A1B', 'A2', 'A2U', 'PDF1.7', 'PDF/A', 'PDF/A1', 'PDF/A1A', 'PDF/A1B', 'PDF/A2', 'PDF/A2U']	canvas
PDF_WORD_ORDER	Set the value of word order to "Reading" to extract text as if read, or "Document" to match original document structure. Type: string:enum Default: READING Possible Values: ['READING', 'DOCUMENT']	document
PDF_XFA_ENABLED	Enable or disable XFA rendering Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PDF_XFA_ERROR	Enable or disable detection of dynamic XFA content. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
PDF_XMPMETA		document

Name	Description	Applies To
	Enable or disable whether to include XMP metadata in the output, requiring an opening document with a metadata flag enabled. Type: bool Default: ON Possible Values: ['ON', 'OFF']	
PGM_MODE	Set the mode for PGM file saving to ASCII or binary. Type: string:enum Default: ASCII Possible Values: ['ASCII', 'BINARY']	canvas
PLIST_FORMAT	Set the output format for extracted PLIST data (JSON or Plain Text). Type: string:enum Possible Values: ['', 'JSON']	document
PNG_BITSPERPIXEL	Set the value of bits-per-pixel when generating PNG. Type: string:enum Default: SOURCE Possible Values: ['SOURCE', 'AUTO', '1', '8', '16', '24', '32']	canvas
PNG_ENCODER	Set the value of image encoder for JPEG files on Windows using WIC. Type: string:enum Default: NATIVE Possible Values: ['NATIVE', 'WIC']	canvas
PPM_MODE	Set the mode for PPM file saving to ASCII or binary. Type: string:enum Default: ASCII Possible Values: ['ASCII', 'BINARY']	canvas
PROCESS_TIMEOUT	Set the time limit for processing documents, in milliseconds. Type: number Default: 0	document
PSTMESSAGE TYPE	Set whether to extract embedded messages from PST/OST archives in MSG format or EML format. Type: string:enum Default: MSG Possible Values: ['MSG', 'EML']	document
		canvas

Name	Description	Applies To
RASTER_DRAWINGS	Enable or disable the conversion of embedded drawings to bitmap in vector formats. Type: bool Default: ON Possible Values: ['ON', 'OFF']	
REDACT	Set the value of redacting words on nominated pages. Type: string	document
RESAMPLE_IMAGES	Enable or disable image resampling for vector-based output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
SCALE	Set the scale that the page is rendered. Type: number Default: 1	document
SHOWHIDDEN	Enable or disable showing normally invisible page elements. Type: unknown Default: "off"	document
SHOWHIDDEN_EXCEL_REFS	Enable or disable the inclusion of hidden Excel references. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
SLIDE_COMMENTS	Enable or disable showing slide comments in presentations. Type: string:enum Possible Values: ['OFF', 'ON', 'APPEND']	document
SLIDE_NOTES	Enable or disable showing note pages in presentations. Type: string:enum Possible Values: ['OFF', 'ON', 'APPEND']	document
SLIDE_NOTES_FORMATTING	Enable or disable slide note formatting. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document

Name	Description	Applies To
SPREADSHEET _COLLAPSE_ ROWS	Enable or disable collapsing of spreadsheet rows with duplicates. Set to 'OFF', 'NO', or less than 1 to disable. Type: number Default: "100"	document
SPREADSHEET _PAGEDIRECTIO N	Set the value of page direction to determine how pages are ordered when a sheet does not fit on one page. Type: string:enum Default: OverThenDown Possible Values: ['OverThenDown', 'DownThenOver']	document
SPREADSHEET _SHEETS	Set to the sheets to render. Set to ALL or a comma-separated list of numbers or ranges. For example, 1,2,5..7 Type: string Default: ALL	document
SVG_ COALESCE_ RUNS	Enable or disable combining similar text styles into a single run when rendering SVG elements. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
SVG_FLATTEN_ TEXT	Enable or disable rendering of outputs text as paths instead of text elements. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
SVG_INCLUDE_ CHAR_OFFSET	Enable or disable inclusion of character offsets. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
SVG_INLINE_ IMAGES	Enable or disable embedded images as inline elements in SVG. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
TEXT_ DEFAULT_FONT	Set the value of default font to use for non-HD image rendering. Type: string Default: Courier New	document
	Set the default font size for non-HD images rendering. Type: number:point Default: 10	document

Name	Description	Applies To
TEXT_ DEFAULT_ FONT_SIZE		
TEXT_ EXTENSION	Specify the default file extension for extracted text output. Type: string Default: .txt	document
TEXT_ FALLBACK	Enable or disable text fallback rendering for documents that do not directly support hi-def processing. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
TEXT_ FALLBACK_ MODE	Set the value of text fallback mode to specify how to handle metadata for non-HD formats. Type: string:enum Possible Values: ['', 'BODYONLY']	document
TEXT_INLINE_ LINKS	Enable or disable the writing of links in text mode inline or separately. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
TEXT_MARGIN_ BOTTOM	Set the bottom margin to use for formats without page or margin info, measured in points. Type: number:point Default: 36	document
TEXT_MARGIN_ LEFT	Set the value of left margin to use for formats without page or margin information, in points. Type: number:point Default: 36	document
TEXT_MARGIN_ RIGHT	Set the value of right margin to use for formats without page or margin information, in points. Type: number:point Default: 36	document
TEXT_MARGIN_ TOP	Set the value of top margin to use for formats without page or margin information, in points. Type: number:point Default: 36	document
TEXT_ METADATA	Enable or disable the extraction of metadata as plain text when processing documents.	document

Name	Description	Applies To
	Type: bool Default: OFF Possible Values: ['ON', 'OFF']	
TEXT_PAGE_CHARS_PER_LINE	Set the character limit for text-only lines. Type: number Default: 0	document
TEXT_PAGE_FOOTER	Enable or disable inclusion of the footer when rendering text pages. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
TEXT_PAGE_HEIGHT	Set the page height to a specified number of points. This value determines the page width for formats without page information. Type: number:point Default: 792	document
TEXT_PAGE_LINES_PER_PAGE	Set the line count per page for formats without layout details. Type: number Default: 0	document
TEXT_PAGE_ORIENTATION	Set the page orientation to portrait or landscape. Type: string:enum Default: PORTRAIT Possible Values: ['PORTRAIT', 'LANDSCAPE']	document
TEXT_PAGE_SIZE	Set the value of output page size using ISO standard or metric page sizes. Cannot be used with TEXT_PAGE_WIDTH or TEXT_PAGE_HEIGHT. Type: string:enum Possible Values: [' ', 'LETTER', 'LEGAL', 'TABLOID', 'LEDGER', 'A0', 'A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10', 'B0', 'B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B9', 'B10', 'C0', 'C1', 'C2', 'C3', 'C4', 'C5', 'C6', 'C7', 'C8', 'C9', 'C10']	document
TEXT_PAGE_WIDTH	Set the page width to a specified number of points. This value determines the page width for formats without page information.	document

Name	Description	Applies To
	Type: number:point Default: 612	
TEXT_PAGE_WORD_WRAP	Enable or disable word wrap for formats without layout information. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
TIFF_BITSPERPIXEL	Set the value of bit depth to specify the number of bits per pixel in the TIFF output. Type: string:enum Possible Values: ['0', '1', '2', '4', '8', '24']	canvas
TIFF_BYTEORDER	Set the byte order of the generated TIFF image. Type: string:enum Default: NATIVE Possible Values: ['NATIVE', 'LE', 'BE']	canvas
TIFF_COMPRESSION	Set the value of compression type for the generated TIFF image. Type: string:enum Default: LZW Possible Values: ['NONE', 'PACKED', 'JPEG', 'LZW', 'GROUP3', 'GROUP4', 'GROUP6']	canvas
TIFF_IGNORE_DECODE_ERR	Enable or disable feature to control error handling for TIFF image decoding. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
TIFF_IGNORE_MEMORY_LIMITS	Enable or disable enforced memory limits when processing large TIFF images. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
TIFF_PHOTOMETRIC_MINISBLACK	Enable or disable feature for controlling black and white TIFF output representation. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas

Name	Description	Applies To
TIFF_REORIENT_PAGES	Enable or disable the reorientation of TIFF images for top-down, left-to-right display. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
TIFF_SUSPECT_DPI_RATIO_ERROR	Enable or disable suspect DPI checks when processing TIFF files. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
USE_MAPPED_FONTS	Enable or disable use of actual fonts instead of stored fonts. This affects how fonts are displayed in generated output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
VISIO_FOOTER	Enable or disable injected footer on Visio documents Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
VISIO_RASTER_TEXT	Enable or disable rendering Visio text in HTML mode as raster images with transparent text overlays. Type: bool Default: ON Possible Values: ['ON', 'OFF']	document
WATERMARK	Set the watermark text to apply to pages when rendering to hi-def image formats. Type: string	canvas
WATERMARK_COLOR	Set the value of watermark text color. Type: string Default: #000000	canvas
WATERMARK_DIRECTION	Set the direction of watermark text when rendering to HD image formats. Type: string:enum Default: LBRT Possible Values: ['LBRT', 'LTRB', 'LR']	canvas
	Set the value of font used for watermark rendering.	

Name	Description	Applies To
WATERMARK_FONT	Type: string Default: Aria	canvas
WATERMARK_OPACITY	Set the transparency level of watermark text as a value between 0 and 1. Type: number Default: 0.1	canvas
XML_FORMATTED	Enable or disable formatting XML input to show only text content. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
XML_FORMATTED_WITH_ATTRIBUTES	Enable or disable the inclusion of attribute names and values in formatted XML. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
XML_FORMATTED_WITH_NODENAMES	Enable or disable including node names in formatted XML. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	document
XML_HEADINGS_ONLY	Enable or disable feature extracting headings from structured XML. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas
XML_IMAGES	Enable or disable inclusion of image data in structured XML outputs. Images will be output inline, base-64 encoded. Type: bool Default: ON Possible Values: ['ON', 'OFF']	canvas
XML_INCLUDE_FORMS	Enable or disable including form elements in structured XML output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas

Name	Description	Applies To
XML_ SUPPRESS_ EMPTY_PARA	Enable or disable suppression of empty paragraphs in XML output. Type: bool Default: OFF Possible Values: ['ON', 'OFF']	canvas

4.7 Environment Variables

The environment variables listed below should be set before initializing Document Filters.

Name	Description
ISYS_CURRENT_DATETIME	Indicates the time that should be used when processing documents that require the current time. Defaults to the current time; set with the format of <code>YYYYMMDD_HHMMSS</code> .
ISYS_DEFAULT_CODEPAGE	Indicates the default codepage to use when the codepage cannot be determined from the file. Defaults to system codepage.
ISYS_DEFAULT_LCID	Indicates the default locale ID to use when processing MS Excel files. Defaults to the current system locale; set with the MS locale ID. Example: 3081=English Australia.
ISYS_FONTS	Indicates the location where the fonts.ini file is located. Defaults to application/fonts.
ISYS_FONTS_CACHE	Indicates a file where font information can be cached improving load performance. Defaults to off; set to a fully qualified name to enable.
ISYS_FONTS_CACHE_AGE	Indicates the maximum age of the font cache before it should be regenerated. Defaults to 24 hours; set to the number of minutes to adjust.
ISYS_FONTS_DEFAULTS	Indicates whether the system fonts should be loaded. Defaults to true; set to 0 to disable.
ISYS_FONTS_DIAG	When enabled, prints diagnostic information to the console of font locations and names. Defaults to off; set to 1 to enable.
ISYS_FONTS_NOEMBEDDED	Indicates whether fonts embedded in a document that do not have a system font available should be processed. Defaults to on; set to 1 to disable.

Name	Description
ISYS_HDHTML_OUTPUT_INJECT_SAFE_PATHS	Indicates the directories from which files can be read using the <code>HDHTML_OUTPUT_INJECT_HEAD</code> and <code>HDHTML_OUTPUT_INJECT_FOOTER</code> options. Use a semicolon (;) to separate path names on Windows systems, and a colon (:) to separate path names on non-Windows systems. Defaults to an empty string.
ISYS_MAX_DOCHANDLES	Indicates the maximum number of documents that can be loaded at a time. Defaults to 64.
ISYS_TMP	Indicates the location where temp files can be generated. Defaults to the system temp directory.
ISYS_XML_MEMORY_LIMIT	Indicates the maximum amount of memory that an XML file can take before erroring. Defaults to 2GB; set to the number of MB.

4.8 Third-Party Libraries

4.8.1 7-zip 23.01

License: LZMA-SDK-9.22

LZMA SDK is written and placed in the public domain by Igor Pavlov.

Some code in LZMA SDK is based on public domain code from another developers: 1) PPMd var.H (2001): Dmitry Shkarin 2) SHA-256: Wei Dai (Crypto++ library)

Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original LZMA SDK code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

LZMA SDK code is compatible with open source licenses, for example, you can include it to GNU GPL or GNU LGPL code.

4.8.2 bcl 1.2.11

```
* Copyright (c) 2003-2006 Marcus Geelnard
*
* This software is provided 'as-is', without any express or implied
* warranty. In no event will the authors be held liable for any damages
* arising from the use of this software.
*
* Permission is granted to anyone to use this software for any purpose,
* including commercial applications, and to alter it and redistribute it
* freely, subject to the following restrictions:
*
* 1. The origin of this software must not be misrepresented; you must not
*    claim that you wrote the original software. If you use this software
*    in a product, an acknowledgment in the product documentation would
*    be appreciated but is not required.
*
* 2. Altered source versions must be plainly marked as such, and must not
*    be misrepresented as being the original software.
*
* 3. This notice may not be removed or altered from any source
*    distribution.
*
* Marcus Geelnard
* marcus.geelnard at home.se
```

4.8.3 bzip2 1.0.8

License: bzip2-1.0.8

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2019 Julian R Seward. All

rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@acm.org
bzip2/libbzip2 version 1.0.8 of 13 July 2019

4.8.4 chrome 103.0.5060.123

License: Apache-2.0

```
// Copyright 2014 PDFium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
```

```
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

4.8.5 cld2

License: Apache-2.0

Apache License

Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

4.8.6 droid-fonts 1

Droid Fonts Overview

<http://www.droidfonts.com/>

The following is a list of Droid fonts found in the Android SDK:

Droid Sans (regular, bold)
Droid Sans Mono (regular, bold)
Droid Serif (regular, italic, bold, bold italic)

The new Droid font collection provides an extensive set of styles, weights and language support. Each of the Droid fonts was custom designed by Ascender and optimized for on-screen legibility with the Android platform.

Each of the Droid fonts contains an extensive character set with coverage including Western Europe, Eastern/Central Europe, Baltic, Cyrillic, Greek and Turkish.

Support for CJK is provided by the Droid Sans "fallback" font. This font contains over 43,000 glyphs and includes support for Simplified Chinese (GB2312), Traditional Chinese (Big 5), Japanese (JIS 0208) and Korean (KSC 5601).

This font uses the Simplified Chinese ideographs for shared Unicode code points.

Font Licenses

The Droid fonts included with the Android SDK are distributed under the Apache License, Version 2.0.

<http://www.apache.org/licenses/LICENSE-2.0>

4.8.7 flashsdk 1.0

License: LicenseRef-flashsdk

MACROMEDIA, INC.

Macromedia Flash Search Engine Software Development Kit (SDK) License Agreement

Please read this document carefully before proceeding. You must agree to this Macromedia Flash Search Engine Software Development Kit (SDK) License Agreement (the "License") in order to use the Macromedia Flash Search Engine Software Development Kit (the "SDK").

1. Definitions

- a. "SDK" means the source code and documentation provided to you pursuant to this License to assist you in the indexing and processing Macromedia Flash Files (SWF).
- b. "Products" means your software product(s) and/or service(s) in which you search Flash Files (SWF) through the use of the SDK.
- c. "Flash File Format (SWF)" or "SWF" means the file format designated by .SWF

2. Licenses

Pursuant to the terms and conditions of this License, you are granted a nonexclusive license to use, modify and create derivative works ("Derivative Works") of the sample code included in the SDK for the sole purpose of developing Products that extract text and link information from a SWF file, and to distribute compiled version of such Derivative Works in object code format solely as incorporated in the object code format of such Products (the "Bundled Products").

3. Restrictions

By using the licenses above, you agree to the following restrictions:

- a. You will not make or distribute copies of the SDK, or electronically transfer the SDK outside your company.
- b. You will not modify, sell, rent, transfer, resell for profit, distribute, or create derivative works based upon the SDK or any part thereof other than as set forth in Section 2.
- c. You will not export or re-export, directly or indirectly, the SDK into any country prohibited by the United States Export Administration Act and the regulations thereunder.
- d. Any direct or indirect distribution of any Bundled Products by you shall be under the terms of a license agreement containing terms that: (i) prohibit any modifications to the Derivative Works or any part thereof, (ii) prohibit any reverse engineering, disassembly or recompilation of the Derivative Works or any part thereof, (iii) disclaim any and all warranties on behalf of Macromedia and each of its licensors, (iv) disclaim, to the extent permitted by applicable law, liability of Macromedia and/or its licensors for any damages, whether direct, indirect, incidental or consequential, arising from the use of the Bundled Products, (v) require the user to comply fully with all relevant export laws and regulations of the United States to assure that the Bundled Products or any part thereof is not exported, directly or indirectly, in violation of United States law, and (vi) include the appropriate copyright notice.

4. Software Defect Reporting

If you find software defects in the SDK, you should report them to Flash_Search@macromedia.com. Macromedia will evaluate and, at its sole discretion, may address them in a future revision of the SDK.

5. Updates

You understand and agree that Macromedia may amend, modify, change, and cease distribution or production of the SDK at any time. You understand that you are

not entitled to receive any upgrades, updates, or future versions of the SDK under this License.

6. Ownership

Macromedia and its suppliers or licensors shall retain all right, title, and interest to the SDK and SWF. All rights not expressly granted herein are reserved by Macromedia.

7. Indemnity

You will indemnify and hold Macromedia harmless from any third party claim, loss, or damage (including attorney's fees) related to your use of the SDK (other than your use of the SDK as expressly permitted under Section 2 hereof but subject to all of the restrictions and limitations set forth in this License) or related to your Product(s).

8. Disclaimer of Warranties and Technical Support

THE SDK IS PROVIDED TO YOU FREE OF CHARGE, AND ON AN "AS IS" BASIS AND "WITH ALL FAULTS", WITHOUT ANY TECHNICAL SUPPORT OR WARRANTY OF ANY KIND FROM MACROMEDIA. YOU ASSUME ALL RISKS THAT THE SDK IS SUITABLE OR ACCURATE FOR YOUR NEEDS AND YOUR USE OF THE SDK IS AT YOUR OWN DISCRETION AND RISK. MACROMEDIA AND ITS LICENSORS DISCLAIM ALL EXPRESS AND IMPLIED WARRANTIES FOR THE SDK INCLUDING, WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ALSO, THERE IS NO WARRANTY OF NON-INFRINGEMENT, TITLE OR QUIET ENJOYMENT.

SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM STATE TO STATE. THESE LIMITATIONS OR EXCLUSIONS OF WARRANTIES AND LIABILITY MAY NOT AFFECT OR PREJUDICE THE STATUTORY RIGHTS OF A CONSUMER; I.E., A PERSON ACQUIRING GOODS OTHERWISE THAN IN THE COURSE OF A BUSINESS.

9. Limitation of Damages

NEITHER MACROMEDIA NOR ITS SUPPLIERS OR LICENSORS SHALL BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES OR LOSS (INCLUDING DAMAGES FOR LOSS OF BUSINESS, LOSS OF PROFITS, OR THE LIKE), ARISING OUT OF THIS LICENSE WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY, PRODUCT LIABILITY OR OTHERWISE, EVEN IF MACROMEDIA OR ITS REPRESENTATIVES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

The limited warranty, exclusive remedies and limited liability set forth above are fundamental elements of the basis of the bargain between Macromedia and you. You agree that Macromedia would not be able to provide the Software on an economic basis without such limitations. IN NO EVENT WILL MACROMEDIA BE LIABLE FOR ANY AMOUNT GREATER THAN WHAT YOU ACTUALLY PAID FOR THE SDK.

10. US Government - Restricted Rights

The SDK and accompanying materials are provided with Restricted Rights. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in this License and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013 (OCT 1988), FAR 12.212(a)(1995), FAR 52.227-19, or FAR 52.227-14, as applicable. Manufacturer: Macromedia, Inc., 601 Townsend, San Francisco, CA 94103.

11. Termination

a. Either party may terminate this License immediately if the other party materially defaults in the performance of any provision of this License, the non-defaulting party gives written notice to the other party of such default,

and the defaulting party fails to cure such default within thirty (30) days after receipt of such notice. Upon the termination of this License, the rights and licenses granted to you by Macromedia pursuant to this License will automatically cease.

b. If this License is terminated for any reason, the provisions of Sections 1, 3, 6, 7, 8, 9, 10, 11(b) and 12 shall survive such termination.

12. General

This License shall be governed by the internal laws of the State of California, without regard to its choice of law rules. This License contains the complete agreement between you and Macromedia with respect to the subject matter of this License, and supersedes all prior or contemporaneous agreements or understandings, whether oral or written. All questions concerning this License shall be directed to: Macromedia, Inc., 601 Townsend Street, San Francisco, CA 94103, Attention: General Counsel.

4.8.8 freetype 2.13.3

License: FTL

The FreeType Project LICENSE

2006-Jan-27

Copyright 1996-2002, 2006 by
David Turner, Robert Wilhelm, and Werner Lemberg

Introduction

=====

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- o We don't promise that this software works. However, we will be interested in any kind of bug reports. ('as is' distribution)
- o You can use this software for whatever you want, in parts or full form, without having to pay us. ('royalty-free' usage)
- o You may not pretend that you wrote this software. If you use

it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. ('credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

"""

Portions of this software are copyright © <year> The FreeType Project (www.freetype.org). All rights reserved.

"""

Please replace <year> with the value from the FreeType version you actually use.

Legal Terms

=====

0. Definitions

Throughout this license, the terms 'package', 'FreeType Project', and 'FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the 'FreeType Project', be they named as alpha, beta or final release.

'You' refers to the licensee, or person using the project, where 'using' is a generic term including compiling the project's source code as well as linking it to form a 'program' or 'executable'. This program is referred to as 'a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

1. No Warranty

THE FREETYPE PROJECT IS PROVIDED 'AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS

BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

2. Redistribution

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- o Redistribution of source code must retain this license file ('FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.
- o Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

3. Advertising

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: 'FreeType Project', 'FreeType Engine', 'FreeType library', or 'FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

4. Contacts

There are two mailing lists related to FreeType:

- o freetype@nongnu.org

Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

o `freetype-devel@nongnu.org`

Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

`http://www.freetype.org`

--- end of FTL.TXT ---

4.8.9 gif-h 0.1

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to `<http://unlicense.org>`

4.8.10 giflib 5.2.1

License: MIT

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

4.8.11 gsl-lite 0.42.0

The MIT License (MIT)

Copyright (c) 2015-2019 Martin Moene
Copyright (c) 2019-2025 Moritz Beutel
Copyright (c) 2015-2018, 2025 Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

4.8.12 gumbo-parser 0.10.2

Apache License

Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

4.8.13 international_components_for_unicode 58.1

License: Unicode-DFS-2016

COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright © 1991–2016 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>

Permission is hereby granted, free of charge, to any person obtaining
a copy of the Unicode data files and any associated documentation
(the "Data Files") or Unicode software and any associated documentation
(the "Software") to deal in the Data Files or Software
without restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, and/or sell copies of
the Data Files or Software, and to permit persons to whom the Data Files
or Software are furnished to do so, provided that either
(a) this copyright and permission notice appear with all copies
of the Data Files or Software, or
(b) this copyright and permission notice appear in associated
Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT OF THIRD PARTY RIGHTS.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale,
use or other dealings in these Data Files or Software without prior
written authorization of the copyright holder.

Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

2. Chinese/Japanese Word Break Dictionary Data (cjdict.txt)

```
# The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice,
# this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
```

```

# the distribution.
# Neither the name of Google Inc. nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
# The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
# * Libtabe (Chinese)
#   - https://sourceforge.net/project/?group\_id=1519
#   - Its license terms and conditions are shown below.
#
# * IPADIC (Japanese)
#   - http://chasen.aist-nara.ac.jp/chasen/distribution.html
#   - Its license terms and conditions are shown below.
#
# -----COPYING.libtabe ---- BEGIN-----
#
# /*
#  * Copyright (c) 1999 TaBE Project.
#  * Copyright (c) 1999 Pai-Hsiang Hsiao.
#  * All rights reserved.
#  *
#  * Redistribution and use in source and binary forms, with or without
#  * modification, are permitted provided that the following conditions
#  * are met:
#  *
#  * . Redistributions of source code must retain the above copyright
#  *   notice, this list of conditions and the following disclaimer.
#  * . Redistributions in binary form must reproduce the above copyright
#  *   notice, this list of conditions and the following disclaimer in
#  *   the documentation and/or other materials provided with the
#  *   distribution.
#  * . Neither the name of the TaBE Project nor the names of its
#  *   contributors may be used to endorse or promote products derived
#  *   from this software without specific prior written permission.
#  *
#  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
#  * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
#  * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
#  * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
#  * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

```

```

# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# /*
# * Copyright (c) 1999 Computer Systems and Communication Lab,
# *      Institute of Information Science, Academia
# *      Sinica. All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# *   notice, this list of conditions and the following disclaimer.
# * . Redistributions in binary form must reproduce the above copyright
# *   notice, this list of conditions and the following disclaimer in
# *   the documentation and/or other materials provided with the
# *   distribution.
# * . Neither the name of the Computer Systems and Communication Lab
# *   nor the names of its contributors may be used to endorse or
# *   promote products derived from this software without specific
# *   prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
#   University of Illinois
# c-tsai4@uiuc.edu http://casper.beckman.uiuc.edu/~c-tsai4
#
# -----COPYING.libtabe-----END-----
#
# -----COPYING.ipadic-----BEGIN-----
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# paragraphs.

```

```
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software. The following conditions for ICOT
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
# on, or be attached to, the Program, which is distributed substantially
# in the same form as set out herein and that such intended
# distribution, if actually made, will neither violate or otherwise
# contravene any of the laws and regulations of the countries having
# jurisdiction over the User or the intended distribution itself.
#
# NO WARRANTY
#
# The program was produced on an experimental basis in the course of the
# research and development conducted during the project and is provided
# to users as so produced on an experimental basis. Accordingly, the
# program is provided without any warranty whatsoever, whether express,
# implied, statutory or otherwise. The term "warranty" used herein
# includes, but is not limited to, any warranty of the quality,
# performance, merchantability and fitness for a particular purpose of
# the program and the nonexistence of any infringement or violation of
# any right of any third party.
#
# Each user of the program will agree and understand, and be deemed to
# have agreed and understood, that there is no warranty whatsoever for
# the program and, accordingly, the entire risk arising from or
# otherwise connected with the program is assumed by the user.
#
# Therefore, neither ICOT, the copyright holder, or any other
# organization that participated in or was otherwise related to the
# development of the program and their respective officials, directors,
# officers and other employees shall be held liable for any and all
# damages, including, without limitation, general, special, incidental
# and consequential damages, arising out of or otherwise in connection
# with the use or inability to use the program or any product, material
# or result produced or otherwise obtained by using the program,
# regardless of whether they have been advised of, or otherwise had
# knowledge of, the possibility of such damages at any time during the
# project or thereafter. Each user will be deemed to have agreed to the
# foregoing by his or her commencement of use of the program. The term
# "use" as used herein includes, but is not limited to, the use,
# modification, copying and distribution of the program and the
# production of secondary products from the program.
#
# In the case where the program, whether in its original form or
# modified, was distributed or delivered to or received by a user from
# any person, organization or entity other than ICOT, unless it makes or
```

```
# grants independently of ICOT any specific warranty to the user in
# writing, such person, organization or entity, will also be exempted
# from and not be held liable to the user for any such damages as noted
# above as far as the program is concerned.
#
# -----COPYING.ipadic-----END-----
```

3. Lao Word Break Dictionary Data (laodict.txt)

```
# Copyright (c) 2013 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: http://code.google.com/p/lao-dictionary/
# Dictionary: http://lao-dictionary.googlecode.com/git/Lao-Dictionary.txt
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
#         (copied below)
#
# This file is derived from the above dictionary, with slight
# modifications.
# -----
# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification,
# are permitted provided that the following conditions are met:
#
#
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in
# binary form must reproduce the above copyright notice, this list of
# conditions and the following disclaimer in the documentation and/or
# other materials provided with the distribution.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.
# -----
```

4. Burmese Word Break Dictionary Data (burmesedict.txt)

```
# Copyright (c) 2014 International Business Machines Corporation
# and others. All Rights Reserved.
#
# This list is part of a project hosted at:
# github.com/kanyawtech/myanmar-karen-word-lists
#
# -----
# Copyright (c) 2013, LeRoy Benjamin Sharon
```

```
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met: Redistributions of source code must retain the above
# copyright notice, this list of conditions and the following
# disclaimer.  Redistributions in binary form must reproduce the
# above copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided
# with the distribution.
#
#   Neither the name Myanmar Karen Word Lists, nor the names of its
#   contributors may be used to endorse or promote products derived
#   from this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
# BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
# TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
# DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
# ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
# TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
# THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
# SUCH DAMAGE.
# -----
```

5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

```
# 7. Database Ownership
#
# The TZ database itself is not an IETF Contribution or an IETF
# document. Rather it is a pre-existing and regularly updated work
# that is in the public domain, and is intended to remain in the
# public domain. Therefore, BCPs 78 [RFC53378] and 79 [RFC3979] do
# not apply to the TZ Database or contributions that individuals make
# to it. Should any claims be made and substantiated against the TZ
# Database, the organization that is providing the IANA
# Considerations defined in this RFC, under the memorandum of
# understanding with the IETF, currently ICANN, may act in accordance
# with all competent court orders. No ownership claims will be made
# by ICANN or the IETF Trust on the database or the code. Any person
# making a contribution to the database or code waives all rights to
# future claims in that contribution or in the TZ Database.
```

4.8.14 leptonica 1.84.1

License: BSD-2-Clause


```

/*=====
- Copyright (C) 2001 Leptonica. All rights reserved.
-
- Redistribution and use in source and binary forms, with or without
- modification, are permitted provided that the following conditions
- are met:
- 1. Redistributions of source code must retain the above copyright
- notice, this list of conditions and the following disclaimer.
- 2. Redistributions in binary form must reproduce the above
- copyright notice, this list of conditions and the following
- disclaimer in the documentation and/or other materials
- provided with the distribution.
-
- THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
- ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
- LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
- A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANY
- CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
- EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
- PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
- PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
- OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
- NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
- SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*=====*/

```

4.8.15 libcurl 7.50.1

License: curl

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2016, Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

4.8.16 liberation-fonts 2.00.5

Digitized data copyright (c) 2010 Google Corporation
with Reserved Font Arimo, Tinos and Cousine.
Copyright (c) 2012 Red Hat, Inc.
with Reserved Font Name Liberation.

This Font Software is licensed under the SIL Open Font License,
Version 1.1.

This license is copied below, and is also available with a FAQ at:
<http://scripts.sil.org/OFL>

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

PREAMBLE The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such.
This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting ? in part or in whole ?
any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

- 1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.
- 2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.
- 3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.
- 4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.
- 5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION

This license becomes null and void if any of the above conditions are not met.

DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

4.8.17 libjpeg 9e

License: IJG

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-2022, Thomas G. Lane, Guido Vollbeding. All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

- (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
- (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
- (3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltmain.sh). Another support script, install-sh, is copyright by X Consortium but is also freely distributable.

4.8.18 libpng 1.6.40

License: libpng-2.0

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE
=====

PNG Reference Library License version 2

- * Copyright (c) 1995-2023 The PNG Reference Library Authors.
- * Copyright (c) 2018-2023 Cosmin Truta.
- * Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson.
- * Copyright (c) 1996-1997 Andreas Dilger.
- * Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or

other liability, whether in contract, tort or otherwise, arising from, out of, or in connection with the software, or the use or other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this software, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux
Eric S. Raymond
Mans Rullgard
Cosmin Truta
Gilles Vollant
James Yu
Mandar Sahastrabuddhe
Google Inc.
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

Some files in the "scripts" directory have other copyright owners, but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

4.8.19 libtiff 4.6.0

License: libtiff

```
# LibTIFF license

Copyright © 1988-1997 Sam Leffler\
Copyright © 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and
its documentation for any purpose is hereby granted without fee, provided
that (i) the above copyright notices and this permission notice appear in
all copies of the software and related documentation, and (ii) the names of
Sam Leffler and Silicon Graphics may not be used in any advertising or
publicity relating to the software without the specific, prior written
permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND,
EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY
WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR
ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND,
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF
LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE
OF THIS SOFTWARE.
```

4.8.20 libwebp 1.3.2

License: BSD-3-Clause

```
Copyright (c) 2010, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in
  the documentation and/or other materials provided with the
  distribution.

* Neither the name of Google nor the names of its contributors may
  be used to endorse or promote products derived from this software
  without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
```

```
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

4.8.21 litehtml 0.3

License: BSD-3-Clause

```
Copyright (c) 2013, Yuri Kobets (tordex)
All rights reserved.
```

```
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
```

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

4.8.22 oda 2023.4

License: COMMERCIAL

```
Portions of this product Copyright Open Design Alliance.
```

4.8.23 openjpeg 2.5.3

License: BSD-2-Clause

```
/*
 * The copyright in this software is being made available under the 2-clauses
 * BSD License, included below. This software may be subject to other third
```



```

* party and contributor rights, including patent rights, and no such rights
* are granted under this license.
*
* Copyright (c) 2002-2014, Universite catholique de Louvain (UCL), Belgium
* Copyright (c) 2002-2014, Professor Benoit Macq
* Copyright (c) 2003-2014, Antonin Descampe
* Copyright (c) 2003-2009, Francois-Olivier Devaux
* Copyright (c) 2005, Herve Drolon, FreeImage Team
* Copyright (c) 2002-2003, Yannick Verschuere
* Copyright (c) 2001-2003, David Janssens
* Copyright (c) 2011-2012, Centre National d'Etudes Spatiales (CNES), France
* Copyright (c) 2012, CS Systemes d'Information, France
*
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS `AS IS'
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/

```

4.8.24 openssl 1.0.2o

License: OpenSSL

LICENSE ISSUES

=====

The OpenSSL toolkit stays under a double license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts.

OpenSSL License

```

/* =====
* Copyright (c) 1998-2019 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions

```

```

* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. All advertising materials mentioning features or use of this
*    software must display the following acknowledgment:
*    "This product includes software developed by the OpenSSL Project
*    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*    endorse or promote products derived from this software without
*    prior written permission. For written permission, please contact
*    openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
*    nor may "OpenSSL" appear in their names without prior written
*    permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
*    acknowledgment:
*    "This product includes software developed by the OpenSSL Project
*    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* =====
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com).  This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License
-----

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*

```

```

* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*    must display the following acknowledgement:
*    "This product includes cryptographic software written by
*    Eric Young (eay@cryptsoft.com)"
*    The word 'cryptographic' can be left out if the rouines from the library
*    being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*    the apps directory (application code) you must include an acknowledgement:
*    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

```

4.8.25 pdfium 5060

License: Apache-2.0

```

// Copyright 2014 PDFium Authors. All rights reserved.
//

```

```
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

4.8.26 protobuf 3.0.0

License: BSD-3-Clause

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
This file is copyrighted by Red Hat Inc.
- Atomicops support for AIX/POWER, located in
src/google/protobuf/stubs/atomicops_internals_power.h.
This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

4.8.27 protobuf-c 1.3.3

License: BSD-2-Clause

Copyright (c) 2008-2016, Dave Benson and the protobuf-c authors.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The code generated by the protoc-gen-c code generator and by the protoc-c compiler is owned by the owner of the input files used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is covered by the above license.

4.8.28 qpdf 8.4.0

License: Apache-2.0

Apache License

Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

4.8.29 safeint 3.0.28

License: MIT

MIT License

Copyright (c) 2018 Microsoft

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

4.8.30 sfntly r239

License: Apache-2.0

Apache License

Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2011 Google Inc. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

4.8.31 skia

License: BSD-3-Clause

Copyright (c) 2011 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

4.8.32 skia-svg

License: BSD-3-Clause

```
// Copyright (c) 2011 Google Inc. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```

```
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

4.8.33 snappy 1.1.3

License: BSD-3-Clause

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===

Some of the benchmark data in testdata/ is licensed differently:

- fireworks.jpeg is Copyright 2013 Steinar H. Gunderson, and
is licensed under the Creative Commons Attribution 3.0 license
(CC-BY-3.0). See <https://creativecommons.org/licenses/by/3.0/>
for more information.
- kppkn.gtb is taken from the Gaviota chess tablebase set, and

is licensed under the MIT License. See <https://sites.google.com/site/gaviotachessengine/Home/endgame-tablebases-1> for more information.

- paper-100k.pdf is an excerpt (bytes 92160 to 194560) from the paper "Combinatorial Modeling of Chromatin Features Quantitatively Predicts DNA Replication Timing in *Drosophila*." by Federico Comoglio and Renato Paro, which is licensed under the CC-BY license. See <http://www.ploscompbiol.org/static/license> for more information.
- alice29.txt, asyoulik.txt, plrabb12.txt and lcet10.txt are from Project Gutenberg. The first three have expired copyrights and are in the public domain; the latter does not have expired copyright, but is still in the public domain according to the license information (<http://www.gutenberg.org/ebooks/53>).

4.8.34 stackwalker 13

License: BSD-2-Clause

BSD-2-Clause

Copyright (c) 2005 - 2019, Jochen Kalmbach
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

4.8.35 teigha 4.0.0

License: COMMERCIAL

Portions of this product Copyright Open Design Alliance.

4.8.36 tesseract 3.02.02

License: Apache-2.0

```
This package contains the Tesseract Open Source OCR Engine.  
Originally developed at Hewlett Packard Laboratories Bristol and  
at Hewlett Packard Co, Greeley Colorado, all the code  
in this distribution is now licensed under the Apache License:
```

```
** Licensed under the Apache License, Version 2.0 (the "License");  
** you may not use this file except in compliance with the License.  
** You may obtain a copy of the License at  
** http://www.apache.org/licenses/LICENSE-2.0  
** Unless required by applicable law or agreed to in writing, software  
** distributed under the License is distributed on an "AS IS" BASIS,  
** WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
** See the License for the specific language governing permissions and  
** limitations under the License.
```

Other Dependencies and Licenses:

=====

Tesseract use Leptonica library (<http://leptonica.com/>) with a very weakly
restricted copyright license (<http://leptonica.com/about-the-license.html>)

4.8.37 tinyspline 0.6.0

License: MIT

MIT License

Copyright (c) 2016 Marcel Steinbeck

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

4.8.38 tinyxml 2.6.2

License: Zlib

```

/*
www.sourceforge.net/projects/tinyxml
Original code by Lee Thomason (www.grinninglizard.com)

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.
*/

```

4.8.39 unrar 5.9.1

License: LicenseRef-unrar

```

*****      *****      *****      UnRAR - free utility for RAR archives
**  **  **  **  **  **  ~~~~~~
*****      *****      *****      License for use and distribution of
**  **  **  **  **  **  ~~~~~~
**  **  **  **  **  **  FREE portable version
                           ~~~~~~

The source code of UnRAR utility is freeware. This means:

1. All copyrights to RAR and the utility UnRAR are exclusively
   owned by the author - Alexander Roshal.

2. UnRAR source code may be used in any software to handle
   RAR archives without limitations free of charge, but cannot be
   used to develop RAR (WinRAR) compatible archiver and to
   re-create RAR compression algorithm, which is proprietary.
   Distribution of modified UnRAR source code in separate form
   or as a part of other software is permitted, provided that
   full text of this paragraph, starting from "UnRAR source code"
   words, is included in license, or in documentation if license
   is not available, and in source code comments of resulting package.

3. The UnRAR utility may be freely distributed. It is allowed

```

to distribute UnRAR inside of other software packages.

4. THE RAR ARCHIVER AND THE UnRAR UTILITY ARE DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS SOFTWARE.
5. Installing and using the UnRAR utility signifies acceptance of these terms and conditions of the license.
6. If you don't agree with terms of the license you must remove UnRAR files from your storage devices and cease to use the utility.

Thank you for your interest in RAR and UnRAR.

Alexander L. Roshal

4.8.40 wavpack 5.6.0

License: BSD-3-Clause

Copyright (c) 1998 - 2022 David Bryant
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Conifer Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

4.8.41 Xpdf 4.05

License: COMMERCIAL

Portions of this product Copyright 1996-2003 Glyph & Cog, LLC

4.8.42 zlib 1.3

License: Zlib

```
/* zlib.h -- interface of the 'zlib' general purpose compression library
   version 1.3, August 18th, 2022
```

Copyright (C) 1995-2023 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly
jloup@gzip.org

Mark Adler
madler@alumni.caltech.edu

```
*/
```

4.8.43 zxing-cpp 1.2.0

Apache License

Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

4.9 Known Issues

- Files compressed with WinZip's ZIPX proprietary JPEG compression are not supported.
- HD Mode: 3D charts will render as their 2D equivalents.
- HD Mode: Apple Pages, Keynote and Numbers can render embedded images only. Files created from templates may have placeholder images that are not embedded in the file.
- HD Mode: Apple Pages, Keynote and Numbers connection lines are not supported.
- HD Mode: Apple Pages, Keynote and Numbers line endpoints (e.g. arrow, circle, or square) are not supported.
- HD Mode: Category rows not supported in Apple Numbers.
- HD Mode: Custom Formats are not support for Apple Numbers.
- HD Mode: Documents will be rendered using the fonts available on the host system. If an original font is not present, a fallback substitute will be applied, which **will affect the final output**.
- HD Mode: Gridlines may appear on some tables without them in Apple Numbers.
- HD Mode: MS Excel files with rotated text within the cell (not a drawing) will be rendered horizontally.
- HD Mode: MS Office files using 'Filename Field' will have text substituted with the name of the file passed to the Open function.
- HD Mode: MS Office Picture Style 3D effects are not supported.
- HD Mode: MS PowerPoint files are rendered shapes first, then text which may affect z-order.
- HD Mode: MS PowerPoint output size will be determined based on the Slide Width and Height, which may be different than the stored Page size.
- HD Mode: MS PowerPoint Slide Notes and Slide Comments will always appear together on a single page.
- HD Mode: The DOCUMENT_HEADERS/FOOTERS options only apply to formats that store headers and footers separately. PDF is **not** one of these formats.
- HD Mode: The following chart types are not supported: Split Pie, Doughnut, Stock, Surface and Radar.
- HD Mode: When combining PDF documents, form fields with duplicate names will have the page number appended to the end of the field name in order to retain its unique value.
- HD Mode: When converting HTML files to images, CSS page breaks are not used.
- HD Mode: When converting to PDF/A and/or PDF1.7, documents containing gradients with transparent steps will be converted to solid color steps.
- HD Mode: When generating PDF files, each individual page cannot be greater than 200x200 inches.
- HD Mode: When processing HTML files, only CSS level 2 and below is supported. CSS level 3 will not be processed.
- HD Mode: When processing HTML files, only embedded CSS is supported. External CSS files will not be processed.

- HD Mode: When processing PDF XFA files, custom Adobe fonts will fallback to the closest available on the host system. This may cause some rendering differences between DocFilters output and Adobe Reader.
- HD Mode: When processing PDF XFA files, embedded Javascript is not executed. This may cause some dynamic content to not render as expected.
- Text Mode: Apple DMG extraction is only supported on MacOS
- Text Mode: MS Excel files processed with EXCELMODE=TEXT (on by default) extract only string information (no numbers or dates). To extract more information, use EXCELMODE=CSV,TSV or CST.

5. Releases

5.1 Releases

- [Document Filters 25.4.0 \(Nov 2025\)](#)
- [Document Filters 25.3.0 \(Aug 2025\)](#)
- [Document Filters 25.2.0 \(May 2025\)](#)
- [Document Filters 25.1.0 \(Feb 2025\)](#)
- [Document Filters 24.4.0 \(Nov 2024\)](#)
- [Document Filters 24.3.0 \(Aug 2024\)](#)
- [Document Filters 24.2.0 \(May 2024\)](#)
- [Document Filters 24.1.0 \(Feb 2024\)](#)
- [Document Filters 23.3.0 \(Sep 2023\)](#)
- [Document Filters 23.2.0 \(May 2023\)](#)
- [Document Filters 23.1.0 \(Feb 2023\)](#)
- [Document Filters 22.4.0 \(Nov 2022\)](#)
- [Document Filters 22.3.0 \(Aug 2022\)](#)
- [Document Filters 22.2.0 \(May 2022\)](#)
- [Document Filters 22.1.0 \(Feb 2022\)](#)
- [Document Filters 21.11.0 \(Nov 2021\)](#)
- [Document Filters 21.8.0 \(Aug 2021\)](#)
- [Document Filters 21.5.0 \(May 2021\)](#)
- [Document Filters 21.2.0 \(Feb 2021\)](#)
- [Document Filters 11.4.20 \(Nov 2020\)](#)
- [Document Filters 11.4.19 \(Jul 2020\)](#)
- [Document Filters 11.4.18 \(May 2020\)](#)
- [Document Filters 11.4.17 \(Feb 2020\)](#)
- [Document Filters 11.4.16 \(December 2019\)](#)
- [Document Filters 11.4.15 \(September 2019\)](#)
- [Document Filters 11.4.14 \(June 2019\)](#)
- [Document Filters 11.4.13 \(March 2019\)](#)

- [Document Filters 11.4 Build 3054 \(November 2018\)](#)
- [Document Filters 11.4 Build 2990 \(September 2018\)](#)
- [Document Filters 11.4 Build 2934 \(July 2018\)](#)
- [Document Filters 11.4 Build 2878 \(May 2018\)](#)
- [Document Filters 11.4 Build 2822 \(March 2018\)](#)
- [Document Filters 11.4 Build 2766 \(January 2018\)](#)
- [Document Filters 11.4 Build 2710 \(November 2017\)](#)
- [Document Filters 11.4 Build 2647 \(September 2017\)](#)
- [Document Filters 11.4 Build 2600 \(August 2017\)](#)
- [Document Filters 11.4 Build 2581 \(July 2017\)](#)
- [Document Filters 11.4 Build 2543 \(June 2017\)](#)
- [Document Filters 11.4 Build 2452 \(March 2017\)](#)
- [Document Filters 11.4 Build 2480 \(April 2017\)](#)
- [Document Filters 11.3 Build 2400 \(January 2017\)](#)
- [Document Filters 11.3 Build 2356 \(December 2016\)](#)
- [Document Filters 11.3 Build 2322 \(November 2016\)](#)
- [Document Filters 11.3 Build 2228 \(August 2016\)](#)
- [Document Filters 11.3 Build 2172 \(June 2016\)](#)
- [Document Filters 11.3 Build 2116 \(April 2016\)](#)
- [Document Filters 11.3 Build 2074 \(March 2016\)](#)
- [Document Filters 11.3 Build 2040 \(January 2016\)](#)
- [Document Filters 11.3 Build 1998 \(December 2015\)](#)
- [Document Filters 11.2 Build 1962 \(November 2015\)](#)
- [Document Filters 11.2 Build 1914 \(September 2015\)](#)
- [Document Filters 11.2 Build 1884 \(September 2015\)](#)
- [Document Filters 11.2 Build 1858 \(August 2015\)](#)
- [Document Filters 11.2 Build 1808 \(June 2015\)](#)
- [Document Filters 11.2 Build 1766 \(May 2015\)](#)
- [Document Filters 11.2 Build 1732 \(March 2015\)](#)
- [Document Filters 11.2 Build 1666 \(January 2015\)](#)
- [Document Filters 11.2 \(November 2014\)](#)

- [Document Filters 11.1 Build 1546 \(October 2014\)](#)
- [Document Filters 11.1 Build 1464 \(July 2014\)](#)
- [Document Filters 11.1 Build 1394 \(April 2014\)](#)
- [Document Filters 11.1 Build 1333 \(March 2014\)](#)
- [Document Filters 11.1 Build 1262 \(December 2013\)](#)
- [Document Filters 11.1 \(October 2013\)](#)
- [Document Filters 11.0 \(July 2013\)](#)

5.2 Document Filters 25.4.0 (Nov 2025)

5.2.1 Enhancements

- Introduced option `PDF_RENDER_MODE` to specify whether PDF output should be rendered for on-screen display or printing. This affects certain rendering behaviors, including how annotations are drawn. (DF-2622)
- Introduced support for OCR of embedded images in Excel (XLS, XLSX, and CELL), PDF, and XFA PDF input files using the `OCR_INLINE_IMAGES` option. Extracted text appears as searchable, accurately positioned content in HiDef output and as inline content in Markdown output, providing enhanced searchability and text extraction without impacting the visual fidelity of documents. (DF-2560)
- Introduced support for specifying top, bottom, left, and right exclusion margins on each page of PDF input, to omit headers, footers, or side notes from extracted text and Markdown output — without affecting HiDef rendering.
 - Use a combined shorthand option (`PDF_EXCLUSION_MARGINS={value}`) or individual options (`PDF_EXCLUSION_MARGIN_TOP` , `PDF_EXCLUSION_MARGIN_BOTTOM` , `PDF_EXCLUSION_MARGIN_LEFT` , `PDF_EXCLUSION_MARGIN_RIGHT`).
 - Margin values can be specified in pixels, points, or as percentages of the page dimensions.
 - This ensures cleaner extracted text for AI workflows, data curation, and search indexing. (DF-2621)

5.2.2 Updates

- HD Mode: Resolved a condition for DOC and DOCX files in HD mode where vanished paragraphs could cause incorrect list numbering during PDF conversion. This ensures paragraph numbering in the output matches the original Word document, improving accuracy for legal and structured documents. (DF-2589)
- HD Mode: Resolved a condition for DOC and DOCX files where floating tables with pinned header rows could repeat previous row when splitting large rows across pages. (DF-2293)
- HD Mode: Resolved a condition where charts with logarithmic scale axes were rendered incorrectly. (DF-2520)
- HD Mode: Resolved a condition where conversion of a non-PDF input file into PDF output fails when the `PDF_PRESERVE_ORIGINAL` option is enabled. (DF-2399)
- HD Mode: Resolved a condition where DOCX files containing bidirectional text could be missing text content. (DF-2347)
- Resolved a condition for DOC files where only the first bullet in each list was retained when converting to Markdown. This ensures all list items are correctly interpreted, improving accuracy and completeness in Markdown exports. (DF-2609)
- Resolved a condition for DOCX files where a segmentation fault could occur during conversion to PDF when processing certain table content. This stability improvement prevents crashes during automated document transformations, ensuring more reliable handling of complex DOCX structures. (DF-2667)

- Resolved a condition for EML files where calendar decline messages produced incorrect output, displaying VCAL metadata instead of the expected footer text. This ensures that converted EML files now retain footer content consistent with how Outlook displays these messages, improving fidelity for end users. (DF-2634)
- Resolved a condition for MS Word 6 files where files created on MacOS may extract or render extended ASCII characters incorrectly. (DF-2532)
- Resolved a condition for PDF files where some tables with full borders may not be extracted as tables. (DF-2620)
- Resolved a condition for PPTX files where a corrupted file could cause a memory leak. (DF-2668)
- Resolved a condition for the Java API where calling getDefaultAppearance on an annotation could cause a SIGSEGV. (DF-2580)
- Resolved a condition for the Java API where Popup annotations could have infinitely nested child Popup annotations. This prevents excessive memory usage and ensures annotation structures are correctly terminated, improving stability and reliability for downstream processing. (DF-2583)
- Resolved a condition for the Java API where setting and then retrieving an action on an annotation object would not return the saved action. (DF-2582)
- Security: 7-zip: CVE-2025-53816: not-affected: The vulnerable code is part of the full 7-Zip application, not the LZMA SDK, and is not compiled into Document Filters. (DF-2572)
- Security: 7-zip: CVE-2025-53817: not-affected: The vulnerable code is not compiled into Document Filters. (DF-2571)
- Security: libtiff: CVE-2025-9165: not-affected: The vulnerable code is in tiff CLI tool tiffcmp, and is not compiled into Document Filters. (DF-2573)
- Security: libtiff: CVE-2025-9900: patched: Applied security patch to address this issue. (DF-2638)
- Security: Protobuf: CVE-2025-10220: CVE affects .NET NuGet packages (Google.Protobuf, DynamicData, etc.) used in AxxonSoft Axxon One VMS. Document Filters uses C++ protobuf library v3.0.0 from source, not .NET NuGet packages. None of the vulnerable .NET packages are present in Document Filters codebase. (DF-2607)
- Security: xpdf: CVE-2025-11896: patched: Applied security patch to address this issue. (DF-2653)

5.3 Document Filters 25.3.0 (Aug 2025)

5.3.1 Enhancements

- Introduced support for file type identification of JSON files. (DF-2457)
- Introduced support for HL7 Medical Records, including detection and processing into both Markdown and HiDef output modes. This enables HL7 content to be transformed into structured, machine-readable formats for use in AI workflows, data integration, and analysis while maintaining fidelity to the original record structure. (DF-2467)
- Introduced support for OCR processing of embedded images within document pages when exporting to HiDef output. The new `OCR_INLINE_IMAGES` option detects and extracts text from inline images, ensuring that more context is preserved for downstream uses such as AI workflows, data extraction, and content transformation. Supported in this release for email and web archive formats (HTML, MSG, EML, MHT, ICS/VCAL, OLK15) and office productivity formats (Apple Keynote, Apple Numbers, Apple Pages, HWPX, OpenOffice, RTF, MS Word, WordPerfect, MS Visio, MS PowerPoint). This capability helps retain meaningful text content from screenshots, diagrams, and other inline images, ensuring it is available in structured outputs like Markdown without altering the original appearance in image-based formats. (DF-2384)
- Introduced support for table detection in PDFs to better recognize tables that contain only inner borders without explicit outer borders. This enhancement ensures more accurate identification of grid-based data for downstream uses such as AI systems, structured data extraction, and document transformation workflows. (DF-2374)

5.3.2 Updates

- HD Mode: Resolved a condition for Apple iWork files where images may rendered with low resolution. (DF-2503)
- HD Mode: Resolved a condition for Apple iWork files where padding may be incorrectly applied to the top of a textbox. (DF-2503)
- HD Mode: Resolved a condition for Apple iWork files where paragraphs may be rendered with a black background. (DF-2503)
- HD Mode: Resolved a condition for DOCX files where floating tables with disabled overlap were overlapping. (DF-2479)
- HD Mode: Resolved a condition for DOCX files where numbered lists were sometimes skipping sequence numbers. (DF-2491)
- HD Mode: Resolved a condition for DOCX files where the layout of inline images could lead to a Page Limit error. (DF-2222)
- HD Mode: Resolved a condition for DOCX where Unicode characters were not rendered properly in non-standard files. UTF-8 is now a default encoding for DOCX internal XML files. Added `MSWORD_DEFAULT_UTF8` option allowing for a revert to previous behavior. (DF-2475)

- HD Mode: Resolved a condition for EML and MSG files where HTML elements with style "display:none" were excluded from output. A new option "HTML_SHOW_DISPLAY_NONE" (defaulting to OFF) can be enabled to force display of hidden elements of HTML structures for EML and MSG files, which mimics the behavior of some email clients, like Outlook. (DF-2421)
- HD Mode: Resolved a condition where Microsoft Word documents created with Microsoft Office 2013 or later could have graphics placed in the wrong position. (DF-2248)
- HD Mode: Resolved a condition where table rows containing graphics and/or text boxes were not sized properly, and text boxes were positioned incorrectly with regards to graphic elements. (DF-2224)
- HD Mode: Resolved an issue where HTML-based files using indirect color values could render as black. (DF-2379)
- Resolved a condition for MSG files where files may render differently between Windows and Linux. (DF-2522)
- Resolved a condition for PDF files where punctuation at the end of the line may appear in the wrong location in RTL documents. (DF-2482)
- Resolved a condition for the OCR image callback where the SourcePageIndex may be incorrectly set when processing PDF files. (DF-2511)
- Resolved a condition for XLS files where files with an invalid SupBook could fail to render. (DF-2512)
- Resolved a condition of OCR callback hooks where a SIGSEGV may be raised when processing images <= 8 bits per pixel. (DF-2547)
- Resolved a condition where converting a PDF file with the only root reference being inside an xref object could cause a segmentation fault. (DF-2476)
- Resolved a condition where DOCX metadata output used "AppVersion" for the application version, while other Office formats used "App Version". DOCX now uses "App Version" for consistency in metadata naming across Office formats. (DF-2248)
- Resolved a condition where processing a corrupted PDF file with a xref stream could result in the file being recognized as unreadable instead of corrupted. (DF-2505)
- Resolved a condition where text returned from the Python bindings method `Extractor.GetText(stripControlCodes=True)` would be missing commas. (DF-2488)
- Security: 7-zip: CVE-2025-55188: Document Filters is not affected because the vulnerable code path is never invoked. The product requires callers to provide an explicit, validated extraction path and does not use p7zip to directly create files on the system. These runtime protections prevent exploitation of the symbolic link handling issue. (DF-2564)
- Security: international_components_for_unicode: CVE-2025-54627: code-not-present: issue is specific to genrb CLI tool build process, not in the ICU library itself. Document Filters does not use the genrb binary or the affected parsing code during runtime operation. (DF-2557)
- Security: libtiff: CVE-2024-13978: not-affected: affected code is not utilized by Document Filters. (DF-2546)
- Security: libtiff: CVE-2025-8176: not-affected: affected code is not utilized by Document Filters. (DF-2540)

- Security: libtiff: CVE-2025-8176: not-affected: affected code is not utilized by Document Filters. (DF-2558)
- Security: libtiff: CVE-2025-8177: not-affected: affected code is not utilized by Document Filters. (DF-2539)
- Security: libtiff: CVE-2025-8534: not-affected: affected code is not utilized by Document Filters. (DF-2550)
- Security: libtiff: CVE-2025-8961: Document Filters vendors libtiff 4.6.0 and does not compile or ship the tiffcrop CLI utility (tools/tiffcrop.c not part of build; no tiffcrop objects found in build output). CVE only impacts tiffcrop in 4.7.0 per NVD/VulDB advisory; our version predates affected release and excludes the tool, making the vulnerable code unreachable. (DF-2565)
- Security: openjpeg: CVE-2025-50952: patched: Applied security patch to address the issue. (DF-2555)
- Security: openjpeg: CVE-2025-54874: patched: Applied security patch to address the issue. (DF-2552)
- Security: Protobuf: CVE-2025-4565: not-affected: affected code is not utilized by Document Filters. (DF-2498)
- Security: protobuf: CVE-2025-53605: not-affected: Vulnerability only affects rust protobuf code, those parts are not incorporated into Document Filters. (DF-2509)
- Security: skia: CVE-2025-54627: code-not-present: issue is specific to Huawei HarmonyOS's implementation of Skia and its associated calling code, therefore Document Filters is not affected by this issue. (DF-2554)
- Security: zlib: CVE-2025-4638: not-affected: affected code is not utilized by Document Filters. (DF-2465)
- Security: zlib: CVE-2025-4638: not-exploitable: the issue is specific to the PointCloudLibrary, which uses a modified version of the ZLib library. (DF-2463)
- Security: zlib: CVE-2025-5087: false-positive: the issue is specific to the Kaleris NAVIS systems, not a vulnerability of the library itself. (DF-2507)
- Security: zlib: CVE-2025-52937: not-affected: issue is specific to the PointCloudLibrary, which uses a modified version of the ZLib library. This issue does not affect Document Filters. (DF-2506)

5.4 Document Filters 25.2.0 (May 2025)

5.4.1 Enhancements

- Improved conversion of Microsoft Outlook files to Markdown with better mapping of lists and paragraphs. (DF-2365)
- Introduced a flexible OCR callback mechanism that allows AI and ML systems to integrate custom OCR engines—such as Python-based solutions like EasyOCR—directly into the Document Filters workflow. This enhancement enables users to invoke OCR functions in the language of their host application, providing greater control and adaptability for image-based document processing, with extracted text flowing seamlessly into standard Document Filters outputs like Markdown, PDF, HTML, and text. See the Document Filters documentation for implementation guidance and sample integrations. (DF-2265)
- Introduced a new `IGR_Open_Ex` callback, `IGR_OPEN_CALLBACK_ACTION_APPROVE_EXTERNAL_RESOURCE`, enabling applications to approve or deny the download of external images in HTML-based documents. (DF-1722)
- Introduced a new `IGR_Open_Ex` callback, `IGR_OPEN_CALLBACK_ACTION_GET_RESOURCE`, allowing applications to provide external resource bytes, such as images, when processing HTML-based documents. This enables greater flexibility by allowing developers to supply resource data directly via an `IGR_Stream`, eliminating dependencies on strict directory structures or URL-based downloads. (DF-2372)
- Introduced support for adding positional metadata to Markdown and JSON MDAST outputs, enabling AI and ML systems to map extracted content back to its original position in the source document. When enabled via "MARKDOWN_INCLUDE_LOCATIONS" or "JSON_INCLUDE_BOUNDS", positional data is embedded as comments in Markdown or as structured properties in JSON MDAST, enhancing contextual understanding for use cases like search, content analysis, and document review. See the Document Filters documentation for more details. (DF-2341)
- Introduced support for converting complex tables—including those with lists and other unsupported Markdown content—by embedding sanitized HTML tables within the Markdown output. This ensures complete and accurate content preservation without data loss, improving reliability for AI, data curation, and documentation workflows. Standard tables without complex content continue to render as native Markdown for optimal readability. (DF-2345)
- Introduced support for normalizing dashes into the content cleaning feature in Markdown and JSON outputs using the `normalize_dashes` cleaning function. This option normalizes dashes and characters that commonly serve as dashes into the ASCII minus sign (45/0x2D) to improve readability and machine-friendliness of generated content, making it ideal for downstream processing in AI/ML systems. (DF-2220)
- Introduced support to identify and extract tables from XFA PDF files, preserving the logical structure of tables, rows, and cells for enhanced AI data analysis. It can be enabled using the `PDF_TABLE_DETECTION` option, defaulting to OFF. (DF-2196)
- Introduced the `MARKDOWN_TABLE_DETECT_TITLES` option to enable detection of table titles when the first row spans all columns. (DF-2206)
- Upgraded FreeType to 2.13.3. (DF-2446)

- Upgraded OpenJPEG to 2.5.3. (DF-2360)

5.4.2 Updates

- HD Mode: Resolved a condition for DOCX files where numbered lists were sometimes skipping sequence numbers on nested levels. (DF-2290)
- HD Mode: Resolved a condition when converting an HTML file where some special characters would not have their size recognized correctly. (DF-2338)
- HD Mode: Resolved a condition where converting a PDF with an invalid page size results in an empty file. Introduced options `PDF_PAGE_SIZE_WIDTH` and `PDF_PAGE_SIZE_HEIGHT` to set the maximum allowed page width and height for PDF input. Both default to 14400 points (200 inches). When the limit is exceeded, Document Filters will return a descriptive error. (DF-2233)
- HD Mode: Resolved a condition where converting an XLSX file could fail with a "bad allocation" error. (DF-2335)
- HD Mode: Resolved an issue in Markdown conversion of Office documents where certain bullet lists could trigger the error: "pop() called with mismatched type". (DF-2435)
- HD Mode: Resolved an issue in RTF formatting where images were mislocated and some paragraphs could disappear. (DF-2373)
- HD Mode: Resolved an issue where paragraph borders were rendered improperly when placed inside a table. (DF-2120)
- HD Mode: Resolved an issue where tables disappeared when processing Open Office Impress files. (DF-2120)
- Resolved an issue in PDF table detection where header rows were not recognized if they were outside a bordered area. (DF-2368)
- Resolved an issue where Microsoft Office documents encrypted using SHA-1 could fail to decrypt. (DF-2430)
- Resolved an issue where references with special characters in XLS could cause "invalid string position" error. (DF-2351)
- Resolved an issue with HTML files that could cause truncation when a file contained improperly encoded multi-byte sequences (e.g., invalid UTF-8). The document may still display "broken" characters in place of the invalid sequences; correcting these requires changes to the source file. (DF-2427)
- Security: bzip2: CVE-2024-3508: not-exploitable: Issue exists for Bombastic sBOM editor; does not impact Document Filters. (DF-2402)
- Security: Chrome: CVE-2025-1918: not-affected: The vulnerable code is not compiled into Document Filters. (DF-2418)
- Security: FreeType: CVE-2025-27363: not-affected: The vulnerable code no longer exists in version 2.13.3 of the library. (DF-2400)
- Security: Giflib: CVE-2024-45993: not-exploitable: Issue exists in CLI, not in the library; does not impact Document Filters. (DF-2401)

- Security: Giflib: CVE-2025-31344: not-exploitable: Issue exists in CLI, not in the library; does not impact Document Filters. (DF-2443)
- Security: OpenJPEG: CVE-2023-39327: patched: Applied security patch to address this issue. (DF-2417)
- Security: OpenJPEG: CVE-2023-39328: patched: Applied security patch to address this issue. (DF-2415)
- Security: OpenJPEG: CVE-2023-39329: patched: Applied security patch to address this issue. (DF-2414)
- Security: OpenJPEG: CVE-2024-56826: not-exploitable: Issue does not exist in version. (DF-2413)
- Security: OpenJPEG: CVE-2024-56827: not-exploitable: Issue does not exist in version. (DF-2412)
- Security: Protobuf: CVE-2024-24786: not-exploitable: Vulnerability requires access to Unmarshal, which is not reachable in Document Filters. (DF-2404)
- Security: Protobuf: CVE-2024-7254: not-exploitable: Vulnerability only affects Java Protobuf code; these parts are not incorporated into Document Filters. (DF-2403)
- Security: Skia: CVE-2024-43768: not-exploitable: Document Filters use zlib library directly to process PDF files, not Skia. (DF-2407)
- Security: Skia: CVE-2025-0436: not-exploitable: Document Filters does not use Skia library for HTML conversions. (DF-2405)
- Security: Skia: CVE-2025-0444: not-exploitable: Issue exists in Blink rendering engine; does not impact Document Filters. (DF-2406)
- Security: Snappy: CVE-2024-36114: not-exploitable: Issue exists for Aircompressor library; does not impact Document Filters. (DF-2408)
- Security: Tesseract: CVE-2024-29511: not-exploitable: Document Filters does not use Artifex Ghostscript software. (DF-2409)
- Security: Xpdf: CVE-2025-2574: patched: Applied security patch to address this issue. (DF-2419)
- Security: Xpdf: CVE-2025-3154: patched: Applied security patch to address this issue. (DF-2425)
- Security: zlib: CVE-2024-57923: not-exploitable: Document Filters does not use zlib for Btrfs or target s390 platforms. (DF-2429)
- Security: zlib: CVE-2025-0725: not-exploitable: Document Filters does not use zlib for processing of HTTP responses. (DF-2411)
- Security: zlib: CVE-2025-25293: not-exploitable: Document Filters does not use zlib for SAML processing. (DF-2410)

5.5 Document Filters 25.1.0 (Feb 2025)

5.5.1 Enhancements

- Introduced file identification for incomplete or truncated files, allowing Document Filters to determine file types using the first 2KB of data in most cases. (DF-2039)
- Introduced headings mapping for Hangul HWP 3 files to improve structured output. (DF-2261)
- Introduced headings mapping for Hangul HWP 5 files to improve structured output. (DF-2198)
- Introduced headings mapping for Hangul HWPX files to improve structured output. (DF-2130)
- Introduced headings mapping for Open Document Presentation files (.odp) to improve structured output. (DF-2124)
- Introduced headings mapping for PDF XFA files to improve structured output. (DF-2197)
- Introduced list item mapping for Hangul HWP files to improve structured output. (DF-2260)
- Introduced list item mapping for Hangul HWPX files to improve structured output. (DF-2262)
- Introduced list item mapping for PDF XFA files to improve structured output. (DF-2259)
- Introduced list item mapping for WordPerfect 6 files to improve structured output. (DF-2263)
- Introduced option `PDF_LIST_DETECTION` for detecting bullet and numbered lists in PDF files, allowing for more accurate mapping and output to structured formats such as Markdown and JSON. (DF-2125)
- Introduced option `PDF_MARKDOWN_UNWRAP_TEXT` for detecting line wrapping in PDF files, allowing for cleaner output when generating structured formats such as Markdown and JSON. (DF-2125)
- Introduced slide title and subtitle mappings to headings for Apple Keynote 09 files to improve structured output. (DF-2126)
- Introduced slide title and subtitle mappings to headings for Apple Keynote 14 files to improve structured output. (DF-2127)
- Upgraded Xpdf to 4.05.02. (DF-2084)

5.5.2 Updates

- HD Mode: Resolved a condition for HTML based files where content after an unexpected `</html>` close tag would not appear in rendered output. (DF-2289)
- HD Mode: Resolved a condition for PostScript files where files containing type-3 fonts may fail to render. (DF-2241)
- HD Mode: Resolved a condition where converting a PPTX file may cause a memory violation. (DF-2272)
- HD Mode: Resolved a condition where email headers may not display correctly when HTML tags are commented. (DF-2147)
- HD Mode: Resolved a condition where processing Microsoft Office Excel files (.xlsx) containing charts could cause a memory violation. (DF-2269)

- HD Mode: Resolved a condition where processing Microsoft Office Word files (.doc) containing Microsoft Office Excel elements could cause a memory violation. (DF-2270)
- Resolved a condition for MS Office documents where improperly formatted headers/footers may cause a crash. (DF-2268)
- Resolved a condition for XPS files where files stored in pieces would fail to identify and extract. (DF-2244)
- Resolved a condition where converting an HTML file to markdown in text-mode may cause an error. (DF-2266)
- Resolved a condition where EML files were identified as Text when opened via stream. (DF-2230)
- Resolved a condition where processing a PDF containing a page with a very large number of characters would take a very long time. A new option, `PDF_MAX_CHARS_PER_PAGE`, which defaults to 50000, can be used to manage how many characters on a page of PDF input are processed by Document Filters; setting the option to a value of 0 removes the limit. (DF-2271)
- Resolved a condition where setting `MARKDOWN_SIMPLE_TABLE_STYLE` to `HTML` would still output pipe tables when generating Markdown. (DF-2288)
- Security: 7-zip: CVE-2024-11477: not-exploitable: unused code block; does not impact Document Filters. (DF-2274)
- Security: 7-zip: CVE-2025-0411: not-exploitable: issue exists in desktop application, not in library; does not impact document filters. (DF-2325)

5.6 Document Filters 24.4.0 (Nov 2024)

5.6.1 Enhancements

- Introduced option "GRAPHIC_BMP_DPI" to control whether DPI stored in BITMAPINFOHEADER is read when processing BMP image files. Defaults to OFF, resulting in a default 96 DPI being used. (DF-2066)
- Introduced support for a new Simplified JSON output canvas type. This streamlines the representation of document structures by leveraging a simplified JSON schema. It reduces the complexity of processing full DOM structures, making it particularly beneficial for AI and ML applications that require structured data, such as headings, tables, paragraphs, and lists, in a flattened hierarchy similar to Markdown output. (DF-2170)
- Introduced support for C++17 in the Document Filters C++ bindings. (DF-2140)
- Introduced support for including image metadata in Markdown output when "MARKDOWN_INCLUDE_METADATA=ON". (DF-2227)
- Introduced support for Markdown output in text mode, allowing users to convert documents into Markdown format through the new "IGR_FORMAT_MARKDOWN" text mode flag. This feature enhances flexibility by not requiring HD Mode for Markdown output, resulting in more performant and efficient output. (DF-2200)
- Introduced support for simplified content cleaning in Markdown and JSON outputs. This feature offers multiple configurable cleaning options, including removing non-ASCII characters and normalizing quotes. These options improve the readability and machine-friendliness of generated content, making it ideal for downstream processing in AI/ML systems. (DF-2175)
- Introduced the Document Filters Python library as a GitHub package that can now be installed using a Python package manager. (DF-1975)

5.6.2 Updates

- HD Mode: Resolved a condition for Apple Numbers files where cell data may be missing when converting to Hi-Def. (DF-2187)
- HD Mode: Resolved a condition for Apple Numbers files where highlighted text maybe shaded with a black background. (DF-2188)
- HD Mode: Resolved a condition for MS Office files where incorrect table headers were repeated if the table doesn't start from the top of the page. (DF-1645)
- HD Mode: Resolved a condition for MS Office files where table header rows are repeated if they don't fit on one page. (DF-1646)
- HD Mode: Resolved a condition for MS Office files where tables split across pages may cause the file to not convert. (DF-2158)
- HD Mode: Resolved a condition where internal links in MS Office files may display as plain text. (DF-2153)

- HD Mode: Resolved a condition where some tables would have missing cells when converting Word95 (or older) files. (DF-2107)
- HD Mode: Resolved a condition where tables in MS Powerpoint 97-2007 files would have their background color displayed incorrectly. (DF-2150)
- Resolved a condition in MS Office files where sub-directories could be reported as subfiles, causing an exception to be thrown. (DF-2223)
- Resolved a condition where attempting to open an rc4 encrypted file with a very long password would cause a segmentation fault. (DF-2179)
- Resolved a condition where non-latin characters would not be converted properly when converting EML files with UTF8 Byte Order Marks. (DF-2056)
- Resolved a condition where processing a Microsoft Office Excel 97-2003 Binary File Format (.xls) file with a self-reference to a range of cells that includes the 65536th row could hang or run indefinitely. (DF-2166)
- Resolved a condition where WordArt text was missing from text-mode output for MS Office binary format files (e.g. .doc .ppt .xls). (DF-2094)
- Resolved a memory leak when converting a corrupted RTF file to JSON. (DF-2114)
- Security: PDFium: CVE-2024-7973: not-exploitable: unused code block; does not impact Document Filters. (DF-2152)
- Security: xpdf: CVE-2024-7866: patched: applied security patch to address this issue. (DF-2143)
- Security: xpdf: CVE-2024-7867: not-exploitable: unused code block; does not impact Document Filters. (DF-2176)
- Security: xpdf: CVE-2024-7868: not-exploitable: issue is already mitigated; does not impact Document Filters. (DF-2177)

5.7 Document Filters 24.3.0 (Aug 2024)

5.7.1 Enhancements

- Introduced option "HTML_IMAGES_SCALE_TO_FIT" to control whether images attached to email messages are scaled to fit the page. Defaults to OFF, resulting in no scaling. (DF-1907)
- Introduced support for a new JSON output canvas type that structures document data in a detailed, hierarchical format for seamless integration with AI and other JSON-compatible applications. This enhancement ensures efficient parsing and utilization of document content, improving AI-driven data analysis and processing. (DF-2043)
- Introduced support for a new Markdown output canvas type. This allows users to seamlessly convert documents to Markdown, providing an efficient way to present basic formatting, along with content. Markdown's lightweight nature makes it ideal for various applications, including displaying content in wiki systems, as well as interactions with AI systems. (DF-2015)
- Introduced support for accessing "IGR_Get_Page_Elements" in the .NET, Java, and Python APIs. (DF-2044)
- Introduced support for accessing the "MimeType" property through the "Extractor" class in the .NET, Java, and Python APIs. (DF-2042)
- Introduced support for enumerating supported configuration options in the Java APIs using the new "GetAvailableOptions" method. (DF-2041)
- Introduced support for enumerating supported file types in the Java APIs using the new "GetSupportedFormats" method. (DF-2041)
- Introduced support for extracting links to external workbooks within XLSX files using the "SHOWHIDDEN_EXCEL_REFS" option, defaulting to OFF. (DF-1902)
- Introduced support for HD rendering of Hancom Hangul HWPX files, enabling enhanced rendering and improved fidelity for these document types. This update ensures accurate display and processing of HWPX documents, maintaining their original formatting and layout. (DF-1932)
- Introduced support for identification and text extraction support for Hancom Hangul HWPX files. (DF-1258)
- Introduced support for sub-file extraction for MSI Installer files. (DF-2038)
- Introduced support to be able to determine if a subfile is password protected, before extracting the file. For the C API, IGR_SUBFILE_INFO_FLAG_PASSWORD_PROTECTED will be set on the flags of the IGR_Subfile_Info structure. For Object APIs, the IsEncrypted property has been added to the SubFile object. (DF-1968)
- Introduced support to identify and extract tables from untagged PDF files, preserving the logical structure of tables, rows, and cells for enhanced AI data analysis. This feature ensures accurate table detection for vector-based PDFs, facilitating better utilization in AI and other applications. It can be enabled using the "PDF_TABLE_DETECTION" option, defaulting to OFF. (DF-2045)

5.7.2 Updates

- HD Mode: Resolved a condition for DOCX files where illustrations (pictures, shapes, etc.) vertically positioned with "Margin" could be placed incorrectly. (DF-1839)
- HD Mode: Resolved a condition for DOCX files where images with a single textbox/graphic in a table resulted in an incorrect size of the row containing the object. (DF-1874)
- HD Mode: Resolved a condition for HTML files where styles on form elements may not be rendered. (DF-1840)
- HD mode: Resolved a condition for PDF files where too large input pages resulted in failed memory allocation. Output page size is limited to 200 inches in any dimension for xpdf engine. (DF-2099)
- HD Mode: Resolved a condition for RTF files where content with nested table may not render as expected. (DF-2095)
- HD Mode: Resolved a condition for RTF files where line spacing inherited from styles may render differently than MS Word. (DF-1702)
- HD mode: Resolved a condition for XLSX files where conversion of large files with a large number of duplicated rows resulted in long processing time and large memory usage, while adding the "SPREADSHEET_COLLAPSE_ROWS" option, defaulting to 100. (DF-1996)
- HD Mode: Resolved a condition where converting a text document to HDHTML may result in text placed in incorrect locations. (DF-1708)
- HD Mode: Resolved a condition where converting an RTF file could result in a crash when using an object API. (DF-2085)
- HD Mode: Resolved a condition where converting DOCX files with images and setting the GRAPHIC_DPI option resulted in the images being shifted. (DF-1983)
- HD Mode: Resolved a condition where converting to HTML5 may cause some images to have an empty 'idf-graphic-data' tag. (DF-1999)
- HD Mode: Resolved a condition where having multiple threads convert text files to PDF could cause a segmentation fault. (DF-2075)
- HD Mode: Resolved a condition where style inheritance could be incorrectly applied for paragraph and character styles in Office Open XML documents (e.g. MS Word .docx, MS Excel .xlsx, MS Powerpoint .pptx). (DF-2009)
- HD Mode: Resolved a condition where text with automatic font color or no font color on dark fill/background would render with font color black instead of white. (DF-1986)
- HD Mode: Resolved a potential memory leak for PDFs when converting to TIFF. (DF-2010)
- HD Mode: Resolved an issue where converting a file with text in tables could overflow its cell and overlap with content in adjacent cells. (DF-2051)
- Security: libtiff: CVE-2024-7006: patched: applied security patch to address this issue. (DF-2137)
- Security: PDFium: CVE-2024-5846: patched: applied security patch to address this issue. (DF-2089)
- Security: PDFium: CVE-2024-5847: patched: applied security patch to address this issue. (DF-2088)

- Security: xpdf: CVE-2024-4141: not-exploitable: issue is already mitigated; does not impact document filters. (DF-1993)
- Security: xpdf: CVE-2024-4568: not-exploitable: issue is already mitigated; does not impact document filters. (DF-2108)
- Security: xpdf: CVE-2024-4976: not-exploitable: issue is already mitigated; does not impact document filters. (DF-2109)

5.8 Document Filters 24.2.0 (May 2024)

5.8.1 Enhancements

- **Enhanced Accessibility Info Extraction:** Document Filters now supports the extraction of accessibility information from MS Office documents, aiding in the development of accessible products. The extracted "alt text" labels include images, drawings, objects, word art, smart art, charts, icons, shapes, and text boxes across Word (DOC and DOCX), PowerPoint (PPT and PPTX), Excel (XLS and XLSX), and Visio (VSDX) files. The extracted alt text is surfaced in various output formats, including HTML5, Classic HTML, XML, and PDF (when creating accessible output). (DF-1926)
- **Introduced support for extracting links to external workbooks within XLS files when EXCELMODE is not 'TEXT',** using the "SHOWHIDDEN_EXCEL_REFS" option, defaulting to OFF. (DF-1662)
- **Introducing Custom Tesseract Integration:** In response to compatibility challenges with the latest Tesseract version, we've enhanced Document Filters to seamlessly integrate with custom versions of Tesseract. Users can now specify their preferred Tesseract executable path via a simple configuration option. If no custom path is provided or if the location is not found, Document Filters intelligently falls back to the bundled Tesseract version. Our updated documentation provides clear guidance on configuring and utilizing custom Tesseract versions with Document Filters, ensuring a smooth integration experience. (DF-1979)
- **Introducing Document Comparison API:** Our latest release introduces a powerful API for comparing documents with precision and flexibility. Users can compare entire documents or individual pages, with options to exclude headers, footers, and specify margins for accurate comparisons. The API detects and highlights changes, including modifications, insertions, and deletions, providing a comprehensive view of document differences. With markup rendering for visual clarity, users can easily review and assess changes. Supporting various formats like MS Word and PDF, this API is well-documented, scalable, and designed for seamless integration into existing systems. Learn more about its capabilities in our documentation here: https://docs.hyland.com/DocumentFilters/en_US/Print/tutorials/how_do_i_compare_documents.html. (DF-1984)
- **Introducing Enhanced File Type Categorization:** In this release, we've expanded the capabilities of `IGR_Get_Format_Attribute` by introducing a more detailed File Format Category system. Now, users can categorize file formats based on the type of applications that created them, allowing for more granular organization and handling of various file types within the Document Filters system. This feature brings added flexibility and efficiency to file format management, enhancing the overall usability of the platform. (DF-1903)
- **Introducing RGBA and ABGR Pixel Formats:** We've enhanced the functionality of `IGR_Get_Page_Pixels` and `IGR_Open_DIB` functions by adding support for RGBA and ABGR pixel formats. (DF-1953)
- **Upgraded Leptonica to 1.84.1.** (DF-1638)
- **Upgraded ODA to 23.12 SP2.** (DF-1829)
- **Upgraded Xpdf to 4.05.** (DF-1897)

5.8.2 Updates

- HD Mode: Resolved a condition for MSG files where black boxes may appear incorrectly. (DF-1928)
- HD Mode: Resolved a condition for PDF files where radio buttons and check boxes were incorrectly marked as ON. (DF-1970)
- HD Mode: Resolved a condition for PDF files where radio buttons were displayed in a rectangle instead of a circle. (DF-1970)
- HD Mode: Resolved a condition for Visio files where fonts scaled below 1 would render as large default font-size. (DF-1965)
- HD Mode: Resolved a condition when converting an RTF file with an image may cause the image to shift. (DF-1951)
- HD Mode: Resolved a condition when using 'DrawImage' or 'DrawScaleImage' with an HTML5 canvas would cause the same image to be drawn on each page. (DF-1900)
- HD Mode: Resolved a condition where converting a DOCX file to PDF on Linux may result in the file size drastically increasing. (DF-2008)
- HD Mode: Resolved a condition where converting an DWG file may cause a segmentation fault. (DF-1985)
- HD Mode: Resolved a condition where converting an MS Excel file may cause a segmentation fault. (DF-1962)
- HD Mode: Resolved a condition where converting an MS Word file may cause a segmentation fault. (DF-1937)
- HD Mode: Resolved a condition where converting an MS Word file may cause a segmentation fault. (DF-1963)
- HD Mode: Resolved a condition where converting an MSG file to an image may cause the body of the MSG to not render. (DF-1893)
- HD Mode: Resolved a condition where converting an RTF file to PDF may cause the output to be empty. (DF-1978)
- HD Mode: Resolved a condition where converting an XLSB file may result in a corrupt sheet error. (DF-1929)
- HD Mode: Resolved a condition where PPTX files containing a corrupted image would cause the file conversion operation to fail. (DF-1967)
- HD Mode: Resolved a condition where setting OCR_REORIENT_PAGES=ON and setting GRAPHIC_DPI to a specific value will cause any page that is 'reoriented' to not respect the GRAPHIC_DPI value. (DF-1973)
- HD Mode: Resolved a condition where text on a PDF may display as an empty box for unicode characters. (DF-1931)
- Resolved a condition for Excel files where hyperlinked images would not output the link when EXCELMODE is not 'text'. (DF-1949)
- Resolved a condition for HTML files where using the IMAGEURL option would not enclose the URL within quotes. (DF-1957)
- Resolved a condition for PDFs when 'OCR_REORIENT_PAGES' is enabled would not reorient the pages. (DF-1933)
- Resolved a condition where a text file may be identified as WordPerfect. (DF-1959)
- Resolved a condition where extracting MSG files from PST files would fail. (DF-1950)

- Security: xpdf: CVE-2024-2971: patched: applied security patch to address this issue. (DF-1990)
- Security: xpdf: CVE-2024-3247: not-exploitable: issue is already mitigated; does not impact document filters. (DF-1994)
- Security: xpdf: CVE-2024-3248: patched: applied security patch to address this issue. (DF-1991)
- Security: xpdf: CVE-2024-3900: patched: applied security patch to address this issue. (DF-1992)

5.9 Document Filters 24.1.0 (Feb 2024)

5.9.1 Notice

iOS and Android Platform Deprecation

Per the [Hyland Community post](#) of December 1st, 2023, starting with the 24.1 release of Hyland Document Filters, the installers for the iOS and Android platforms will no longer be available for download. If needed, they can be requested from your first line of support for the 24.1 release.

Please note the following key points:

- **Discontinuation of Availability:** As of the 24.1.0 release, Hyland Document Filters for the Android and iOS platforms will no longer be available for download. Please reach out to your first line of support if the 24.1.0 installers of these platforms are needed.
- **Deprecation of Platforms:** Following the 24.1.0 release, support for the Android and iOS platforms will be deprecated, and future releases of Hyland Document Filters will no longer support these platforms.
- **Continued Support for Other Platforms:** Hyland Document Filters will continue to support other platforms, and we remain committed to providing a reliable and high-quality experience for our customers.

Solaris Platform Changes

The Solaris Sparc edition has been updated to utilize Solaris Developer Studio 12.6 for its build. This modification includes an update to the C++ runtime prerequisite, potentially necessitating the installation or integration of the Solaris Developer Studio runtime alongside your application.

Refer to the [Platform Requirements](#) for additional information.

5.9.2 Enhancements

- Introduced HD support for PDF XFA using the "PDF_XFA_ENABLED" option, defaulting to false. (DF-1263)
- Introduced option "OLM_DATE_FIELD" to set a comma separated list of fields containing message time in OLM files, with the order of the field names defining search priority. Defaults to empty, resulting in the first field found being used. (DF-1826)
- Introduced support for extracting comments from RAR files. (DF-1248)
- Introduced support for extracting comments from ZIP files. (DF-1250)
- Introduced support for extracting links to external workbooks within XLS files using the "SHOWHIDDEN_EXCEL_REFS" option, defaulting to OFF. (DF-1251)
- Introduced support for preservation of alpha channel (transparency) in BMP input. (DF-1738)

- Introduced support for preservation of alpha channel (transparency) in TIFF output when option "TIFF_BITSPERPIXEL" is '0' and option "TIFF_COMPRESSION" is 'NONE' or 'LZW'. (DF-1737)
- Introduced support for running Document Filters on Linux ARM64 for MUSL/Alpine. (DF-1522)
- Upgraded libtiff to 4.6.0. (DF-1673)
- Upgraded libwebp to 1.3.2. (DF-1674)
- Upgraded Xpdf to 4.04.03. (DF-1456)

5.9.3 Updates

- HD Mode: Resolved a condition for CGM files where the file may not render correctly. (DF-1550)
- HD Mode: Resolved a condition for DOCX files where a crash may occur when processing headers/footers with missing attributes. (DF-1701)
- HD Mode: Resolved a condition for HTML files where inline embedded images with an incomplete MIME type would not render. (DF-1742)
- HD Mode: Resolved a condition for PDFs with corrupt page size values, where an error would be given instead of setting a default page size. (DF-1500)
- HD Mode: Resolved a condition for SVG files that caused them to not render on Windows when the regional settings are set to a locale that does not use period and comma as their decimal and thousand separators. (DF-1733)
- HD Mode: Resolved a condition for Visio files where some .vdx and .vsdx files would render incorrectly with various non-"en_US" locales. (DF-1853)
- HD Mode: Resolved a condition for XPS files that caused them to not render on Windows when the regional settings are set to a locale that does not use period and comma as their decimal and thousand separators. (DF-1732)
- HD Mode: Resolved a condition where 16-bit PICT files may not render. (DF-1485)
- HD Mode: Resolved a condition where calling "IGR_Canvas_DrawImage" with an SVG on a page created by "IGR_Canvas_Blank_Page" causes a SIGSEGV. (DF-1878)
- HD Mode: Resolved a condition where certain MS Visio files may not render. (DF-1653)
- HD Mode: Resolved a condition where certain PICT files may not render. (DF-1485)
- HD Mode: Resolved a condition where long footnotes may overlap content of the document. (DF-1032)
- HD Mode: Resolved a condition where MSG files with 1000+ people in the recipients list may not render. (DF-1832)
- HD Mode: Resolved a condition where option "OCR_REORIENT_PAGES=ON" could result in blank pages in rendered output. (DF-1845)
- HD Mode: Resolved a condition where OS2 Bitmap files don't render. (DF-1484)
- HD Mode: Resolved a condition where output of PDF files may have the text incorrectly spaced. (DF-1528)
- HD Mode: Resolved a condition where some DICOM files may not render. (DF-1780)

- HD Mode: Resolved a condition where some embedded images within MSG files may not render. (DF-1849)
- HD Mode: Resolved a condition where the text of the "WATERMARK" option may be cut off. (DF-1884)
- Resolved a condition for HTML files where entity reference characters may not be processed correctly. (DF-1852)
- Resolved a condition for ODT files where "ENUMERATE_IMAGES=ON" may not work as expected. (DF-1580)
- Resolved a condition for PDF documents where having "PDF_PRESERVE_ORIGINAL=ON" may cause relative links within the document to not work. (DF-1703)
- Resolved a condition that may cause memory leaks when using the Python bindings. (DF-1889)
- Resolved a condition where .mbox files were not identified correctly. (DF-1846)
- Resolved a condition where an HTML file containing an SVG may be identified as an SVG. (DF-1736)
- Resolved a condition where converting a tabbed text file to classic HTML removed all of the tabs between the text. (DF-1888)
- Security: glib: CVE-2023-48161: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1825)
- Security: libtiff: CVE-2023-3164: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1882)
- Security: libtiff: CVE-2023-3576: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1706)
- Security: libtiff: CVE-2023-40745: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1705)
- Security: libtiff: CVE-2023-41175: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1704)
- Security: libtiff: CVE-2023-52356: patched: applied security patch to address this issue. (DF-1883)
- Security: libtiff: CVE-2023-6277: patched: applied security patch to address this issue. (DF-1830)
- Security: ODA: CVE-2023-5179: patched: applied security patch to address this issue. (DF-1786)
- Security: ODA: CVE-2023-5180: patched: applied security patch to address this issue. (DF-1859)
- Security: tinyxml: CVE-2023-34194: patched: applied security patch to address this issue. (DF-1855)
- Security: zlib: CVE-2023-45853: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1711)

5.10 Document Filters 23.3.0 (Sep 2023)

5.10.1 Notice

Per the [Hyland Community post](#) of July 18, 2023, starting with the 23.3.0 release of Hyland Document Filters, the installers for the Solaris Intel and HP-UX platforms will no longer be available for download. If needed, they can be requested from your first line of support for the 23.3.0 release.

Please note the following key points:

- **Discontinuation of Availability:** As of the 23.3.0 release, Hyland Document Filters for Solaris Intel and HP-UX platforms will no longer be available for download. Please reach out to your first line of support if the 23.3.0 installers of these platforms are needed.
- **Depreciation of Platforms:** Following the 23.3.0 release, support for the Solaris Intel and HP-UX platforms will be deprecated, and future releases of Hyland Document Filters will no longer support these platforms.
- **Continued Support for Other Platforms:** Hyland Document Filters will continue to support other platforms, and we remain committed to providing a reliable and high-quality experience for our customers.

5.10.2 Enhancements

- Improved symbol character to Unicode character conversions. (DF-1531)
- Introduced option "SPREADSHEET_SHRINKTOFIT_LIMIT" to control the lowest scale allowed when "SPREADSHEET_SHRINKTOFIT" is enabled, defaulting to 25%. (DF-1559)
- Introduced password support for ODF files using "IGR_Open_Ex" callbacks. (DF-1257)
- Introduced password support for PDF files using "IGR_Open_Ex" callbacks. (DF-1252)
- Introduced support for "OpenDocument Drawing" isys:subType metadata. (DF-1621)
- Introduced support for easily providing localization values for metadata names using "IGR_Open_Ex" callbacks for all modes. (DF-272)
- Introduced support for the use of the Document Filters NuGet package within Native C++ projects. (DF-1060)
- Upgraded bzip to 1.0.8. (DF-1615)
- Upgraded libjpeg to 9e. (DF-1616)
- Upgraded libpng to 1.6.40. (DF-1617)
- Upgraded libtiff to 4.5.1. (DF-1642)
- Upgraded libwebp to 1.3.1. (DF-1618)
- Upgraded LZMA SDK to 23.01. (DF-1629)
- Upgraded openjpeg to 2.5.0. (DF-1620)

- Upgraded wavpack to 5.6.0. (DF-1619)
- Upgraded zlib to 1.3.0. (DF-1643)

5.10.3 Updates

- HD Mode: Resolved a condition for DWG files where the first page is rendered incorrectly. (DF-1572)
- HD Mode: Resolved a condition for MSG files where a space character would be incorrectly included as part of a word's positional information. (DF-1602)
- HD Mode: Resolved a condition for PDFs where having incorrect mask data would cause embedded images to not render. (DF-1578)
- HD Mode: Resolved a condition for TIFFs where images with either the x or y DPI being 0 caused the image to not render. (DF-1614)
- HD Mode: Resolved a condition where 4 bit per pixel TIFF images without a palette and using MIN_IS_WHITE were not rendering correctly. (DF-1613)
- HD Mode: Resolved a condition where bitmaps would be promoted to 24-bit. (DF-1509)
- HD Mode: Resolved a condition where certain DOCX files may fail to render with the error message "Page limit exceeded" due to a paragraph-anchored graphic with text wrap forcing its anchor paragraph to the next page. (DF-1424)
- HD Mode: Resolved a condition where certain DOCX files may fail to render with the error message "Page limit exceeded" due to a table with multiple repeat header rows being split across pages. (DF-1573)
- HD Mode: Resolved a condition where certain DOCX files may not render in FreeBSD. (DF-1537)
- HD Mode: Resolved a condition where CGM files may not render. (DF-1549)
- HD Mode: Resolved a condition where converting an MSG file with embedded Excel objects to HTML may cause the objects to not render. (DF-1541)
- HD Mode: Resolved a condition where converting documents containing EMF/WMF images to PDFs may cause a segmentation fault. (DF-1561)
- HD Mode: Resolved a condition where CR/LF characters would be removed from the subject of an EML file. (DF-1597)
- HD Mode: Resolved a condition where DOC files converted to PNG may have the style of the title used for the first entry of a table of contents. (DF-1113)
- HD Mode: Resolved a condition where JPEGs would be promoted to 24-bit. (DF-1590)
- HD Mode: Resolved a condition where palletted PNGs would be promoted to 32-bit. (DF-1589)
- HD Mode: Resolved a condition where PDF redactions with PDF_PRESERVE_ORIGINAL may incorrectly scale, rotate, or mirror the content after successful redaction. (DF-1527)

- HD Mode: Resolved a condition where rendering images to a TIFF canvas would promote the image to 24-bit. (DF-1591)
- HD Mode: Resolved a condition where specific TIFFs using OJPEG compression would not render. (DF-1592)
- HD Mode: Resolved a condition where the conversion of an Excel file to .html would result in entire columns of values not visible in the .html output. (DF-1347)
- Resolved a condition where metadata for WebP files would not be extracted. (DF-1639)
- Resolved a condition where reported image sizes may be off by 1 due to pixel density stored as dots per centimeter. (DF-1012)
- Resolved a condition where the .NET API would incorrectly return "IGR_CANCELLED" instead of throwing an exception. (DF-1637)
- Security: 7-zip: CVE-2017-17969: not-exploitable: issue exists in desktop application, not in library; does not impact document filters. (DF-1624)
- Security: 7-zip: CVE-2018-10115: not-exploitable: issue exists in desktop application, not in library; does not impact document filters. (DF-1627)
- Security: 7-zip: CVE-2018-10172: not-exploitable: issue exists in desktop application, not in library; does not impact document filters. (DF-1623)
- Security: 7-zip: CVE-2018-5996: not-exploitable: issue exists in desktop application, not in library; does not impact document filters. (DF-1628)
- Security: giflib: CVE-2023-39742: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1650)
- Security: Leptonica: CVE-2020-36277: not-exploitable: unused code block; does not impact document filters. (DF-1569)
- Security: Leptonica: CVE-2020-36278: not-exploitable: unused code block; does not impact document filters. (DF-1569)
- Security: Leptonica: CVE-2020-36279: not-exploitable: unused code block; does not impact document filters. (DF-1569)
- Security: Leptonica: CVE-2020-36281: not-exploitable: unused code block; does not impact document filters. (DF-1569)
- Security: libtiff: CVE-2020-18768: not-exploitable: issue does not exist in version. (DF-1649)
- Security: libtiff: CVE-2022-40090: not-exploitable: issue does not exist in version. (DF-1648)
- Security: libtiff: CVE-2023-25433: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1575)
- Security: libtiff: CVE-2023-25434: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1564)

- Security: libtiff: CVE-2023-25435: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1570)
- Security: libtiff: CVE-2023-26965: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1563)
- Security: libtiff: CVE-2023-26966: patched: applied security patch to address this issue. (DF-1574)
- Security: libtiff: CVE-2023-2731: not-exploitable: issue does not exist in version. (DF-1542)
- Security: libtiff: CVE-2023-2908: patched: applied security patch to address this issue. (DF-1583)
- Security: libtiff: CVE-2023-30774: patched: applied security patch to address this issue. (DF-1544)
- Security: libtiff: CVE-2023-30775: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1543)
- Security: libtiff: CVE-2023-3316: not-exploitable: issue does not exist in version. (DF-1568)
- Security: libtiff: CVE-2023-3618: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1595)
- Security: libwebp: CVE-2023-1999: patched: applied security patch to address this issue. (DF-1577)
- Security: libwebp: CVE-2023-4863: patched: applied security patch to address this issue. (DF-1670)
- Security: unrar: CVE-2022-48579: not-exploitable: unused code block; does not impact document filters. (DF-1625)
- Security: Xpdf: CVE-2023-2662: not-exploitable: issue does not exist in version. (DF-1526)
- Security: Xpdf: CVE-2023-2663: not-exploitable: issue does not exist in version. (DF-1524)
- Security: Xpdf: CVE-2023-2664: not-exploitable: issue does not exist in version. (DF-1523)
- Security: Xpdf: CVE-2023-3044: not-exploitable: issue does not exist in version. (DF-1557)

5.11 Document Filters 23.2.0 (May 2023)

5.11.1 Enhancements

- Introduced callback parameter to `DocumentFilters.OpenExtractor()` and `Extractor.Open()` in .NET API to enable "IGR_Open_Ex" support. (DF-1458)
- Introduced callback parameter to `DocumentFilters.OpenExtractor()` and `Extractor.Open()` in Java API to enable "IGR_Open_Ex" support. (DF-1460)
- Introduced callback parameter to `DocumentFilters.OpenExtractor()` and `Extractor.Open()` in Python API to enable "IGR_Open_Ex" support. (DF-1459)
- Introduced expanded support of "isys:SubType" metadata for Microsoft Excel documents that can help distinguish between the different types of Microsoft Excel documents. (DF-1201)
- Introduced HD and metadata support for Canon RAW Version 2 (CR2) files. (DF-1469)
- Introduced HD and metadata support for Canon RAW Version 3 (CR3) files. (DF-1482)
- Introduced option "TIFF_SUSPECT_API_RATIO_ERROR" to allow disabling the error that is given when X and Y DPIs are not equal and their ratio is below the threshold, defaulting to ON. (DF-1479)
- Introduced password support for DOC files using "IGR_Open_Ex" callbacks. (DF-1253)
- Introduced password support for DOCX files using "IGR_Open_Ex" callbacks. (DF-1244)
- Introduced password support for PPT files using "IGR_Open_Ex" callbacks. (DF-1254)
- Introduced password support for PPTX files using "IGR_Open_Ex" callbacks. (DF-1245)
- Introduced password support for XLS files using "IGR_Open_Ex" callbacks. (DF-1255)
- Introduced password support for XLSX files using "IGR_Open_Ex" callbacks. (DF-1246)
- Introduced password support for ZIP files using "IGR_Open_Ex" callbacks. (DF-1242)

5.11.2 Updates

- HD Mode: Resolved a condition for Apple Pages where bulleted lists may render an incorrect bullet symbol when converting to HTML. (DF-636)
- HD Mode: Resolved a condition in RTF files where Unicode characters stored as surrogate pairs did not render correctly. (DF-1462)
- HD Mode: Resolved a condition when converting EML to HTML where email address lists that don't contain spacing would result in incorrect line breaks. (DF-1454)
- HD Mode: Resolved a condition where certain TIFF documents may not render. (DF-1477)
- HD Mode: Resolved a condition where missing TIFF data may cause the image to not render. (DF-1501)
- HD Mode: Resolved a condition where numbered lists may display as bulleted lists in HTML. (DF-424)

- HD Mode: Resolved a condition where Old-Style JPEG compressed TIFF files may not render. (DF-1476)
- HD Mode: Resolved a condition where Photoshop images containing additional channels may not render. (DF-1445)
- HD Mode: Resolved a condition where Word 6 files may not render the entire document. (DF-1452)
- HD Mode: Resolved a condition where word-breaking when the “word” is longer than the available space would cause incorrect spacing. (DF-1470)
- Resolved a condition for EML files where attachment names may be truncated if they are not stored in quotes. (DF-1471)
- Resolved a condition for MS Excel 2007+ documents where the subtype may not be indicated. (DF-961)
- Resolved a condition for RAR archives where password decryption may fail if the subfiles require different passwords. (DF-1487)
- Resolved a condition where a DOCX document could consume an excess amount of memory. (DF-1474)
- Resolved a condition where calling IGR_Extract_Subfile_Stream after calling IGR_Get_Subfiles_Enumerator would result in a crash with subsequent calls. (DF-1499)
- Resolved a condition where error codes would be incorrectly replaced with the generic error code 4 in the Python bindings. (DF-1498)
- Resolved a condition where images may be misidentified as PICT. (DF-1439)
- Security: freetype: CVE-2023-2004: not-exploitable: issue does not exist in version. (DF-1512)
- Security: libtiff: CVE-2022-4645: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1472)
- Security: libtiff: CVE-2023-1916: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1506)
- Security: ODA: CVE-2023-22669: patched: applied security patch to address this issue. (DF-1514)
- Security: ODA: CVE-2023-22670: patched: applied security patch to address this issue. (DF-1510)
- Security: ODA: CVE-2023-26495: patched: applied security patch to address this issue. (DF-1503)
- Security: protobuf: CVE-2022-48468: patched: applied security patch to address this issue. (DF-1511)
- Security: snappy: CVE-2023-28115: not-exploitable: unused code block; does not impact document filters. (DF-1488)

5.12 Document Filters 23.1.0 (Feb 2023)

5.12.1 Enhancements

- Introduced Attachment extraction for ICS/VCAL calendar files. (DF-1131)
- Introduced HD support for Adobe Illustrator files. (DF-1266)
- Introduced HD support for Apple Numbers 2009 files. (DF-1205)
- Introduced HD support for ICS/VCAL calendar files. (DF-1131)
- Introduced HD support for Postscript files. (DF-1264)
- Introduced option "HTML_MAX_NESTED_TABLES" to control the maximum number of nested tables allowed when loading HTML, defaulting to 150. (DF-1398)
- Introduced option "PDF_TRANSPARENT_BACKGROUND" to provide the ability to render PDF files with a transparent background, defaulting to off. (DF-343)
- Introduced password support for 7-Zip using "IGR_Open_Ex" callbacks. (DF-1241)
- Introduced password support for RAR using "IGR_Open_Ex" callbacks. (DF-1240)
- Introduced support for cancellation when opening an MS Excel document using "IGR_Open_Ex" callbacks. (DF-1239)
- Introduced support for multi-event ICS files where each event is extracted as a subfile. (DF-1130)

5.12.2 Updates

- HD Mode: Resolved a condition for Apple Keynote 2009 where text vertical alignment may be incorrect. (DF-1316)
- HD Mode: Resolved a condition for converting an HTML file with an invalid CSS value for "line-height" to an image would cause the height to be set to 0. (DF-416)
- HD Mode: Resolved a condition for DOCX files where font styling may not be applied to hyperlinks. (DF-1342)
- HD Mode: Resolved a condition for MS Excel 2007+ where drawings may not render. (DF-159)
- HD Mode: Resolved a condition for MSG files when outputted as HTML using the HDHTML_WORD_RUNS=on setting would put each word into a separate DIV. (DF-1054)
- HD Mode: Resolved a condition for PDFs where a segmentation fault may occur after rendering a corrupted document. (DF-1396)
- HD Mode: Resolved a condition for PDFs where the Words collection may be incomplete. (DF-1356)
- HD Mode: Resolved a condition for when TIFF files containing incorrect metadata may result in the image appearing black. (DF-1412)
- HD Mode: Resolved a condition in Apple iWork 09 files where multi-column textboxes could be missing text. (DF-1325)

- HD Mode: Resolved a condition where converting a PDF to TIFF may not render correctly when the PDF is missing a required Length property. (DF-1430)
- HD Mode: Resolved a condition where converting an MSG file to PNG may not succeed. (DF-1426)
- HD Mode: Resolved a condition where converting from a DOC file to a PNG could cause a segfault. (DF-1283)
- HD Mode: Resolved a condition where converting from an HWP file to PNG could cause a memory leak. (DF-1287)
- HD Mode: Resolved a condition where converting MSG files to HD HTML may result in the document having a black background. (DF-1366)
- HD Mode: Resolved a condition where documents may produce the error message "Page limit exceeded" due to a graphic in the first paragraph exceeding the maximum page size. (DF-1411)
- HD Mode: Resolved a condition where images with transparency would incorrectly respect HDHTML_IMAGE_FORMAT when the user provided format does not have transparency support. Images with transparency now revert to the default of PNG if the user provided format does not support transparency. (DF-1429)
- HD Mode: Resolved a condition where JPEG2000 files without the JPX header may not render correctly. (DF-250)
- HD Mode: Resolved a condition where Shapes in Apple iWork 2009 documents would appear with rounded instead of sharp edges. (DF-1323)
- HD Mode: Resolved a condition where TIFFs using JPEG compression with an invalid sampling were resulting in white documents appearing pink. (DF-1365)
- HD Mode: Resolved a condition where transparent text on a PDF may display as white. (DF-1055)
- Resolved a condition for EML files where attachments may be incorrectly identified as being corrupt files. (DF-1367)
- Resolved a condition for MS Access files where Attachment List fields were not being enumerated and extracted. (DF-1122)
- Resolved a condition for MS PowerPoint 2018 where modern comments are not extracted. (DF-1446)
- Resolved a condition for MS PowerPoint where extracted comments would not display the commenter's name when SHOWHIDDEN is enabled. (DF-662)
- Resolved a condition for PDFs where hyperlinks incorrectly reported their destination locations on the document. (DF-1438)
- Resolved a condition for TIFF files where specifying the same file for both input and output will cause an Access Violation. (DF-1443)
- Resolved a condition for VCAL files where some fields may be missing or have been incorrectly formatted. (DF-904)
- Resolved a condition for Win16 based DLL and EXE files where the file type was not returned. (DF-1368)
- Resolved a condition in Apple Numbers which resulted in a segmentation fault. (DF-1447)
- Resolved a condition on Linux where exceptions thrown from custom IGRStream implementations would terminate the process instead of marshaling the exception. (DF-1391)
- Resolved a condition where an MSG file with tens of thousands of items may result in a stack-overflow. (DF-1437)

- Resolved a condition where EML files extracted from MBOX/Sendmail could have the wrong line-endings. (DF-1125)
- Resolved a condition where extracting empty DOCX attachments would throw an error. (DF-1431)
- Resolved a condition where large Visio files would display HTML in text-mode. (DF-1370)
- Resolved a condition where parsing HTML with a nested table depth greater than 11 results in an error. (DF-900)
- Resolved a condition where setting PSTMESSAGE TYPE=EML may cause the extracted EML file to contain invalid multi-part sections. (DF-1115)
- Resolved a condition where setting PSTMESSAGE TYPE=EML would cause the extracted EML file to not contain the sub-file. (DF-1115)
- Security: Leptonica: CVE-2022-38266: not-exploitable: unused code block; does not impact document filters. (DF-1423)
- Security: libtiff: CVE-2015-1239: not-exploitable: issue fixed in version OpenJPEG 2.1.0-2+deb8u4. (DF-1374)
- Security: libtiff: CVE-2015-7313: not-exploitable: issue was fixed in libtiff 4.0.7. (DF-1375)
- Security: libtiff: CVE-2022-3970: not-exploitable: unused code block; does not impact document filters. (DF-1349)
- Security: libtiff: CVE-2022-48281: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1414)
- Security: libtiff: CVE-2023-0795: not-exploitable: unused code block; does not impact document filters. (DF-1448)
- Security: libtiff: CVE-2023-0796: not-exploitable: unused code block; does not impact document filters. (DF-1448)
- Security: libtiff: CVE-2023-0797: not-exploitable: unused code block; does not impact document filters. (DF-1448)
- Security: libtiff: CVE-2023-0798: not-exploitable: unused code block; does not impact document filters. (DF-1448)
- Security: libtiff: CVE-2023-0799: not-exploitable: unused code block; does not impact document filters. (DF-1448)
- Security: libtiff: CVE-2023-0800: not-exploitable: unused code block; does not impact document filters. (DF-1448)
- Security: libtiff: CVE-2023-0801: not-exploitable: unused code block; does not impact document filters. (DF-1448)
- Security: libtiff: CVE-2023-0802: not-exploitable: unused code block; does not impact document filters. (DF-1448)
- Security: libtiff: CVE-2023-0803: not-exploitable: unused code block; does not impact document filters. (DF-1448)
- Security: libtiff: CVE-2023-0804: not-exploitable: unused code block; does not impact document filters. (DF-1448)
- Security: Xpdf: CVE-2022-24106: patched: applied security patch to address this issue. (DF-1428)
- Security: Xpdf: CVE-2022-24107: patched: applied security patch to address this issue. (DF-1427)
- Security: Xpdf: CVE-2022-38171: patched: applied security patch to address this issue. (DF-1361)

5.13 Document Filters 22.4.0 (Nov 2022)

5.13.1 Enhancements

- Introduced HD support for Apple Keynote 2009 files. (DF-1022)
- Introduced a default font color of black when rendering as HTML. (DF-1210)
- Introduced option "JPEG_REORIENT_PAGES" to control if the stored orientation should be honored when loading JPEG files, defaulting to OFF. (DF-1222)
- Introduced option "TEXT_PAGE_CHARS_PER_LINE". When given a value greater than 0, this will indicate the maximum number of characters per-line. (DF-1069)
- Introduced option "TEXT_PAGE_LINES_PER_PAGE". This will indicate the maximum number of lines per-page. (DF-1069)
- Introduced option "TEXT_PAGE_WORD_WRAP". When enabled, lines will be word wrapped. (DF-1069)

5.13.2 Updates

- HD Mode: Resolved a condition for Apple Pages 2009 where pie chart explosion did not render correctly. (DF-1160)
- HD Mode: Resolved a condition for MS Excel files where images in the header and footer may not render. (DF-1107)
- HD Mode: Resolved a condition for MS Word files where a drop cap is positioned incorrectly. (DF-476)
- HD Mode: Resolved a condition for MS Word files where lines with line spacing less than 1.0 would not be placed correctly. (DF-688)
- HD Mode: Resolved a condition for Office 2007+ where pie chart explosion did not render correctly. (DF-1160)
- HD Mode: Resolved a condition for RTF files where characters may be missing from output. (DF-1291)
- HD Mode: Resolved a condition for RTF files where embedded images may not render. (DF-1167)
- HD Mode: Resolved a condition for RTF files where watermarks partially obscured part of the text. (DF-1066)
- HD Mode: Resolved a condition for RTF files where watermarks were not centered correctly. (DF-1066)
- HD Mode: Resolved a condition for TIFF files where invalid planar config values may cause them to not render. (DF-1221)
- HD Mode: Resolved a condition for XPS files where text context may not be aligned. (DF-691)
- HD Mode: Resolved a condition where Charts and other graphics were not rendered with SHOW_HIDDEN=on in MS Office documents if the paragraph they were contained in was marked as hidden. (DF-1123)
- HD Mode: Resolved a condition where MS Word 2007+ documents with a very large number of text runs sharing the same numbering style caused very high memory usage. (DF-1228)
- HD Mode: Resolved a condition where certain OXPS files would not render. (DF-1229)

- HD Mode: Resolved a condition where converting PICT to PDF without explicitly setting the font name would cause a crash. (DF-1271)
- HD Mode: Resolved a condition where using TEXT_FALLBACK option could result in the text rendering past the margins and off the bottom of the page. (DF-1232)
- Resolved a condition for EML files where messages with a large amount of X headers may fail to identify. (DF-921)
- Resolved a condition for MS Access files where 2016 versioned files may fail to extract. (DF-1224)
- Resolved a condition for MS Excel files where header/footer content is not extracted when content is left justified. (DF-681)
- Resolved a condition for PDF files containing invalid shading patterns which may cause memory leaks. (DF-1289)
- Resolved a condition for PDF files containing links which may cause memory leaks. (DF-1285)
- Resolved a condition for RPM files where archives compressed with XZ would fail to extract. (DF-1225)
- Resolved a condition for Word Perfect files that resulted in a crash when processing documents with nested frames. (DF-1272)
- Resolved a condition that caused a memory leak in Apple Pages 2009 when converting to PNG. (DF-1284)
- Resolved a condition where ICU may segfault in low memory environments. (DF-1309)
- Resolved a condition where MS Office XML files may be misidentified as zip files. (DF-1338)
- Resolved a condition where MS PowerPoint 2003 files would display bullets instead of numbers. (DF-659)
- Resolved a condition where TIFF files would fail to decode if the BitPerSample tag was missing. (DF-1236)
- Resolved a condition where a corrupt PDF file could cause a heap-buffer-overflow and possibly result in a crash of the application. (DF-1290)
- Resolved a condition where a corrupt PNG file could cause a heap-buffer-overflow and possibly result in a crash of the application. (DF-1288)
- Resolved a condition where generating classic HTML could cause textbox backgrounds to be incorrectly rotated. (DF-835)
- Resolved a condition where the incorrect DPI was asserted for TIFF, PNG, or JPEG with less than 48 DPI. (DF-1336)
- Resolved a regression where HDHTML_IMAGE_FORMAT was not honored for HTML5 output causing all inline images to be written as PNG. (DF-1308)
- Security: libtiff: CVE-2022-1354: patched: applied security patch to address this issue. (DF-1234)
- Security: libtiff: CVE-2022-1355: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1233)
- Security: libtiff: CVE-2022-2953: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1219)

- Security: libtiff: CVE-2022-3570: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1326)
- Security: libtiff: CVE-2022-3597: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1327)
- Security: libtiff: CVE-2022-3598: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1329)
- Security: libtiff: CVE-2022-3599: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1330)
- Security: libtiff: CVE-2022-3626: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1331)
- Security: libtiff: CVE-2022-3627: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1332)

5.14 Document Filters 22.3.0 (Aug 2022)

5.14.1 Enhancements

- Improved support for redaction of PDFs when using PDF_PRESERVE_ORIGINAL option. (DF-1104)
- Introduced HD support for Apple Pages 2009 files. (DF-806)
- Introduced identification and extraction support for self-extracting 7Zip EXE files. (DF-1136)
- Introduced identification and extraction support for self-extracting RAR EXE files. (DF-1139)
- Introduced identification and extraction support for self-extracting ZIP EXE files. (DF-1137)
- Introduced identification and metadata support for Adobe Photoshop Large Document Files. (DF-118)
- Introduced option "ARCHIVE_IGNORE_CHECKSUM" to allow for data from corrupt or invalid archives to still be extracted. (DF-765)
- Introduced option "TEXT_INLINE_LINKS". When enabled, text mode will output the URL at the location it was found, rather than after the page content. This is available for Word 97-2003, Word 2007+, Excel 97-2003, Excel 2007+, PowerPoint 97-2003, PowerPoint 2007+. (DF-1072)
- Introduced options "XML_FORMATTED", "XML_FORMATTED_WITH_ATTRIBUTES" and "XML_FORMATTED_WITH_NODENAMES" that allow for the toggling on and off of XML tags for text output of XML files. (DF-770)
- Introduced support for AutoCAD 2018+. These documents are supported on all platforms, except the following: AIX, FreeBSD 9, HPUX, Solaris, Linux-PPC, Linux-ARM32. (DF-92)
- Introduced support for AutoCloseable on Java objects allowing for try-with-resource blocks for easier object management. (DF-1052)
- Introduced support for Named Destination annotations, which allow for sections to be marked and then referenced by Hyperlink annotations. (DF-997)
- Introduced support for text extraction in Windows Metafile and Windows Enhanced Metafile formats. (DF-1073)

5.14.2 Updates

- HD Mode: Resolved a condition for HTML files where `<table>` centered in `<div>` may not be positioned correctly. (DF-437)
- HD Mode: Resolved a condition for MS Word files where drawings and highlights created using a touch sensitive pen may not render. (DF-1124)
- HD Mode: Resolved a condition for MS Word files where floating objects anchored to a paragraph with a page break may be incorrectly moved to the following page. (DF-566)
- HD Mode: Resolved a condition for MS Word files where floating tables may incorrectly overlap the footer of a page. (DF-1048)

- HD Mode: Resolved a condition for MS Word files where floating tables may incorrectly overlap the header when spacing pages. (DF-1050)
- HD Mode: Resolved a condition for MSG Files where messages containing multiple HTML bodies may show the incorrect version. (DF-1112)
- HD Mode: Resolved a condition for Office documents where bullet characters from Symbol fonts may be incorrectly positioned in right-to-left based documents. (DF-470)
- HD Mode: Resolved a condition for RTF documents where clipping on embedded images may not be correctly applied. (DF-1005)
- HD Mode: Resolved a condition of Office files where numbered lists may force line breaks in the middle of the number when inserted into narrow table cells. (DF-1050)
- HD Mode: Resolved a condition that prevented some images from displaying in MS Word documents after they had been modified by third party applications. (DF-886)
- HD Mode: Resolved a condition that would cause chapters in MS Word documents to be incorrectly rendered. (DF-1121)
- HD Mode: Resolved a condition when partially redacting an embedded image in a PDF does not permanently remove the underlying image data of the embedded object. (DF-1102)
- HD Mode: Resolved a condition where Apple Pages/Keynote/Numbers 2009 files converted to Apple Pages/Keynote/Numbers 2010+ do not display embedded images. (DF-1019)
- HD Mode: Resolved a condition where bar graphs with logarithmic scaling would not display data labels correctly. (DF-1058)
- HD Mode: Resolved a condition where fields in textboxes inside headers/footers would not have their content updated before being rendered. (DF-1056)
- HD Mode: Resolved a condition where right to left text was not properly aligned in Microsoft Office documents. (DF-481)
- HD Mode: Resolved a condition with Charts in Office 2007+ not respecting the reverse display order property. (DF-643)
- Resolved a condition for Hangul Files where sub-files may not extract correctly. (DF-1120)
- Resolved a condition for WordPerfect for Mac 1 where files may fail to identify or fail to extract. (DF-1151)
- Resolved a condition for the SHOW_HIDDEN option in text mode where it was not defaulting to OFF when expected. (DF-948)
- Resolved a condition in MS Office documents with comments that resulted in a crash. (DF-1075)
- Resolved a condition that was preventing password protected documents from getting reported as encrypted when using the extractor open API method. (DF-928)
- Resolved a condition where EML files were being incorrectly identified as text files. (DF-1106)

- Resolved a condition where MS Excel binary files were not getting identified correctly. (DF-881)
- Resolved a condition where converting a PDF document to text may cause space characters to be lost. (DF-832)
- Resolved a condition where the file type of MSG files missing their stored MessageType are not able to be identified. (DF-1044)
- Security: PDFium: CVE-2022-0306: not-exploitable: issue does not exist in version. (DF-1153)
- Security: libtiff: CVE-2022-2867: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1164)
- Security: libtiff: CVE-2022-2868: not-exploitable: issue exists in cli tool, not in library; does not impact document filters. (DF-1163)
- Security: wavpack: CVE-2022-2476: not-exploitable: issue does not exist in version. (DF-1119)
- Security: zlib: CVE-2022-37434: not-exploitable: issue does not exist in version. (DF-1156)

5.15 Document Filters 22.2.0 (May 2022)

5.15.1 Enhancements

- Introduced new "isys:subType" metadata type to OpenOffice documents that can help distinguish between the different classes of documents. (DF-801)
- Introduced new Python bindings that support Python 3.7+. The new bindings move from requiring Python version-specific layer to using ctypes to call into the Document Filters API, this makes them more portable, and will work against future versions of Python without change. See the implementation guide for details. (DF-803)
- Introduced option "HTML_RASTER_TEXT" which controls how HTML is rendered. When enabled, the text is flattened into the output and allows for z-index adjustments between other HTML elements. (DF-771)
- Introduced option PDF_XFA_ERROR. When enabled, dynamic XFA forms will return an error, rather than rendering the "please upgrade your viewer" page. (DF-775)
- Introduced support for Aztec barcode generation via Annotation API. (DF-807)
- Introduced support for decoding multi-frame GIFs. Pass the option "GIF_DECODE_ALL_FRAMES=on" when opening the file. Each frame will be available as a page. (DF-776)
- Introduced support for international numbering for MS Word numbered lists. (DF-48)
- Introduced support for international numbering for Open Office numbered lists. (DF-8)

5.15.2 Updates

- Added new option "TIFF_IGNORE_DECODE_ERR" option to allow partially corrupt or malformed documents to render. Defaults to off. (DF-861)
- HD Mode: Resolved a condition for EML/MSG/HTML files where content relying on media selectors may not render as expected. (DF-939)
- HD Mode: Resolved a condition for EML/MSG/HTML files where tables containing long sequences without word breaks may render wider than expected. (DF-868)
- HD Mode: Resolved a condition for EML/MSG/HTML where images larger than a page may be clipped. (DF-959)
- HD Mode: Resolved a condition for EML/MSG/MHT files where large images may cause blank pages to be created. (DF-1007)
- HD Mode: Resolved a condition for HD output where original DPI may not be copied into generated files when GRAPHIC_DPI=AUTO is used. (DF-855)
- HD Mode: Resolved a condition for MS Office charts where axis labels may incorrectly be rendered when they are disabled in the document. (DF-592)
- HD Mode: Resolved a condition for MS Office charts where labels may appear in the wrong position. (DF-633)

- HD Mode: Resolved a condition for MS Word files where borders may not render correctly on merged tables. (DF-895)
- HD Mode: Resolved a condition for MS Word files where empty shapes may not take their reserved space. (DF-943)
- HD Mode: Resolved a condition for MS Word files where floating elements within tables may incorrectly impact content outside of the table. (DF-876)
- HD Mode: Resolved a condition for MS Word files where inline images may not be correctly position based on parent paragraphs justification. (DF-853)
- HD Mode: Resolved a condition for MS Word files where text boxes may be incorrectly rendered with a border. (DF-936)
- HD Mode: Resolved a condition for PDF files where malformed documents may cause blank pages to be rendered when used with PDF_PRESERVE_ORIGINAL option. (DF-857)
- HD Mode: Resolved a condition for TIFF files containing JPEG compressed pages with mismatched metadata failing to render. (DF-1033)
- HD Mode: Resolved a condition for TIFF images where files with dimensions greater than 65,535 could fail to render. (DF-854)
- HD Mode: Resolved a condition of MS Office charts where charts may render smaller than expected. (DF-849)
- HD Mode: Resolved a condition of MS Word files where floating textboxes may appear on the wrong page. (DF-598)
- Resolved a condition for EML files where files starting with a large amount of non-standard headers may not be identified. (DF-898)
- Resolved a condition for EML/MSG/HTML where content encoded with "gbk" charset may not correctly decode. (DF-880)
- Resolved a condition for EMLX files where files may fail to identify correctly. (DF-848)
- Resolved a condition for MS Access files where password protected files could return a generic error code. (DF-158)
- Resolved a condition for MS Excel files where older versions may incorrectly format numbers as dates. (DF-1018)
- Resolved a condition for Office 2007 files where hidden content may be shown when SHOWHIDDEN=on was not set. (DF-905)
- Resolved a condition for PDF files where documents containing a large number of OCPProperty objects may take long processing times. (DF-925)
- Resolved a condition for PNG files where images with a transparent background may fail to OCR. (DF-945)
- Resolved a condition for RAW images where files may be misidentified as multi-byte text. (DF-569)
- Resolved a condition for RTF files where metadata may be truncated. (DF-915)
- Resolved a condition for SVG files where text content may not be extracted. (DF-877)
- Resolved a condition for Unsupported Binary files where they may misidentify as Uniplex. (DF-950)

- Resolved a condition for Unsupported Binary files where they may misidentify as UTF-16 text files. (DF-874)
- Resolved a condition for Unsupported Binary files where they may misidentify as Word Perfect 4.2. (DF-858)
- Resolved a condition for Unsupported Binary files where they may misidentify as XYWrite. (DF-890)
- Resolved a condition for Visio files where certain subfile types may fail to extract. (DF-885)
- Resolved a condition of MS Excel files where system formatted date/times were not formatted using the provided LCID. (DF-196)
- Resolved a condition where unsupported binary files may misidentify as MS Word. (DF-847)
- Security: freetype2: CVE-2022-27404: patched: issue exists from api misuse, not based on content. (DF-1013)
- Security: freetype2: CVE-2022-27405: patched: issue exists from api misuse, not based on content. (DF-1013)
- Security: freetype2: CVE-2022-27406: patched: issue exists from api misuse, not based on content. (DF-1013)
- Security: giflib: CVE-2022-28506: not-exploitable: this issue exists in a command line tool, not the library. (DF-1016)
- Security: libtiff: CVE-2022-0561: not-exploitable: issue does not exist in version. (DF-960)
- Security: libtiff: CVE-2022-0562: patched: applied security patch to address this issue. (DF-960)
- Security: libtiff: CVE-2022-1622: not-exploitable: issue does not exist in version. (DF-1041)
- Security: libtiff: CVE-2022-1623: not-exploitable: issue does not exist in version. (DF-1041)
- Update to zlib 1.2.12. (DF-788)

5.16 Document Filters 22.1.0 (Feb 2022)

5.16.1 Enhancements

- Introduced improved font fallback and discovery when mapping fonts on systems where the original is not available, or when fonts have missing glyphs. (DF-316)
- Introduced support for comments and annotation extraction for MS Excel 2007+ via the Annotation API in HD Mode. (DF-778)
- Introduced support for comments and annotation extraction for MS Excel 97-2003 via the Annotation API in HD Mode. (DF-789)
- Introduced support for comments and annotation extraction for MS PowerPoint 2007+ via the Annotation API in HD Mode. (DF-777)
- Introduced support for comments and annotation extraction for MS PowerPoint 97-2003 via the Annotation API in HD Mode. (DF-760)
- Introduced support for comments and annotation extraction for MS Word 2007 via the Annotation API in HD Mode. (DF-808)
- Introduced support for comments and annotation extraction for MS Word 97-2003 via the Annotation API in HD Mode. (DF-754)
- Introduced support for MacOS on Apple's M1 architecture. (DF-782)

5.16.2 Updates

- HD Mode: Resolved a condition for Hangul files where certain files may fail to render with "Page Limit Exceeded" error. (DF-919)
- HD Mode: Resolved a condition for MS Excel files where borders on pictures may not render. (DF-439)
- HD Mode: Resolved a condition for MS Excel files where cell backgrounds may be incorrectly rendered as black. (DF-958)
- HD Mode: Resolved a condition for MS Excel files where certain merged cells could cause a sigsegu. (DF-938)
- HD Mode: Resolved a condition for MS Excel where chart titles or labels maybe clipped. (DF-899)
- HD Mode: Resolved a condition for MS PowerPoint files where embedded WMF images could be cropped. (DF-916)
- HD Mode: Resolved a condition for MS PowerPoint where certain files may cause a sigsegu. (DF-917)
- HD Mode: Resolved a condition for MS Word files where floating elements may appear in the wrong position or page. (DF-654)
- HD Mode: Resolved a condition for MS Word where certain files may cause a sigsegu. (DF-891)
- HD Mode: Resolved a condition for RTF files where line shapes were not rendered. (DF-668)

- HD Mode: Resolved a condition for XPS files where watermarks/wordart may not render correctly when converting to PDF. (DF-565)
- Resolved a condition for 7zip files where certain archives created with PPMd compression may cause a sigsegv on Linux Intel-64. (DF-933)
- Resolved a condition for Adobe Flash/SWF where corrupt files could cause the program to hang. (DF-920)
- Resolved a condition for EML files where attachments without a Content-Type tag may not extract. (DF-884)
- Resolved a condition for MS Excel files where a sigsegv may occur on MacOS when fonts have not been configured, or fallbacks are not found. (DF-903)
- Resolved a condition for MS Excel files where certain numbers expressed as fractions may cause the document to take longer to process. (DF-954)
- Resolved a condition for MS Excel files where dates formatted in Korean may not display correctly. (DF-897)
- Resolved a condition for MS Office files where documents created in third-party applications may identify as ZIP rather than their actual format. (DF-924)
- Resolved a condition for MS Office files where metadata from documents created in third-party applications may not extract. (DF-889)
- Resolved a condition for MS Word files where non-compliant DOCX files may cause a crash. (DF-865)
- Resolved a condition for PDF files where embedded JBIG2 images may cause a sigsegv on Linux. (DF-845)
- Resolved a condition for PDF files where word spacing may not appear correctly. (DF-937)
- Resolved a condition for PST files where extracted MSG messages may not show the recipients correctly when viewed in Outlook. (DF-956)
- Resolved a condition for Text files where certain files may be incorrectly identified as WordPerfect for Mac. (DF-935)
- Resolved a condition for the API where option string values were case sensitive. (DF-957)
- Resolved a condition in ICU where certain low-memory conditions could cause a sigsegv. (DF-843)

5.17 Document Filters 21.11.0 (Nov 2021)

5.17.1 Enhancements

- Added FreeBSD 13 to the supported platforms list. (DF-799)
- Added Windows Server 2022 to the supported platforms list. (DF-800)
- Introduced a new C API for modifying Bookmarks in PDF output. (DF-140)
- Introduced a new COM API for Hyperlinks, Annotations, RenderPageProperties and BlankPage. (DF-505)
- Introduced a new Object API for modifying Bookmarks in PDF output. (DF-795)
- Introduced PDF_SET_OWNER_PASSWORD, PDF_SET_USER_PASSWORD and PDF_ENCRYPT options for creating password protected PDF files. (DF-524)

5.17.2 Updates

- HD Mode: Resolved a condition for Apple Pages files where files may fail to render. (DF-902)
- HD Mode: Resolved a condition for DGN files where a corrupt file may cause a crash. (DF-846)
- HD Mode: Resolved a condition for DGN files where corrupt files may cause a memory leak. (DF-883)
- HD Mode: Resolved a condition for DGN files where corrupt files may hang. (DF-926)
- HD Mode: Resolved a condition for HTML files where padding or margins may be incorrectly calculated. (DF-673)
- HD Mode: Resolved a condition for HTML files where Unicode characters above 0xFFFF would not render. (DF-879)
- HD Mode: Resolved a condition for ICO files where not all icons would be extracted. (DF-856)
- HD Mode: Resolved a condition for JPEG files where truncated images may fail to open. (DF-869)
- HD Mode: Resolved a condition for MS Office files where charts axis and legend may not render in the correct position. (DF-859)
- HD Mode: Resolved a condition for MS PowerPoint files where documents containing corrupt or missing hyperlink information may fail. (DF-922)
- HD Mode: Resolved a condition for MS Word files where floating graphics in headers/footers with negative offsets may render in the wrong position. (DF-669)
- HD Mode: Resolved a condition for MS Word files where footnote/endnotes may have the incorrect indent. (DF-576)
- HD Mode: Resolved a condition for MS Word files where paragraphs containing consecutive line breaks may be truncated. (DF-867)
- HD Mode: Resolved a condition for MS Word files where table containing negative offsets may render in the wrong position. (DF-378)
- HD Mode: Resolved a condition for MSG files where certain messages containing compressed rtf may render truncated. (DF-878)

- HD Mode: Resolved a condition for PCX files where corrupt files may cause a crash. (DF-578)
- HD Mode: Resolved a condition for PDF files where annotations may report in the wrong position for pages containing a crop box. (DF-906)
- HD Mode: Resolved a condition for PDF files where documents containing legacy embedded fonts may not render correctly. (DF-665)
- HD Mode: Resolved a condition for PDF files where files containing incomplete Japanese font references may fail to render. (DF-204)
- HD Mode: Resolved a condition for the API where long running calls to IGR_Close_Canvas could block new canvases from being created. (DF-830)
- HD Mode: Resolved a condition for the GetPageAttribute API where incomplete information maybe returned for a page. (DF-577)
- HD Mode: Resolved a condition for TIFF files where non-compliant images compressed with JPEG compression and an incorrect bit-depth may fail to render. (DF-955)
- HD Mode: Resolved a condition where output images larger than 65,536 pixels tall may fail to compress when using TIFF_COMPRESSION=JPEG. (DF-918)
- Patched Tinyxml2 used by ODA in ISYSautocad addressing CVE-2021-42260. (DF-871)
- Resolved a condition for MS Media formats where truncated files may not output metadata in text mode. (DF-913)
- Resolved a condition for MS Word files where hyperlinks extracted when SHOWHIDDEN is enabled may cause a crash for certain documents. (DF-641)
- Resolved a condition for MS Word files where old versions may not correctly detect the files code page. (DF-683)
- Resolved a condition for PDF files where documents with a corrupt xref table may fail to open. (DF-949)
- Resolved a condition for RTF files where specifying BODY_ONLY would still include metadata. (DF-644)
- Resolved a condition for Text Files where files may be incorrectly identified as FITS image. (DF-952)
- Resolved a condition for ZIP files where split archives may not extract. (DF-946)
- Resolved a condition of MS Excel where cells containing the error value "#N/A" may incorrectly render as "#####". (DF-596)
- Resolved a condition to MS PowerPoint files where files containing corrupt or incomplete shape information may fail to extract in text mode. (DF-944)
- Resolved a condition where files may be incorrectly identified as Lotus Notes (NSF). (DF-838)
- Updated default font locations to include user fonts on Windows. (DF-626)

5.18 Document Filters 21.8.0 (Aug 2021)

5.18.1 Enhancements

- Introduces new API `IGR_Get_Page_Pixels` for accessing low-level pixel data of a page. (DF-519)

5.18.2 Updates

- HD Mode: Resolved a condition for EMF/WMF files where embedded bitmaps may not render. (DF-635)
- HD Mode: Resolved a condition for HTML files where malformed `<table>` elements may fail to render. (DF-567)
- HD Mode: Resolved a condition for HTML5 output where annotation/markup SVG layer may be incorrectly positioned. (DF-698)
- HD Mode: Resolved a condition for Lotus 1-2-3 files where content may be missing when converting to HD with `TEXT_FALLBACK` enabled. (DF-707)
- HD Mode: Resolved a condition for MS Excel files where files generated by third-party systems may render with text striked out. (DF-663)
- HD Mode: Resolved a condition for MS Excel files where XLSB with comments may cause an error. (DF-590)
- HD Mode: Resolved a condition for MS Office files where bullets using the Symbol font may not render on systems without the System font installed. (DF-623)
- HD Mode: Resolved a condition for MS Office files where checkboxes may not be rendered on systems without an adequate Unicode font. (DF-613)
- HD Mode: Resolved a condition for MS PowerPoint files where slides may incorrectly render with a black background. (DF-697)
- HD Mode: Resolved a condition for MS PowerPoint files where slides with malformed embedded images may fail to render. (DF-622)
- HD Mode: Resolved a condition for MS PowerPoint files where text may render at an incorrect height where font style information is inherited from the master slide. (DF-589)
- HD Mode: Resolved a condition for MS Word files where content deleted with Track Changed enabled may still appear in the output. (DF-572)
- HD Mode: Resolved a condition for PDF files where certain PDF forms would fail to merge when using `PDF_PRESERVE_ORIGINAL` option. (DF-580)
- HD Mode: Resolved a condition for PNG and JPG image files where DPI stored in EXIF metadata may not be read. (DF-696)
- HD Mode: Resolved a condition for TIFF images where files incorrectly encoded with "Old JPEG" compression may fail to render. (DF-680)

- Resolved a condition for 7zip Archives where some sub-files compressed with LZMA+Delta may fail to extract. (DF-214)
- Resolved a condition for Apple Pages where some footnote/endnotes may fail to extract in text-mode. (DF-677)
- Resolved a condition for EML files where files with few headers may fail to identify as EML. (DF-575)
- Resolved a condition for HTML files where files that have mismatched byte-order-marker and charset directives may identify as ASCII. (DF-637)
- Resolved a condition for MS PowerPoint files where certain Unicode ranges may not extract correctly in text-mode. (DF-701)
- Resolved a condition for MSG files where metadata maybe truncated if the body or subject contain control characters. (DF-675)
- Resolved a condition for PST files where extracted MSG files may show the wrong date when opened in Outlook. (DF-651)
- Resolved a condition for PST files where large extracted MSG files may be malformed. (DF-625)
- Upgraded third-party libwebp library addressing several potential vulnerabilities. (DF-684)

5.19 Document Filters 21.5.0 (May 2021)

5.19.1 Enhancements

- Added Hi-Def support for Apple Numbers files. (DF-115)
- Added support for running on Windows Server Core. (DF-339)
- Improved handling of charts in Apple Pages and Keynote. (DF-254)
- Improved performance of processing TIFF files when creating JPG or PNG images. (DF-533)
- Introduce HiDef support for X-Windows XWD images. (DF-269)
- Introduces ID and Text support for X-Window DMP image files. (DF-310)
- Introduces support for generating PDF/A-2U files; specify option PDFVERSION=PDFA2U when creating the PDF canvas. (DF-35)
- Introduces support for OTF and OTC font files. (DF-511)

5.19.2 Updates

- HD Mode: Resolved a condition for DICOM images where files compressed with lossless huffman may fail to render. (DF-216)
- HD Mode: Resolved a condition for EPS files where 32 bit-per-pixel images may fail to render. (DF-451)
- HD Mode: Resolved a condition for MS Excel files where hidden columns may be treated incorrectly when processed with SHOW_HIDDEN option. (DF-224)
- HD Mode: Resolved a condition for MS Excel files where single value pie charts may fail to render. (DF-475)
- HD Mode: Resolved a condition for MS Excel spreadsheets where files containing no cell content would produce a zero-page file. (DF-631)
- HD Mode: Resolved a condition for MS Word files where date fields may not be localized. (DF-586)
- HD Mode: Resolved a condition for MS Word files where documents containing a continuous-break followed by a page-break may fail to create a new page. (DF-694)
- HD Mode: Resolved a condition for MS Word files where emoji characters may fail to render. (DF-602)
- HD Mode: Resolved a condition for MS Word files where internal links generated in HTML output maybe wrong. (DF-391)
- HD Mode: Resolved a condition for MS Word files where malformed footnotes may cause a crash. (DF-620)
- HD Mode: Resolved a condition for MS Word files where space-before settings may be incorrectly applied to the first page of a section. (DF-619)
- HD Mode: Resolved a condition for MS Word files where text maybe rendered with the wrong font. (DF-384)

- HD Mode: Resolved a condition for MSG files where messages containing CSS overrides for the display attribute tags may fail to render. (DF-582)
- HD Mode: Resolved a condition for PDF files where combining multiple PDFs containing forms with the same field names may not render correctly. (DF-632)
- HD Mode: Resolved a condition for PDF files where converting PDF to PDF with rotation may report incorrect word coordinates. (DF-645)
- HD Mode: Resolved a condition for PDF Files where files containing malformed instructions could fail to render all content. (DF-583)
- HD Mode: Resolved a condition for PDF files where non-embedded fonts located on the local machine, but contained in a TTC file, were not used. (DF-573)
- HD Mode: Resolved a condition for Windows Bitmap files where 4 bit-per-pixel compressed images may be truncated. (DF-397)
- HiDef: Resolved a condition for MS Excel files where chart labels may be missing. (DF-703)
- HiDef: Resolved a condition for BMP files where OS/2 images may fail to render. (DF-245)
- HiDef: Resolved a condition for BMP generation where the file scanline may miss padding bytes. (DF-629)
- HiDef: Resolved a condition for HTML files where bgcolor would not set the page color. (DF-487)
- HiDef: Resolved a condition for MS Excel files where diagonal borders were not rendered. (DF-218)
- HiDef: Resolved a condition for MS Excel files where embedded images may be missing from output. (DF-452)
- HiDef: Resolved a condition for MS Excel files where stacked charts with negative values may not render correctly. (DF-182)
- HiDef: Resolved a condition for MS Word files where embedded PICT files may render at the wrong size. (DF-435)
- HiDef: Resolved a condition for MS Word files where headers/footers containing images larger than the page may not render correctly. (DF-213)
- HiDef: Resolved a condition for MS Word files where paragraphs may incorrectly render as a numbered list. (DF-563)
- HiDef: Resolved a condition for MS Word files where version 6.0 may fail to render. (DF-192)
- HiDef: Resolved a condition for MS Word files where XML 2003 versions may fail to render. (DF-223)
- HiDef: Resolved a condition for PDF files where merging multiple PDF forms with same field names could collide. (DF-587)
- HiDef: Resolved a condition for PICT images where 1, 4 and 8 bit-per-pixels images may fail to render. (DF-373)
- HiDef: Resolved a condition for the Canvas APIs where colors with an alpha channel would render opaque. (DF-401)
- HiDef: Resolved a condition for WMF files where large images may fail to render. (DF-185)
- HiDef: Resolved a condition for XBM images where files may be misidentified as ASCII. (DF-175)
- HiDef: Resolved a condition for XPS files where some embedded images may fail to render. (DF-617)

- HiDef: Resolved a condition of MS Word files where table cells may incorrectly clip their content. (DF-156)
- Patched libtiff to address vulnerability CVE-2020-35523. (DF-647)
- Resolved a condition for MS Access files where password protected files could return a generic error code. (DF-158)
- Resolved a condition for PDF files where files encoded as Japanese Adobe-Japan1 would map hyphen to non-breaking hyphen. (DF-234)
- Resolved a condition for PST/OST files where all messages may not be enumerated. (DF-650)
- Resolved a condition for TAR files where archives created on MacOS, with long files, may show truncated filenames. (DF-695)
- Resolved a condition for the Canvas API where calling DrawScaledImage or DrawImage with a buffer less than 2K may fail to render the image. (DF-674)
- Resolved a condition for ZIP files where archives containing invalid dates may fail to extract. (DF-649)
- Upgraded Wavpack to 5.4.0 to address multiple potential CVEs. (DF-676)

5.20 Document Filters 21.2.0 (Feb 2021)

5.20.1 Enhancements

- Added Hi-Def support for Apple Keynote files. (DF-39)
- Added Hi-Def support for Apple Pages files. (DF-526)
- Introduced Annotation Extraction API to the Object APIs. (DF-252)
- Introduced new Canvas type IGR_DEVICE_IMAGE_BRK to allow saving single-page images as Fax images. (DF-514)
- Introduced new Canvas type IGR_DEVICE_IMAGE_DCX to allow saving multi-page images as DCX file. (DF-525)
- Introduced new Canvas type IGR_DEVICE_IMAGE_GIF to allow saving single-page images as GIF images. (DF-545)
- Introduced new Canvas type IGR_DEVICE_IMAGE_JPEG2000 to allow saving single-page images as JPEG2000 images. (DF-552)
- Introduced new Canvas type IGR_DEVICE_IMAGE_PCX to allow saving single-page images as PCX images. (DF-540)
- Introduced new Canvas type IGR_DEVICE_IMAGE_TGA to allow saving single-page images as TGA images. (DF-543)
- Introduced new Canvas type IGR_DEVICE_IMAGE_WEBSAFE which can be used to save a single-page to the smallest, portable web-safe format; including GIF, JPG or PNG. (DF-527)
- Introduced new Option BMP_COMPRESSION to control if canvas type IGR_DEVICE_IMAGE_BMP saves compressed or uncompressed images. (DF-504)
- Introduced new option TIFF_REORIENT_PAGES to control if the stored orientation should be honored when loading TIFF files, defaulting to ON. The default behavior has not changed from previous builds. (DF-548)
- Upgraded OpenJPEG library to 2.4. (DF-539)

5.20.2 Updates

- HD Mode: Resolved a condition for MS Excel files where text boxes containing Unicode characters may not render correctly. (DF-702)
- HD Mode: Resolved a condition for MS Office files where the files created with third-party applications may fail to render. (DF-672)
- HD Mode: Resolved a condition for MS PowerPoint files where hidden content may render without SHOWHIDDEN being enabled. (DF-603)
- HD Mode: Resolved a condition for MS Visio files where gradients may not render correctly when saving to PDF. (DF-153)

- HD Mode: Resolved a condition for MS Word files where text may not be selectable on pages with borders when creating HTML5 output. (DF-601)
- HD Mode: Resolved a condition for MS Word files where WordArt may render with the wrong font. (DF-197)
- HD Mode: Resolved a condition for PDF files where errors may occur when joining multiple files into one when using PDF_PRESERVE_ORIGINAL. (DF-612)
- HD Mode: Resolved a condition for PDF files where large pages may not be correctly scaled when converting to HTML5. (DF-682)
- HD Mode: Resolved a condition for PDF files where rotated text may report the wrong dimensions in Structured XML. (DF-219)
- HD Mode: Resolved a condition for Photoshop files where some 32-bpp images fail to render. (DF-486)
- HD Mode: Resolved a condition for Sun Raster files where images may not render correctly. (DF-418)
- HD Mode: Resolved a condition for XPS Files where gradients maybe missing when converting to PDF. (DF-472)
- HD Mode: Resolved a condition where an uncompressed 24-bit tiff with asymmetric DPI may render incorrectly. (DF-615)
- HD Mode: Resolved a condition where saving to a TIFF file with JPEG Compression and 8-bits per pixel would create a blank image. (DF-581)
- Patched third-party PDF library to address potential vulnerabilities. (DF-585)
- Patched xpdf library to resolve a condition where corrupt PDF files may cause a crash. (DF-574)
- Patched xpdf library to resolve a condition where PDF files containing certain embedded PostScript may cause a crash. (DF-571)
- Patched xpdf library to resolve a condition where PDF files with malformed bookmarks may cause a stack-overflow. (DF-664)
- Patched xpdf library to resolve a condition where PDF files with malformed fonts may cause a crash. (DF-646)
- Patched xpdf library to resolve a condition where PDF files with malformed shaders may cause a crash. (DF-642)
- Patched xpdf library to resolve a condition where PDF files with malformed xref entries may cause a stack overflow. (DF-692)
- Patched xpdf library to resolve potential vulnerability CVE-2020-35376. (DF-638)
- Resolve a condition where a PDF annotation created by Document Filters may have issues when modified in other PDF viewers. (DF-413)
- Resolved a condition for MS PowerPoint files where not all images may be returned when enumerating subfiles with ENUMERATE_IMAGES enabled. (DF-496)
- Resolved a condition for PST files where some extracted MSG files may not display their HTML body in Outlook. (DF-618)
- Resolved a condition for RAR files where incomplete multi-part archives may report invalid sub files. (DF-685)

5.21 Document Filters 11.4.20 (Nov 2020)

5.21.1 Enhancements

- Introduced file identification support for X-Windows Dump files. (DF-30)
- Introduced hi-def support for SciTex CT images. (DF-67)
- Introduced metadata support for CRW images. (DF-3)
- Introduced subfile support for ACE archive files. (DF-42)
- Introduced support for a source rectangle and a destination rectangle when rendering pages with IGR_Render_Page_Ex. (DF-251)
- Introduced support for Alpine Linux, allowing for easier deployment within Docker containers. (DF-305)
- Introduced support for SUSE Linux Enterprise Server 15 SP2 for ARM (aarch64). (DF-283)

5.21.2 Updates

- HD Mode: Resolved a condition for HTML files where tags styles with "display: inline" may fail to render. (DF-467)
- HD Mode: Resolved a condition for HTML files where CSS class selectors may cause text to be rendered larger than expected. (DF-403)
- HD Mode: Resolved a condition for HTML files where images may be cropped in output. (DF-395)
- HD Mode: Resolved a condition for HTML files where single
 tags in a <div> may fail to add spacing. (DF-376)
- HD Mode: Resolved a condition for IGR_Get_Page_Hyperlinks API where wrong coordinates may be returned. (DF-464)
- HD Mode: Resolved a condition for MS Excel files where vertical text may not be aligned correctly. (DF-202)
- HD Mode: Resolved a condition for MS PowerPoint where embedded images may fail to render. (DF-453)
- HD Mode: Resolved a condition for MS Word files where floating tables may be rendered in the wrong position. (DF-370)
- HD Mode: Resolved a condition for MS Word files where redacting a word in a header or footer may redact the same word on subsequent pages. (DF-474)
- HD Mode: Resolved a condition for MSG files where files containing a HTML body with no <body> tag may render incorrectly. (DF-394)
- HD Mode: Resolved a condition for PDF files where rotated pages may report the wrong word coordinates. (DF-495)
- HD Mode: Resolved a condition for Redacted files where hyperlinks over the redacted area may still be clickable. (DF-426)
- HD Mode: Resolved a condition for the Annotation APIs where dash borders may be rendered incorrectly. (DF-396)

- HD Mode: Resolved a condition for TIFF files where processing the file may cause more than required I/O. (DF-444)
- HD Mode: Resolved a condition for Word Perfect where line spacing maybe inconsistent. (DF-460)
- HD Mode: Resolved a condition where specifying GRAPHIC_WIDTH may not scale output correctly in HTML5. (DF-433)
- Resolved a condition for 7zip files where encrypted files may return a generic error message when extracting. (DF-420)
- Resolved a condition for 7zip files where files compressed with Deflate may fail to extract. (DF-420)
- Resolved a condition for Apple Pages where files saved from iCloud may fail to identify. (DF-477)
- Resolved a condition for IGR_Get_Subfiles_Enumerator where an error may be returned for files that do not support Subfiles. (DF-469)
- Resolved a condition for Image Files where converting to a raster based format with OCR enabled may cause an error. (DF-488)
- Resolved a condition for MS Excel files where relative hyperlinks may fail to extract. (DF-445)
- Resolved a condition for MS Word files where internal hyperlinks may fail to be extracted. (DF-447)
- Resolved a condition for MSG files where attachments may be identified by their Content-ID. (DF-410)
- Resolved a condition for PDF files where annotations with Unicode text may fail to extract using the Annotation APIs. (DF-393)
- Resolved a condition for PICT files where 32 bit per pixel images may contain red artifacts. (DF-468)
- Resolved a condition for TAR files where GNU TAR files with long file names may report truncates names. (DF-409)
- Resolved a condition for tar.gz files where the stored date was not reported as UTC. (DF-438)
- Resolved a condition for TGA images where some targa16 may fail to render. (DF-404)
- Resolved a condition for TIFF files where "TIFF_COMPRESSION=SOURCE" may incorrectly use CCITTFAX4 compression. (DF-465)
- Resolved a condition for Unix Compress files where subfiles may fail to extract when loading from a Stream. (DF-458)
- Resolved a condition where AM/PM may be swapped. (DF-419)

5.22 Document Filters 11.4.19 (Jul 2020)

5.22.1 Enhancements

- Added identification support for AppleDouble files. (DF-284)
- Added identification support for MO:DCA files. (DF-332)
- Added identification support for SysInternals ProcMon files. (DF-296)
- Added identification support for Windows RES resource files. (DF-297)
- Added support for extracting OpenAction URLs for PDF files. (DF-348)
- Added support for generating 1D Code 128 barcode via the Canvas Annotation API. (DF-282)
- Added support for generating 1D Code 39 barcode via the Canvas Annotation API. (DF-314)
- Added support for generating 1D GS1-128 barcode via the Canvas Annotation API. (DF-342)
- Added support for generating 2D Datamatrix barcode via the Canvas Annotation API. (DF-344)
- Added support for generating 2D PDF417 barcode via the Canvas Annotation API. (DF-262)
- Added support for generating 2D QR barcode via the Canvas Annotation API. (DF-263)
- Added support for URL extraction for HTML files. (DF-345)
- Added support for URL extraction for MSG files. (DF-345)
- Added support for URL extraction for RTF files. (DF-345)
- Introduced IGR_Canvas_Blank_Page API that can be used to insert a blank page onto a Canvas. (DF-293)
- Introduced IGR_Open_DIB API for loading raw-pixel data as new document. (DF-295)
- Introduces HiDef support for Kodax Photo CD. (DF-256)
- Patched libtiff to address CVE-2019-17546. (DF-277)
- Updated URL extraction to be able to separate the URL and Text. (DF-346)
- Upgraded to libjpeg-9d addressing vulnerabilities CVE-2020-1415 and CVE-2020-14153. (DF-340)

5.22.2 Updates

- HD Mode: Resolved a condition for HTML files where <abbr> tags may not render correctly. (DF-423)
- HD Mode: Resolved a condition for HTML files where <button> tags may fail to render. (DF-428)
- HD Mode: Resolved a condition for HTML files where <cite> tags may not render text italicized. (DF-361)
- HD Mode: Resolved a condition for HTML files where <dfn> tags may not render text italicized. (DF-377)
- HD Mode: Resolved a condition for HTML files where <input> placeholder attributes may fail to render. (DF-431)
- HD Mode: Resolved a condition for HTML files where <ins> and <del> tags may not be correctly marked. (DF-417)

- HD Mode: Resolved a condition for HTML files where <legend> tag may fail to render. (DF-390)
- HD Mode: Resolved a condition for HTML files where <mark> tag may not render text as highlighted. (DF-364)
- HD Mode: Resolved a condition for HTML files where <meter> tag may not render correctly. (DF-459)
- HD Mode: Resolved a condition for HTML files where <pre> tags may not render with a fixed pitch font. (DF-411)
- HD Mode: Resolved a condition for HTML files where <q> tag may not render text as quoted. (DF-482)
- HD Mode: Resolved a condition for HTML files where <s> tags may not render text as strikethrough. (DF-422)
- HD Mode: Resolved a condition for HTML files where <small> tags may not render text smaller. (DF-491)
- HD Mode: Resolved a condition for HTML files where <sub> tags may not render text as subscript. (DF-366)
- HD Mode: Resolved a condition for HTML files where <sup> tags may not render text as superscript. (DF-381)
- HD Mode: Resolved a condition for HTML files where <thead>, <tbody> and <tfoot> may render out of order. (DF-478)
- HD Mode: Resolved a condition for HTML files where <u> may not render text with an underline. (DF-405)
- HD Mode: Resolved a condition for HTML files where <var> tag may not render text as italicized. (DF-387)
- HD Mode: Resolved a condition for HTML files where inline base64 encoded images may not render. (DF-455)
- HD Mode: Resolved a condition for HTML files where inline svg images may fail to render. (DF-414)
- HD Mode: Resolved a condition for HTML files where table <caption> maybe missing. (DF-380)
- HD Mode: Resolved a condition for MS Excel files where implied column alignment maybe incorrect. (DF-164)
- HD Mode: Resolved a condition for MS Excel files where shapes and drawings may not scale correctly when using SPREADSHEET_SHRINKTOFIT. (DF-221)
- HD Mode: Resolved a condition for MS Office files where bad VML paths may cause a segmentation fault. (DF-363)
- HD Mode: Resolved a condition for MS Word 6 files where bullet lists may not render correctly. (DF-198)
- HD Mode: Resolved a condition for MS Word files where incorrect indenting maybe applied to paragraphs. (DF-225)
- HD Mode: Resolved a condition for MS Word files where malformed containers would fail to load. (DF-479)
- HD Mode: Resolved a condition for RTF files where large embedded images may be rendered at a lower resolution. (DF-489)
- HD Mode: Resolved a condition for XPS files where images may not be extracted in Classic HTML. (DF-371)
- Resolved a condition for 7z archives where 0-byte fails would return an error during extraction. (DF-494)
- Resolved a condition for Adobe files where corrupt XMP metadata may cause a file to fail to process. (DF-400)
- Resolved a condition for EML files where large files maybe incorrectly identified as TXT. (DF-382)
- Resolved a condition for Excel Files where converting to CSV may fail if fonts cannot be found. (DF-471)
- Resolved a condition for HTML files where high-range Unicode entities may not be extracted correctly. (DF-466)

- Resolved a condition for MS Excel files where complex diagrams may cause a crash. (DF-434)
- Resolved a condition for MS Word files where custom metadata may not be extracted. (DF-427)
- Resolved a condition for PDF files where files containing 64-bit xref tables may fail to convert. (DF-383)

5.23 Document Filters 11.4.18 (May 2020)

5.23.1 Enhancements

- Added Hi-Def support for SGI Raster Images. (DF-291)
- Added identification support for EOT Font files. (DF-121)
- Added identification support for FujiXerox Docuworks files. (DF-253)
- Added identification support for QuarkExpress files. (DF-271)
- Added identification support for SQLite Database files. (DF-307)
- Added identification support for Visual Studio SUO files. (DF-266)
- Added identification support for Windows Thumbs.db files. (DF-351)
- Introduced extension to "IGR_Render_Page_Ex" that allows for new form-values to be passed when rendering a page. See Implementation Guide for details. (DF-138)
- Introduced new API "IGR_Get_Page_Hyperlinks" that can be used to extract information about the Hyperlinks available on any given page. See Implementation Guide for details. (DF-41)

5.23.2 Updates

- HD Mode: Resolved a condition for EML files where HTML tables may render incorrectly if the CSS display tag is changed. (DF-212)
- HD Mode: Resolved a condition for HTML files where form elements with manually set CSS display may not render. (DF-177)
- HD Mode: Resolved a condition for MS Excel files where content of check boxes maybe clipped or misaligned. (DF-230)
- HD Mode: Resolved a condition for MSG and RTF files where extended Unicode may not be output correctly when converting to Classic HTML. (DF-190)
- HD Mode: Resolved a condition for PDF files where text may be missing on documents that embed the same font multiple times. (DF-238)
- HD Mode: Resolved a condition for PDF files where text position may be incorrect on certain malformed files. (DF-226)
- HD Mode: Resolved a condition for SVG files where textLength attribute was not processed. (DF-203)
- HD Mode: Resolved a condition for SVG output where rotated text maybe placed in the wrong location. (DF-446)
- HD Mode: Resolved a condition for XPS files where extra images may be generated when converting to HTML. (DF-239)

- HD Mode: Resolved a condition for the Canvas Drawing functions where brush and stroke was inconsistently applied to Rect, Ellipse, Pie and RoundRect. (DF-173)
- Patched ICU (International Components for Unicode) to address CVE-2017-17484. (DF-195)
- Patched ICU (International Components for Unicode) to address CVE-2020-10531. (DF-154)
- Resolved a condition for 7z files where AES encrypted files would not report as password protected. (DF-244)
- Resolved a condition for DWG files where corrupt files may cause a crash. (DF-235)
- Resolved a condition for MS Excel files where files containing corrupt comment data may cause a crash. (DF-242)
- Resolved a condition for MS Office files where dates in some metadata fields were incorrectly formatted. (DF-493)
- Resolved a condition for MS Visio files where certain files may hang when processing. (DF-237)
- Resolved a condition for MS Word files where headers and footers may not be extracted on documents created by Office 365 web-version. (DF-157)
- Resolved a condition for PDF files where corrupt files may cause a crash. (DF-229)
- Resolved a condition for PDF files where files containing corrupt AcroForms could cause the process to hang. (DF-205)
- Resolved a condition for RAR files where certain files may cause the process to hang. (DF-207)
- Resolved a condition for XML files where the wrong charset encoding may be detected. (DF-367)
- Resolved a condition for XPS files where text may fail to extract from some documents. (DF-415)

5.24 Document Filters 11.4.17 (Feb 2020)

5.24.1 Enhancements

- Added Hi-Def support for WordPerfect Graphics (version 1). (REN-855)
- Added new "GRAPHIC_ROTATE" option that can be used to output pages at different rotations. The option can be used on the Canvas to apply to all pages, or via the new IGR_Render_Page_Ex function. (DF-10)
- Added new IGR_Canvas_Annotate_JSON API that allows for page-level annotations to be applied to output. Annotations are user-interactive when generating PDF output. (DF-103)
- Added new IGR_Render_Page_Ex API that allows for per-page options to be passed. (DF-74)
- Improved handling of vector EMF/WMF images when generating PDF files where image file is no longer converted to a Bitmap first. (REN-1025)

5.24.2 Updates

- HD Mode: Resolved a condition for Excel files where text in cells may be clipped. (DF-247)
- HD Mode: Resolved a condition for MHT files where embedded images may not be output with IMAGEURL prefix. (DF-1419)
- HD Mode: Resolved a condition for MS Excel files where text boxes, drawing, or images may create extra rows at the end of a worksheet. (DF-1420)
- HD Mode: Resolved a condition for MS Office where borders may appear thicker when rendering at lower DPIs. (DF-206)
- HD Mode: Resolved a condition for MS Word files where multi-level lists may not be formatted correctly. (DF-233)
- HD Mode: Resolved a condition for MS Word files where paragraphs containing page breaks may be missing from output. (DF-1418)
- HD Mode: Resolved a condition for TIFF images where files with 64-bit colorspace may fail to render. (DF-1417)
- HD Mode: Resolved a condition where PDF files may crashing when processing corrupt files with PDF_PRESERVE_ORIGINAL enabled. (DF-180)
- Patched libtiff to address CVE-2019-17546. (DF-232)
- Patched openjpeg to address CVE-2020-8112 and CVE-2020-6851. (DF-184)
- Resolved a condition for DXF CAD files where certain files may fail to identify. (DF-193)
- Resolved a condition for MS Office Binder files where extracted filenames did not use stored filenames. (DF-167)
- Resolved a condition where MS DOS executable files may fail to identify. (DF-241)
- Security: UnRAR: CVE-2017-12938: patched: applied security patch to address this issue. (DF-189)
- Security: UnRAR: CVE-2017-12940: patched: applied security patch to address this issue. (DF-189)

- Security: UnRAR: CVE-2017-12941: patched: applied security patch to address this issue. (DF-189)
- Security: UnRAR: CVE-2017-12942: patched: applied security patch to address this issue. (DF-189)
- Upgraded libpng to 1.6.37, addressing CVE-2017-12652. (DF-161)

5.25 Document Filters 11.4.16 (December 2019)

5.25.1 Enhancements

- Added Bookmark APIs for extraction for processing of hierarchical bookmark information. (REN-975)
- Added Form APIs for extraction and processing of PDF form data. (REN-580)
- Added GRAPHIC_EFFECT options to for extended image manipulation. (REN-39)
- Added HDHTML_IMAGE_FORMAT=SVG option to generate HTML5 content with embedded SVG images, rather than converting drawings to a bitmap. (REN-559)
- Added Hi-Def support for Paint Shop Pro images. (REN-956)
- Added Hi-Def support for Windows Cursor images. (REN-64)
- Added Hi-Def support for X-Windows Bitmap images. (REN-951)
- Added Hi-Def support for X-Windows Pixmap images. (REN-954)
- Improved handling of HTML documents with words that do not fit on a page. (REN-814)

5.25.2 Updates

- HD Mode: Resolved a condition for MS Excel files where cells may be incorrectly marked as right-to-left when generating HTML5 output. (REN-931)
- HD Mode: Resolved a condition for MS Word files where borders may be incorrectly rendered as single line. (REN-927)
- HD Mode: Resolved a condition for MS Word files where nested tables may not be correctly clipped. (REN-294)
- HD Mode: Resolved a condition for MS Word files where paragraph borders may be incorrectly rendered as a single line. (REN-293)
- HD Mode: Resolved a condition for MS Word files where text frames may not correctly clip contained content. (REN-295)
- HD Mode: Resolved a condition for PDF files where ink markup may not render. (REN-317)
- HD Mode: Resolved a condition for RTF files where files that contained both Unicode and Ansi characters may not render correctly. (REN-955)
- HD Mode: Resolved a condition for SVG files where gradients may not render correctly. (REN-917)
- Resolved a condition for Apple PList files where extracted values may be incorrect. (REN-1012)
- Resolved a condition for CAB files where some sub files may fail to extract. (REN-1044)
- Resolved a condition for MP3 files where some files may fail to identify without a file extension. (REN-1043)
- Resolved a condition for MP4 files where reported dates may be incorrect. (REN-1023)

- Resolved a condition for MS Office documents where unescaped HTML entities may appear in text output.
(REN-922)

5.26 Document Filters 11.4.15 (September 2019)

5.26.1 Enhancement

- Added Hi-Def support for Microsoft Outlook for Mac OLK15MsgSource files. (REN-357)
- Added Hi-Def support for Microsoft Windows Clipboard (CLP) files. (REN-376)
- Added Hi-Def support for MS Project Gantt Charts. (REN-343)
- Added Hi-Def support for NCR images. (REN-379)
- Added IGR_Get_Option_Attribute function for enumerating available Document Filters options. (REN-587)
- Added option DETECT_MACROS that outputs a metadata value "isys:Macros" if macros are detected in MS Office documents. (REN-346)
- Added option RESAMPLE_IMAGE to control if embedded images are scaled up/down when generating PDF files. (REN-694)
- Added subfile extraction support for Microsoft Outlook for Mac OLM files. (REN-358)
- Added support for markup and drawing functions on to a HTML5 canvas. (REN-730)
- Added text support for Apple iBook file type. (REN-680)
- Added text support for Apple PList binary files. (REN-491)
- Added text support for EPUB ebook file type. (REN-679)
- Added text support for Quattro Pro Spreadsheet files. (REN-682)
- Extended IGR_Get_Format_Attribute to be able to return a MIME type per format. (REN-588)
- Improved handling of PDF files where page count and dimension information is processed faster. (REN-564)
- Patched bzip2 to address CVE-2019-12900. (REN-705)
- Patched freetype2 to address CVE-2017-8105. (REN-710)
- Upgraded openjpeg to 2.3.1 to address multiple vulnerabilities. (REN-707)

5.26.2 Updates

- HD Mode: Resolved a condition for IGR_Draw_Image where content type "image/svg+xml" was not recognized. (REN-787)
- HD Mode: Resolved a condition for MS Excel files where cells that span multiple pages may have words repeated in XML or HTML5 output. (REN-193)
- HD Mode: Resolved a condition for MS Excel files where drawings may be miss positioned in Classic HTML output. (REN-495)

- HD Mode: Resolved a condition for MS Excel files where locale aware dates would not use the value in `ISYS_DEFAULT_LCID`. (REN-515)
- HD Mode: Resolved a condition for MS Word files where documents containing headers/footers with images larger than the page would fail to render. (REN-790)
- HD Mode: Resolved a condition for MS Word files where embedded graphics may be missing in documents created by third-party tools. (REN-286)
- HD Mode: Resolved a condition for MS Word files where large tables in headers or footer may cause a crash. (REN-866)
- HD Mode: Resolved a condition for MS Word files where paragraphs or table cells may contain extra padding. (REN-280)
- HD Mode: Resolved a condition for MS Word files where tables containing extended ASCII characters may not render correctly. (REN-517)
- HD Mode: Resolved a condition for MS Word files where text fields may not be correctly aligned. (REN-290)
- HD Mode: Resolved a condition for RTF files where a file would fail to render if it contained non-conformant tags. (REN-493)
- HD Mode: Resolved a condition for TIFF files where certain corrupt images would return Page Limit Exceeded error. (REN-561)
- HD Mode: Resolved a condition for TIFF images where some non-compliant images may fail to render. (REN-887)
- Resolved a condition for EMLX files where they maybe incorrectly identified as ASCII files. (REN-826)
- Resolved a condition for MS PowerPoint files where text on master slides may not be extracted. (REN-868)
- Resolved a condition for MS Project files where resource names may not be extracted. (REN-654)
- Resolved a condition for MSO files where subfiles may fail to extract. (REN-686)
- Resolved a condition for Multi-Byte text files where some CJK encodings may not be detected. (REN-497)
- Resolved a condition for Open Office Spreadsheets where comments may not be extracted. (REN-674)
- Resolved a condition for PDF files where text from watermark may be merged with the overlaying text. (REN-188)
- Resolved a condition for text files where ISO-2022-KR may not decode correctly. (REN-510)
- Resolved a condition for Text Files where they may be misidentified as UUencoded, and hang when processing. (REN-897)
- Resolved a condition where .NET Framework Web Applications may fail to find the Document Filters DLLs when hosted on a network share. (REN-785)
- Resolved a condition where Python binary files for Windows were missing from the installer. (REN-836)
- Resolved a condition where the Implementation Guide did not correctly indicate XPS support. (REN-201)
- Updated giflib to 5.2.1 addressing CVE-2019-15133. (REN-885)

5.27 Document Filters 11.4.14 (June 2019)

5.27.1 Enhancements

- Added metadata and HD support for Computer Graphics Metafile (CGM) images. (REN-250)
- Added metadata and HD support for FlashPix image files. (REN-33)
- Added option EXCEL_PAGEBREAK_BETWEEN_SHEETS which adds a page-break marker between sheets when processing spreadsheets as CSV. (REN-507)
- Added option GRAPHIC_DESKEW to deskew image formats when converting to HD. (REN-37)
- Added option GRAPHIC_DESPECKLE to despeckle image formats when converting to HD. (REN-38)
- Added options IMNET_OFFSET_X and IMNET_OFFSET_Y for controlling the position of the text layer for IMNET COLD files. (REN-395)
- Added sub-file support for XXencoded archives. (REN-251)
- Added support for Japanese Reiwa era. (REN-460)
- Added text and sub-file extraction support for Domino XML (DXL). (REN-344)

5.27.2 Updates

- HD Mode: Resolved a condition for BMP files where 32-bit images may swap color channels. (REN-392)
- HD Mode: Resolved a condition for FlashPix image files where colors may be tinted incorrectly. (REN-381)
- HD Mode: Resolved a condition for IMNET COLD files where extra pages may be produced. (REN-305)
- HD Mode: Resolved a condition for MS Excel files where hidden columns may cause rendering issues. (REN-494)
- HD Mode: Resolved a condition for MS Excel files where hidden rows and cells may be incorrectly rendered when converting to Classic HTML. (REN-478)
- HD Mode: Resolved a condition for MS Word files where borders may be incorrectly added to tables. (REN-278)
- HD Mode: Resolved a condition for MS Word files where borders may be incorrectly calculated for tables containing Row Spans. (REN-276)
- HD Mode: Resolved a condition for MS Word files where corrupt documents may consume more memory than expected. (REN-499)
- HD Mode: Resolved a condition for MS Word files where internal border style was not applied to split tables. (REN-282)
- HD Mode: Resolved a condition for Office files where an incorrect fill maybe used when rendering shapes. (REN-341)
- HD Mode: Resolved a condition for Office files where rows may be incorrectly split on page boundaries. (REN-289)
- HD Mode: Resolved a condition for Office files where table cell borders may render with the incorrect width in PDF output. (REN-275)

- HD Mode: Resolved a condition for PDF files where documents containing complex Ink annotations may take longer than expected to render. (REN-498)
- HD Mode: Resolved a condition for PDF files where scaling output using GRAPHIC_WIDTH or HEIGHT on the canvas may scale incorrectly. (REN-391)
- HD Mode: Resolved a condition for Visio files where images may not scale properly when the GRAPHIC_WIDTH options is used as a canvas option. (REN-473)
- HD Mode: Resolved a condition for XPS files where a crash may occur when rendering the same page more than once. (REN-327)
- HD Mode: Resolved a condition for XPS files where images may not scale properly with the GRAPHIC_WIDTH option. (REN-464)
- Resolved a condition for MS Works files where spreadsheets may fail to identify if extension is not present. (REN-479)
- Resolved a condition for MSI files where some files may fail to identify. (REN-512)
- Resolved a condition for MSO files where subfiles may fail to extract. (REN-508)
- Resolved a condition for Text files where they could be incorrectly identified as Apple Quicktime. (REN-500)
- Resolved a condition for the Java API where an exception was thrown on empty containers when calling GetFirstSubFile. (REN-374)

5.28 Document Filters 11.4.13 (March 2019)

5.28.1 Notifications

- Version numbers have been updated include a release version as the third digit

5.28.2 Enhancements

- Added high-definition support for MacPaint (.MAC) images
- Added high-definition support for MS Paint (.MSP) images
- Added high-definition support for IMNET G4 (.IMT) images
- Added high-definition support for IMNET COLD (.IMT) files
- Added support for Outlook MSO object files
- Introduced option “PDF_PRESERVE_ORIGINAL” that can be used when converting PDF to PDF. When enabled, the original PDF page elements are copied “as-is”, there is no rendering or modification.
- Introduced option “METADATA_WHEN_ENCRYPTED” that can be used with Microsoft Office documents with protected with “Azure Information Protection”. It allows for metadata to be extracted even if the body of the document is encrypted.

5.28.3 Updates

- Added missing option TEXT_METADATA to implementation guide (REN-331)
- Applied patches to ICU addressing CVE-2017-15396 and CVE-2017-15422 (REN-215)
- Fixed an issue where theme colors may not render correctly for XLS files (REN-187)
- HD Mode: Resolved a condition for HTML files where CSS page breaks were not used when rendering to image. (REN-176)
- Resolved a condition where DXF files may be identified as ASCII. (REN-190)
- HD Mode: Resolved a condition where Word 95 files may fail with error code 4. (REN-192)
- HD Mode: Resolved a condition for MS Excel files where cells may have the wrong background color (REN-178)
- HD Mode: Resolved a condition for MS Office files where images may not be correctly scaled when setting GRAPHIC_WIDTH or GRAPHIC_HEIGHT options. (REN-196)
- HD Mode: Resolved a condition for MS Word files were some bulleted lists may repeat the bullet character. (REN-292)
- HD Mode: Resolved a condition for MS Word files were text alignment may be wrong (REN-198)
- HD Mode: Resolved a condition for MS Word files where hyperlinks may miss the #fragment (REN-313)
- HD Mode: Resolved a condition for PDF files were highlight annotations may not render. (REN-315)

- HD Mode: Resolved a condition for PDF files where vector graphics may fail to render. (REN-306)
- HD Mode: Resolved a condition for TIFF files where files encoding with JPEG compression may not render. (REN-296)
- Resolved a condition for archives where modified dates of subfiles may be incorrect on Linux. (REN-326)
- Resolved a condition for Email Messages where header maybe missing. (REN-199)
- Resolved a condition for Email Messages where monospaced text may not wrap correctly. (REN-199)
- Resolved a condition for Email Messages where tables may not wrap correctly. (REN-199)
- Resolved a condition for EML files where files with a long delivery history may fail to identify correctly. (REN-204)
- Resolved a condition for MS Excel files where Excel 2003 XML files may cause a segfault (REN-212)
- Resolved a condition for MS Excel files where Japanese formatted dates may not convert correctly (REN-269)
- Resolved a condition for MS Project files where certain files may cause a segfault. (REN-216)
- Resolved a condition for MS Word files where hyperlinks created via fields may not be extracted (REN-329)
- Resolved a condition for MS Word files where hyperlinks on drawings may not be extracted. (REN-324)
- Resolved a condition for MSG files where certain files may cause a segfault (REN-211)
- Resolved a condition for Office documents where incomplete URLs may be extracted (REN-312)
- Resolved a condition for PDF files where Japanese files with corrupt embedded fonts may not render as expected. (REN-200)
- Resolved a condition for PST files where some messages may be missed when enumerating and extracting (REN-205)
- Resolved a condition for RAR files where files created with encrypted file name would not throw an exception when using Java (REN-207)
- Resolved a condition where Text files without byte-order-marker (BOM) may identify as the wrong format (REN-210)
- Resolved a condition with the Implementation Guide where ISYS_DEFAULT_LCID example was incorrect. (REN-226)

5.29 Document Filters 11.4 Build 3054 (November 2018)

5.29.1 Enhancements

- Added high-definition support for Brooktrout 301 Fax Images
- Added high-definition support for JEDMICS images
- Added improved support for Lotus 1-2-3 spreadsheets
- Extended reported “capabilities” flags to be able to distinguish between Office 2007+ and earlier versions

5.29.2 Updates

- Added OCR_REORIENT_PAGES to the implementation guide. (TFS12506)
- HD Mode: Resolved a condition for HTML files where elements with style 'line-height: initial' did not render correctly. (TFS12508)
- HD Mode: Resolved a condition for MS PowerPoint files where runs of text may be split across multiple elements when generating classic HTML. (TFS12398)
- HD Mode: Resolved a condition for MS Word files where footnotes may not appear in HTML5 output. (TFS12530)
- HD Mode: Resolved a condition for MS Word styles where malformed documents may cause an access violation. (TFS12552)
- HD Mode: Resolved a condition for WordPerfect files where fontnotes may not appear in HTML5 output. (TFS12529)
- Resolved a condition for Document Filters Samples where license key was not set in Java samples on install. (TFS12515)
- Resolved a condition for Document Filters Samples where the ConvertDocument Sample did not correctly output (classic) HTML. (TFS12514)
- Resolved a condition for Lotus 1-2-3 files where documents loaded via stream may fail to identify as the correct format. (TFS12547)
- Resolved a condition for MS Excel files where corrupt embedded objects or files may cause an access violation. (TFS12531)
- Resolved a condition for MS Excel files where hyperlink URLs may not be extracted when SHOWHIDDEN is enabled. (TFS12541)
- Resolved a condition for MS Excel files where malformed hyperlinks may cause an access violation. (TFS12518)
- Resolved a condition for MS Project files where files containing invalid task relationships may cause an access violation. (TFS12517)
- Resolved a condition for MS Project files where files may cause an access violation. (TFS12540)

- Resolved a condition for PDF files where documents containing accessibility information may take larger than expected amounts of memory. (TFS12467)
- Resolved a condition for PDF files where documents using Japanese fonts would replace ideographic space with space. (TFS12537)
- Resolved a condition for PDF files where Japanese documents without embedded fonts may extract certain punctuation incorrectly. (TFS12534)
- Resolved a condition for PDF files where malformed documents may cause an access violation. (TFS12512)
- Resolved a condition for PDF files where malformed documents may cause long processing times, or for text to be repeated. (TFS12491)
- Resolved a condition for PDF output where files generated with PDF_VERSION=A2 would be marked as PDF/A-2a but not contain accessibility information. (TFS12553)
- Resolved a condition for Sendmail/MBOX where files may be misidentified as ASCII. (TFS12486)
- Resolved a condition where certain EML files may identify as ASCII. (TFS12504)
- Resolved a condition where IGR_Get_Format_Attribute returns the incorrect 'Search Short Name' for MS Word. (TFS12536)

5.30 Document Filters 11.4 Build 2990 (September 2018)

5.30.1 Enhancements

- Added high-definition support for IFF files
- Added high-definition support for MS Word 6 and MS Word 95
- Added high-definition support for WBMP files
- Added identification support for additional CAD formats - CATIA CAD, Stereolithography CAD, STEP 3D CAD, Initial Graphics Exchange Specification, Pro/ENGINEER Model, Pro/ENGINEER Assembly, Pro/ENGINEER Drawing, Pro/ENGINEER Drawing Form, SolidWorks CAD, Jupiter Tessellation, Parasolid Model Part, Wavefront OBJ, Autodesk WHIP
- Added metadata support for Corel (Perfect Office) Presentation
- Added text mode support for Apple Keynote
- Added text mode support for Apple Numbers
- Added text mode support for Apple Pages
- Improved high-definition support for 1-bit-per-pixel images

5.30.2 Updates

- HD Mode: Resolved a condition for PDF output where the conversion could fail with higher DPI values. (TFS12487)
- HD Mode: Resolved a condition for EML files where malformed embedded HTML may cause text to be missing. (TFS12466)
- HD Mode: Resolved a condition for MS EXCEL where text that is not visible on the page may be findable within the browser. (TFS12460)
- HD Mode: Resolved a condition for PDF output where pages larger than 200" where being clipped rather than scaled. (TFS12436)
- HD Mode: Resolved a condition for MS Word files where graphics may not render in headers and footers. (TFS12394)
- Resolved a condition for classic HTML output where empty table cell height was measured incorrectly. (TFS11835)
- Resolved a condition for EML output where CC lines after the first line may not render. (TFS12440)

5.31 Document Filters 11.4 Build 2934 (July 2018)

5.31.1 Notice

Linux Shared Libraries are now built with RPATH set to \$ORIGIN. This enables the libraries to find dependent libraries independent of the LD_LIBRARY_PATH settings.

C header file has been renamed to "DocumentFilters.h". There are no other functional changes to this file.

C++ object files have been renamed to "DocumentFiltersObjects.cpp/h". There are no other functional changes to this file.

5.31.2 Enhancements

- Added high-definition support for Hangul 5 word processor
- Added high-definition support for Windows icon files
- Added high-definition support for Adobe Photoshop image files
- Added text and high-definition support for CALS Raster image files
- Added text and high-definition support for GEM Raster image files
- Added high-definition support for DICOM medical image files
- Added option PNG_BITSPERPIXEL which can control the output size of saved PNG files
- Added option EXTRACT_EMBEDDED_NAME which extract nested object names when processing subfiles on MS Office files.
- Added option TIFF_IGNORE_MEMORY_LIMITS which disables memory limit checks when processing extremely large TIFF images.
- Introduced .NET core compatible library called Hyland.DocumentFilters, which is unusable from .NET Framework 2.0 and above, and .NET core. To target .NET core, you must update your project to use Hyland.DocumentFilters. Perceptive.DocumentFilters will remain for backwards compatibility.
- C# samples now support .NET core on Windows, Linux and MacOS.

5.31.3 Updates

- HD Mode: Added new option PAGE_TEXT_SYMBOLS_TO_UNICODE. When set, symbols (Wingdings, Webdings, etc) will be mapped to a Unicode representation when calling IGR_Get_Page_Text.
- HD Mode: Implemented option TIFF_IGNORE_MEMORY_LIMITS, which can be set if processing extremely large TIFF files. When not set, and error will be generated if the image would take more than 1GB of memory to process. (TFS12433)

- HD Mode: Introduced option TEXT_FALLBACK_MODE=bodyonly to control whether metadata should be included in renditions when converting non-hd formats to HD. (TFS12401)
- HD Mode: Resolved a condition for HTML and EML files where files containing malformed HTML may not render as expected. (TFS12421)
- HD Mode: Resolved a condition for MS Word files where ADVANCE fields were not processed resulting overlapped content. (TFS12395)
- HD Mode: Resolved a condition for MS Word files where segfault/access violation may occur on documents containing complex image wrapping in nested tables. (TFS12430)
- HD Mode: Resolved a condition for MS Word files where the wrong version may be reported in error messages when conversion to HD.
- HD Mode: Resolved a condition for RTF files where embedded images may render an low resolution. (TFS12425)
- HD Mode: Resolved a condition for WordPerfect files where full-justified paragraphs may render incorrect.
- HD Mode: Resolves a condition for MS Word documents where saved in compatibility mode may render as blank pages.
- Resolved a condition for JPEG files where short metadata fields may be missing from output. (TFS12393)
- Resolved a condition for MS Excel files where multi-part web view files may hang during extraction. (TFS12414)
- Resolved a condition for MS Word files where the wrong sub-version may be reported during extraction
- Resolved a condition for PDF files where with partial font data may cause the process hang. (TFS12435)
- Resolved a condition for WordStar 2000 files where extended characters may be missing from output
- Resolved a condition for WordStar 4 and 5 files where files may be misidentified

5.32 Document Filters 11.4 Build 2878 (May 2018)

5.32.1 Enhancements

- Added metadata and high-definition support for DCX (multi-page PCX) images
- Added metadata and high-definition support for EPS with Preview images
- Added metadata and high-definition support for JPEG2000 (JP2, J2C and JPF) images
- Added metadata and high-definition support for PCX images
- Added metadata and high-definition support for Targa (TGA) images
- Added support for generating PDF/A-1a
- Added text and high-definition support for Hangul 3 word processor
- Improved handling of right-to-left languages, such as Hebrew and Arabic, when converting to high-definition

5.32.2 Updates

- HD Mode: Resolved a condition for MS Word files where graphics embedded in fields may incorrectly show. (TFS12351)
- HD Mode: Resolved a condition for Office documents where subscript and superscript would be the wrong size when rendering at higher DPI. (TFS12352)
- HD Mode: Resolved a condition for PDF files where certain embedded Type 1 fonts may fail to render. (TFS12322)
- HD Mode: Resolved a condition for Word Perfect files where text may be spaced incorrectly, and incorrect bullets may be used. (TFS11837)
- HD-Mode: Resolved a condition for EML files where attachments may not be shown in the attachment list if they were marked as 'inline'. (TFS12323)
- Resolved a condition for EML and EMLX files where files containing long lines may not be correctly identified. (TFS12076)
- Resolved a condition for PDF files where extremely large files may report as corrupt. (TFS11800)
- Text-Mode: Resolved a condition for OpenOffice Spreadsheets where collapsed rows may report the wrong number of collapses lines. (TFS12335)
- Text-Mode: Resolved a condition for PDF files where extra-long, single page documents may take significant time to process. (TFS12319)
- Text-Mode: Resolved a condition for PDF files where hyperlink URL text was not extracted if the link contained no text. (TFS12324)
- Text-Mode: Resolved a condition for PDF files where text of adjacent lines may be merged together. (TFS12320)

5.33 Document Filters 11.4 Build 2822 (March 2018)

5.33.1 Notice

The following vulnerabilities have been reported against Document Filters and have been addressed in this release.

- HD Mode: Resolved a condition for MS Word files where specially crafted documents could cause heap corruption and free-after-use errors. (TALOS-2018-0527) (TFS11840)
- HD Mode: Resolved a condition for OpenOffice documents where specially crafted files may cause heap corruption. (TALOS-2018-0528) (TFS11841)
- HD Mode: Resolved a condition for MS Word files where specially crafted documents could cause heap corruption. (TALOS-2018-0534) (TFS11848)
- HD Mode: Resolved a condition for MS Word files where specially crafted files may cause a buffer-overwrite when converting to classic html (TALOS-2018-0538) (TFS12302)

The following open-sources libraries have been patched to address disclosed vulnerabilities.

- ICU (International Components for Unicode) 58.1 : patched for CVE-2017-14952, CVE-2017-7868 and CVE-2017-7867.
- libTIFF 4.0.8 : patched for CVE-2017-9935, CVE-2017-9936, CVE-2017-10688, CVE-2017-11335, CVE-2017-12944, CVE-2017-13726 and CVE-2017-13727.
- freetype2 2.6.5 : patched for CVE-2016-10244.
- skia : patched for CVE-2013-6648.

5.33.2 Enhancements

- Added support for Android.
- Added support for generating linearized PDF files (PDF_LINEARIZE=on).
- Added support for generating PDF outlooks/bookmarks (PDF_OUTLINES=on).
- Added support for Word for Mac versions 4 and 5.

5.33.3 Updates

- Applied patches to open-source ICU and TIFF libraries to address vulnerabilities. (TFS11846)
- HD Mode: Improved the memory utilization for MS Excel files when converting to CSV. (TFS11828)
- HD Mode: Resolved a condition for HTML files where form elements would not render. (TFS11839)
- HD Mode: Resolved a condition for MS Excel files where corrupt documents may cause a crash. (TFS11843)

- HD Mode: Resolved a condition for MS Excel files where documents containing corrupted comments may take longer than expected to process. (TFS11829)
- HD Mode: Resolved a condition for MS PowerPoint files where backgrounds with multiple overlays may render incorrectly. (TFS11743)
- HD Mode: Resolved a condition for MS PowerPoint files where fill patterns may render incorrectly. (TFS11776)
- HD Mode: Resolved a condition for MS PowerPoint files where subscript footer elements may be positioned incorrectly. (TFS11734)
- HD Mode: Resolved a condition for MS PowerPoint files where tables with merged cells may render incorrectly. (TFS11749)
- HD Mode: Resolved a condition for MS Visio files where connection lines may incorrectly render as curved lines. (TFS12030)
- HD Mode: Resolved a condition for MS Word files where table shading may incorrectly cover text. (TFS11814)
- HD Mode: Resolved a condition for PDF files where documents with mixed Acro forms and annotations may not render all text elements. (TFS11821)
- HD Mode: Resolved a condition for Visio files where rounded rectangles and dashed lines may not render correctly. (TFS11836)
- Improved memory utilization for OpenOffice Spreadsheet files when converting to CSV. (TFS11842)
- Resolved a condition for ZIP files where multi-part archives that only contain a single part could fail to extract. (TFS11824)
- Text Mode: Resolved a condition for RTF files where hidden content may still be extracted when the SHOWHIDDEN setting is set to off. (TFS12029)

5.34 Document Filters 11.4 Build 2766 (January 2018)

5.34.1 Enhancements

- Added support for iOS
- Added support for extracting hyperlink URL from PDF files
- Added support for Excel files saved as 'Single File Web Page'
- Added support for Excel files saved as 'XML Spreadsheet 2003'

5.34.2 Updates

- HD Mode: Resolved a condition for MS Office files where charts with Unicode characters may not render correctly.
- HD Mode: Resolved a condition for MS PowerPoint files where bullet points containing ideographic characters may insert extra line breaks. (TFS11746)
- HD Mode: Resolved a condition for MS PowerPoint files where ideographic characters may not word wrap . (TFS11748)
- HD Mode: Resolved a condition for MS PowerPoint files where images may incorrectly have a border applied. (TFS11751)
- HD Mode: Resolved a condition for MS Word documents where TEXT_FALLBACK may not work for older versions. (TFS11741 / SF24029545)
- HD Mode: Resolved a condition for PDF files where words may be incorrectly split. (TFS11539 / SF02231140)
- HD Mode: Resolved a condition where HTML files may produce the wrong number of pages when rendering at high DPI. (TFS11797 / SF1032495)
- Resolved a condition where closing a document object before a canvas object may cause a crash. (TFS11775)
- Resolved a condition where MS Word version 2 files may fail to extract. (TFS11793 / SF24041188)
- Resolved a condition where using incorrect OCR data files may cause a crash. (TFS11783 / SF24032310)

5.35 Document Filters 11.4 Build 2710 (November 2017)

5.35.1 Enhancements

- Added support for Linux ARMv8 64-bit.
- Added support for PostScript HD output.
- Added support for SVG HD output.
- Added support for SVG input.
- Updated default PDF output to PDF1.7. Includes improved handling of Office Drawing, allowing for higher quality when zoomed.

5.35.2 Updates

- HD Mode: Resolved a condition for HTML and EML files where files containing multiple nested blocks may not render. (TFS11667 / SF24019014)
- HD Mode: Resolved a condition for HTML files where memory may be leaked when converting to an image. (TFS11632)
- HD Mode: Resolved a condition for Image files where OCR_REORIENT_PAGES may not work when outputting to Classic HTML. (TFS11649)
- HD Mode: Resolved a condition for MS Excel files where column widths may be incorrectly set. (TFS11713 / SF24026986)
- HD Mode: Resolved a condition for MS Excel files where embedded objects may not render in Classic HTML. (TFS11615 / SF24005187)
- HD Mode: Resolved a condition for MS Excel files where embedded WordArt may render incorrectly. (TFS11642 / SF02328616)
- HD Mode: Resolved a condition for MS Excel files where memory may be leaked if the input file is corrupt. (TFS11650)
- HD Mode: Resolved a condition for MS Excel files where number formatting may be incorrect. (TFS11712 / SF24024507)
- HD Mode: Resolved a condition for MS Word documents where floating graphics in footers may not appear. (TFS11596 / SF02318881)
- HD Mode: Resolved a condition for MS Word files where documents generated from reporting engines may render incorrectly. (TFS11655 / SF24015883)
- HD Mode: Resolved a condition for MS Words files where text marked as hidden may still be shown. (TFS11754)
- HD Mode: Resolved a condition for OpenOffice documents where different page sizes on sections may not be correct. (TFS11735)

- HD Mode: Resolved a condition for PDF files where documents with corrupt embedded fonts may not render correctly. (TFS11577 / SF02287968)
- HD Mode: Resolved a condition for PDF files where line text may be merged together. (TFS11740 / SF24029418)
- HD Mode: Resolved a condition for PDF files where words may be broken into multiple pieces. (TFS11739 / SF24029420)
- HD Mode: Resolved a condition for PDF files where words maybe incorrectly merged together. (TFS11684 / SF24009163)
- HD Mode: Resolved a condition for RTF files where dynamic page numbers may not render correctly. (TFS11733)
- HD Mode: Resolved a condition for StarOffice files where content maybe missing. (TFS11737)
- HD Mode: Resolved a condition for Text files where content may be missing when converting to HD. (TFS11738 / SF24029553)
- HD Mode: Resolved a condition for TIFF files where files encoding images with JPEG compression, and 1 bit per pixel, would generate an empty image. (TFS11604)
- HD Mode: Resolved a condition for TIFF images where the process may stall if the input file is corrupt. (TFS11648)
- Resolved a condition for DMG where Apple Partition Map may not process correctly. (TFS11564 / SF02263432)
- Resolved a condition for DXF files where corrupt files may cause the process to crash. (TFS11661)
- Resolved a condition for Flash files where corrupt files may cause large memory allocations. (TFS11656)
- Resolved a condition for Microsoft Access files where corrupt files may cause memory errors. (TFS11657)
- Resolved a condition for Microsoft PowerPoint files where corrupt documents may cause large memory allocations. (TFS11658)
- Resolved a condition for MS Cabinet files where the process may hang, or fail to extract content. (TFS11679 / SF24016622)
- Resolved a condition for Open Office files where corrupt documents may cause a buffer over-read. (TFS11660)
- Resolved a condition for PDF files where documents containing Japanese may return wrong quote characters. (TFS11471 / SF02138062)
- Resolved a condition for PDF files where corrupt documents may cause a memory leak. (TFS11644)
- Resolved a condition for PDF files where corrupt documents may cause a memory leak. (TFS11666)
- Resolved a condition for PDF files where corrupt documents may cause a stack overflow. (TFS11659)
- Resolved a condition for PST files where corrupt files may cause a stack overflow. (TFS11662)
- Resolved a condition for XML files generated on MacOS where new lines may be suppressed. (TFS11663 / SF24011156)
- Resolved a condition for XML Files where corrupt documents may cause a memory leak. (TFS11665)

- Resolved a condition for ZIP files where files with the central directory in the middle of the file may fail to process. (TFS11729 / SF02321416)
- Text Mode: Resolved a condition for MS Excel files where WordArt may be missing from text output. (TFS11640 / SF02328616)
- Text Mode: Resolved a condition for MSG and RTF files where Unicode may be missing from output. (TFS11556 / SF02260396)
- Text Mode: Resolved a condition for Paradox Database files where the process may crash if the input file is corrupt. (TFS11646)
- Text Mode: Resolved a condition for Windows Media files where the process may stall if the input file is corrupt. (TFS11647)
- Text Mode: Resolved a condition for XML files where documents with a byte-order mark would be identified as text. (TFS11755 / SF24032369)

5.36 Document Filters 11.4 Build 2647 (September 2017)

5.36.1 Updates

- HD Mode: Resolved a condition for HTML and EML files where files containing multiple nested blocks may not render. (TFS11667 / SF24019014)
- HD Mode: Resolved a condition for HTML files where memory may be leaked when converting to an image. (TFS11632)
- HD Mode: Resolved a condition for Image files where OCR_REORIENT_PAGES may not work when outputting to Classic HTML. (TFS11649)
- HD Mode: Resolved a condition for MS Excel files where embedded objects may not render in Classic HTML. (TFS11615 / SF24005187)
- HD Mode: Resolved a condition for MS Excel files where embedded WordArt may render incorrectly. (TFS11642 / SF02328616)
- HD Mode: Resolved a condition for MS Excel files where memory may be leaked if the input file is corrupt. (TFS11650)
- HD Mode: Resolved a condition for MS Word documents where floating graphics in footers may not appear. (TFS11596 / SF02318881)
- HD Mode: Resolved a condition for MS Word files where documents generated from reporting engines may render incorrectly. (TFS11655 / SF24015883)
- HD Mode: Resolved a condition for PDF files where documents with corrupt embedded fonts may not render correctly. (TFS11577 / SF02287968)
- HD Mode: Resolved a condition for TIFF files where files encoding images with JPEG compression, and 1 bit per pixel, would generate an empty image. (TFS11604)
- HD Mode: Resolved a condition for TIFF images where the process may stall if the input file is corrupt. (TFS11648)
- Resolved a condition for DMG files where an Apple Partition Map may not process correctly. (TFS11564 / SF02263432)
- Resolved a condition for DXF files where corrupt files may cause the process to crash. (TFS11661)
- Resolved a condition for Flash files where corrupt files may cause large memory allocations. (TFS11656)
- Resolved a condition for Microsoft Access files where corrupt files may cause a heap-use-after-free. (TFS11657)
- Resolved a condition for Microsoft PowerPoint files where corrupt documents may cause large memory allocations. (TFS11658)
- Resolved a condition for MS Cabinet files where the process may hang, or fail to extract content. (TFS11679 / SF24016622)
- Resolved a condition for Open Office files where corrupt documents may cause a buffer over-read. (TFS11660)

- Resolved a condition for PDF files where documents containing Japanese may return wrong quote characters. (TFS11471 / SF02138062)
- Resolved a condition for PDF files where corrupt documents may cause a memory leak. (TFS11644)
- Resolved a condition for PDF files where corrupt documents may cause a memory leak. (TFS11666)
- Resolved a condition for PDF files where corrupt documents may cause a stack overflow. (TFS11659)
- Resolved a condition for PST files where corrupt files may cause a stack overflow. (TFS11662)
- Resolved a condition for XML files generated on MacOS that may suppress new lines. (TFS11663 / SF24011156)
- Resolved a condition for XML files where corrupt documents may cause a memory leak. (TFS11665)
- Text Mode: Resolved a condition for MS Excel files where WordArt may be missing from text output. (TFS11640 / SF02328616)
- Text Mode: Resolved a condition for MSG and RTF files where Unicode may be missing from output. (TFS11556 / SF02260396)
- Text Mode: Resolved a condition for Paradox Database files where the process may crash if the input file is corrupt. (TFS11646)
- Text Mode: Resolved a condition for Windows media files where the process may stall if the input file is corrupt. (TFS11647)

5.37 Document Filters 11.4 Build 2600 (August 2017)

5.37.1 Updates

- Resolved a condition where concurrent calls to `PDF_Multiplex()` may result in thread unsafe behavior. (TFS11627 / SF24009936)
- HD Mode: Resolved a condition for MS PowerPoint PPTX files where some background images may be vertically flipped. (TFS11623 / SFProspect)
- Resolved a condition for PDF files where some files may incorrectly be reported as corrupt. (TFS11621 / SF24005025)
- Resolved a condition where dates may not properly get converted to the current locale on Windows. (TFS11620 / SF02330132)
- HD Mode: Resolved a condition for PDF files where some documents may use an incorrect font. (TFS11614)
- Resolved a condition where a text file may be identified as an MP3 file. (TFS11610 / SF24003143)
- Resolved a condition for PDF files where some files may incorrectly be reported as corrupt. (TFS11576 / SF2286655)

5.38 Document Filters 11.4 Build 2581 (July 2017)

5.38.1 Updates

- Resolved security vulnerability CVE-2017-2822 (TALOS-2017-0323). (TFS11603)
- Resolved security vulnerability CVE-2017-2821 (TALOS-2017-0322). (TFS11602)
- HD Mode: Resolved a condition for MS Word DOCX files where conversion may product an extra page at the end. (TFS11597 / SF02175656)

5.39 Document Filters 11.4 Build 2543 (June 2017)

5.39.1 Notice

The following updates have been made for this release.

- wavpack upgraded to 5.1.0
- libtiff upgraded to 4.0.8

5.39.2 Enhancements

- Added option `LOCALTIME_CONVERSION` to enable or disable conversion of time stamp metadata fields to UTC from local time for documents where the timezone is not specified. (TFS11557 / SF02259016)

5.39.3 Updates

- HD Mode: Resolved a condition for MS PowerPoint PPTX files where images may render in black and white instead of in color. (TFS11566 / SF02280031)
- HD Mode: Resolved a condition for MS Word documents where underlined text may not render correctly. (TFS8321)
- HD Mode: Resolved a condition for MS Word DOCX files where a page border may display incorrectly. (TFS11392)
- HD Mode: Resolved a condition for MS Word DOCX files where a table cell may consume too much vertical space. (TFS9897)
- HD Mode: Resolved a condition for MS Word DOCX files where an error may occur on loading. (TFS11551 / SF02248621)
- HD Mode: Resolved a condition for MS Word files where paragraph spacing may not match the original document. (TFS11500 / SF02175656)
- HD Mode: Resolved a condition for OpenOffice ODT files where text rotation may not be respected. (TFS8467)
- HD Mode: Resolved a condition for PDF files where rendering larger DPI resolutions may cause an error. (TFS11535 / SF2183076)
- HD Mode: Resolved a condition where long words may render at a slightly incorrect size. (TFS7787)
- Resolved a condition for ARJ files where a corrupt file may cause a crash. (TFS11591 / SF02317672)
- Resolved a condition for XML files where the option `ISYS_XML_MEMORY_LIMIT` may not behave as expected on FreeBSD. (TFS11580 / SF02296141)

5.40 Document Filters 11.4 Build 2452 (March 2017)

5.40.1 Notice

The following updates have been made for this release.

- libwebp upgraded to 0.6.0
- zlib upgraded to 1.2.11
- libjpeg upgraded to 9.9b
- libpng updated to 1.6.28

5.40.2 Enhancements

- This release added support for Linux PPC64LE.
- The following HD-specific options have been added.
- Added the EMAIL_FONT_FAMILY_OVERRIDE option to control the default font used when rendering email messages, including HTML.
- Added the EMAIL_FONT_SIZE_OVERRIDE option to control the default font size used when rendering email messages, including HTML.
- Added the PDFANNOTATIONS option to control when annotations are rendered when loading PDF files.
- Added the PDF_VERSION option to control the generated PDF compliance level, including PDF/A.
- Added the SPREADSHEET_PAGEDIRECTION option to control the order of pages when the worksheet does not fit on a single page.
- Added the SPREADSHEET_SHRINKTOFIT option to control spreadsheet page scaling.

5.40.3 Updates

- HD Mode: Resolved a condition for EML files where duplicate lines sometimes appeared across multiple pages. (TFS11211 / SF01963174)
- HD Mode: Resolved a condition for EML files where inline images would sometimes extract as an inline image and also as a sub-file. (TFS11275 / SF1983030)
- HD Mode: Resolved a condition for HTML files where conversions sometimes stopped if an invalid CSS was encountered. (TFS11461 / SF02136328)
- HD Mode: Resolved a condition for MS Excel files where the fill color and box borders would sometimes not render. (TFS11359)
- HD Mode: Resolved a condition for MS Excel XLSX files where some long strings of text would render incorrectly. (TFS9488)

- HD Mode: Resolved a condition for MSG files where text would sometimes display incorrectly when converted to PDF. (TFS9963)
- HD Mode: Resolved a condition for PDF files where the text within separate columns sometimes merged when rendered. (TFS7992)
- HD Mode: Resolved a condition for RTF files where text from multiple pages sometimes merged when converted. (TFS11474 / SF02149295)
- HD: Resolved a condition where MS Word DOCX files with complex tables would sometimes render incorrectly when converted to TIFF. (TFS11455 / SF02128915)
- Resolved a condition for EMLX files using a custom stream where the returned file capabilities were inconsistent versus using a file stream. (TFS11346 / SF02024317)
- Resolved a condition for MS Excel XLSX files where files generated by a third-party tool may have failed to convert. (TFS11479 / SF02155059)
- Resolved a condition for MS Excel XLSX files where the contents would sometimes not fully extract. (TFS11467 / SF02143108)
- Resolved a condition for MS Word DOCX files where an error was sometimes incorrectly received when extracting embedded images. (TFS11465 / SF02143105)
- Resolved a condition for MS Word DOCX files where custom user properties were sometimes not extracted. (TFS11464 / SF02143099)
- Resolved a condition for PDF files indexed in standard HTML or HD where extra spaces would sometimes appear in words. (TFS9462)
- Resolved a condition for PDF files where extra spaces would sometimes appear in words when converted to text or classic HTML. (TFS8935)
- Resolved a condition for PDF files where extra spaces would sometimes appear in words when converted to text or HD. (TFS9055)
- Resolved a condition for PDF files where text of scanned documents may have failed. (TFS11201 / SF01960015)
- Resolved a condition for PDF files where words would sometimes merge when converted to classic HTML. (TFS7971)
- Resolved a condition for RTF files where metadata output was sometimes not consistent for each output mode. (TFS11411 / SFN/A)
- Resolved a condition where an error message was not reported when ISYS_TMP exceeded the disk quota. (TFS11475 / SF01217776)
- Resolved a condition where Document Filters sometimes failed to detect malicious archives. (TFS11469 / SF02143102)
- Resolved a condition where EML files sometimes caused IGR_Open_Stream to fail with an error code 4. (TFS11481 / SF2156621)

- Resolved a condition where ENC2 files may have been misidentified as XYwrite. (TFS11498 / SF02178255)
- Resolved a condition where incorrect error codes were sometimes returned on corrupt files in RAR, ZIP, and 7ZIP archives. (TFS11466 / SF02143097)
- Text Mode: Resolved a condition for MS Excel files where text may not have extracted correctly. (TFS11391 / SF02072048)
- Text Mode: Resolved a condition for PDF files where extra spaces would sometimes appear in words when converted to text. (TFS11169 / SF01948838)
- Text Mode: Resolved a condition for PDF files where some commas were misplaced when converted to text. (TFS8977)
- Text Mode: Resolved a condition for RTF files where accented characters would sometimes not extract. (TFS11470 / SF02143161)
- Text Mode: Resolved a condition for RTF files where Unicode characters were sometimes not included in the output. (TFS11463 / SF02140571)
- Text Mode: Resolved a condition where some TIFF files would cause Document Filters to stop responding when attempting to convert to text. (TFS11495 / SF02176227)

5.41 Document Filters 11.4 Build 2480 (April 2017)

5.41.1 Updates

- HD Mode: Resolved a condition for MS Excel files where borders may be thicker than the original document. (TFS11489 / SF2163083)
- HD Mode: Resolved a condition for MS Excel files where documents saved with 1904 based dates may not render correctly. (TFS11517 / SF02194079)
- HD Mode: Resolved a condition for MS Excel files where the fill color and box borders would sometimes not render. (TFS11359)
- HD Mode: Resolved a condition for MS PowerPoint files where some clipart images may fail to render. (TFS11506 / SF02181750)
- HD Mode: Resolved a condition for MS Word DOCX files where tables and page backgrounds were sometimes merged with the previous page. (TFS11389)
- HD Mode: Resolved a condition for MS Word files where images may be missing on documents generated by third-party tools. (TFS11522 / SF2166577)
- HD Mode: Resolved a condition for MS Word files where paragraph spacing may not match the original document. (TFS11500 / SF02175656)
- HD Mode: Resolved a condition for MS Word files where table cells may be rendered with a black background. (TFS11510 / SF02163067)
- HD Mode: Resolved a condition for MS Word files where tables would sometimes render incorrectly when the file was converted to Classic HTML. (TFS11412 / SFN/A)
- HD Mode: Resolved a condition for OpenOffice files where processing may cause a crash. (TFS11540 / SF02231209)
- HD Mode: Resolved a condition for PDF files where text may render with an incorrect size when using a DPI greater than 300. (TFS11480 / SF02155399)
- HD Mode: Resolved a condition for some MS Word DOCX files where text-wrapped graphics would sometimes overlap when rendered. (TFS11386)
- Resolved a condition for MS Excel files where corrupt documents could cause a crash. (TFS11536 / SF02200996)
- Resolved a condition for MS Excel files where specially crafted documents could cause a buffer over-read (TALOS-2017-0302). (TFS11534)
- Resolved a condition for OpenOffice Spreadsheets where some content may be missing from extracted text. (TFS11530 / SF02220982)
- Resolved a condition for PDF files where documents with complex charts may take longer than expected to process. (TFS11531 / SF02221692)

- Resolved a condition for PST files where extracted MSG files with attachments may fail to open. (TFS11524 / SF02210625)
- Resolved a condition for RTF files where metadata output was sometimes not consistent for each output mode. (TFS11411 / SFN/A)
- Resolved a condition for Send Mail files where the file may be misidentified as text. (TFS11523 / SF01966637)
- Resolved a condition where file descriptors were sometimes not cleared when processing DGN files through a custom stream. (TFS11404)
- Resolved a condition where MSG files with Unicode recipients would sometimes extract incorrectly. (TFS11408 / SF02053895)
- Resolved a potential buffer overflow in the C++ samples. (TFS11514 / SF2191936)
- Text Mode: Resolved a condition for MS Excel files where files generated by third party tools may use more memory than necessary when converting to CSV. (TFS11511 / SF2188703)
- Text Mode: Resolved a condition for PDF files where some commas were misplaced when converted to text. (TFS8977)
- Text Mode: Resolved a condition for Shift_JIS encoded XML files with XML charset declarations where characters would sometimes extract incorrectly. (TFS11378)
- Text-Mode: Resolved a condition where Office files may take longer to process than on previous builds. (TFS11537 / SF2222206)

5.42 Document Filters 11.3 Build 2400 (January 2017)

5.42.1 Notice

The following updates have been made for this release.

- LibTIFF upgraded to 4.0.7. This upgrade resolves 15 CVEs.
- LZMA SDK updated to 16.04. This update resolves 1 CVE.

After review of vendor support, the minimum supported versions have been updated for the following operating systems. For more information, see supported platforms in the Document Filters Implementation Guide.

- FreeBSD (version 9 and higher)
- MacOS (version 10.8 and higher)

5.42.2 Updates

- HD Mode: Resolved a condition for MS Excel XLSX file documents with merged cells would sometimes not render. (TFS11435 / SF02095676)
- HD Mode: Resolved a condition for MS PowerPoint PPTX files where files saved in strict mode may not have rendered correctly. (TFS11440 / SF02095700)
- HD Mode: Resolved a condition for MS Word DOCX files where tables and page backgrounds were sometimes merged with the previous page. (TFS11389)
- HD Mode: Resolved a condition for MS Word files where tables would sometimes render incorrectly when the file was converted to Classic HTML. (TFS11412)
- HD Mode: Resolved a condition for MS Word files where the word dimensions sometimes changed when rendered to canvas. (TFS11262 / SF01970929)
- HD Mode: Resolved a condition for PDF files where annotations would render when converted to HTML or HTML5 when the PDFANNOTATIONS option was set to OFF. (TFS9286)
- HD Mode: Resolved a condition for PDF files where check marks would sometimes not render when converting to HTML5. (TFS11398 / SF2031468)
- HD Mode: Resolved a condition for PDF files where some forms would not render when converted to TIFF. (TFS9962)
- HD Mode: Resolved a condition for RTF files where pages with a colored background would sometimes render incorrectly as white. (TFS11439 / SF02095687)
- HD Mode: Resolved a condition for TIFF files where a page would sometimes render as a solid black page. (TFS11410 / SF02079166)

- HD Mode: Resolved a condition where the text may have extended past the page number on the same page when converting from TXT files to TIFF. (TFS11447 / SF02118113)
- Resolved a condition for PDF files where spaces were sometimes not displayed in text extraction. (TFS11390 / SF02072048)
- Resolved a condition that prevented the registration of ISYS11df.dll on Windows Server 2008 Standard SP2 operating systems. (TFS11441 / SF02101151)
- Resolved a condition where not all text may have been extracted from HTML files. (TFS11443 / SF02100295)
- Resolved a condition where small UTF-8 text files may not have correctly detect the encoding. (TFS11235 / SF01969656)
- Resolved a condition where unknown or unsupported file types may have been incorrectly identified as WordStar. (TFS11454 / SF02129078)

5.43 Document Filters 11.3 Build 2356 (December 2016)

5.43.1 Enhancements

- Added support for Extended Zip(X) archives. Document Filters now supports the following Zip compression modes.
- Deflate, Deflate64, Reduce, Implode, Shrink
- BZip2, LZMA, PPMd, WavPack
- Added HD support for Microsoft Visio VDX and VSDX documents.
- Added HD support for Microsoft XPS documents.
- Added HD support for Microsoft WordArt (2D).
- Added support for deflate64 compressed zip archives. (TFS9541)

5.43.2 Updates

- HD Mode: Resolved a condition for broken HTML files where some tables and pages would render incorrectly. (TFS11407 / SF02084108)
- HD Mode: Resolved a condition for EML files converted to TIFF where text sometimes rendered off the page. (TFS11403 / SF02077751)
- HD Mode: Resolved a condition for MS Word DOCX files where patterned backgrounds sometimes rendered incorrectly. (TFS11387)
- HD Mode: Resolved a condition for PDF files where check marks would sometimes not render when converting to HTML5. (TFS11398 / SF2031468)
- HD Mode: Resolved a condition for PDF files where files containing only images would sometimes take longer than usual to process. (TFS11284 / SF01988817)
- HD Mode: Resolved a condition for some MS Word DOCX files where pages with clipping or column-balancing could cause an access violation. (TFS11395)
- HD Mode: Resolved a condition for some MS Word DOCX files where text would sometimes render incorrectly. (TFS11394)
- HD Mode: Resolved a condition for some MS Word DOCX files where text-wrapped graphics would sometimes overlap when rendered. (TFS11386)
- HD Mode: Resolved a condition for some MS Word DOCX files where the footer sometimes rendered in an incorrect location. (TFS11393)
- HD Mode: Resolved a condition for some MS Word DOCX files where white backgrounds would sometimes tile incorrectly. (TFS11388)
- HD Mode: Resolved a condition where PDF files converted to HTML5 would sometimes render the font size incorrectly. (TFS11281 / SF01971282)

- Resolved a condition for EML files where the header information would sometimes display an incorrect attachment name. (TFS11343 / SF2030351)
- Resolved a condition for EMLX files using a custom stream where the returned file capabilities were inconsistent versus using a file stream. (TFS11346 / SF02024317)
- Resolved a condition for MS Word DOCX files where italic characters would sometimes render incorrectly when the file was converted to Classic HTML. (TFS11380 / SF2060388)
- Resolved a condition where an "OSAtomicIncrement32Barrier symbol not found" error could occur on Mac OS X 10.8/10.9. (TFS11338 / SF02019015)
- Resolved a condition where an unsupported CIT file type was sometimes misidentified as an Intergraph/MicroStation CAD (DGN) file. (TFS11400 / SF02063503)
- Resolved a condition where file descriptors were sometimes not cleared when processing DGN files through a custom stream. (TFS11404)
- Resolved a condition where MSG files with Unicode recipients would sometimes extract incorrectly. (TFS11408)
- Resolved a condition where PDF files with custom encodings would sometimes display incorrectly. (TFS11279 / SF01971282)
- Text Mode: Resolved a condition for MS Excel files where text may not have extracted correctly. (TFS11391 / SF02072048)
- Text Mode: Resolved a condition for Shift_JIS encoded XML files with XML charset declarations where characters would sometimes extract incorrectly. (TFS11378)
- Text Mode: Resolved a condition for XIB (XML) files where subfile text sometimes did not extract. (TFS11399 / SF02076495)

5.44 Document Filters 11.3 Build 2322 (November 2016)

5.44.1 Updates

- HD Mode: Resolved a condition for EML files where the text between tables and paragraphs would sometimes overlap when rendered. (TFS11214 / SF01964036)
- HD Mode: Resolved a condition for HTML5 output where changing the region settings on a computer would sometimes cause the decimal mark to convert incorrectly. (TFS11245 / SF01970052)
- HD Mode: Resolved a condition for MS Excel XLSX files where several extra pages would sometimes render. (TFS11243 / SF01972989)
- HD Mode: Resolved a condition for MS Office 2007 files where the Pie Chart legend colors would sometimes render incorrectly. (TFS11337)
- HD Mode: Resolved a condition for MS PowerPoint files where dashed lines sometimes rendered as solid lines. (TFS11302)
- HD Mode: Resolved a condition for MS PowerPoint files where graph colors would sometimes render incorrectly. (TFS11300)
- HD Mode: Resolved a condition for MS PowerPoint PPT files where the fill color would sometimes render incorrectly. (TFS11295)
- HD Mode: Resolved a condition for MS PowerPoint PPT files where the same bar graph would sometimes render multiple times. (TFS11297)
- HD Mode: Resolved a condition for MS PowerPoint PPT files where when converted to TIFF, PNG, JPG, or PDF, images would sometimes render incorrectly. (TFS11273 / SF01982747)
- HD Mode: Resolved a condition for MS PowerPoint PPT files where when converted to TIFF, PNG, JPG, or PDF, the date would sometimes not render. (TFS11288 / SF01982747)
- HD Mode: Resolved a condition for MS PowerPoint PPTX files where graphs would sometimes render incorrectly. (TFS11301)
- HD Mode: Resolved a condition for MS Word DOC files where extraction sometimes resulted in an invalid "Page limit exceeded" error. (TFS11189 / SF01952351)
- HD Mode: Resolved a condition for MS Word DOC files where page numbers would sometimes display multiple times. (TFS11310 / SF02016323)
- HD Mode: Resolved a condition for MS Word DOCX files where if the machine was set to German or Russian, extra pages were sometimes output. (TFS11308 / SF01987732)
- HD Mode: Resolved a condition for MS Word DOCX files where oval Callout shapes that were rotated would sometimes render with an incorrect rotation. (TFS11263)

- HD Mode: Resolved a condition for MS Word DOCX files where oval Callout shapes would sometimes render incorrectly. (TFS9929)
- HD Mode: Resolved a condition for MS Word files where an incorrect error code was returned when the Microsoft Rights Management Service protection was enabled for the file. (TFS11241)
- HD Mode: Resolved a condition for OpenOffice ODP files where the font style and images would sometimes not render correctly. (TFS11203 / SF01962077)
- HD Mode: Resolved a condition for PDF files where check marks and other symbols would sometimes not render. (TFS11347 / SF2031468)
- HD Mode: Resolved a condition for PDF files where the hit-highlighting alignment sometimes covered the original text. (TFS8221)
- HD Mode: Resolved a condition for PowerPoint files where PICT images would not render. (TFS9349)
- HD Mode: Resolved a condition for RTF files where a “Page limit exceeded” error would sometimes erroneously return when converted. (TFS11240 / SF1970678)
- HD Mode: Resolved a condition for some Japanese, Simplified Chinese, and Traditional Chinese fonts where the font would sometimes not convert correctly on non-Windows platforms. (TFS11340)
- HD Mode: Resolved a condition for WordPerfect files where a “Page limit exceeded” error would sometimes erroneously return when converted. (TFS11242 / SF01971543)
- HD Mode: Resolved a condition where Japanese text in PPT files would sometimes render incorrectly on Linux (non-Windows). (TFS11348 / SF2029952)
- HD Mode: Resolved a condition where WMF files converted to HTML5 would sometimes return an error code 4. (TFS11234 / SF01969475)
- Resolved a condition for EML files where Japanese header text would sometimes not render correctly. (TFS11163 / SF01946757)
- Resolved a condition for HTML files where some Cyrillic characters would not extract properly. (TFS11175 / SF01953628)
- Resolved a condition for MS Excel CSV output where numbers were inconsistently rounded across platforms. (TFS10037)
- Resolved a condition for MS Excel XLSX files where some scripting elements were not escaped when converted to Classic HTML. (TFS11307 / SF02013425)
- Resolved a condition for MS Word DOC files where converting to a text or image would sometimes causes Document Filters to stop responding in Linux and Windows environments. (TFS11282 / SF01988119)
- Resolved a condition for MS XPS files where some files were not identified correctly. (TFS11266)
- Resolved a condition for MSG files where attachments were indexed but sometimes the parent document was not indexed. (TFS11233)

- Resolved a condition where large MBOX files, greater than 2 GB, would sometimes result in an error code 4 during email extraction. (TFS11267 / SF01978119)
- Resolved a condition where PPM image files were sometimes misidentified as text. (TFS11342)
- Resolved a condition where Russian HTML files would sometimes extract with an unrecognized character. (TFS11274)
- Resolved a condition where the application stopped responding when converting OpenOffice ODS files to text or HTML using the EXCELMODE=CSV option. (TFS11258 / SF01977321)
- Text Mode: Resolved a condition for HTML files where text extraction would sometimes cause words to appear merged. (TFS11363 / SF02044335)
- Text Mode: Resolved a condition for MS Excel files where different DPI values sometimes changed the text output. (TFS11237 / SF01969477)
- Text Mode: Resolved a condition where a malformed line in an HTML header would sometimes stop text extraction. (TFS11176 / SF01953628)
- Text Mode: Resolved a condition where decoding ANSI or single-byte codepage text or charset streams required stateful encodings. (TFS11172)
- Text Mode: Resolved a condition where PPT files with sub-files would sometimes result in an error code 4 when extracting to text. (TFS11303 / SF02008883)

5.45 Document Filters 11.3 Build 2228 (August 2016)

5.45.1 Notice

Due to lack of customer demand and vendor support, this is the final build for Document Filters for the following architectures.

- Microsoft Windows (Itanium 64 bit)
- Linux (Itanium 64 bit)

If you require builds for these platforms, contact your account representative.

5.45.2 Updates

- HD Mode: Resolved a condition for broken HTML files where some text may not have been rendered. (TFS9970)
- HD Mode: Resolved a condition for EMF files where image boxes would sometimes render incorrectly. (TFS9925)
- HD Mode: Resolved a condition for MS Excel files where check boxes may not have rendered correctly. (TFS9952)
- HD Mode: Resolved a condition for MS Excel files where radio buttons may not have rendered correctly. (TFS9951)
- HD Mode: Resolved a condition for MS Visio files where NURBS curves would not render. (TFS8412)
- HD Mode: Resolved a condition for MS Word DOC files where header rows would sometimes not repeat to subsequent pages. (TFS9901)
- HD Mode: Resolved a condition for MS Word DOCX files where a corrupt file sometimes caused an error when opening. (TFS9933)
- HD Mode: Resolved a condition for MS Word DOCX files where double brace shapes may not have rendered correctly. (TFS9928)
- HD Mode: Resolved a condition for MS Word DOCX files where header rows may not have repeated to subsequent pages. (TFS9924)
- HD Mode: Resolved a condition for MS Word DOCX files where line numbers would sometimes render incorrectly. (TFS9922)
- HD Mode: Resolved a condition for MS Word DOCX files where paragraph spacing may have been incorrect. (TFS9932)
- HD Mode: Resolved a condition for MS Word DOCX files where some table rows may have rendered incorrectly. (TFS9931)
- HD Mode: Resolved a condition for MS Word DOCX files where text box placement may have been incorrect. (TFS9921)
- HD Mode: Resolved a condition for MS Word DOCX files where text from some sections would not render. (TFS9923)

- HD Mode: Resolved a condition for MS Word DOCX files where text may have rendered upside down in Callout shapes. (TFS9930)
- HD Mode: Resolved a condition for MS Word files where command buttons may not have rendered correctly. (TFS9946)
- HD Mode: Resolved a condition for MS Word files where drop-down menus would not render selected items correctly. (TFS9945)
- HD Mode: Resolved a condition for MS Word files where graphics within a floating table would sometimes not render correctly. (TFS8594)
- HD Mode: Resolved a condition for MS Word files where graphics would sometimes not render on the correct page. (TFS8593)
- HD Mode: Resolved a condition for MS Word files where page numbers may not have rendered correctly. (TFS9937)
- HD Mode: Resolved a condition for MS Word files where table cell borders would sometimes not render. (TFS9935)
- HD Mode: Resolved a condition for MS Word files where table cell text was sometimes missing when rendered. (TFS9936)
- HD Mode: Resolved a condition for MS Word files where table cells sometimes shifted when rendered. (TFS9934)
- HD Mode: Resolved a condition for MS Word files where text would not render on the correct page. (TFS9947)
- HD Mode: Resolved a condition for MS Word files where text would sometimes not render. (TFS9903)
- HD Mode: Resolved a condition for MS Word files where the text direction in some tables would sometimes not render correctly. (TFS9902)
- HD Mode: Resolved a condition for MSG files where Chinese characters may not have rendered correctly. (TFS9964)
- HD Mode: Resolved a condition for PDF files where some cell text would not render. (TFS9584)
- HD Mode: Resolved a condition for PDF files where some text bounding boxes were not positioned correctly. (TFS7995)
- HD Mode: Resolved a condition for PDF files where some text may not have rendered correctly. (TFS9413)
- HD Mode: Resolved a condition for PDF files where the word x/y coordinates were sometimes incorrect and would not highlight the entire word. (TFS7976)
- HD Mode: Resolved a condition where a corrupt PDF file sometimes caused the engine to stop responding. (TFS7906)
- HD Mode: Resolved a condition where the same HTML document processed multiple times could have contained different output. (TFS10029)
- Improved performance when extracting text from PDF files with corrupt bookmarks. (TFS9959)
- Resolved a condition for Apple Pages files where some files were not identified correctly. (TFS10030 / SF01928071)
- Resolved a condition for Apple Pages files where some files were not identified correctly. (TFS10035)

- Resolved a condition for BMP files that were recognized as WordPerfect files. (TFS9957)
- Resolved a condition for BZ2 archives where a stack buffer overflow could occur (TALOS-CAN-0173). (TFS11159)
- Resolved a condition for encrypted MS Excel XLS files where text extraction sometimes caused an error. (TFS11162 / SF01946222)
- Resolved a condition for MBOX files where GetSubFile returned ANSI characters instead of Unicode characters. (TFS9914)
- Resolved a condition for MS Excel files where some files may have failed to open. (TFS10034)
- Resolved a condition for MS Excel files where some files were not identified correctly. (TFS10033)
- Resolved a condition for MS Excel XLS files where a stack buffer overflow could occur (TALOS-CAN-0172). (TFS11158)
- Resolved a condition for MS Project files where the extracted text may have contained incorrect ANSI characters. (TFS10038)
- Resolved a condition for MS Visio files where text extraction may have caused an Access Violation. (TFS10039 / SF01921809)
- Resolved a condition for MS Word DOC files where a heap buffer overflow could occur (TALOS-CAN-0185). (TFS11165)
- Resolved a condition for PST files where output was sometimes inconsistent between Windows and non-Windows systems. (TFS10025 / SF01921015)
- Resolved a condition when calling IGR_Get_Page_Attribute that included thousands separators in numerical values. (TFS10027 / SF01927119)
- Text Mode: Resolved a condition for OpenOffice ODS files where images may not have extracted correctly when using ENUMERATE_IMAGES=ON. (TFS10043 / SF01916031)

5.46 Document Filters 11.3 Build 2172 (June 2016)

5.46.1 Enhancements

- Improved performance when converting large MS Excel XLSX files to CSV.
- Improved support for JustSystems Ichitaro files, versions now supported are 5, 6, 8+.
- HD Mode: Added the ability to perform HD conversions of MS Word 2003 XML formatted files.
- HD Mode: Added support for MS Excel files with "doughnut" charts.
- HD Mode: The automatic scaling of watermarks is now allowed for strings up to approximately 80 characters.
- HD Mode: Added support for hyperlinks in PDF output files.
- HD Mode: Now allows the STRINGS_FALLBACK option to apply to HD Mode.

5.46.2 Updates

- HD Mode: Added support for hyperlinks in PDF output files. (FB10025)
- HD Mode: Added support for MS Excel files with "doughnut" charts. (SF01584208 / FB10332)
- HD Mode: Added the ability to perform HD conversions of MS Word 2003 XML formatted files. (SF01511517 / FB10510)
- HD Mode: Now allows the STRINGS_FALLBACK option to apply to HD Mode. (SF928105 / FB9144)
- HD Mode: Resolved a condition for MS Excel files where borders may not render correctly. (FB10368)
- HD Mode: Resolved a condition for MS Excel files where borders were incorrectly rendered. (FB10476)
- HD Mode: Resolved a condition for MS Excel files where bubble charts rendered incorrectly. (FB10467)
- HD Mode: Resolved a condition for MS Excel files where cell borders may not have been rendered. (FB10468)
- HD Mode: Resolved a condition for MS Excel files where hyperlinked text may have been misplaced. (FB10465)
- HD Mode: Resolved a condition for MS Excel files where the footer text sometimes overlapped the last line of text when rendered. (FB10577)
- HD Mode: Resolved a condition for MS Excel files where the x-axis and y-axis text in a chart with empty ranges did not render correctly. (SF01599720 / FB10363)
- HD Mode: Resolved a condition for MS Excel where charts may not have rendered correctly. (FB10372)
- HD Mode: Resolved a condition for MS Office files where some Math Equation symbols did not render. (FB10521)
- HD Mode: Resolved a condition for MS PowerPoint files where dashed line shapes sometimes rendered incorrectly. (FB10458)
- HD Mode: Resolved a condition for MS PowerPoint files where date formats rendered incorrectly. (FB10457)

- HD Mode: Resolved a condition for MS PowerPoint files where hyperlinks may not have worked correctly in HTML5 output. (SF01602509 / FB10500)
- HD Mode: Resolved a condition for MS PowerPoint files where image fill colors may not have rendered correctly. (FB10456)
- HD Mode: Resolved a condition for MS PowerPoint files where images may not have been rotated correctly. (FB10453)
- HD Mode: Resolved a condition for MS PowerPoint files where shapes may have been scaled incorrectly. (FB10455)
- HD Mode: Resolved a condition for MS PowerPoint files where WordArt text was incorrectly sized. (FB10437)
- HD Mode: Resolved a condition for MS PowerPoint where embedded WMF images sometimes rendered incorrectly. (SF01536194 / FB10483)
- HD Mode: Resolved a condition for MS PowerPoint where the cell gradient colors for tables did not render. (FB10383)
- HD Mode: Resolved a condition for MS Word DOCX files where auto dates rendered incorrectly in the footer. (FB10587)
- HD Mode: Resolved a condition for MS Word DOCX files where cell borders in tables may not have rendered correctly. (FB10552)
- HD Mode: Resolved a condition for MS Word DOCX files where check box selections may have rendered incorrectly. (FB10561)
- HD Mode: Resolved a condition for MS Word DOCX files where numbered lists may have rendered incorrectly. (FB10562)
- HD Mode: Resolved a condition for MS Word DOCX files where oval shapes may have rendered in an incorrect location. (FB10549)
- HD Mode: Resolved a condition for MS Word DOCX files where page background images may not have rendered. (FB10565)
- HD Mode: Resolved a condition for MS Word DOCX files where some text within a table may not have rendered correctly. (FB10550)
- HD Mode: Resolved a condition for MS Word DOCX files where some text would overlap tables when rendered. (FB10564)
- HD Mode: Resolved a condition for MS Word DOCX files where text sometimes overlapped table information. (FB10551)
- HD Mode: Resolved a condition for MS Word DOCX files where the cell shading color for tables rendered incorrectly. (FB10573)
- HD Mode: Resolved a condition for MS Word DOCX files where the numbers were missing in numbered and multilevel lists. (FB10548)

- HD Mode: Resolved a condition for MS Word files where automatic dates may not have rendered correctly. (FB10538)
- HD Mode: Resolved a condition for MS Word files where paragraph spacing may have been incorrect. (FB10471)
- HD Mode: Resolved a condition for MS Word files where some graphics may not have rendered. (FB10531)
- HD Mode: Resolved a condition for MS Word files where some icons may not have rendered. (FB10547)
- HD Mode: Resolved a condition for MS Word files where some images incorrectly overlapped. (FB10478)
- HD Mode: Resolved a condition for MS Word files where some images may have shifted off the page. (FB10546)
- HD Mode: Resolved a condition for MS Word files where some tables overlapped when rendered. (FB10582)
- HD Mode: Resolved a condition for MS Word files where some text was incorrectly rendered. (FB10480)
- HD Mode: Resolved a condition for MS Word files where some underlined fields may not have rendered correctly. (FB10543)
- HD Mode: Resolved a condition for MS Word files where strikethrough styles may not have been rendered. (FB10470)
- HD Mode: Resolved a condition for MS Word files where table dimensions and some text rendered incorrectly. (FB10540)
- HD Mode: Resolved a condition for MS Word files where table headers were missing and some tables overlapped. (FB10560)
- HD Mode: Resolved a condition for MS Word files where text box text may have been incorrectly placed when rendered. (FB10581)
- HD Mode: Resolved a condition for MS Word files where the page header placement sometimes rendered incorrectly. (FB10530)
- HD Mode: Resolved a condition for MS Word files where track changes deleted text would render. (FB10567)
- HD Mode: Resolved a condition for MS Word files where watermarks may not have rendered. (FB10544)
- HD Mode: Resolved a condition for PDF files where fractional font sizes were rounded to the nearest integer. (SF01762267 / FB10520)
- HD Mode: Resolved a condition for PDF files where text may have rendered incorrectly. (SF01773121 / FB10532)
- HD Mode: Resolved a condition where floating point numbers may have displayed with too many zeros. (FB10289)
- HD Mode: Resolved a condition where floating point rounding errors may have occurred. (FB10513)
- HD Mode: Resolved a condition where some EML files converted to images did not render correctly. (SF01837424 / FB10585)
- HD Mode: Resolved a condition where the alpha channel (transparency) may have been lost on images when HDHTML_IMAGE_FORMAT=AUTO was used. (FB10499)

- HD Mode: Resolved a condition where TIFF conversions to PDF sometimes rendered pages with the incorrect orientation. (SF01717220 / FB10506)
- HD Mode: Resolved a condition where WMF and EMF image files were incorrectly rendered. (FB10441)
- HD Mode: Resolved a condition where WMF rotated text was not anchored properly. (SF01536194 / FB10484)
- HD Mode: Resolved a condition which caused an IGR13 error code to display for some password protect RAR files. (SF01583969 / FB10335)
- HD Mode: The automatic scaling of watermarks is now allowed for strings up to approximately 80 characters. (SF01527695 / FB10188)
- Improved performance when converting large MS Excel XLSX files to CSV. (SF01845622 / FB10579)
- Improved support for JustSystems Ichitaro files, versions now supported are 5, 6, 8+. (SF01831754 / FB10559)
- Resolved a condition for .OST files where text extraction returned zero characters. (SF01754536 / FB10511)
- Resolved a condition for MS Excel (XLSX) files where text alignment rendered incorrectly when converted to Classic HTML. (SF01768045 / FB10523)
- Resolved a condition for MS Excel files which may not have opened when corrupt. (SF01642641 / FB10472)
- Resolved a condition for PDF files where processing a corrupt document may have caused the engine to stop responding. (SF01648807 / FB10496)
- Resolved a condition where an error occurred when using Document Filters 11.3.0.2116 on Windows XP Pro SP3. (SF01832035 / FB10545)
- Text Mode: Resolved a condition for damaged PDF files where some converted text did not output correctly. (SF01841620 / FB10569)
- Text Mode: Resolved a condition for MS PowerPoint files where some text in SmartArt diagrams did not extract correctly. (FB10517)
- Text Mode: Resolved a condition for MS PowerPoint files where text may not have extracted correctly when using ENUMERATE_IMAGES=ON. (SF01846231 / FB10578)

5.47 Document Filters 11.3 Build 2116 (April 2016)

5.47.1 Updates

- HD Mode: Added option HDHTML_FIELD_ATTRIBUTES to mark fielded text blocks in HTML5. (SF01548495 / FB10266)
- HD Mode: Added option HTML_IMAGE_ATTRIBUTES to include data attribute containing the embedded name of each image. (SF01548605 / FB10265)
- HD Mode: Resolved a condition for HTML files where 'background-color:window' would render as black. (SF01609686 / FB10381)
- HD Mode: Resolved a condition for MS Excel files where 'Angle of 1st Slice' was not applied when rendering Pie Charts. (SF01602819 / FB10357)
- HD Mode: Resolved a condition for MS Excel files where 'Line with Markers' charts may not draw connecting lines. (FB10377)
- HD Mode: Resolved a condition for MS Excel files where borders may not render correctly. (FB10367)
- HD Mode: Resolved a condition for MS Excel files where borders may not render correctly. (FB10368)
- HD Mode: Resolved a condition for MS Excel files where cell alignment may be incorrect. (FB10461)
- HD Mode: Resolved a condition for MS Excel files where cells with formula errors rendered as 0, rather than #N/A. (SF01599720 / FB10350)
- HD Mode: Resolved a condition for MS Excel files where charts may be missing bar outlines. (FB10378)
- HD Mode: Resolved a condition for MS Excel files where conditional formatting may not be applied. (SF01584192 / FB10333)
- HD Mode: Resolved a condition for MS Excel files where headers and footers may not render correctly. (FB10369)
- HD Mode: Resolved a condition for MS Excel files where images may be incorrectly rendered at the bottom of a cell, rather than the top. (SF01599720 / FB10351)
- HD Mode: Resolved a condition for MS Excel files where larger than expected memory may be used when processing Feature/Themed Tables. (FB10398)
- HD Mode: Resolved a condition for MS Excel files where manually sized and positioned chart legends may not render correctly. (FB10373)
- HD Mode: Resolved a condition for MS Excel files where text in scaled text boxes may be the wrong size. (SF01599720 / FB10362)
- HD Mode: Resolved a condition for MS Excel where Feature Tables/Themes were not rendered correctly. (SF01599720 / FB10352)
- HD Mode: Resolved a condition for MS Office files where chart legend may draw over chart plot area. (FB10386)

- HD Mode: Resolved a condition for MS Office files where embedded EMF/WMF images may fail to render at higher DPI. (FB10394)
- HD Mode: Resolved a condition for MS Office files where WingDings may render incorrectly when using a 3rd-party substitute font. (FB10403)
- HD Mode: Resolved a condition for MS PowerPoint files where 3D box shape may render incorrectly. (FB10389)
- HD Mode: Resolved a condition for MS PowerPoint files where EMF/WMF image recoloring via brush was missing. (FB10436)
- HD Mode: Resolved a condition for MS PowerPoint files where flowchart shapes may render incorrectly. (FB10391)
- HD Mode: Resolved a condition for MS PowerPoint files where inner hyperlinks may not work correctly. (SF01602509 / FB10359)
- HD Mode: Resolved a condition for MS PowerPoint files where some shape text was rotated incorrectly. (FB10435)
- HD Mode: Resolved a condition for MS PowerPoint files where table backgrounds may not render correctly. (FB10319)
- HD Mode: Resolved a condition for MS PowerPoint files where table borders may be missing. (FB10390)
- HD Mode: Resolved a condition for MS PowerPoint files where table borders may render thicker than expected. (FB10392)
- HD Mode: Resolved a condition for MS PowerPoint files where Teardrop shapes may render incorrectly (FB10385)
- HD Mode: Resolved a condition for MS PowerPoint files where textbox line spacing may be incorrect. (FB10384)
- HD Mode: Resolved a condition for MS PowerPoint files where the master slide background may be used, when it should not. (FB10348)
- HD Mode: Resolved a condition for MS PowerPoint files where trapezoid images may be incorrectly rendered. (FB10454)
- HD Mode: Resolved a condition for MS Word files where 'Through' text-wrapping was incorrectly rendered. (FB10421)
- HD Mode: Resolved a condition for MS Word files where documents with negative margins would cause a 'Page Limit Exceeded' error. (SF01566039 / FB10347)
- HD Mode: Resolved a condition for MS Word files where Drop Cap may fail to render correctly. (FB10360)
- HD Mode: Resolved a condition for MS Word files where embedded fonts with strikeout or underline may not render correctly. (FB10361)
- HD Mode: Resolved a condition for MS Word files where footnotes may be missing. (FB10433)
- HD Mode: Resolved a condition for MS Word files where gradient fills may not be applied. (FB10423)
- HD Mode: Resolved a condition for MS Word files where legacy drawings may be incorrectly positioned. (FB10426)
- HD Mode: Resolved a condition for MS Word files where legacy shapes may be incorrectly positioned. (FB10422)

- HD Mode: Resolved a condition for MS Word files where legacy shapes may be incorrectly positioned. (FB10424)
- HD Mode: Resolved a condition for MS Word files where legacy text boxes may be incorrectly positioned. (FB10425)
- HD Mode: Resolved a condition for MS Word files where multi-column text may be incorrectly positioned. (FB10431)
- HD Mode: Resolved a condition for MS Word files where processing a corrupt document may cause a hang. (SF01589489 / FB10343)
- HD Mode: Resolved a condition for MS Word files where table borders may be missing when applied via Themes. (FB10376)
- HD Mode: Resolved a condition for MS Word files where table borders may be rendered thicker than expected. (FB10375)
- HD Mode: Resolved a condition for MS Word files where table headers may be missing. (SF01637084 / FB10452)
- HD Mode: Resolved a condition for MS Word files where text may be missing. (FB10442)
- HD Mode: Resolved a condition for MS Word files where text styling may be incorrect when applied via Table Themes. (FB10365)
- HD Mode: Resolved a condition for MS Word files where track changes did not appear as red/strikeout in HTML5. (SF01540719 / FB10240)
- HD Mode: Resolved a condition for PDF files where incorrect font may be used, if the embedded font contained a style name. (FB10412)
- HD Mode: Resolved a condition where HTML_MAX_FILESIZE was incorrectly applied to EML and MSG files. (SF01609873 / FB10382)
- Resolved a condition where JVM may crash on Linux. (SF01598228 / FB10354)
- Text Mode: Resolved a condition for EML message files where upper range ANSI characters may have been lost. (SF01638918 / FB10444)
- Text Mode: Resolved a condition for misidentified legacy files that may cause the engine to hang. (SF01637382 / FB10450)
- Text Mode: Resolved a condition for MS Excel files where larger than expected memory may be used when processing Feature/Themed Tables. (SF01611401 / FB10395)

5.48 Document Filters 11.3 Build 2074 (March 2016)

5.48.1 Enhancements

- Added option HDHTML_IMAGE_FORMAT to control embedded image types in HTML5 output, can be PNG, JPG or AUTO.
- Added option HTML_IMAGE_ATTRIBUTES to include data attribute containing embedded image names.
- Added option MARKUP_OPTIONS to control how track-changes are rendered when SHOWHIDDEN is enabled.

5.48.2 Updates

- HD Mode: Added option HTML_IMAGE_ATTRIBUTES to include data attribute containing the embedded name of each image. (SF01548605 / FB10265)
- HD Mode: Improved performance of rendering embedded EMF/WMF images. (FB10325)
- HD Mode: Resolved a condition for Hancom Hancell files may fail to load. (FB10280)
- HD Mode: Resolved a condition for MS Excel files where charts may be missing. (SF01511524 / FB10290)
- HD Mode: Resolved a condition for MS Excel files where document may hang. (SF01549043 / FB10257)
- HD Mode: Resolved a condition for MS Excel files where text in WordArt may be missing. (FB10245)
- HD Mode: Resolved a condition for MS Office files where chart data labels may appear in the wrong position. (FB9165)
- HD Mode: Resolved a condition for MS Office files where chart data labels may be missing. (SF01562062 / FB10278)
- HD Mode: Resolved a condition for MS Office files where chart labels with new lines may not wrap correctly. (FB9160)
- HD Mode: Resolved a condition for MS Office files where chart legends may be incorrectly placed. (SF01562062 / FB10279)
- HD Mode: Resolved a condition for MS Office files where charts with background images may not render correctly. (FB10243)
- HD Mode: Resolved a condition for MS Office files where Home Plate shape may render incorrectly. (FB10248)
- HD Mode: Resolved a condition for MS Office files where shape colors may be incorrect on AIX, Linux PPC and Solaris Sparc. (FB10281)
- HD Mode: Resolved a condition for MS Office files where Wingdings arrows may not render correctly. (FB10264)
- HD Mode: Resolved a condition for MS PowerPoint files where footnotes may be rendered on the wrong page. (FB10299)
- HD Mode: Resolved a condition for MS PowerPoint files where grouped shapes may not rotate correctly. (FB10327)
- HD Mode: Resolved a condition for MS PowerPoint files where grouped shapes may not rotate correctly. (FB10328)

- HD Mode: Resolved a condition for MS PowerPoint files where images may have the wrong rotation. (FB10296)
- HD Mode: Resolved a condition for MS PowerPoint files where Notched Circular Arrows may not render correctly. (FB10273)
- HD Mode: Resolved a condition for MS PowerPoint files where numbered lists may use the wrong font. (FB10259)
- HD Mode: Resolved a condition for MS PowerPoint files where pages may fail to render when using FIRST_PAGE option. (SF01548530 / FB10263)
- HD Mode: Resolved a condition for MS PowerPoint files where shapes with grayscale effect may not render in grayscale. (FB10275)
- HD Mode: Resolved a condition for MS PowerPoint files where slides may be rendered with extra bullets. (FB10295)
- HD Mode: Resolved a condition for MS PowerPoint files where table backgrounds may not render correctly. (FB10319)
- HD Mode: Resolved a condition for MS PowerPoint files where text in rotated shapes may be misaligned. (FB10288)
- HD Mode: Resolved a condition for MS PowerPoint files where text in rotated shapes may be placed incorrectly. (FB10303)
- HD Mode: Resolved a condition for MS PowerPoint files where text may not be correctly aligned within a text box. (FB10301)
- HD Mode: Resolved a condition for MS Word files where 'Page limit exceeded' may erroneously be returned. (SF01566039 / FB10292)
- HD Mode: Resolved a condition for MS Word files where 'Washed out' effect was not applied to images. (FB10212)
- HD Mode: Resolved a condition for MS Word files where certain documents may cause an access violation. (FB10297)
- HD Mode: Resolved a condition for MS Word files where documents containing multiple columns may fail to process. (SF01580191 / FB10329)
- HD Mode: Resolved a condition for MS Word files where gradient backgrounds may use the wrong color. (SF01537021 / FB10216)
- HD Mode: Resolved a condition for MS Word files where HTML5 output may not render deleted text as strike-through when SHOWHIDDEN=ON. (SF1540719 / FB10239)
- HD Mode: Resolved a condition for MS Word files where page margins may be incorrect. (SF01555606 / FB10272)
- HD Mode: Resolved a condition for MS Word files where styles spanning fields may not be correctly applied. (SF01548499 / FB10267)
- HD Mode: Resolved a condition for MS Word files where text with negative alignment may not position correctly. (FB9966)
- HD Mode: Resolved a condition for MS Word files where track changes did not appear as red/strikeout in HTML5. (SF01540719 / FB10240)

- HD Mode: Resolved a condition for MS Word files where watermarks may not be positioned correctly. (FB10306)
- HD Mode: Resolved a condition for PDF files where text may fail to render. (SF01531577 / FB10262)
- Resolved a condition for 7zip files where files greater than 2GB would fail to extract. (SF01583972 / FB10334)
- Resolved a condition for PDF files where files containing a corrupt Unicode CMAP may extract bad text. (SF01490930 / FB10109)
- Resolved a condition where temp files may not be deleted on UNIX based operating systems. (SF1575235 / FB10339)

5.49 Document Filters 11.3 Build 2040 (January 2016)

5.49.1 Enhancements

- Added support for XZ archive files.
- Added support for RAR 5 archive files.

5.49.2 Updates

- Add DBF_TABLE_VIEW option to control how dBASE files are extracted. (SF767836 / FB8696)
- Added Strong Name signing to Perceptive.DocumentFilters.dll (FB9901)
- Added support for RAR 5 archive files. (SF01329218 / FB9876)
- Added support for XZ archive files. (SF1334381 / FB9893)
- HD Mode: Resolved a condition for DGN files where complex drawings may take longer than usual time to process. (SF01537204 / FB10218)
- HD Mode: Resolved a condition for EML files where charset may be incorrectly identified. (SF01531558 / FB10206)
- HD Mode: Resolved a condition for HTML files where lines may draw over each other when CSS is incorrectly formatted. (FB9861)
- HD Mode: Resolved a condition for MS Excel files where files created by third-party applications may fail to convert. (SF01514358 / FB10154)
- HD Mode: Resolved a condition for MS Excel files where gradients may render with the wrong colors. (SF798279 / FB8817)
- HD Mode: Resolved a condition for MS Excel files where text in drawings may be inverted. (FB10126)
- HD Mode: Resolved a condition for MS Excel files where tri-state date masks may not render correctly. (SF01512795 / FB10167)
- HD Mode: Resolved a condition for MS Office files where files created by third-party applications may fail to convert. (SF01512791 / FB10155)
- HD Mode: Resolved a condition for MS Office files where nested Math Equation objects may cause a SegFault. (FB10189)
- HD Mode: Resolved a condition for MS Office files where Notched Arrow shapes may not render correctly. (FB10127)
- HD Mode: Resolved a condition for MS Office files where Pie Charts may draw over the chart's title. (SF01536212 / FB10233)
- HD Mode: Resolved a condition for MS Office files where SmartArt does not position correctly. (FB10203)
- HD Mode: Resolved a condition for MS Office files where text in Flow Chart shapes may not be centered correctly. (FB10227)

- HD Mode: Resolved a condition for MS Office files where text may be missing from SmartArt diagrams. (FB10193)
- HD Mode: Resolved a condition for MS Office files where text may not be centered correctly in Callout shapes. (FB10222)
- HD Mode: Resolved a condition for MS Office files where text may not be correctly centered in shapes. (FB10204)
- HD Mode: Resolved a condition for MS Office files where the line height for blank lines may be incorrectly calculated. (FB9945)
- HD Mode: Resolved a condition for MS PowerPoint files where distance between bullets and text may be incorrect. (FB10207)
- HD Mode: Resolved a condition for MS PowerPoint files where horizontal flipping may not be correctly applied to images. (FB10237)
- HD Mode: Resolved a condition for MS PowerPoint files where incorrect theme colors may be used. (FB10200)
- HD Mode: Resolved a condition for MS PowerPoint files where list items may be missing a bullet. (FB10219)
- HD Mode: Resolved a condition for MS PowerPoint files where rotated shape text may not render correctly. (FB10151)
- HD Mode: Resolved a condition for MS PowerPoint files where shapes may be incorrectly rotated. (FB10246)
- HD Mode: Resolved a condition for MS PowerPoint files where tables with Row Spans may not render correctly. (FB10176)
- HD Mode: Resolved a condition for MS PowerPoint files where text may not be correctly centered in hexagon shapes. (FB10234)
- HD Mode: Resolved a condition for MS PowerPoint where image borders may render with the wrong color. (FB10191)
- HD Mode: Resolved a condition for MS Visio files where images with transparent backgrounds may not render correctly. (FB9077)
- HD Mode: Resolved a condition for MS Word files where a section break inside track-changes may not be processed correctly. (SF01531307 / FB10225)
- HD Mode: Resolved a condition for MS Word files where documents generated by third-party applications may cause a SegFault. (SF1536616 / FB10229)
- HD Mode: Resolved a condition for MS Word files where files saved as "Strict Open XML" may fail to convert. (SF01512787 / FB10159)
- HD Mode: Resolved a condition for MS Word files where files saved as "Word 2003 XML" may fail to convert. (SF01511552 / FB10158)
- HD Mode: Resolved a condition for MS Word files where footnotes may cause a new page to be created. (SF01512790 / FB10162)

- HD Mode: Resolved a condition for MS Word files where footnotes on columns may be positioned incorrectly. (SF01512790 / FB10164)
- HD Mode: Resolved a condition for MS Word files where HTML5 output may not render deleted text as strike-through when SHOWHIDDEN=ON. (SF1540719 / FB10239)
- HD Mode: Resolved a condition for MS Word files where list numbering resets may not be processed correctly. (SF01531309 / FB10224)
- HD Mode: Resolved a condition for MS Word files where runs of underscores may be too wide. (SF01512791 / FB10161)
- HD Mode: Resolved a condition for MS Word files where text shading may not be correctly applied. (SF01512790 / FB10163)
- HD Mode: Resolved a condition for MS Word files where word-fields may not render. (SF01537021 / FB10217)
- HD Mode: Resolved a condition for multi-page TIFF files where pages with different DPI may not scale correctly when using GRAPHIC_DPI=AUTO. (SF01538767 / FB10231)
- HD Mode: Resolved a condition for TIFF files where images without a stored DPI may render smaller than expected. (SF1545204 / FB10247)
- HD Mode: Resolved a condition for WMF files where drawings with Unicode symbols may not render correctly. (SF01531314 / FB10214)
- HD Mode: Resolved a condition for WMF files where text may not render correctly. (FB10215)
- HD Mode: Resolved a condition where files with fractional font sizes may render smaller than expected. (FB10178)
- Improved processing time of some complex PDF files. (SF01506963 / FB10169)
- Resolved a condition for Archive files where archives with file names > 4K may cause a SegFault. (SF01510012 / FB10190)
- Resolved a condition for MS Excel files where protected workbooks may fail to convert. (SF01531177 / FB10197)
- Resolved a condition where files may be incorrectly identified as dBASE. (SF01531709 / FB10198)

5.50 Document Filters 11.3 Build 1998 (December 2015)

5.50.1 Enhancements

Conversion profiles

Conversion profiles allow for different options to be specified based on (1) the type of input file or document, (2) the desired conversion mode (text, classic HTML or HD) and (3) the desired output format. This is convenient, for example, if different options are needed for inputs that are bitmaps versus documents.

See the comments in ISYS11df.ini for full details and example usage.

Email date formats and timezones

To specify date format and timezone for HD email conversion using the EMAIL_DATE_FORMAT and EMAIL_DATE_TIMEZONE_OFFSET options.

Image extraction

To extract images (of HD supported formats) during text-mode conversions using the ENUMERATE_IMAGES option. Embedded images will appear as attachments.

Image compression

To minimize the size of TIFF and PDF output files, use the TIFF_COMPRESSION=AUTO, PDF_COMPRESSION=AUTO and JPEG_QUALITY=AUTO options.

5.50.2 Updates

- Added ability to specify datetime format (EMAIL_DATE_FORMAT) and timezone (EMAIL_DATE_TIMEZONE_OFFSET) when rendering email files. (SF900700 / FB9145)
- HD Mode: Added hyperlink support for MS PowerPoint files. (FB10099)
- HD Mode: Resolved a condition for MS Excel files where files with 'Password to Edit' may fail to process. (SF1485105 / FB10128)
- HD Mode: Resolved a condition for MS Excel files where text in drawings may be inverted. (FB10126)
- HD Mode: Resolved a condition for MS PowerPoint files where conversion may hang. (SF01476759 / FB10091)
- HD Mode: Resolved a condition for MS PowerPoint files where gradients may render incorrectly. (FB10063)

- HD Mode: Resolved a condition for MS PowerPoint files where hidden master content may be shown even when SHOWHIDDEN is off. (FB10118)
- HD Mode: Resolved a condition for MS PowerPoint files where margins may not be applied. (FB10022)
- HD Mode: Resolved a condition for MS PowerPoint files where text in tables may not be aligned correctly. (FB10098)
- HD Mode: Resolved a condition for Open Office Presentation files where footers may not render. (FB10072)
- HD Mode: Resolved a condition where generated PDFs may render incorrectly with Unicode characters. (FB10083)
- HD Mode: Resolved a condition where MS Excel files may cause a segmentation fault. (SF01474251 / FB10086)
- HD Mode: Resolved a condition where MS PowerPoint files where date labels in charts may render incorrectly. (FB10096)
- HD Mode: Resolved a condition where out of memory errors may not be reported. (SF01444108 / FB10038)
- Resolved a condition for MS Visio files where sub-enumerating attachments may fail. (SF01506433 / FB10149)
- Resolved a condition for Paradox files where extraction may hang. (SF1501112 / FB10138)

5.51 Document Filters 11.2 Build 1962 (November 2015)

5.51.1 Updates

- HD Mode: Implemented DOCUMENT_HEADERS/FOOTERS option for MS Excel and MS PowerPoint files. (SF01464615 / FB10065)
- HD Mode: Resolved a condition where PDF files may fail to render at higher DPIs. (SF01463628 / FB10062)
- HD Mode: Resolved a condition where WMF files may fail to render. (FB10060)
- HD Mode: Resolved a condition for MS Office files where text may be misaligned in callout shapes. (FB10058)
- HD Mode: Resolved a condition for MS Office files where chart series colors may be incorrect. (FB10056)
- Resolved a condition for MS PowerPoint files where slide page numbers were not rendered. (SF01455192 / FB10053)
- HD Mode: Resolved a condition for MS Office files where Brace shapes may render incorrectly. (FB10051)
- HD Mode: Resolved a condition for MS PowerPoint files where images on Master Slides may not be present on rendered slides. (FB10044)
- HD Mode: Resolved a condition for MS PowerPoint files where shapes may render with the wrong color. (FB10043)
- HD Mode: Resolved a condition for Image files where quality may degrade if image is scaled. (FB10037)
- HD Mode: Resolved a condition for black and white TIFF files where colors may be inverted when converting to PDF. (FB10036)
- HD Mode: Resolved a condition where TIFF_BYTEORDER may not be used for all TIFF output types. (SF01432340 / FB10033)
- Resolved a condition where some self-extracting archives may fail to identify. (SF01433358 / FB10028)
- HD Mode: Resolved a condition where setting HTML_MAX_FILESIZE may still result in a document being processed. (SF01443957 / FB10024)
- HD Mode: Resolved a condition where setting TIFF_COMPRESSION=GROUP3..6 without setting TIFF_BITSPERPIXEL=1 would result in an invalid TIFF file. (SF01443808 / FB10023)
- Resolved a condition for PDF files where files with large blocks of script may not be identified correctly. (SF1439141 / FB10014)
- HD Mode: Resolved a condition for MS PowerPoint files where line height may be calculated incorrectly. (FB10013)
- HD Mode: Added support for PBM, PGM and PPM as an input file type. (FB10010)
- HD Mode: Resolved a condition for MS Office files where cylinder objects may render incorrectly. (FB10007)
- HD Mode: Resolved a condition for MS PowerPoint files where text alignment on Master Slides may not be reflected on Child Slides. (FB10006)
- Resolved a condition for TEXT files where they may be mis-identified as QuickBooks. (SF01433464 / FB10005)

- HD Mode: Resolved a condition for MS PowerPoint files where gradient backgrounds may not render. (FB10004)
- HD Mode: Resolved a condition for MS PowerPoint files where text in Callouts may not center correctly. (FB10001)
- Resolved a condition for MS Excel files produced by third-party applications may cause a crash. (SF1419409 / FB9999)
- Resolved a condition for EML files without an extension, and have been routed through multiple mail servers, may not identify correctly. (SF1430718 / FB9998)
- HD Mode: Resolved a condition for MS Office files where text contained in rounded rectangles may not be positioned correctly. (FB9992)
- HD Mode: Resolved a condition for MS Office files where line heights of blank lines may not be calculated correctly. (FB9990)
- HD Mode: Resolved a condition for MS PowerPoint files where auto-fit text may not be placed correctly. (FB9989)
- HD Mode: Resolved a condition for MS Word files where image clipping may not be applied correctly. (FB9988)
- HD Mode: Resolved a condition for MS PowerPoint files where text-breaks may not be applied. (FB9986)
- HD Mode: Resolved a condition for MS Office files where bent shape connectors may render incorrectly. (FB9985)
- HD Mode: Resolved a condition for MS PowerPoint files where multi-column shapes may not render correctly. (FB9974)
- HD Mode: Resolved a condition for MS PowerPoint files where multi-level lists may apply the wrong font size. (FB9973)
- HD Mode: Resolved a condition for MS Office files where embedded TIFF images may not render on all platforms. (FB9972)
- HD Mode: Resolved a condition for MS Office files where the scroll shape may render incorrectly. (FB9971)
- HD Mode: Resolved a condition for MS Office files where grouped shapes may not align correctly. (FB9970)
- HD Mode: Resolved a condition for MS Office files where some embedded EMF images may not render. (FB9969)
- HD Mode: Resolved a condition for MS Office files where vertical shape text may be rendered as horizontal. (FB9967)
- HD Mode: Resolved a condition for MS Excel files where pie chart labels may be missing. (FB9965)
- Resolved a condition for MS Excel files, generated by third-party applications, may fail to convert. (FB9964)
- Resolved a condition for MS Excel files where full page charts may be placed incorrectly. (FB9963)
- HD Mode: Resolved a condition where SetOpacity with a value of 255, rendered incorrectly. (SF1386233 / FB9958)
- HD Mode: Resolved a condition for MS PowerPoint files where custom tab stops may not be applied. (FB9956)
- HD Mode: Resolved a condition for MS Office files where the line height for blank lines may be incorrectly calculated. (FB9945)
- Improved performance of converting MS Excel files to CSV, TSV and CST (SF01365277 / FB9940)

- Add the ability to sub-enumerate images on HD supported formats (ENUMERATE_IMAGES) (SF1348180 / FB9923)
- Resolved a condition where some EML files may be misidentified. (SF01317124 / FB9836)
- HD Mode: Resolved a condition where charts may render with incorrect colors. (FB9696)
- HD Mode: Resolved a condition for MS Visio files where .EMF images may not render correctly on non-Windows platforms. (FB9672)
- HD Mode: Resolved a condition for MS Office files where 3D area charts may render out of order. (FB9660)
- HD Mode: Resolved a condition for MS Excel files where data labels may be missing from charts. (FB9659)
- HD Mode: Resolved a condition for MS Office files where icons may have a dark shadow. (FB8885)
- HD Mode: added ability to specify PHOTOMETRIC_INTERPRETATION when creating TIFF files (TIFF_PHOTOMETRIC_MINISBLACK) (SF00705032 / FB8622)

5.52 Document Filters 11.2 Build 1914 (September 2015)

5.52.1 Updates

- Added Strong Name signing to Perceptive.DocumentFilters.dll (FB9901)
- Added support to extract Open Office User Fields as metadata. (FB9446)
- HD Mode: Added new HTML class to comment anchors (idf-comment-anchor) when converting MS PowerPoint files to HTML5. (SF1082358 / FB9584)
- HD Mode: General chart rendering improvements. (SF01194065 / FB9846)
- HD Mode: General chart rendering improvements. (SF01194069 / FB9511)
- HD Mode: Improved handling of "Center Across Selection" in MS Excel files. (FB9152)
- HD Mode: Resolved a condition for EML and MSG files where conversion may fail if message contains large nested HTML tags. (SF1331146 / FB9880)
- HD Mode: Resolved a condition for EML and MSG files where redaction bounding box may be incorrectly calculated. (SF01338340 / FB9903)
- HD Mode: Resolved a condition for IGR_Canvas_Rotation where rotation may not be applied as expected. (SF873909 / FB9072)
- HD Mode: Resolved a condition for MS Excel charts where an incorrect font could be used on legend and axis labels. (FB9553)
- HD Mode: Resolved a condition for MS Excel files where text may render past the extent of the cell. (SF1194060 / FB9496)
- HD Mode: Resolved a condition for MS PowerPoint files where text within arrows may not be placed correctly. (FB9952)
- HD Mode: Resolved a condition for MS PowerPoint files where arrow heads may not render correctly. (FB9934)
- HD Mode: Resolved a condition for MS PowerPoint files where arrow heads may render larger than expected. (FB9947)
- HD Mode: Resolved a condition for MS PowerPoint files where bent connectors may not render correctly. (FB9941)
- HD Mode: Resolved a condition for MS PowerPoint files where borders on images may not render correctly when the image is rotated. (FB9922)
- HD Mode: Resolved a condition for MS PowerPoint files where bullet lists may apply incorrect indentation. (FB9918)
- HD Mode: Resolved a condition for MS PowerPoint files where curved arrows may not render correctly. (FB9928)
- HD Mode: Resolved a condition for MS PowerPoint files where incorrect font may be used when rendering drawings. (FB9916)

- HD Mode: Resolved a condition for MS PowerPoint files where RoundDiagonalCornerRectangles may render with all corners rounded. (FB9949)
- HD Mode: Resolved a condition for MS PowerPoint files where rounded corners may be larger than expected. (FB9943)
- HD Mode: Resolved a condition for MS PowerPoint files where semi-transparent fills may not render correctly. (FB9919)
- HD Mode: Resolved a condition for MS PowerPoint files where tab indented content may be misaligned. (FB9951)
- HD Mode: Resolved a condition for MS PowerPoint files where table cell shading may not be applied. (FB9927)
- HD Mode: Resolved a condition for MS PowerPoint files where text shapes may be incorrectly rotated. (FB9899)
- HD Mode: Resolved a condition for MS Word files where documents with track-changes may not render correctly. (FB9910)
- HD Mode: Resolved a condition for MS Word where comments may incorrectly repeat. (FB9909)
- HD Mode: Resolved a condition where chart legends may render incorrectly. (FB9695)
- HD Mode: Resolved a condition where charts may render with incorrect colors. (FB9696)
- Improved error handling when system drive is running low on space. (SF01217776 / FB9569)
- Resolved a condition for 7-zip files where extraction may fail on archives created with bzip2 compression. (SF01350136 / FB9921)
- Text Mode: Resolved a condition for MSG files, containing iso-2022-jp characters, may not convert correctly on non-Windows platforms. (SF01365322 / FB9937)

5.53 Document Filters 11.2 Build 1884 (September 2015)

5.53.1 Enhancements

- Added ability to extract Open Office spreadsheets in EXCELMODE=CSV,TSV, or CST. (FB9863)
- Added ability to specify an HTML attribute for HDHTML_INCLUDE_WORD_INDEXES. (SF01194108 / FB9528)
- Added identification support for Unix AR Archives (filetype 179). (FB9692)
- Added IMAGE_PROCESSOR option to control which internal image renderer is used on MS Windows. (SF01257462 / FB9687)
- Added support for Windows Icon ICO files. (SF01204454 / FB9527)
- Added the ability to control the amount of text passed to the character encoding detection engine (ENCODING_DETECTION_BUFFER) (SF1321142 / FB9862)
- Added the ability to extract Duration metadata from WAV files. (SF01231092 / FB9611)

5.53.2 Updates

- HD Mode: Added basic support for WordArt in OpenOffice HD. (FB9128)
- HD Mode: Added new EXCELMODE=CST, where cell contents (commas and quotes) are not escaped. (SF01177544 / FB9456)
- HD Mode: Added support for custom shapes and graphic styling in OpenOffice HD. (FB9125)
- HD Mode: Added the ability to process Duotone, Brightness, and Contrast color modifications in MS Office file formats. (FB9664)
- HD Mode: Improved handling of gradient colors in MS Excel charts. (FB9564)
- HD Mode: Improved page clipping and overflowing for OpenOffice ODS documents. (FB9365)
- HD Mode: Improved support for embedded drawings in Open Office documents. (FB9126)
- HD Mode: Resolved a condition for EMF files where mirrored vector graphics may not render correctly. (FB9726)
- HD Mode: Resolved a condition for EML files where 7-bit encoded non-US character sets may not decode correctly. (SF01177594 / FB9576)
- HD Mode: Resolved a condition for EML files where images may not render correctly using IMAGEURL. (SF1272930 / FB9731)
- HD Mode: Resolved a condition for HTML files where alpha transparency may not be applied when converting to an image. (FB9887)
- HD Mode: Resolved a condition for HTML files where lines may draw over each other when CSS is incorrectly formatted. (FB9861)
- HD Mode: Resolved a condition for ICS files where metadata begin markers may not be present. (FB9535)

- HD Mode: Resolved a condition for MS Excel (.XLS) files where files containing brackets may hang during extraction. (SF01287137 / FB9759)
- HD Mode: Resolved a condition for MS Excel files where cell run-styles were not applied. (SF1292042 / FB9786)
- HD Mode: Resolved a condition for MS Excel files where chart labels on the horizontal and vertical axes may not render correctly. (SF01194065 / FB9505)
- HD Mode: Resolved a condition for MS Excel files where column borders may not render correctly in HTML5. (FB9119)
- HD Mode: Resolved a condition for MS Excel files where fractions may not render correctly. (SF01287095 / FB9760)
- HD Mode: Resolved a condition for MS Excel files where worksheets with "password to edit" may not extract. (SF1133559 / FB9439)
- HD Mode: Resolved a condition for MS Excel files, created by 3rd-party applications, where conversion may fail. (SF01327818 / FB9878)
- HD Mode: Resolved a condition for MS Office files where arrows may not render correctly for some drawings. (FB9745)
- HD Mode: Resolved a condition for MS PowerPoint 2007 files where table cell alignment may render incorrectly. (FB9767)
- HD Mode: Resolved a condition for MS PowerPoint files where drawings may be incorrectly scaled. (FB9879)
- HD Mode: Resolved a condition for MS PowerPoint files where images may not render with the correct rotation. (FB9674)
- HD Mode: Resolved a condition for MS PowerPoint files where page/slide numbers may not render correctly. (FB9729)
- HD Mode: Resolved a condition for MS PowerPoint files where some embedded images may fail to render. (FB9869)
- HD Mode: Resolved a condition for MS PowerPoint files where vertically aligned text may not render correctly. (FB9754)
- HD Mode: Resolved a condition for MS PowerPoint PPT/PPS files where a table with an unequal number of rows may cause an access violation. (SF01272085 / FB9716)
- HD Mode: Resolved a condition for MS Visio files where EMF images may not render correctly on non-Windows platforms. (FB9672)
- HD Mode: Resolved a condition for MS Word DOCX files where images may render off the page due to incorrectly applied text wrapping. (FB9697)
- HD Mode: Resolved a condition for MS Word files where extraction may result in an erroneous "Page limit exceeded" error. (SF1275155 / FB9720)
- HD Mode: Resolved a condition for MS Word files where image rotation and flipping may not be applied correctly. (FB8802)

- HD Mode: Resolved a condition for MS Word files where paragraphs and numbered lists may not be numbered correctly. (FB9753)
- HD Mode: Resolved a condition for MS Word files, created by 3rd-party applications, may fail to convert to HD. (SF01328665 / FB9873)
- HD Mode: Resolved a condition for MSG files where conversion may fail. (SF01328513 / FB9875)
- HD Mode: Resolved a condition for OpenOffice files where lines (shapes) may not render correctly. (FB8727)
- HD Mode: Resolved a condition for OpenOffice ODS files where sheet names may not render. (FB9131)
- HD Mode: Resolved a condition for OpenOffice ODS files where the DPI setting may negatively affect image size and cropping. (FB9297)
- HD Mode: Resolved a condition for TIFF files where setting compression to Group4 may erroneously encode with Group3. (SF1275300 / FB9721)
- HD Mode: Resolved a condition for WordPerfect documents where text may not flow around images correctly. (SF01225395 / FB9698)
- HD Mode: Resolved a condition where IGR_Canvas_Rotation incorrectly used radians instead of degrees. (FB9066)
- HD Mode: Resolved a condition where IGR_Get_Page_Attribute may not return the original document's page dimensions. (SF00764881 / FB8695)
- Improved memory consumption when processing Microsoft Office documents in text-mode. (SF1314064 / FB9827)
- Resolved a condition for EML files where unprintable characters may not display correctly. (SF01217754 / FB9619)
- Resolved a condition for MS Excel files where cells containing multiple formatting styles may not render correctly. (SF01194106 / FB9489)
- Resolved a condition for MS Excel files where files exported from Google Sheets may not process correctly in CSV mode. (FB9860)
- Resolved a condition for PDF files where multi-column paragraphs may split incorrectly causing the paragraph-per-document limit to be exceeded. (SF01261873 / FB9709)
- Resolved a condition for PDF files where warnings about nedpfree may be written to the console. (SF1118158 / FB9843)
- Resolved a condition for UCS2 text files where line feeds were not being applied correctly. (SF1333693 / FB9891)
- Resolved a condition for UTF-8 Text files without a Byte-Order-Marker that may process incorrectly. (SF1321141 / FB9855)
- Resolved a condition where corrupt PNG files may cause IGR_Open to freeze. (FB9702)
- Resolved a condition where corrupt PST files may hang the process. (SF01333286 / FB9890)
- Resolved a condition where MSG files containing Hebrew character encoding may not extract correctly. (SF01296237 / FB9796)
- Resolved a condition where MSO files were misidentified as XYwrite. (SF01306790 / FB9815)

- Resolved a condition where OutBufSize may be uninitialized when IGR_Get_Page_Text returns IGR_NO_MORE. (SF1232918 / FB9732)
- Resolved a condition where text extraction may hang with a .patch file if using the Python API. (SF01273433 / FB9717)
- Text Mode: Resolved a condition for corrupt EML files where body text may incorrectly render in metadata. (SF1273604 / FB9718)
- Text Mode: Resolved a condition for MS Excel files where full-sheet charts may not extract text. (SF01319145 / FB9853)

5.54 Document Filters 11.2 Build 1858 (August 2015)

5.54.1 Enhancements

- Added basic support for WordArt in OpenOffice HD. (FB9128)
- Added identification support for Unix AR Archives (filetype 179). (FB9692)
- Added support for custom shapes and graphic styling in OpenOffice HD. (FB9125)
- Added the IMAGE_PROCESSOR option to control which internal image renderer is used on Windows. (SF01257462 / FB9687)

5.54.2 Updates

- HD Mode: Added the ability to process Duotone, Brightness, and Contrast color modifications in MS Office file formats. (FB9664)
- HD Mode: Improved handling of gradient colors in MS Excel charts. (FB9564)
- HD Mode: Improved page clipping and overflowing for OpenOffice .ODS documents. (FB9365)
- HD Mode: Resolved a condition for .EMF files where mirrored vector graphics may not render correctly. (FB9726)
- HD Mode: Resolved a condition for .EML files where 7-bit encoded non-US character sets may not decode correctly. (SF01177594 / FB9576)
- HD Mode: Resolved a condition for .EML files where images may not render correctly using IMAGEURL. (SF1272930 / FB9731)
- HD Mode: Resolved a condition for .ICS files where metadata begin markers may not be present. (FB9535)
- HD Mode: Resolved a condition for MS Excel .XLS files where files containing brackets may hang during extraction. (SF01287137 / FB9759)
- HD Mode: Resolved a condition for MS Excel files where chart labels on the horizontal and vertical axes may not render correctly. (SF01194065 / FB9505)
- HD Mode: Resolved a condition for MS Excel files where column borders may not render correctly in HTML5. (FB9119)
- HD Mode: Resolved a condition for MS Excel files where fractions may not render correctly. (SF01287095 / FB9760)
- HD Mode: Resolved a condition for MS Excel files where worksheets with "password to edit" may not extract. (SF1133559 / FB9439)
- HD Mode: Resolved a condition for MS PowerPoint .PPT/.PPS files where a table with an unequal number of rows may cause an access violation. (SF01272085 / FB9716)
- HD Mode: Resolved a condition for MS PowerPoint files where images may not render with the correct rotation. (FB9674)

- HD Mode: Resolved a condition for MS PowerPoint files where page/slide numbers may not render correctly. (FB9729)
- HD Mode: Resolved a condition for MS Visio files where .EMF images may not render correctly on non-Windows platforms. (FB9672)
- HD Mode: Resolved a condition for MS Word .DOCX files where images may render off the page due to incorrectly applied text wrapping. (FB9697)
- HD Mode: Resolved a condition for MS Word files where extraction may result in an erroneous "Page limit exceeded" error. (SF1275155 / FB9720)
- HD Mode: Resolved a condition for OpenOffice .ODS files where the DPI setting may negatively affect image size and cropping. (FB9297)
- HD Mode: Resolved a condition for TIFF files where setting compression to Group4 may erroneously encode with Group3. (SF1275300 / FB9721)
- Resolved a condition for .EML files where unprintable characters may incorrectly display. (SF01217754 / FB9619)
- Resolved a condition for OpenOffice .ODS files where sheet names may not render. (FB9131)
- Resolved a condition for OpenOffice files where lines (shapes) may not render correctly. (FB8727)
- Resolved a condition for PDF files where multi-column paragraphs may split incorrectly causing the paragraph-per-document limit to be exceeded. (SF01261873 / FB9709)
- Resolved a condition where corrupt PNG files may cause IGR_Open to freeze. (FB9702)
- Resolved a condition where IGR_Canvas_Rotation incorrectly used radians instead of degrees. (FB9066)
- Resolved a condition where IGR_Get_Page_Attribute may not return the original document's page dimensions. (SF00764881 / FB8695)
- Resolved a condition where OutBufSize may be uninitialized IGR_Get_Page_Text returns IGR_NO_MORE. (SF1232918 / FB9732)
- Resolved a condition where text extraction may hang with a .patch file if using the Python API. (SF01273433 / FB9717)
- Text Mode: Resolved a condition for corrupt .EML files where body text may incorrectly render in metadata. (SF1273604 / FB9718)

5.55 Document Filters 11.2 Build 1808 (June 2015)

5.55.1 Enhancements

- Added extraction and HD support for MS Outlook for Mac 2011. (FB9592)
- Added the ability to extract Duration metadata from .WAV files. (SF01231092 / FB9611)

5.55.2 Updates

- HD Mode: Improved scaling of embedded EMF/WMF images. (FB9598)
- HD Mode: Resolved a condition for .EML files where 7-bit encoded non-US character sets may not decode correctly. (SF01177594 / FB9576)
- HD Mode: Resolved a condition for .EML files where content may not render correctly. (FB9629)
- HD Mode: Resolved a condition for .EML files where inline images may incorrectly show as Attachments. (SF1194085 / FB9594)
- HD Mode: Resolved a condition for .EML files where quotation marks may not render correctly. (SF01201153 / FB9595)
- HD Mode: Resolved a condition for .EMLX files where content may not render correctly. (FB9630)
- HD Mode: Resolved a condition for .MSG files where inline images may not render correctly. (SF1218715 / FB9566)
- HD Mode: Resolved a condition for .MSG files where keyword metadata values may have extraneous commas. (FB9632)
- HD Mode: Resolved a condition for .MSG files where the keywords metadata field may not render correctly. (FB9631)
- HD Mode: Resolved a condition for .RTF files where IGR_Extract_Page_Image_Stream may fail. (SF1225111 / FB9610)
- HD Mode: Resolved a condition for charts where automatically generated titles may not render. (FB9599)
- HD Mode: Resolved a condition for charts where data labels may not render correctly. (FB9628)
- HD Mode: Resolved a condition for charts where legends may incorrectly render horizontal instead of vertical. (FB9642)
- HD Mode: Resolved a condition for charts where text labels may incorrectly render as numbers. (FB9635)
- HD Mode: Resolved a condition for HTML files where Unicode characters may not render correctly. (SF01218855 / FB9565)
- HD Mode: Resolved a condition for MS Excel files where cell background colors may not render on Linux 64-bit. (FB9597)
- HD Mode: Resolved a condition for MS Excel files where cell contents may be cut off by the grid lines. (SF1199488 / FB9537)

- HD Mode: Resolved a condition for MS Excel files where cell lines may not render correctly. (FB9627)
- HD Mode: Resolved a condition for MS Excel files where dates may not render correctly or may be missing. (SF1240123 / FB9644)
- HD Mode: Resolved a condition for MS Office files where bracket and double bracket shapes may not render correctly. (FB9661)
- HD Mode: Resolved a condition for MS PowerPoint .PPT files where hidden objects may not recognize the SHOWHIDDEN setting. (FB9578)
- HD Mode: Resolved a condition for MS PowerPoint .PPTX files where image transparency may not render correctly. (FB9633)
- HD Mode: Resolved a condition for MS PowerPoint .PPTX files where placeholder styles may not process correctly. (FB9670)
- HD Mode: Resolved a condition for MS PowerPoint files where background colors may not render correctly. (FB9572)
- HD Mode: Resolved a condition for MS PowerPoint files where bullet colors may not render correctly. (FB9574)
- HD Mode: Resolved a condition for MS PowerPoint files where default text styles may not apply correctly. (FB9653)
- HD Mode: Resolved a condition for MS PowerPoint files where hidden text may incorrectly show. (FB9567)
- HD Mode: Resolved a condition for MS PowerPoint files where images may incorrectly stretch. (FB9634)
- HD Mode: Resolved a condition for MS PowerPoint files where line spacing may not render correctly. (FB9636)
- HD Mode: Resolved a condition for MS PowerPoint files where table cells may not render correctly on subsequent page loads. (FB9575)
- HD Mode: Resolved a condition for MS PowerPoint files where text borders may incorrectly render. (FB9637)
- HD Mode: Resolved a condition for MS PowerPoint files where the theme may not apply correctly. (FB9657)
- HD Mode: Resolved a condition for MS Project files where metadata may contain extraneous characters. (SF1232292 / FB9620)
- HD Mode: Resolved a condition for MS Visio files where text may apply z-order incorrectly. (FB9591)
- HD Mode: Resolved a condition for MS Word files where conversion on Linux may result in a crash. (SF1223551 / FB9585)
- HD Mode: Resolved a condition for MS Word files where strikethrough text may not render correctly. (FB9580)
- HD Mode: Resolved a condition for PDF files where conversion to HTML5 may not complete. (SF1214430 / FB9612)
- HD Mode: Resolved a condition for StarOffice .SXC files where content may not render. (FB9606)
- HD Mode: Resolved a condition for StarOffice .SXW files where content may not render correctly. (FB9608)
- HD Mode: Resolved a condition for StarOffice .SXW files where content may not render. (SF9606 / FB9609)
- HD Mode: Resolved a condition for WordPerfect files where bullets/check marks may not render. (SF1195067 / FB9680)

- HD Mode: Resolved a condition for WordPerfect files where header text may not render correctly. (FB9623)
- HD Mode: Resolved a condition for WordPerfect files where images may not render correctly. (FB9550)
- HD Mode: Resolved a condition for WordPerfect files where page numbering may not render correctly. (SF1195037 / FB9618)
- HD Mode: Resolved a condition for WordPerfect files where table cell values may not align correctly. (FB9625)
- HD Mode: Resolved a condition for WordPerfect files where table spacing may not render correctly. (SF1195067 / FB9651)
- HD Mode: Resolved a condition for WordPerfect files where text columns may not render correctly. (SF1195071 / FB9521)
- HD Mode: Resolved a condition for WordPerfect files where text may incorrectly overwrite images. (FB9624)
- HD Mode: Resolved a condition for WordPerfect files where white space may highlight incorrectly. (SF1195067 / FB9665)
- HD Mode: Resolved a condition where .RTF files may not render correctly. (SF01225111 / FB9590)
- HD Mode: Resolved a condition where IGR_Get_Page_Text returns the wrong buffer size. (SF1232918 / FB9643)
- HD Mode: Resolved a condition where WINMAIL.DAT files may not extract correctly. (SF1258500 / FB9688)
- Resolved a condition for .EML files where unprintable characters may incorrectly display. (SF01217754 / FB9619)
- Resolved a condition for .MSG files where "From" metadata may not extract correctly. (SF01243538 / FB9652)
- Resolved a condition for Ichitaro files where text may not extract. (SF01243242 / FB9656)
- Resolved a condition for legacy MS Excel files where extraction may result in Error Code 4. (SF01220284 / FB9571)
- Resolved a condition for MS Excel files where EXCELMODE=CSV may not work correctly when passed with IGR_FORMAT_TEXT. (SF1238986 / FB9673)
- Resolved a condition for MS Excel files where shapes may not have rendered correctly in Classic HTML. (SF01216843 / FB9559)
- Resolved a condition for PDF files where extraction may fail on Linux 32-bit systems. (SF1237938 / FB9638)
- Resolved a condition where Chinese characters may not render correctly in PDF output mode. (FB9654)

5.56 Document Filters 11.2 Build 1766 (May 2015)

5.56.1 Enhancements

- Added ability to specify an HTML attribute for HDHTML_INCLUDE_WORD_INDEXES. (SF01194108 / FB9528)
- Added support for RFC822 .EML files. (SF1155562 / FB9534)
- Added support for Windows Icon .ICO files. (SF01204454 / FB9527)

5.56.2 Updates

- HD Mode: Improved conversion logic for .EML files by adding an Attachments field. (SF1194018 / FB9531)
- HD Mode: Improved logic for rendering older versions of SmartArt in MS PowerPoint files. (SF01194621 / FB9477)
- HD Mode: Resolved a condition for .EML files where special characters may not render. (SF01201203 / FB9518)
- HD Mode: Resolved a condition for .ICS files where metadata begin markers may not be present. (FB9535)
- HD Mode: Resolved a condition for .MSG files where JPEGs may not extract. (SF1194085 / FB9530)
- HD Mode: Resolved a condition for .MSG files where the To and CC fields may not render. (SF01208386 / FB9532)
- HD Mode: Resolved a condition for charts where axis titles may not render correctly. (FB9536)
- HD Mode: Resolved a condition for MS Excel files where cell borders may not render correctly. (SF798268 / FB8823)
- HD Mode: Resolved a condition for MS Excel files where chart labels on the horizontal and vertical axes may not render correctly. (SF01194065 / FB9505)
- HD Mode: Resolved a condition for MS Excel files where table headers may incorrectly overlap. (SF01194073 / FB9504)
- HD Mode: Resolved a condition for MS Outlook files where hyperlink formatting may be incorrect. (SF1194090 / FB9482)
- HD Mode: Resolved a condition for MS PowerPoint files where background images may incorrectly render outside of shape boundaries. (SF01199028 / FB9503)
- HD Mode: Resolved a condition for MS PowerPoint files where background images may not render correctly. (FB9498)
- HD Mode: Resolved a condition for MS PowerPoint files where bullet list line height may be incorrect. (FB9459)
- HD Mode: Resolved a condition for MS PowerPoint files where bullet lists may incorrectly render as number lists. (FB9471)
- HD Mode: Resolved a condition for MS PowerPoint files where default tab positions may be incorrectly placed. (FB9469)
- HD Mode: Resolved a condition for MS PowerPoint files where hyperlink text may not render correctly. (FB9540)

- HD Mode: Resolved a condition for MS PowerPoint files where some objects/images may render with incorrect color. (SF01194621 / FB9478)
- HD Mode: Resolved a condition for MS PowerPoint files where text borders may not render. (SF01196782 / FB9525)
- HD Mode: Resolved a condition for MS PowerPoint files where text box borders may not render correctly. (SF01196765 / FB9512)
- HD Mode: Resolved a condition for MS PowerPoint files where text may render upside down. (FB8875)
- HD Mode: Resolved a condition for MS PowerPoint files where title text may not render correctly. (FB9542)
- HD Mode: Resolved a condition for MS Visio files where pages may incorrectly overlap. (SF1195033 / FB9479)
- HD Mode: Resolved a condition for MS Word files where extraneous text may render in footers. (SF1194079 / FB9495)
- HD Mode: Resolved a condition for MS Word files where footers may not render correctly. (SF1194076 / FB9487)
- HD Mode: Resolved a condition for MS Word files where hyperlinks may not appear as hyperlinks. (SF1106944 / FB9472)
- HD Mode: Resolved a condition for MS Word files where images/graphics in headers may not render. (SF1194055 / FB9492)
- HD Mode: Resolved a condition for MS Word files where images/graphics may not render. (SF01194041 / FB9490)
- HD Mode: Resolved a condition for MS Word files where images/graphics may not render. (SF1194046 / FB9491)
- HD Mode: Resolved a condition for PDF files where spaces may not render in HTML5. (SF01204232 / FB9529)
- HD Mode: Resolved a condition for PowerPoint files where Sans-Serif fonts might render as Serif fonts. (FB9497)
- HD Mode: Resolved a condition for RTF files where table content may not render. (SF1195032 / FB9481)
- HD Mode: Resolved a condition for WordPerfect files where conversion may result in an error. (SF1194104 / FB9483)
- HD Mode: Resolved a condition for WordPerfect files where footers may not render. (SF1195037 / FB9506)
- HD Mode: Resolved a condition for WordPerfect files where footnotes may not render correctly. (SF1195059 / FB9523)
- HD Mode: Resolved a condition for WordPerfect files where headers may not render. (SF01195040 / FB9484)
- HD Mode: Resolved a condition for WordPerfect files where tables may not render correctly. (SF1195067 / FB9514)
- HD Mode: Resolved a condition where converting files to HTML5 with external images, may not create an external image file. (SF01127057 / FB9485)
- Improved JPEG metadata extraction. (SF1186898 / FB9473)
- Resolved a condition for .ALZ archives with subfiles containing Korean characters where file names may not extract properly. (SF1168153 / FB9452)
- Resolved a condition for .DGN files where text may not be extracted. (SF1174175 / FB9448)
- Resolved a condition for .EML files where extraction may result in an IGR 4 error. (FB9541)

- Resolved a condition for .MHT files where metadata may not render. (SF01213375 / FB9544)
- Resolved a condition for .MSG files where keyword metadata may not render. (FB9513)
- Resolved a condition for MS Excel files where text may extract differently between platforms. (FB9461)
- Resolved a condition for PDF files where text extraction may crash. (SF01184333 / FB9467)
- Resolved a condition where .GIF files may be misidentified as XYwrite. (SF01197198 / FB9539)
- Resolved a condition where word indexes may not be consistent between HD modes (potentially affecting redaction). (FB9562)
- Text Mode: Resolved a condition for .EML files where attached .MSG files may not render correctly. (SF01177617 / FB9464)
- Text mode: Resolved a condition for AutoCAD .DWG files where extraction may fail. (SF1200957 / FB9520)

5.57 Document Filters 11.2 Build 1732 (March 2015)

This section contains a summary of improvements and fixes available since Document Filters 11.2.1666.

5.57.1 Enhancements

- Added character encoding detection for text files. (SF766487 / FB8749)
- Added ID-only support for Material Exchange Format (MXF). (FB6287)
- Added resolution unit for generated TIFF images. (SF01162329 / FB9425)

5.57.2 Updates

- HD Mode: Added the ability in MS PowerPoint files to render comments with the SLIDE_COMMENTS option. (FB9228)
- HD Mode: Added the ability to include or exclude the footer when rendering text pages via TEXT_PAGE_FOOTER. (SF800907 / FB8850)
- HD Mode: Improved .PNG file output control by adding pHYs headers (Physical Pixel Dimensions) to output. (SF1135131 / FB9383)
- HD Mode: Improved dithering algorithm for generated monochrome TIFFs. (SF1074620 / FB9306)
- HD Mode: Improved efficiency when processing MS PowerPoint files by internally re-using Master Slide images. Previously, the Master Slide images would be re-processed for each slide that used them. (FB9300)
- HD Mode: Improved exception handling when invoking the Canvas API on unsupported (non-bitmap) output types. (FB9202)
- HD Mode: Improved tagging of MS PowerPoint slide notes pages for HTML5 and structured XML. (SF1082358 / FB9319)
- HD Mode: Reduced output file size when converting documents to PDF. (SF00953420 / FB9207)
- HD Mode: Resolved a condition for .MHT files where embedded images may be missing. (SF01127057 / FB9358)
- HD Mode: Resolved a condition for charts where data label text may overlap. (FB9248)
- HD Mode: Resolved a condition for charts where data labels may be missing or misaligned. (FB9245)
- HD Mode: Resolved a condition for charts where markers may be missing. (FB8881)
- HD Mode: Resolved a condition for charts where non-solid markers may be missing. (FB9166)
- HD Mode: Resolved a condition for charts where Right and Top axis locations may not render correctly. (FB9162)
- HD Mode: Resolved a condition for horizontal bar charts where data labels may not render correctly. (FB9163)
- HD Mode: Resolved a condition for MS Excel files where chart items may be disordered in the legend. (FB9396)
- HD Mode: Resolved a condition for MS Excel files where charts may render with category labels missing. (FB9158)

- HD Mode: Resolved a condition for MS Excel files where conversion to TIFF may reduce image quality at higher DPIs. (SF1165908 / FB9445)
- HD Mode: Resolved a condition for MS Excel files where headers may not scale correctly. (SF1165908 / FB9432)
- HD Mode: Resolved a condition for MS Excel files where large numbers of empty rows may appear in the output. (SF1133804 / FB9380)
- HD Mode: Resolved a condition for MS Excel files where scaling may be incorrect for files in landscape orientation. (SF1069621 / FB9296)
- HD Mode: Resolved a condition for MS Excel files where SHOWHIDDEN=OFF may not hide cells correctly. (SF1082354 / FB9315)
- HD Mode: Resolved a condition for MS Office files where images with padding may not render. (FB8874)
- HD Mode: Resolved a condition for MS PowerPoint files where blank pages may render when the SLIDE_NOTES option was enabled. (FB9344)
- HD Mode: Resolved a condition for MS PowerPoint files where conversion may fail due to "invalid string position" error. (SF01154769 / FB9412)
- HD Mode: Resolved a condition for MS PowerPoint files where embedded EMF/WMF files may render with an incorrect background. (SF1147424 / FB9389)
- HD Mode: Resolved a condition for MS PowerPoint files where gradient backgrounds may not render properly. (FB9409)
- HD Mode: Resolved a condition for MS PowerPoint files where header sections may not be identified in Structured XML. (SF1127441 / FB9368)
- HD Mode: Resolved a condition for MS PowerPoint files where headers and footers on slide notes pages may not render. (FB9367)
- HD Mode: Resolved a condition for MS PowerPoint files where patterns in embedded .EMF images may not render correctly. (FB8882)
- HD Mode: Resolved a condition for MS PowerPoint files where shapes may be missing solid fills. (FB9415)
- HD Mode: Resolved a condition for MS PowerPoint files where some slides may not be converted (depending on SHOWHIDDEN). (SF00626521 / FB8286)
- HD Mode: Resolved a condition for MS PowerPoint files where Text Box fill color may not render correctly. (FB9397)
- HD Mode: Resolved a condition for MS PowerPoint files where Text Box margins may be incorrect. (SF1154767 / FB9402)
- HD Mode: Resolved a condition for MS PowerPoint files where the solid fill of a text box may not render. (SF1155901 / FB9403)
- HD Mode: Resolved a condition for MS PowerPoint files where top and bottom cell margins may not be considered in cell spacing and alignment. (FB9327)

- HD Mode: Resolved a condition for MS PowerPoint files where vertical alignment may be incorrect, causing text to overlap. (SF1154767 / FB9399)
- HD Mode: Resolved a condition for MS Word (.DOC) files where file hyperlinks may not appear. (SF1106944 / FB9329)
- HD Mode: Resolved a condition for MS Word (.DOCX) files where file hyperlinks may not appear. (SF1106944 / FB9328)
- HD Mode: Resolved a condition for MS Word files where documents may cause segfaults on Linux. (SF1176092 / FB9454)
- HD Mode: Resolved a condition for MS Word files where embedded EMF/WMF images may not render correctly. (FB8880)
- HD Mode: Resolved a condition for MS Word files where embedded EMF/WMF images may not render correctly. (FB8883)
- HD Mode: Resolved a condition for MS Word files where headings may be misaligned. (SF00701992 / FB8612)
- HD Mode: Resolved a condition for MS Word files where images in the header and/or footer were rendered on top of the text. (FB9301)
- HD Mode: Resolved a condition for MS Word files where inline images may be missing after conversion to HTML5. (SF01160780 / FB9422)
- HD Mode: Resolved a condition for MS Word files where lines may be incorrectly drawn over text boxes. (FB9420)
- HD Mode: Resolved a condition for MS Word files where tables may render off the page causing data loss. (FB9307)
- HD Mode: Resolved a condition for MS Word files where text may be incorrectly positioned under images. (FB9419)
- HD Mode: Resolved a condition for OpenOffice .ODP files where text boxes containing subscripts may render text incorrectly. (FB9122)
- HD Mode: Resolved a condition for OpenOffice .ODS files where some cell format settings may not render correctly. (FB9129)
- HD Mode: Resolved a condition when processing MS Word documents where some internal document links may not work after conversion. (SF700147 / FB8551)
- HD Mode: Resolved a condition where .EMF files may not render correctly. (FB9074)
- HD Mode: Resolved a condition where email (.EML) files with accented characters may not render correctly. (FB9444)
- HD Mode: Resolved a condition where email metadata may not be present. (FB9440)
- HD Mode: Resolved a condition where embedded vector drawings may render incorrectly. (FB8884)
- Improved Date metadata logic for PDF files, replacing Z with UTC. (SF769856 / FB8687)
- Improved tagging of slide notes and slide comments in HTML5, XML, and Classic HTML. (FB9337)

- Improved the built-in support for Wingdings, Wingdings 2, and Webdings fonts and now map them to Unicode equivalent characters. (FB9194)
- MS Excel metadata field "Last Saved By" renamed to "Last Accessed By." (SF1129865 / FB9363)
- Resolved a condition for .MHT files where they may be identified as .EML. (SF1051453 / FB9266)
- Resolved a condition for .WAV files where bitrate metadata may be incorrect. (FB9387)
- Resolved a condition for MS Excel files where conversion to Classic HTML on Unix systems may stall. (SF1158158 / FB9417)
- Resolved a condition for MS PowerPoint files where conversion may not retain custom metadata. (SF01127442 / FB9357)
- Resolved a condition for MS Word files where footnote numbering may render incorrectly. (FB9335)
- Resolved a condition for PDF files where conversion with OCR enabled may lead to image rotation and failed extraction. (SF1122485 / FB9345)
- Resolved a condition for PDF files where extraction may result in an Access Violation error. (SF1155846 / FB9401)
- Resolved a condition for vCard (.VCF) files where version metadata may not be extracted. (SF1143791 / FB9435)
- Resolved a condition where IGR_FILE_SUPPORTS_TEXT was not set for .BMP images. (SF1120636 / FB9339)
- Resolved a condition where MBOX/Sendmail files may be misidentified as .EML files and therefore slow to extract. (SF1112548 / FB9336)
- Resolved a condition where MS Word files may fail to extract when multiple graphics on a page are anchored to paragraphs and have negative vertical position. (SF1113563 / FB9332)
- Resolved a condition where MS Works Spreadsheets files may be recognized as an incorrect file type. (SF1143781 / FB9386)
- Resolved a condition where some files may be incorrectly recognized as MP3. (FB9250)
- Resolved a condition where text files may be misidentified as Transcript files. (SF1117024 / FB9408)
- Resolved a condition where text files with Thai characters may be misidentified as WordPerfect files. (SF766487 / FB9423)
- Resolved a condition where using a Chinese OCR dictionary would produce an error. (FB9325)
- Resolved an issue for AutoCAD files where text may not be extracted on Unix platforms. (SF1143791 / FB9438)

5.58 Document Filters 11.2 Build 1666 (January 2015)

This section contains a summary of improvements and fixes available since Document Filters 11.2.

5.58.1 Enhancements

- Added support for processing of MS Word 2003 XML and MS Excel 2003 XML format. (SF00764806 / FB8675)
- Added the ability for HTML file titles to be added to the metadata section rather than the body section. (SF983403 / FB9223)
- Added the ability to inject custom attachment processing code inside the HTML of converted MSG file using EMAIL_ATTACHMENT_LINKS options. (FB9214)
- Added the ability to specify file name prefixes for generated images. (FB9260)
- HD Mode: Added support for redaction when the output mode is HD HTML. (FB9201)
- HD Mode: Implemented Page.GetAttribute(), a new API to retrieve attributes of a given page. (SF825499 / FB8938)
- HD Mode: Improved efficiency when processing PowerPoint files by internally re-using Master Slide images. Previously, the Master Slide images would be re-processed for each slide that used them. (FB9300)
- HD Mode: Improved exception handling when invoking the Canvas API on unsupported output types. (FB9202)
- HD Mode: Improved footer text placement and wrapping logic in MS Word (DOCX). (SF00800921 / FB8836)
- HD Mode: Improved handling of links within text boxes when processing OpenOffice documents. (FB9134)
- HD Mode: Improved OpenOffice master styles logic to properly resolve inherited text and paragraph styles. (FB9124)
- HD Mode: Improved processing of MS PowerPoint files to correctly create tables in output, rather than creating separate floating shape objects. (FB9287)
- HD Mode: Improved rendering of footers in OpenOffice ODP files. (FB9123)
- Improved conversion from MS Excel to Classic HTML by rendering table height. (SF1048824 / FB9261)
- Improved detection support for CPIO files. (FB9254)
- Improved exception handling when using custom streams in Java and Python. (FB9270)
- Improved support of text files with non-Western encoding. (SF1037474 / FB9255)

5.58.2 Updates

- HD Mode: Resolved a condition for MS Excel and PowerPoint files where custom metadata properties were not extracted. (SF01061362 / FB9273)
- HD Mode: Resolved a condition for MS Excel files where footer information extracted into structured-XML format rendered incorrectly. (SF1063757 / FB9281)

- HD Mode: Resolved a condition for MS Excel files where formatting masks may be incorrectly processed and produce extra characters. (FB8901)
- HD Mode: Resolved a condition for MS PowerPoint .PPTX files where rendered slides may contain an incorrect background. (FB9302)
- HD Mode: Resolved a condition for MS PowerPoint files where embedded EMF files may render incorrectly. (FB8877)
- HD Mode: Resolved a condition for MS PowerPoint files where footer information extracted into structured-XML format rendered incorrectly. (SF1063757 / FB9295)
- HD Mode: Resolved a condition for MS PowerPoint files where resized EMF files may render incorrectly. (FB8878)
- HD Mode: Resolved a condition for MS PowerPoint files where tables rendered with incorrect font styles. (FB9112)
- HD Mode: Resolved a condition for MS PowerPoint files where top-aligned text may render incorrectly as uppercase. (FB9154)
- HD Mode: Resolved a condition for MS Word DOCX files where hyperlinks may not work in HTML and HTML5 output. (SF955228 / FB9198)
- HD Mode: Resolved a condition for MS Word files where HD HTML output may incorrectly show field codes. (FB9213)
- HD Mode: Resolved a condition for MS Word files where multi-column text may incorrectly render as a single column. (SF00783520 / FB8762)
- HD Mode: Resolved a condition for OpenOffice ODS files where charts and images may incorrectly stack. (FB9130)
- HD Mode: Resolved a condition for RTF files where tables may incorrectly render extraneous horizontal lines. (SF821812 / FB8909)
- HD Mode: Resolved a condition when processing MS Word documents where some internal document links may not work after conversion. (SF700147 / FB8551)
- HD Mode: Resolved a condition where anchored column text boxes may erroneously render off the page. (FB9284)
- HD Mode: Resolved a condition where the following anchor types for floating objects were not implemented: AnchorText, AnchorNone, AnchorLeftMargin, AnchorRightMargin, AnchorTopMargin, and AnchorBottomMargin. (FB8832)
- HD Mode: Resolved a condition where the page sourceWidth/Height could not be retrieved using Page.GetAttribute/IGR_Get_Page_Attribute. (FB9290)
- Resolved a condition for HTML files where embedded base64 sub-files may not extract. (SF983226 / FB9216)
- Resolved a condition for Ichitaro files (.JTD) where processing Japanese text may render missing characters. (SF1065424 / FB9309)
- Resolved a condition for Ichitaro files (.JTD) where processing Japanese text may render missing characters. (SF1065424 / FB9310)

- Resolved a condition for JPG files where Cyrillic metadata may not extract properly. (SF00942522 / FB9200)
- Resolved a condition for MS Access files where numbers and dates may be inconsistently formatted across platforms. (FB9280)
- Resolved a condition for MS Excel files where a memory leak may occur. (SF827937 / FB8935)
- Resolved a condition for MS Excel files where content spanned across two pages may be duplicated. (SF980022 / FB9215)
- Resolved a condition for MS Excel files where custom property field names may incorrectly render spaces as erroneous characters. (SF983242 / FB9217)
- Resolved a condition for MS Excel files where images converted to classic HTML may incorrectly overlap the toolbar while scrolling. (SF00985437 / FB9221)
- Resolved a condition for MS PowerPoint files where embedded charts or spreadsheets may not extract correctly. (SF1015335 / FB9237)
- Resolved a condition for MS PowerPoint files where embedded OLE objects may sometimes not be identified correctly when extracting sub-files. (SF971353 / FB9210)
- Resolved a condition for MS PowerPoint files where images may convert with missing colors. (SF01017424 / FB9242)
- Resolved a condition for MS Word 97 files where linked documents may not extract successfully. (SF983390 / FB9220)
- Resolved a condition for MS Word files where extracting documents with embedded sub-files may incorrectly output extra/hidden sub-file content. (SF983506 / FB9232)
- Resolved a condition for MS Word files where Russian metadata may extract incorrectly. (SF983287 / FB9234)
- Resolved a condition for MS Word files where soft hyphens may be incorrectly extracted. (SF983286 / FB9222)
- Resolved a condition for MS Word files where uppercase fonts may not render correctly. (FB9258)
- Resolved a condition for OpenOffice ODT files where metadata values may not output correctly. (SF983435 / FB9219)
- Resolved a condition for PDF files where accented characters within annotations may render incorrectly. (SF1033790 / FB9272)
- Resolved a condition for PDF files where generated embedded image quality may decrease. (SF1015936 / FB9241)
- Resolved a condition for PDF files where text annotations may not be extracted correctly. (SF779680 / FB8731)
- Resolved a condition where calling IGR_Render_Page() from multiple threads may produce incorrect or corrupt output. (FB9274)
- Resolved a condition where files may be incorrectly recognized as MP3. (FB9250)
- Resolved a condition where files with Unicode names may not extract from ALZ or EGG archives. (SF1016473 / FB9249)

- Resolved a condition where ISYS ASCII Transcripts may not be identified correctly. (FB9252)
- Resolved a condition where processing files in multi-threaded files may produce different or wrong output. (FB9275)
- Updated CONTAINERS_LIST_DIR option to work when passing a parameter with mixed case. (FB9211)

5.59 Document Filters 11.2 (November 2014)

This section contains a summary of new features in Document Filters 11.2. For detailed API usage, refer to the *Document Filters Implementation Guide*.

5.59.1 Enhancements

This section contains a summary of new features available in Document Filters 11.2.

Smart font mapping

Document Filters has been enhanced to apply smarter font substitution logic for fonts not embedded or available on the running application's server. Additionally, the product now includes several fallback fonts, including a TrueType Unicode font, which provide optimal font mapping across platforms and Unicode-supported languages when the original font is not available.

Developers may also optionally take control of the font fallback logic by specifying supplementary fallback fonts and fallback order in the fonts.ini config file.

Viewer code samples

Compile-ready code samples that demonstrate common viewing-related actions are now available for all API languages across all platforms.

The ConvertDocumentToInteractiveHDHTML sample converts a document to High Definition paginated HTML5, then demonstrates how individual pages may be served as a user scrolls or as the application requires. It also allows common viewing capabilities of zoom, rotation, page navigation, text highlighting, printing, and redaction.

Optical Character Recognition (OCR)

You can use Optical Character Recognition (OCR) to extract text from document image formats. The option is available for text-mode and high-definition outputs. Text-mode supports OCR on the following graphic types: JPEG, TIFF, GIF, PNG, BMP, and Scanned PDFs. High-definition mode supports OCR on the following graphic types: JPEG, TIFF, GIF, PNG, WMF, EMF, BMP, and Scanned PDFs.

The built-in OCR engine also takes advantage of dictionaries to improve accuracy. The default dictionary is English, however additional languages may be installed. Download the appropriate language pack from the Tesseract project to apply these languages.

Office charts

Support for High Definition rendering of 2D charts now includes all platforms. High Definition rendering of Office Charts has been improved in both performance and fidelity. Primary categories of 2D chart types supported include the following items.

- Column
- Line
- Pie
- Bar
- Area
- Scatter (XY)

Note Any 3D charts converting to HD will become 2D charts.

HTML and email to HD

HTML-based input, including MSG and EML documents, now renders in High Definition across all platforms; previously this was only possible in Windows.

TIFF specific output options

HD TIFF output now includes the ability to specify black and white as well as color options. You can also control TIFF compression type. The GRAPHIC_DPI option now supports an additional AUTO parameter for TIFF input documents.

Image specific output options

HD TIFF, BMP, JPG, and PNG now support graphic effects of grayscale and desaturation.

Microsoft PowerPoint specific output options

HD Microsoft PowerPoint now has several options for rendering speaker notes.

Password protected document handling - PDF

A new API option has been introduced to allow applications to submit a password when opening a PDF document.

Platforms

Certified platform support for Redhat Enterprise Linux V7 (x86/x64).

Additional format support

The following formats have been added or are now supported by additional output modes.

- HTML, EML, and MSG as input formats now have full HD support across all platforms; previously Windows only.
- Added full HD support for MS Office 365 Word documents.
- Added full HD support for MS Office 365 Excel documents.
- Added full HD support for MS Office 365 PowerPoint documents.
- Added full HD support MS Excel Binary documents (XLSB), including EXCELMODE=CSV option.
- Added full HD support for MS Excel v4.0 documents (XLS).
- Added full HD support for MS Visio v5.0 documents (VSD).
- Added full HD support for TIFF documents.
- Added full HD support for BMP documents.
- Added full HD support for JPG documents.
- Added full HD support for PNG documents.
- Added full HD support for WMF files and clip art on all platforms, previously Windows only.
- Added full HD support for EMF files and clip art on all platforms, previously Windows only.
- Added support for multi-part RAR extraction.
- Added support for multi-part ZIP extraction.
- Added extraction and full HD support for Hancom Office .CELL.
- Added extraction and full HD support for Hancom Office .SHOW.
- Added extraction support for MS Access 95 - 2010 (MDB/ACCDB).
- Added extraction support for MS Visio 2013 (VSDX).
- Added extraction support for ESTSoft EGG archives and sub-files.
- Added extraction support for ESTSoft ALZ archives and sub-files.
- Added extraction support for UUencoded archives and sub-files.
- Added extraction support for UNIX Compress archives and sub-files.
- Added extraction support for dBaseIV.
- Added identification and extraction support for vCalendar.

- Certified extraction support for MS OneNote 2013.
- Certified extraction support for MS Project 2013
- Added identification support for MS PowerPoint 3.x and 4.x documents.
- Added identification and metadata extraction support for JungUm documents (.GUL).
- Added identification support for Symbian executable files.
- Added identification support for ISYS Index files.
- Added identification support for FITS Image (.FTS).
- Added identification support for Wordstar for Windows version 2 (.WSD).

Note Full HD support implies both the Classic HTML mode and paginated High Definition outputs.

APIs

HD TIFF Output now includes the ability to specify Black and White, Grayscale. You can now control TIFF compression type (PACKED, JPEG, LZW, GROUP3, GROUP6).

HD TIFF, BMP, JPG and PNG now support post processing graphic effects.

Other features

- Exception handling has been improved with the addition of the IGRException class. Previously undefined or lightly defined errors will now return a definitive description string and error code.

5.59.2 Updates

This section contains a summary of improvements and fixes available in Document Filters 11.2.

- Added chart data extraction support for MS Office formats (FB8435)
- Added extraction of page-level orientation, source width and source height metadata from multi-page TIFF documents to the structured XML output.
- Added options for generating black & white and grayscale TIFF output. (FB8466)
- Added support for extraction of sequence metadata from DICOM documents. (FB9088)
- Added Unicode support for AutoCAD documents.
- Allow override of maximum open document handles via ISYS_MAX_DOCHANDLES environment variable (default: 64). (FB8507)
- AutoCAD HD support has been extended to all platforms, except HP-UX and AIX.
- HD Mode: Added the ability to process embedded fonts in MS Word 2007+ documents. (FB8652)

- HD Mode: Implemented chart support on all platforms for Microsoft Office files. (FB8649)
- HD Mode: Implemented processing of transparency layers in MS Visio resulting in much clearer visual effects applied to multi-layer text and images. (FB9085)
- HD Mode: Improved font mapping logic for PDF. (SF843811 / FB8977)
- HD Mode: Improved placement of footnotes when processing MS Word documents. (FB9150)
- HD Mode: Improved tab spacing, line spacing, and text placement in MS Visio. (FB8713)
- HD Mode: MS Excel processing has been improved resulting in a lower memory footprint, added support for additional font styles and enhanced chart rendering capabilities.
- HD Mode: Resolved a condition in MS PowerPoint where substituted fonts caused text to render too wide. (FB9137)
- HD Mode: Resolved a condition where "space before" in the paragraph settings of PowerPoint (PPT) files was not recognized when converted to HD. (SF783662 / FB8873)
- HD Mode: Resolved a condition where cell content in MS Excel may truncate. (SF640928 / FB8385)
- HD Mode: Resolved a condition where charts in MS PowerPoint (PPTX) files may not render correctly. (FB8824)
- HD Mode: Resolved a condition where end notes in Microsoft Word may appear at the end of a section rather than the end of a document. (FB9155)
- HD Mode: Resolved a condition where exporting .DWG files with multiple sheets only rendered the first sheet. (SF924320 / FB9102)
- HD Mode: Resolved a condition where Microsoft Word documents converting to HD may sometimes omit the second page. (SF843466 / FB9173)
- HD Mode: Resolved a condition where MSG file From, To, Cc, and Bcc fields would display disappear when rendering in black-and-white mode. (SF705032 / FB8568)
- HD Mode: Resolved a condition where page borders on MS Word (DOCX) documents may not appear. (FB8650)
- HD Mode: Resolved a condition where page numbers in MS Word DOC files may render incorrectly. (SF725478 / FB8623)
- HD Mode: Resolved a condition where pie charts with a single value rendered nothing in the plot area. (FB9178)
- HD Mode: Resolved a condition where RTF did not correctly process highlighted text. (FB9168)
- HD Mode: Resolved a condition where some MS Office-embedded EMF images would not render. (FB9109)
- HD Mode: Resolved a condition where substituted fonts in MS PowerPoint would sometimes cause paragraph spacing issues. (FB9110)
- HD Mode: Resolved a condition where table text in MS Word (DOCX) files would sometimes overlap or not render. (SF800423 / FB8826)
- HD Mode: Resolved a condition where tables in MS PowerPoint rendered with incorrect font styles. (FB9112)
- HD Mode: Resolved a condition where tables in MS Word might not render correctly. (FB9139)

- HD Mode: Resolved a condition where text color, alignment and font size of Word Art may not render correctly. (FB8983)
- HD Mode: Resolved a condition where text in MS PowerPoint tables would sometimes incorrectly render in center vertical alignment rather than correctly in top vertical alignment. (FB9111)
- HD Mode: Resolved a condition where the footer of an MS Word DOC would display in the header. (SF843466 / FB8972)
- HD Mode: Resolved a condition where the HD Viewer may not open in OSX 10.9.4. (SF830693 / FB8954)
- HD Mode: Resolved a condition where the last column of an XLS spreadsheet may be duplicated when the output flows onto a subsequent page. (SF764002 / FB8669)
- HD Mode: Resolved a potential condition causing cell content to not display. (FB8672)
- HD Mode: Resolved an issue that caused some WMF images not to render correctly. (SF721653 / FB9090)
- Improved handling of sub-file extraction of OLE based documents (MS Office 97-2003). (SF923513 / FB9115)
- Java exceptions are now checked and error number is returned in addition to error string. (FB8462)
- MS Visio document support has improved, resulting in more performant extraction and improved HD rendering.
- Resolved a condition in MSG files where html content containing a <pre> tag would render incorrectly in HTML and would not render at all in HD. (SF732616 / FB8632)
- Resolved a condition that may result in an access violation when processing an XML document. (SF923521 / FB9095)
- Resolved a condition where big-endian text files may not convert correctly. (FB9184)
- Resolved a condition where cells may overlap when converting MS Excel files to Classic HTML on terminal services enabled servers. (SF798449 / FB8904)
- Resolved a condition where column separators in MS Word might not render correctly. (FB9113)
- Resolved a condition where converting a .DWG file to Classic HTML could cause the system to become unresponsive. (FB9118)
- Resolved a condition where deleted markup was sometimes not removed from MS Word files. (FB9170)
- Resolved a condition where embedded Microsoft graph charts stored in MS Office files may not output in text-mode. (FB9054)
- Resolved a condition where GZ files may fail to extract. (SF923516 / FB9096)
- Resolved a condition where hit-highlighting and text selection may not work for PDF files converted to HD HTML. (SF903976 / FB9089)
- Resolved a condition where HTML documents processed in text-only mode may contain too much whitespace. (SF927994 / FB9140)
- Resolved a condition where Microsoft Word (DOC) files may not render German umlaut characters correctly. (SF891197 / FB9078)

- Resolved a condition where MS Excel (XLS) files may not extract successfully using EXCELMODE=CSV. (SF935784 / FB9180)
- Resolved a condition where MS PowerPoint (PPT) generated to Classic HTML, may obscure the scroll bar when it has a large number of slides. (SF935590 / FB9181)
- Resolved a condition where processing a corrupt WMV may result in an out of memory error. (SF947925 / FB9191)
- Resolved a condition where processing a Microsoft Project (MPP) file may result in a "Property size too big" error. (SF931656 / FB9142)
- Resolved a condition where processing an XLS file may result in an "Invalid block size" error. (SF923520 / FB9097)
- Resolved a condition where RTF incorrectly reported false on getSupportsSubFiles. (SF908246 / FB9094)
- Resolved a condition where some fonts would not render properly in MS PowerPoint. (FB9046)
- Resolved a condition where some MS Excel (XLS) files could take significant time to process, appearing to hang. (SF935852 / FB9172)
- Resolved a condition where some ZIP files would not extract correctly. (SF923587 / FB9103)
- Resolved a condition where spacing in SGML files was not rendered correctly. (FB9146)
- Resolved a condition where text extraction may fail for .ONE files. (SF923550 / FB9100)
- Resolved a condition where text in MS PowerPoint (PPT) files may overlap when rendered to Classic HTML. (FB9171)
- Resolved a condition where text may not be extracted from generic spreadsheet file types. (SF764806 / FB8675)
- Resolved a condition where the getID() method may return null for subfiles. (SF923592 / FB9116)
- Resolved a condition where TXT files attached to EML files may not be recognized and/or extracted. (SF816743 / FB8896)
- Resolved a condition where Word Art may not be re-sized correctly. (FB8984)
- Resolved a condition where word spacing following a new line may be incorrect when extracting RTF documents to text. (SF923584 / FB9101)
- Text Mode: Added the extraction of embedded files within TNEF EML documents. (FB8457)
- Text Mode: Added the option to turn off slide numbers in MS PowerPoint (PPT, PPTX) and PDF. (FB8697)
- Text Mode: Resolved a condition where extracted text of PDF annotations may contain unreadable characters. (SF951680 / FB9193)
- Updated Classic HTML output for MS Word documents (DOC,DOCX) to remove global DIV and OL style that may impact rendering of other elements. (SF943780 / FB9183)

5.60 Document Filters 11.1 Build 1546 (October 2014)

5.60.1 Enhancements

New Output Modes

- Several High Definition (HD) Raw Bitmap modes have been added to provide additional developer flexibility. New API options are exercised using the IGR_Make_Output_Canvas method. The color depth is implied by the output mode chosen.

Forced Document Extraction

- A new API option has been added to allow developers to attempt document extraction on unknown format types. This option is exercised when using the IGR_Open_File_Ex or Extractor.OpenEx methods.

Additional Format Support

Several formats have been added or enhanced. Note that “full HD support” implies both the Classic HTML mode and High Definition outputs. Added formats include:

- HTML, EML, MSG as input formats now have full HD support across all platforms; previously Windows only.
- Added full HD support for MS Office 365 Excel documents.
- Added full HD support for MS Office 365 PowerPoint documents.
- Added full HD support MS Excel Binary documents (XLSB), including EXCELMODE=CSV option.
- Added full HD support for MS Excel v4.0 documents (XLS).
- Added full HD support for MS Visio v5.0 documents (VSD).
- Added full HD support for TIFF documents.
- Added full HD support for BMP documents.
- Added full HD support for JPG documents.
- Added full HD support for PNG documents.
- Added extraction support for ESTSoft EGG archives and sub-files.
- Added extraction support for ESTSoft ALZ archives and sub-files.
- Added extraction support for UUencoded archives.
- Added identification support for MS PowerPoint 3.x and 4.x documents.
- Added identification and metadata extraction support for JungUm documents (GUL).

- Added identification support for Symbian executable files.
- Added identification support for ISYS Index files.

Enhanced Formats

- AutoCAD HD support has been extended to all platforms, except HP-UX and AIX.
- Added Unicode support for AutoCAD documents.
- MS Visio document support has vastly improved, resulting in better, more performant extraction, and significant improvement in HD rendering.

New Platforms

- Certified platform support for Redhat Enterprise Linux V7 (x86/x64).
- Additionally, over 65 engineering improvements and bug fixes to extraction capabilities, performance, viewing, and user experience.

5.60.2 Updates

- Added extraction support for .EGG and .ALZ archive types. (SF837073 / FB8951)
- Added identification and metadata extraction support for JungUm documents. (FB8952)
- Added the ability for graphics formats to be converted to any supported output. (SF663636 / FB8469)
- Added the ability to turn off page numbers in PDF text only output using PDFPAGENUMBERS=ON|OFF (default ON). (FB7956)
- HD Mode: Added support for embedded objects when rendering MS Visio (VSD) documents. (FB8933)
- HD Mode: Added support for rendering linear gradients in MS PowerPoint (PPT). (FB9003)
- HD Mode: Implemented support for rendering borders on shapes in MS Office formats. (US15454)
- HD Mode: Implemented Unicode support for AutoCAD. (SF848432 / FB8999)
- HD Mode: Improved ability to render abstract lines in MS Visio. (FB9001)
- HD Mode: Improved handling of paragraph styles on hyperlinks in MS PowerPoint (PPTX). (FB9058)
- HD Mode: Improved line spacing in MS PowerPoint. (SF821810 / FB8912)
- HD Mode: Improved object placement and sizing when rendering MS Visio documents. (FB8957)
- HD Mode: Improved rendering of rotated shapes in MS Office formats. (FB8837)
- HD Mode: Improved rendering of text in drawings in MS Excel. (FB9002)
- HD Mode: Improved style inheritance when rendering MS PowerPoint. (FB8799)

- HD Mode: Improved support for rotated text in MS PowerPoint. (FB8918)
- HD Mode: Improved support for rotated text in MS Word. (FB8917)
- HD Mode: Improved tab spacing, line spacing, and text placement in MS Visio. (FB8713)
- HD Mode: Resolved a condition causing cell borders to appear incorrectly when processing MS Visio documents. (FB8946)
- HD Mode: Resolved a condition causing some formulas in MS Excel documents to display a value of zero rather than the computed output. (SF625761 / FB8283)
- HD Mode: Resolved a condition in MS Word where numbered lists inside text boxes were the incorrect size. (SF843466 / FB8991)
- HD Mode: Resolved a condition in MS Word where overlapping images may not render in the output. (SF843466 / FB8980)
- HD Mode: Resolved a condition when converting to HTML5 that could cause a missing <div> tag resulting in hidden slides when rendering MS PowerPoint documents. (SF873170 / FB9028)
- HD Mode: Resolved a condition where an EMF embedded in a PPT would not render in the correct color. (FB8925)
- HD Mode: Resolved a condition where background colors in MS Excel may cause a cascading effect across cells resulting in incorrect background shading. (SF625761 / FB8282)
- HD Mode: Resolved a condition where body text may overlap with a document header in MS Word. (FB9004)
- HD Mode: Resolved a condition where corrupt MS Word files could cause an access violation. (FB8975)
- HD Mode: Resolved a condition where diagram labels in MS PowerPoint rendered with the incorrect orientation. (SF783616 / FB8778)
- HD Mode: Resolved a condition where fonts in MS Visio documents scaled incorrectly. (FB8948)
- HD Mode: Resolved a condition where images within MS Word documents were missing in the HTML output. (SF721653 / FB8959)
- HD Mode: Resolved a condition where labels rendered incorrectly in MS Visio. (FB8934)
- HD Mode: Resolved a condition where MS PowerPoint documents may display an extra blank page. (FB8964)
- HD Mode: Resolved a condition where MS Visio documents contained extraneous lines. (FB8946)
- HD Mode: Resolved a condition where MS Word documents may output extraneous pages. (FB9070)
- HD Mode: Resolved a condition where structured XML may contain invalid characters. (SF800405 / FB8834)
- HD Mode: Resolved a condition where Structured XML output was invalid when document contained hyperlinks with non-escaped characters (&<>). (FB8944)
- HD Mode: Resolved a condition where text may not render correctly on drawings in MS Visio. (FB8950)
- HD Mode: Resolved a condition where the text within a MS Word document would be positioned over the page header. (FB9004)

- HD Mode: Resolved a condition where TIFF images rendered with the incorrect color. (SF846323 / FB8979)
- HD Mode: Resolved a condition where transparency in Microsoft PowerPoint (PPT) files was handled incorrectly. (FB8919)
- HD Mode: Resolved a condition where vertical alignment of cells in MS Excel was not rendered correctly. (SF798269 / FB8820)
- HD Mode: Resolved a condition where vertical text in MS PowerPoint rendered horizontal. (SF821815 / FB8908)
- Improved detection of misnamed tab-separated text files resulting in more precise extraction. (SF880514 / FB9044)
- Improved header/footer placement and scaling when rendering MS Visio documents with variable scale factors across pages. (FB8967)
- Improved the extractor.Open call by changing the default behavior to IGR_BODY_AND_META for all language bindings. (SF877000 / FB9041)
- Resolved a condition causing a hang on certain password protected RAR files. (SF833483 / FB8996)
- Resolved a condition introduced in engineering build 11.1.0.1464 causing the Java API to throw an exception when converting to TIFF using a custom IGRStream through ByteArrayOutputStream. (SF838124 / FB8982)
- Resolved a condition that could result in extra content being hidden when the option SHOWHIDDEN=HIDDEN is invoked. (FB9015)
- Resolved a condition where corrupt DGN files caused a crash. (FB8574)
- Resolved a condition where extra characters rendered in EML files that contain EML subfiles. (SF820346 / FB8961)
- Resolved a condition where images might not render in MS Visio. (FB9073)
- Resolved a condition where Japanese text in OLE objects could be missing when converting MS PowerPoint files to classic HTML. (SF874214 / FB9029)
- Resolved a condition where MSG header labels would not render when converting to an HD format. (SF819302 / FB8926)
- Resolved a condition where negative numbers in MS Excel were incorrectly being treated as fractions. (SF822333 / FB8911)
- Resolved a condition where not all recipients of an EML file displayed in the output. (SF825659 / FB8930)
- Resolved a condition where older MS Visio documents may fail to open. (FB8945)
- Resolved a condition where re-colored images in MS PowerPoint were rendered incorrectly. (FB8937)
- Resolved a condition where some files could be incorrectly recognized as WordPerfect files. (FB9035)
- Resolved a condition where some MS Visio files may get an IGR error when opening the document. (FB9076)
- Resolved a condition where some text files were not detected correctly. (SF876537 / FB9032)
- Resolved a condition where some VMBX files were misidentified as ASCII files. (SF814805 / FB8906)
- Resolved a condition where text extraction of TNEF EML subfiles did not extract correctly. (SF820346 / FB8905)

- Resolved a condition where text justification in MS Word files rendered incorrectly. (SF843466 / FB8992)
- Resolved a condition where thumbnails rendered incorrectly. (FB8929)
- Resolved a potential crash when processing certain MS Excel documents. (SF833483 / FB9000)
- Resolved conditions where objects may be rendered incorrectly and text indentation was incorrect in MS Visio. (FB8714)
- Text Mode: Resolved a condition where processing a XLSX file using text mode and EXCELMODE=CSV would cause Document Filters to hang. (SF883987 / FB9056)
- Text Mode: Resolved a condition where subfile text extraction is performed with EMAILMODE=MAPIHEADERS when a MSG subfile embedded in another MSG would have missing metadata. (SF889707 / FB9079)
- Text Mode: Updated MSG metadata to output “Date” as the message delivery time, “Sent” as the message submit time if it differs from “Date,” and SMTP email address information. To extract all email metadata content without field mapping interpretation, EMAILMODE=ALLHEADERS may be used. (FB8518)

5.61 Document Filters 11.1 Build 1464 (July 2014)

5.61.1 Enhancements

New APIs

- The GRAPHIC_DPI option now supports an additional parameter for TIFF input documents: AUTO.

Added Formats

- Added full HD support for WMF files and clip art on all platforms.
- Added full HD support for EMF files and clip art on all platforms.
- Certified extraction support for MS OneNote 2013.

Enhanced Formats

- Added extraction of page-level orientation, source width and source height metadata from multi-page TIFF documents to the structured XML output.

5.61.2 Updates

- HD Mode: Added WordNo to the title attribute of each word in HD output when hOCR is enabled. (FB8943)
- HD Mode: Enhanced text placement when text was wrapped around an image in MS Word. (FB8838)
- HD Mode: Fixed an issue with horizontal line rendering in MS Word files. (800425 / FB8831)
- HD Mode: Implemented strikethrough support in MS Excel. (800404 / FB8830)
- HD Mode: Improved ability to render complex rotation and shapes with arcs in MS Visio. (FB8712)
- HD Mode: Improved adherence to text box style attributes when processing MS PowerPoint (PPTX) documents. (FB8914)
- HD Mode: Improved alignment logic of multi-column bulleted lists in MS PowerPoint. (FB8862)
- HD Mode: Improved alignment of the contents of text boxes in MS PowerPoint. (FB8899)
- HD Mode: Improved arrow placement when processing MS Word documents. (783520 / FB8764)
- HD Mode: Improved background color support for font styles in MS Excel. (FB7628)
- HD Mode: Improved cell alignment in MS Excel. (798262 / FB8819)
- HD Mode: Improved cell indentation logic on MS Excel. (798254 / FB8816)
- HD Mode: Improved cell width rendering for tables within text boxes in MS Excel. (FB7637)

- HD Mode: Improved color mapping in MS PowerPoint. (FB8728)
- HD Mode: Improved conversion of images with an asymmetric DPI. (786072 / FB8806)
- HD Mode: Improved font rendering in MS PowerPoint (PPTX) in multi-level lists. (FB8815)
- HD Mode: Improved footer text placement and wrapping logic in MS Word (DOCX). (800921 / FB8836)
- HD Mode: Improved handling of numbering and Table of Contents in MS Word documents. (FB5972)
- HD Mode: Improved handling of space before the first paragraph in MS PowerPoint text boxes. (FB8798)
- HD Mode: Improved header and footer object placement when processing MS PowerPoint documents. (FB8915)
- HD Mode: Improved image positioning in MS Word when no text is present. (FB8825)
- HD Mode: Improved inheritance of font size and styles from master shapes in MS Visio. (771369 / FB8845)
- HD Mode: Improved page order when processing an MS Visio document. (771369 / FB8843)
- HD Mode: Improved processing of shapes in MS Visio. (FB8718)
- HD Mode: Improved shape rendering and placement for MS Visio. (FB8715)
- HD Mode: Improved style handling of master shapes in MS Visio. (771369 / FB8846)
- HD Mode: Improved support of Russian MS Excel. (815848 / FB8891)
- HD Mode: Improved table border processing in MS Word. (783520 / 8761)
- HD Mode: Improved table row formatting and table border logic when processing MS Word documents. (783520 / FB8761)
- HD Mode: Improved text wrapping in MS PowerPoint. (FB8898)
- HD Mode: Improved the ability to render custom lines in MS Word documents. (800425 / FB8831)
- HD Mode: Improved the handling of cell overflow in HTML mode for MS Excel documents. (FB8869)
- HD Mode: Improved the text alignment for MS PowerPoint. (783662 / FB8790)
- HD Mode: Improved vertical alignment of shapes in MS PowerPoint. (FB8849)
- HD Mode: Resolved a condition in MS Excel (XLSX) where custom cell formatting information may incorrectly appear in the rendered output. (800414 / FB8829)
- HD Mode: Resolved a condition in MS Word in which images placed inside of a table may have rendered outside of the table. (800111 / FB8852)
- HD Mode: Resolved a condition where an embedded table in MS Word may not have rendered. (FB8774)
- HD Mode: Resolved a condition where custom bullets in MS PowerPoint may not be rendered correctly. (783662 / FB8791)
- HD Mode: Resolved a condition where near zero values in MS Excel may round to zero. (788993 / FB8801)
- HD Mode: Resolved a condition where numeric cells formatted as Accounting format may display as a hyphen instead of zero. (801611 / FB8841)

- HD Mode: Resolved a condition with MS PowerPoint where footer text may be missing. (783616 / FB8776)
- HD Mode: Resolved a potential condition where cell background in MS Excel may incorrectly render as black. (FB8860)
- HD Mode: Resolved an issue in MS Excel where month values may be incorrectly extracted. (FB8805)
- HD Mode: Resolved an issue in MS PowerPoint where arrow sizes may have been incorrect. (FB8730)
- HD Mode: Resolved an issue in MS PowerPoint where bulleted lists may display the incorrect bullet symbol. (FB8870)
- HD Mode: Resolved an issue in MS PowerPoint where fill color for images may be incorrect. (FB8857)
- HD Mode: Resolved an issue in MS PowerPoint where overlapping shapes may appear incorrectly overlaid. (FB8803)
- HD Mode: Resolved an issue in MS PowerPoint where page numbers may render in the output when they were hidden in the original document. (783662 / FB8789)
- HD Mode: Resolved an issue in MS PowerPoint where text may have overlapped an image when converting the document to TIFF. (783662 / FB8792)
- HD Mode: Resolved an issue in MS Word where document headers may be missing text or missing parts of images. (800424 / FB8828)
- HD Mode: Resolved an issue in which drawings in some MS Excel files failed to render. (663636 / FB8409)
- HD Mode: Resolved an issue in which some MS Visio files may not convert to HD. (771369 / FB8698)
- HD Mode: Resolved an issue where converting text files to HD at a high DPI caused page numbers to overlap document text. (FB8848)
- Implemented the ability to stop reading a custom stream by the calling application returning -1 from the IGR_Stream read/seek methods. (820344 / FB8940)
- Improved identification capabilities of MS OneNote table of contents. (791279 / FB8821)
- Improved memory usage and speed when processing MS Excel documents with drawings and images in EXCELMODE=CSV. (806720 / FB8861)
- Improved performance of EML sub-files. (781959 / FB8768)
- Improved scientific notation number formatting for MS Excel files. (800403 / FB8827)
- Resolved a condition causing words to appear merged when processing certain PDF documents. (779677 / 8711)
- Resolved a condition in MS Word where output may include extraneous text. (FB8688)
- Resolved a condition where redundant metadata blocks were output during extraction of TNEF documents. (SF801572 / FB8859)
- Resolved a condition where Russian characters in a path name or file name were incorrectly encoded. (FB8786)
- Resolved an issue in MS Excel where some cells may be shifted. (791361 / FB8818)

- Resolved an issue with text box placement in headers in MS Word. (767405 / FB8681)
- Text Mode: Added the ability to extract ALT text for IMG files by default. SHOWHIDDEN=OFF and SHOWHIDDEN=HIDDEN may be used to suppress the ALT text. (791777 / FB8812)
- Text Mode: Enhanced extraction support for ZIP containers with Unicode member names authored by InfoZip, PkZip and WinRAR. Resolved a condition causing extraction of sub-files of a ZIP authored by ALZip to fail. (800389 / FB8854)
- Text Mode: Resolved a condition resulting in spaces being removed when processing MS XPS documents. (831444 / FB8942)
- Text Mode: Resolved a potential high memory usage condition when processing email attachments on Linux PPC. (FB8894)
- Text mode: Resolved potential segmentation fault when processing certain MS Excel (XLSX) files generated by OpenOffice. (791278 / FB8866)

5.62 Document Filters 11.1 Build 1394 (April 2014)

5.62.1 Enhancements

New APIs

- HD TIFF output now includes the ability to specify Black and White as well as color options. TIFF compression type can also now be controlled.
- HD TIFF, BMP, JPG and PNG now support graphic effects of grayscale and desaturation.
- HD Microsoft PowerPoint now has several options associated to rendering speaker notes.

Added Formats

- Added extraction and full HD support for Hancom Office .CELL
- Added extraction and full HD support for Hancom Office .SHOW
- Added extraction and full HD support for MS Office 365 Word
- Added extraction support for dBaseIV
- Added extraction support for UNIX Compress archives
- Added identification support for FITS Image (.FTS)
- Added identification support for Wordstar for Windows version 2
- Certified extraction support for MS Project 2013

Enhanced Formats

- Added chart data extraction support for MS Office formats
- HD Mode: Added the ability to process embedded fonts in MS Word 2007+ documents. (FB8652)
- HD Mode: MS Excel processing has been improved resulting in a lower memory footprint, added support for additional font styles and enhanced chart rendering capabilities.

5.62.2 Updates

- Added support for Office 365 MS Word. (FB8734)
- Corrected a potential crash on WMF documents when converting to HTML on Linux. (FB8601)
- Fixed a potential crash when processing certain PDF documents. (FB8744)
- Fixed an issue where PNG transparency would render as black on Linux ARM. (FB8636)

- HD Mode: Added the ability to extract slide notes of MS PowerPoint in High Definition. (FB8779)
- HD Mode: Addressed an issue where some shapes in an XLSX file may cause a large amount of memory to be consumed and a failure to render. (FB8732)
- HD Mode: Addressed an issue with Unicode charts in MS Excel. (FB8737)
- HD Mode: Addressed issue where groups of shapes may not render correctly in MS Word (DOCX) documents. (FB8723)
- HD Mode: Fixed an issue with text box placement when anchored on the bottom margin. (FB8725)
- HD Mode: Fixed rendering issues of some shapes in MS PowerPoint. (FB8729)
- HD Mode: Improved ability to detect and render text that overlaps one or more cells in MS Excel. (FB8665)
- HD Mode: Improved memory usage when processing some large MS Excel documents. (FB8643)
- HD Mode: Improved placement of text in and around shape elements in MS PowerPoint. (FB8757)
- HD Mode: Improved positioning of x-axis on column charts in MS Excel. (FB8674)
- HD Mode: Improved rendering accuracy of table of contents when processing MS Word documents converted from PDF. (FB8708)
- HD Mode: Improved rendering capabilities of partially corrupt MS Word documents. (FB8760)
- HD Mode: Improved rendering of fill color and text in MS Visio. (FB8719)
- HD Mode: Improved rendering of MS Word documents converted from PDF. (FB8659)
- HD Mode: Improved rendering of text within text boxes in MS Excel (XLSX). (FB8670)
- HD Mode: Improved the handling of cascading styles in MS Word documents. (FB8752)
- HD Mode: Improved the precision of text layout in relation to drawn elements in MS PowerPoint. (FB8758)
- HD Mode: Improved the rendered output when converting foreign-language MS Word documents to TIFF. (FB8754)
- HD Mode: Resolved a condition in MS Excel where background color would not render. (FB8703)
- HD Mode: Resolved a condition in MS PowerPoint where extra lines may be drawn on complex drawings. (FB8766)
- HD Mode: Resolved a condition in MS PowerPoint where the incorrect slide background may display on Linux. (FB8702)
- HD Mode: Resolved a condition in MS Visio where hidden images may be displayed. (FB8722)
- HD Mode: Resolved a condition in MS Word where cell width in tables may be incorrectly calculated. (FB8677)
- HD Mode: Resolved a condition where Creation Date and Modified Date metadata in MS Excel will have the correct date, but the incorrect day of the week. (FB8682)
- HD Mode: Resolved a condition where elements of an MS Word document may be positioned outside of the visible right margin. (FB8673)
- HD Mode: Resolved a condition where images may not display when processing RTF. (FB8721)

- HD Mode: Resolved a condition where page borders on MS Word (DOCX) documents may not appear. (FB8650)
- HD Mode: Resolved a condition where subscript may be rendered as superscript when converting MS Word documents on Linux. (FB8773)
- HD Mode: Resolved a condition where tables in MSG files may not render correctly. (FB8671)
- HD Mode: Resolved a condition where text boxes in headers may not render correctly in HTML mode. (FB8676)
- HD Mode: Resolved a condition where values formatted as percent may appear as decimal. (FB8747)
- HD Mode: Resolved a potential condition causing cell content not to display. (FB8672)
- HD Mode: Resolved a potential condition in MS PowerPoint where bold fonts may be rendered as italics. (FB8709)
- HD Mode: Resolved a potential condition in MS Word where headers and footers may be omitted. (FB8686)
- HD Mode: Resolved an issue in MS PowerPoint where borders and arrows may be omitted. (FB8767)
- HD Mode: Resolved an issue in MS Visio where some colored icons may render colorless. (FB8717)
- HD Mode: Resolved an issue in MS Visio where some shapes and text may not render correctly. (FB8720)
- HD Mode: Resolved an issue where charts may not render in some XLS documents. (FB8738)
- HD Mode: Resolved an issue where hard line breaks may not be honored in PPTX files. (FB8735)
- HD Mode: Resolved an issue with bullet placement in some DOCX files. (FB8724)
- HD Mode: Resolved an issue with line wrapping in MS Word headers. (FB8765)
- HD Mode: Resolved an issue with mis-placed text in rotated text boxes in MS PowerPoint. (FB8736)
- HD Mode: Resolved an issue with overflowing cell text and missing text in some XLSX documents. (FB8739)
- Improved encoding detection and handling of Unicode characters in MSG files. (FB8745)
- Improved exception handling and memory usage on very large MS Excel documents. (FB8741)
- Improved exception handling to prevent a possible segmentation fault when processing certain PDF documents on Linux. (FB8746)
- Resolved a possible segmentation fault on Linux (intel-64) when processing an MS Office 2007+ document modified by LibreOffice. (FB8635)
- Resolved an issue where slide notes in PPT files failed to render. (FB8779)
- Resolved an issue where the stored DPI in generated TIFF images may be incorrect, causing it to report an incorrect page size. (FB8658)
- Text Mode: Resolved an issue in RTF where encoded image data may appear in the output. (FB8691)
- Text Mode: Updated identification logic to prevent mis-identification of some text files as WordPerfect 4.2. (FB8651)
- Updated the Java API to provide more detailed exception messages. (FB8775)

5.63 Document Filters 11.1 Build 1333 (March 2014)

5.63.1 Enhancements

- Text Mode: Added a new SubType code for Lotus 1-2-3 files for more descriptive identification. (FB8506)
- Now support vCalendar format identification and extraction. (FB8582)

5.63.2 Updates

- Added identification capability for FTS formats. (FB8630)
- Corrected an issue when extracting title strings from PDF documents in certain encodings. (FB8586) (Perceptive Enterprise Search)
- Enhanced the image output for embedded image files within RTF files. (FB8116)
- Fixed a problem Document Filters had with processing the TextRulerAtom in PPT files. Fixed a problem with the space between bullets and the text following bullets. (FB8559)
- Fixed a problem where document filters was not handling derived master styles properly in PPT files. (FB8558)
- Fixed an issue where DBF files may be incorrectly identified as ASCII/text. (FB8602)
- Fixed an issue where JVM crashes when processing multiple threads of AutoCAD documents. (FB8585)
- Fixed an issue where SIS files may be incorrectly identified as type 3 (Wordstar). (FB8587)
- Fixed an issue where some special characters may not convert correctly in Classic HTML mode. (FB8556)
- Fixed an issue where the LC_TIME locale may be incorrect when using EXCELMODE=CSV. (FB8198)
- HD Mode: Check box selection in MS Word is now respected in the rendered output. (FB8591)
- HD Mode: Corrected an issue where HD conversion of TIFF input documents would only generate the first page during conversion. (FB8468)
- HD Mode: Corrected an issue where HTML output may display html code around email addresses. (FB8555)
- HD Mode: Corrected issue where text may be missing from text boxes in MS PowerPoint. (FB8571)
- HD Mode: Fixed an issue for MS PowerPoint in HD mode where HD output may contain overlapping text and lines over text. (FB8324)
- HD Mode: Fixed an issue with MS Word documents in HD mode where some links may not work. (FB8551)
- HD Mode: Improved MS Word line space rendering in HD mode. (FB8552)
- HD Mode: Resolved a condition where images could be missing in MS PowerPoint. (FB8560)
- HD Mode: Resolved an issue in HD mode for MS Word documents where page numbers in the footer may be missing. (FB8312)
- HD Mode: Resolved an issue with extraneous line breaks when converting EML to HD. (FB8554)

- HD Mode: Resolved an issue with line breaks when converting MSG to TIF in HD mode. (FB8569)
- HD Mode: Resolved potential incorrect page number calculations when processing a MS Word document generated by a non-English version of MS Word. (FB8534)
- Improved exception handling during MS Excel chart processing. (FB8619)
- Improved identification of MP3 and M3U files. (FB8541) (Perceptive Search Integration Kit)
- Improved text extraction for SXI. (FB8603)
- Improved the column spacing and column wrapping for MS Word documents. (FB8233)
- Improved the exception handling when opening document streams and running in multiple threads. (FB8131)
- Improved the handling of embedded images in EML files when converting to HTML. (FB8607)
- PDFDEHYPHENATE option now only removes hyphens at the end of lines. (FB8588)
- Resolved a condition where formatting information may be extracted as text when using EXCELMODE=CSV. (FB8539)
- Resolved a condition where the Author metadata field may not be extracted in older versions of MS Word. (FB8509)
- Resolved a potential memory access violation on Linux and Mac when using PSTMESSAGE TYPE=EML. (FB8532)
- Resolved a potential memory leak when processing sub-files within a MS Word document. (FB8547)
- Resolved an issue in MS Office 2003 files where accented character conversion might not be consistent across platforms. (FB8181)
- Resolved an issue in MS Word formats where line breaks may appear to be missing. (FB8237)
- Resolved an issue where cells in MS Excel documents, that contain both inline and shared string data, may not convert correctly. (FB8561)
- Resolved an issue where text may be missing from text boxes in HTML mode. (FB8309)
- Resolved an issue where ZIP files with Korean file names may cause a segmentation fault. (FB8189)
- Text Mode: Extraction of DBASE4 files is now supported. (FB8440)
- Text Mode: Fixed an issue in Text mode where, in some scenarios, codepage text could be output in the incorrect location. (FB8584)
- Text Mode: Fixed an issue where a DOC file within a DOC file may be misidentified as "unknown". (FB8581)
- Text Mode: Fixed an issue where WordPerfect 1.0 files may be incorrectly identified on the Mac OS. (FB8463)
- Text Mode: Identification for Wordstar 2.0 is now supported. (FB8449)
- Text Mode: Improved the identification of text files without extensions. (FB8624)
- Text Mode: Password protected MS Word files now return a correct error code (IGR5). (FB8578)
- Text Mode: Resolved a condition in MS OneNote where the first line of text may be truncated. (FB8531)

- Text Mode: Resolved a logic error that may prevent some text from being extracted, when reading XLSB files. (FB8605)
- Text Mode: Resolved an issue where VSDX files may be incorrectly identified as ZIP. (FB8594)
- Text Mode: Resolved potential missing content when processing an EML file. (FB8599)

5.64 Document Filters 11.1 Build 1262 (December 2013)

5.64.1 Enhancements

- Improved text extraction from charts. (FB8435)
- Text Mode: Improved the extraction of embedded files within TNEF EML documents. (FB8457)
- Java exceptions are now checked and error number is returned in addition to error string. (FB8462)
- Added options for generating black & white and grayscale TIFF output. (FB8466)
- Allow override of maximum open document handles via `ISYS_MAX_DOCHANDLES` environment variable (default: 64). (FB8507)

5.64.2 Updates

- Added support for Ichitaro author field extraction. (FB8446)
- Document Filters now identifies MS PowerPoint 4.0 files for Mac; the capabilities for this file type are zero. (FB8476)
- Fixed a potential condition where text would not be extracted from WordPerfect files. (FB8508)
- Fixed a problem with child shape placement not taking into account parent shape rotation in group shape containers (Microsoft Office Shapes). (FB8493)
- Fixed a problem with non-breaking spaces being inserted where they should not be in classic HTML. (FB8478)
- Fixed an issue concerning PPTX files where user-added text in layout and master slides was not being output when using doc2text in text mode. (FB8437)
- Fixed an issue where Adobe Postscript may be misidentified as text type 2. (FB8482)
- Fixed an issue where empty subfolders, within RAR archives, may be incorrectly interpreted as documents. (FB8491)
- Fixed an issue where HTML content may be appended into a tag and effectively skipped from the output. (FB8447)
- Fixed an issue where SMTP addresses were not being extracted from PST files. (FB8487)
- Fixed an issue where text extraction of a WordStar file may create extraneous, unreadable text. (FB8515)
- HD Mode: Corrected Classic HTML PPT bullet alignment. Improved HD handling of words that are split immediately following a bullet. (FB8488)
- HD Mode: Fixed a problem on PPT files, in HD and HTML5, where text was showing that should have been hidden by an image. (FB8417)
- HD Mode: Fixed a problem where the symbol font was being applied to non-symbol text in some cases. (FB8438)
- HD Mode: Fixed an issue of parsing words incorrectly when converting to HTML. (FB8451)
- HD Mode: Fixed an issue where page numbers would not be displayed in classic HTML and HD modes on some PPT files. Fixed a problem with displaying footer information on the title page when the option "Don't show on title page" was checked. (FB8434)

- HD Mode: Improved chart fidelity across formats when multiple columns are present. (FB8425)
- HD Mode: Improved indentation and pagination for MSG documents. (FB8490)
- HD Mode: Improved output images / raster scaling based on DPI (default: 96). (FB8525)
- HD Mode: Improved overall text placement in MS Word and addressed potential image ghosting behavior. (FB8414)
- HD Mode: Improved processing of VML images/drawings in MS Word. (FB8404)
- HD Mode: Improved the alignment of text and image callouts in MS Word. (FB8416)
- HD Mode: Improved the chart scaling in MS Excel when changing DPI settings. (FB8453)
- HD Mode: Improved the handling of character offsets in word processing formats when sections are variable in length. (FB8448)
- HD Mode: Improved the rendering of axis labels for charts. (FB8402)
- HD Mode: Improved the rendering of text box borders. (FB8415)
- HD Mode: Improved the support for complex header/footer content in MS Word. (FB8418)
- HD Mode: Improved the text layout of bullet lists in MS PowerPoint. (FB8421)
- HD Mode: Improved the text rendering in MS Word when appearing in a table with same background color. (FB8424)
- HD Mode: Improved viewing of images in MSG files. (FB8412)
- HD Mode: Resolved a condition where percentages may be converted to decimal in MS Excel. (FB8420)
- HD Mode: Resolved a potential condition where page numbers in MS Word were not calculated correctly. (FB8413)
- HD Mode: Resolved a potential overlapping text condition in MS PowerPoint. (FB8474)
- HD Mode: Resolved a potential segmentation fault when processing MS Excel on Linux in Classic HTML mode. (FB8500)
- HD Mode: Resolved an issue of missing characters in text files when outputting in paginated HD modes. (FB8526)
- HD Mode: Resolved an issue where cells in charts may incorrectly appear when a background color is present. (FB8422)
- HD Mode: Resolved an issue where hidden MS Excel sheets would display when the SHOWHIDDEN option was set to OFF. (FB8423)
- HD Mode: Resolved an issue where incorrect sizing of charts in XLS would cause repeating charts to display. (FB8464)
- HD Mode: Resolved an issue where some images may not render in PPTX files. (FB8419)
- HD Mode: Resolved an issue where table borders may not render correctly in MS Word. (FB8428)
- Improved character mapping for MSG documents. (FB8520)
- Improved character set detection on EML documents. (FB8498)
- Improved character set mapping on VCF documents. (FB8499)

- Improved non-Latin font mapping support for EML documents. (FB8465)
- Improved support of KOI8-R Russian/Cyrillic character encoding. (FB8492)
- Improved text extraction speed of MS Word, MS Excel, and MS PowerPoint documents. (FB8501)
- Improved the ability to extract text from VML shapes. (FB8455)
- Improved the chart rendering in MS Excel when extracting single digit double values. (FB8452)
- Improved the metadata and text extraction of Framework WP (FW3). (FB8443)
- Improved the metadata and text extraction of IBM DCA/FFT files. (FB8441)
- Resolved a potential extraction failure on WordPerfect for Mac. (FB8459)
- Resolved a scenario where IBM DisplayWrite files may be identified as text. (FB8504)
- Resolved an issue where hard spaces may not display in text extraction. (FB8521)
- Resolved an issue where some emails extracted from PST files would be missing some metadata. (FB8337)
- Resolved an issue where text extraction of an MS Excel file may create extraneous, unreadable text. (FB8514)
- Resolved an issue where the extracted text, for some embedded images in Word Write files, would generate unexpected tokens and control characters. (FB8460)
- Resolved an issue where TXT files may be incorrectly converted as WordPerfect 4.2. (FB8489)
- Text Mode: Fixed potential mis-identification of Windows Metafile. (FB8513)
- Text Mode: Improved MS PowerPoint extraction to check for TEXT_CONTENT stream even when no slide data exists. (FB8481)
- Text Mode: Improved the extraction of hidden binary content from WinWord 2.0. (FB8510)
- Text Mode: Improved the text extraction of Multimate documents. (FB8450)
- Text Mode: Resolved a potential scenario where metadata may not be extracted correctly in MS Excel. (FB8516)
- Unsupported MS PowerPoint 3.0 files are now identified but have zero capabilities. (FB8483)

5.65 Document Filters 11.1 (October 2013)

5.65.1 Enhancements

Below is a summary of new features in Document Filters 11.1. Detailed API usage information can be found in the Document Filters Implementation Guide.

New APIs

Added support for Python versions 2.6, 2.7 and 3.x. Includes bindings, code samples and function prototypes.

Additional Platform Support

Added support for ARM v7 (Marvell Armada PJ4B). Implemented infrastructure to expedite the certification of additional ARM architectures when requested by customers.

Additional Format Support

New Formats

- Added support for AutoCAD 2013.
- Added support for DMG containers when running on Mac OSX 32/64.

Enhanced Formats

- Enhanced metadata extraction and viewing fidelity for all AutoCAD versions.
- Extended AutoCAD support to all platforms (meta-only where HD is not supported).
- Extended support for iWork 09 (Pages, Numbers, Keynote) to provide chart data extraction.

Other Features

- All Output modes contain significant performance and viewing fidelity improvements. Notables include:
- Major enhancements to speed, memory usage, and viewing fidelity of MS Excel.
- High Definition conversion to PDF now generates much smaller output.
- Improved consistency of output across all formats on all platforms.
- Perceptive HD Viewer now has enhanced page controls including point and click element selection and page drag.

5.65.2 Updates

- Fixed issue where certain Unicode characters in HTML could be filtered out. (FB8353)
- HD Mode: Fixed a potential crash when running Java 6/7 on Solaris / Intel-64. (FB8369)
- HD Mode: Fixed floating point rounding errors in MS Excel documents. (FB8390)
- HD Mode: Fixed issue where images may not display in MS Word documents. (FB8376)
- HD Mode: Fixed issue with text overlapping in MS Excel. (FB8347)
- HD Mode: Fixed issue with Word document processing: bullet points and inherited styles. (FB8348)
- HD Mode: Fixed problem with processing certain MS Excel format strings. (FB8273)
- HD Mode: Improved font handling when converting to HTML. Added an option, `USE_MAPPED_FONTS`, allowing usage of the substituted layout font rather than the referenced font. (FB8375)
- HD Mode: Improved image fidelity of images embedded in MS Word documents on Solaris Intel 32/64. (FB8358)
- HD Mode: Improved line spacing on MS Word for some scenarios. (FB8338)
- HD Mode: Improved MS Word support for Korean hyperlinks. (FB8249)
- HD Mode: Improved output consistency across locales for MS Excel documents. (FB8373)
- HD Mode: Improved precision of generated HD HTML for all formats. (FB8340)
- HD Mode: Improved the handling of text extraction for ODP documents. (FB8222)
- HD Mode: Improved the output of tables in MS Word documents. (FB8335)
- HD Mode: Improved the PDF output quality when saving TXT files to PDF. (FB8308)
- HD Mode: Improved the readability of PDF-converted TXT files. (FB8322)
- HD Mode: Improved the rendering of EML files. (FB8288)
- HD Mode: Improved the rendering of PDFs with in-line images with text. (FB8341)
- HD Mode: Improved the search-ability of EML and MSG files that are saved as HTML. (FB8339)
- HD Mode: Improved the spacing between list items in MS Word documents. (FB8359)
- HD Mode: Improved title colors and the display of Chinese characters for Word documents. (FB8275)
- HD Mode: Improved tolerance of corrupt MSG files. (FB8367)
- HD Mode: Resolved an error where certain MS Word documents may not convert due to too many paragraph splits. (FB8336)
- HD Mode: Resolved an issue where converting to PDF might cause excessively large output files. (FB8330)
- HD Mode: Resolved an issue where dashed lines were not properly aligned with text in MS Word documents. (FB8289)
- HD Mode: Resolved an issue where `IGR_Redact_Page_Text()` will not work unless the page wordlist is pre-filled. (FB8318)

- HD Mode: Resolved an issue where images in EML documents were not displaying. (FB8378)
- HD Mode: Resolved an issue where images may not be extracted when using Java on Linux. (FB8360)
- HD Mode: Resolved an issue where PPT conversion could produce "not enough points for curve" error on Linux. (FB8332)
- HD Mode: Resolved an issue where some MS PowerPoint PPT slides were not converted. (FB8286)
- HD Mode: Resolved an issue where text may overlap when converting MS PowerPoint to HD HTML. (FB8363)
- HD Mode: Resolved an issue where the background color for DOC files was incorrect. (FB8315)
- HD Mode: Resolved an issue where the body of an attached MSG file would not convert. (FB8247)
- HD Mode: Resolved an issue where the output, when converting PDF to HD PDF, would interleave words and characters. (FB8319)
- HD Mode: Resolved an issue with corrupt JPEGs when converting all MS PowerPoint on Linux. (FB8331)
- HD Mode: Resolved an issue with PPT resulting in "invalid string position error". (FB8329)
- HD Mode: Resolved issue with full-screen charts in MS Excel documents. (FB8396)
- HD Mode: Resolved issue with MS Word SmartArt (colors and fonts). (FB7897)
- HD Mode: Resolved issue with sparsely populated columns in MS Excel documents. (FB8406)
- HD Mode: Resolved issue with spreadsheets that do not have stored defaults for column width and row height. (FB8388)
- HD Mode: Resolved issue with TIFF creation on Big-Endian platforms (e.g. Solaris SPARC). (FB8343)
- HD Mode: Resolved potential issue with PDF generation. (FB8399).
- HD Mode: The Open_Document method now returns an error (IGR_E_FONTS_NOT_FOUND) when no fonts exist. (FB8354)
- Improved C++ samples to be more compatible with compiling C programs across platforms. (FB8366)
- Improved consistency of CAD metadata across platforms. (FB8295)
- Improved extraction of metadata from Project documents. (FB8383)
- Improved extraction of MS PowerPoint PPTX files that contain embedded PDF files. (FB8303)
- Improved font mapping and missing font exception handling across all platforms. (FB8356)
- Improved handling of number and percent formatting in MS Excel documents when EXCELMODE=CSV option is invoked. (FB8379)
- Improved PDF Export for non-HD supported formats. (FB8349)
- Improved text alignment, multi-level lists and image rendering in PowerPoint. (FB8250)
- Improved the highlighting for HD HTML-converted PDF documents. (FB8311)
- Resolved a pagination synchronization issue for MS Word documents in the HD Viewer. (FB8334)

- Resolved an issue where additional spaces and commas were added in Excel files when using the EXCELMODE=CSV option. (FB8270)
- Resolved an issue where all file names in Joliet ISO format were upper case. (FB8272)
- Resolved an issue where some extracted PST files would be missing some metadata. (FB8337)
- Resolved an issue with the Perceptive HDViewer on French Windows 7. (FB8326)
- Resolved issue with missing fonts on MacOS X. (FB8387)
- Text Mode: Improved extraction of legacy MS Excel content. (FB8259)
- Text Mode: Resolved issue when processing Japanese text inside HTML tags. (FB8392)

5.66 Document Filters 11.0 (July 2013)

5.66.1 Enhancements

Below is a summary of new features in Document Filters 11.0. Detailed API usage information can be found in the Document Filters Implementation Guide.

High Definition (HD) Output Modes

- HD includes seven new output modes that provide absolute pixel positioning in order to preserve the original look and feel. Any supported input format can be converted to the following output modes:
- JPG, BMP, PNG — Flexible page level image output and drawing control.
- Single and Multipage TIFF — Ideal for eDiscovery, compliance and archiving solutions that require precise output manipulation. Support for TIFF 6.0 options included.
- Searchable PDF — Convert documents to this universal format for publication and information sharing.
- Paginated/HD HTML — Convert and extract documents with page-level fidelity, access, and control.
- Structured XML — hOCR-compliant structured output for easy consumption by applications. This method exposes a full-featured document object model (DOM), including pixel placement, giving total processing control to developers. This mode provides all the information required to convert the document to additional output formats.
- HD output preserves complex formatting, fonts, styles, images, headers/footers, lists, shapes, tables, charts (2D & 3D), and automatic near perfect pagination.
- Document formats with full HD capability, including preservation of virtually all original formatting, include:

Microsoft Word (DOC)	Microsoft Word 2007+ (DOCX)	HTML (HTM/ HTML / MHT)	Email (MSG)
Microsoft PowerPoint (PPT)	Microsoft PowerPoint 2007+ (PPTX)	AutoCAD (DWG/ DGN)	Email (EML)
Microsoft Excel (XLS)	Microsoft Excel 2007+ (XLSX)	Word Perfect (WPD)	
Adobe Portable Document (PDF)	Open Document Text (ODT)	Rich Text (RTF)	
Open Document Presentation (ODP)	Open Document		
Spreadsheet (ODS)	Microsoft Visio (VSD)		

New APIs

- Over 20 new API functions focused on output control and image manipulation.
- Function highlights include:
 - Page Functions: When a document is opened in HD mode, you are able to work with it using the new Page functions. These include getting page counts, the words on a page, or converting a page to an image, HTML or PDF.
 - Canvas Functions: Document Filters introduces the concept of a Canvas. A Canvas is an object on which pages are rendered.
 - Your application can create a canvas for the desired output type and render one or more pages to it.
 - Once a page has been rendered, you can then markup, annotate or redact content by using the new drawing API included with the Canvas functions.
 - New Abilities:
 - Redaction (text, images, and any page elements)
 - Hit-Highlighting
 - Watermarks
 - Bates Stamps
 - Thumbnail generation
 - Annotation/markup support
 - Support for drawing and image attributes
 - Code samples include use of primary API functions.

Perceptive HD Viewer

- Perceptive HD Viewer allows viewing and export to any supported HD output mode.
- Delphi source code included to assist developers with API usage samples.

Additional Platform Support

- Added support for Windows Itanium 64-bit
- Added support for Linux Itanium 64-bit
- Added support for Linux PowerPC (PPC) 32-bit / 64-bit
- Added support for Linux IBM POWER 64
- Added support for FreeBSD 32-bit / 64-bit (x86/x64)

- All APIs and all output modes work across all platforms

Additional Format Support

- Certified support for Microsoft Office 2013
- Certified support for Microsoft Outlook 2013 (.PST/.OST) formats are now supported
- Added support for identification of:
 - Korean archive types .ALZ and .EGG
 - Outlook Express .DBX
 - Microsoft OneNote .TOC
 - WinMail data file
 - Java class .CLASS
 - Korean Word Processor .NDOC
- Added support for DICOM metadata .DCM
- Extended support for embedded object extraction from Hangul (.HWP)
- Extended support for EMLX file identification and extraction of text and metadata

Other Features

- High Definition HTML output from previous versions is now referred to as “Classic HTML” output. Output is now XHTML 1.0 compliant.
- Improved performance, rendering fidelity and lowered memory footprint.
- Charts are now supported in Word, PowerPoint, and Excel on Windows, Linux & Mac OSX.

API changes (since Document Filters 10.x)

- .NET Assembly renamed to Perceptive.DocumentFilters
- .NET FileReaders global factory object has been renamed to DocumentFilters
- Added enforcement of maximum memory allocations when using the .NET API of Document Filters. (FB7851)
- Added ID only support for OneNote TOC files. (FB6925)
- Added support for embedded files in RTF documents. (FB7459)
- Added the ability to generate thumbnails by page. (FB7419)
- Added the ability to hide hyperlinks, bookmarks and comments via the SHOWHIDDEN parameters. (FB7717)

- Added the option to display hidden content in Classic HTML mode using the SHOWHIDDEN option. (FB6611)
- Addressed an issue in Classic HTML mode on a Windows 64-bit platform where some RARs would consume a large amount of memory. (FB7725)
- Addressed an issue in Classic HTML output where table layout rendered extra borders and misplaced content for Word documents. (FB7899)
- Addressed an issue in Excel Classic HTML mode where an extremely large number of columns cause a sheet not to render. (FB7557)
- C header has been renamed to PerceptiveDocumentFilters.h
- C++ exceptions are now derived from std::exception
- C++ FileReaders global factory object has been renamed to DocumentFilters
- C++ HD samples (to images, to PDF, to Structured XML, etc) use auto pointers to simplify managing object lifetimes and exception handling
- C++ license key include file renamed to PerceptiveDocumentFiltersLicense.inc
- C++ namespace has been renamed from ISYS to Perceptive
- C++ objects implementation & header files renamed to PerceptiveDocumentFiltersObjects.cpp and PerceptiveDocumentFiltersObjects.h
- Enhanced the processing of headers and footers in Excel documents. (FB7578)
- Expanded the error information reported in the Document Filters API. (FB7859)
- Extended the SHOWHIDDEN option for the RTF file type in both Text and HD modes. (FB8124)
- Fixed an issue in Classic HTML mode for RTF documents where German umlaut letters caused the output to split incorrectly. (FB7779)
- Fixed an issue in Word where bullets and indentation would not render correctly in Classic HTML and HD HTML. (FB7630)
- Fixed an issue when processing MBOX/Sendmail files containing different date encodings. (FB7925)
- Fixed an issue where certain WPD file could cause a hang. (FB8215)
- Fixed an issue where the Unicode apostrophe character in ODT is converted to white space on a 64-bit Solaris Sparc platform. (FB8197)
- Improved handling of embedded document parts in OpenOffice. (FB5716)
- Improved handling of Excel document that include different text encodings. (FB7970)
- Improved handling of Word documents that have partial or missing style information. (FB7773)
- Improved metadata date normalization in DOC / DOCX / PDF (FB6330)
- Improved scaling handling on Excel documents. (FB7633)
- Improved the auto text color handing for Word documents. (FB7756)

- Improved the border logic for Word documents in Classic HTML mode. (FB7760)
- Improved the detection of temporary environment variables in HD output modes. (FB8216)
- Improved the font mapping for Classic HTML and HD modes. (FB7627)
- Improved the handling of auto-sized text boxes in PowerPoint documents. (FB7696)
- Improved the handling of cell formatting in Excel documents. (FB7638)
- Improved the handling of cell spacing in Excel documents. (FB7636)
- Improved the handling of corruptProject files. (FB7784)
- Improved the handling of CRLF in Excel cell data. (FB7962)
- Improved the handling of embedded clip-art in PowerPoint files. (FB7817)
- Improved the handling of embedded files within RTF documents. (FB7551)
- Improved the handling of Footnotes for Open Office documents. (FB7988)
- Improved the handling of grid lines in Excel documents. (FB7566)
- Improved the handling of password protected ZIP and RAR files. (FB7878)
- Improved the handling of Shift-JIS in HTML documents. (FB7879)
- Improved the handling of Slide Numbers in PowerPoint files. (FB7819)
- Improved the handling of small text files. (FB7930)
- Improved the handling of Table of Contents in Open Office documents. (FB7990)
- Improved the handling of text-alignment in PowerPoint documents. (FB7818)
- Improved the handling of VCF files to include missing fields. (FB7729)
- Improved the line-break logic for Open Office documents. (FB7992)
- Improved the processing of dBASE files when processing tables with deleted rows. (FB7858)
- Improved the space handling for Word Perfect documents. (7943)
- Improved the support for OLM files. (FB7926)
- Improved the text spacing for some bulleted lists and right-side alignment of justified text in HD output for Word documents. (FB8236)
- Java global factory object has been renamed to DocumentFilters
- Java package renamed to com.perceptive.documentfilters
- Mac OSX binaries now load dependencies from @loader_path, which enables embedding in packages. (FB7794)
- Reduced memory usage when processing large UTF-8 or Unicode text files. (FB7839)
- Reduced the memory utilization while process Excel documents in Classic HTML mode. (FB7645)
- Removed the requirement for Administration rights when installing Document Filters. (FB7921)

- Resolved an issue in Classic HTML where attempting to open an XLS file would give a "17" error. (FB8219)
- Resolved an issue in which corrupt or unknown document formats would cause an error when scanning Word documents using Classic HTML mode. (FB8147)
- Resolved an issue in Word where the spacing between a bullet and text was too large in Classic HTML and HD. (FB7625)
- Resolved an issue where some AVI files could cause a crash. (FB7891)
- Resolved an issue when processing corrupt compound-files on non-Windows platforms. (FB7738)
- Resolved an issue where cell background colors would display incorrectly in Excel documents for Classic HTML mode. (FB8173)
- Resolved an issue where date/time formatting would produce truncated results for Excel documents. (FB7993)
- Resolved an issue where dates for files within RAR may be incorrectly reported. (FB7972)
- Resolved an issue where extraneous underlining and bold text would be present in Word documents in Classic HTML. (FB7761)
- Resolved an issue where files were misidentified as Framework3. (FB7758)
- Resolved an issue where HTML was generated with incorrect decimal place symbol, when in Dutch locale. (FB7714)
- Resolved an issue where items within RAR files may be reported as 0 size. (FB7929)
- Resolved an issue where line numbers displayed incorrectly in Word documents using HD mode. (FB8235)
- Resolved an issue where OLE objects were not extracted from HWP files. (FB8176)
- Resolved an issue where shape colors were missing in Word documents using Classic HTML and HD modes. (FB7928)
- Resolved an issue where some calculated numbers in Excel showed as 0 on non-Windows platforms. (FB7967)
- Resolved an issue where some Excel documents would fail to process. (FB7940)
- Resolved an issue where some files were not extracted from CHM files. (FB7868)
- Resolved an issue where Table of Contents may appear twice for Open Office documents. (FB7989)
- Resolved an issue where text extraction of large numbers from Excel on a Linux C platform would cause the incorrect output. (FB8208)
- Resolved an issue where text files were identified incorrectly between 32-bit and 64-bit Windows. (FB7743)
- Resolved an issue where the modified date in a RAR file was off by one month. (FB8096)
- Resolved an issue with Classic HTML where OLE objects may report the processing of subfiles even though the Supports Subfiles parameter is set to False. (FB8230)
- Resolved inconsistency between 32-bit and 64-bit Windows when processing MBOX files. (FB7800)
- Resolved issue where some formula results may not be rendered in XLSX documents. (FB7639)

- Resolved issues with bad characters and output differences between Linux 32-bit and Linux 64-bit platforms for Word documents in Classic HTML. (FB7814)
- Resolved RPM processing issues on different platforms. (FB7751)
- Resolved word and line spacing issues in Word documents in Classic HTML mode where the text was not aligned to corresponding text boxes. (FB7747)
- Resolves the issue of processing Word documents with missing style information in Classic HTML mode. (FB7850)

5.66.2 Updates

- Added external process isolation mode for use when processing large documents in Classic HTML mode. Enabled via option string: HDISOLATION=MEDIUM; HDISOLATION=HIGH. (FB7682)
- Added extra metadata to RTF documents. Improved handling of Arabic RTF documents. (FB7537)
- Added methods to calculate hashes on ISYS streams, functions implemented are: IGR_Calculate_MD5(), GR_Calculate_SHA1(), CExtractor::getHashMD5 CExtractor::getHashSHA1 (FB6635)
- Added support for rotated text in Word documents in Classic HTML mode. (FB6831)
- Added support for sub-enumeration of OneNote documents. (FB6879)
- Added the ability to enforce page margins for Word via Classic HTML mode. Enabled via option string: ADDPAGEMARGINS=YES ; ADDPAGEMARGINS=NO. (FB7490)
- Addressed a metadata order issue in Visio reader. (FB7584)
- Addressed an issue in the Word HTML reader where character padding may be different on different platforms. (FB7397)
- Addressed an issue of Hangul documents not being processed on HP-UX or AIX. (FB7586)
- Addressed an issue of large memory consumption when extracting some messages from PST on 64-bit non-Windows platforms. (FB7610)
- Addressed an issue of MSG in MSG in PST on non-Windows platforms. (FB6864)
- Addressed an issue of processing off-screen images in Excel documents in Classic HTML mode. (FB7612)
- Addressed an issue when GD warnings may be streamed to the console when processing some word documents. (FB7608)
- Addressed an issue when processing images what have negative margins in Word in Classic HTML mode. (FB7609)
- Addressed an issue where sub-enumerated messages from a SendMail file reported the incorrect date. (FB7400)
- Addressed an issue where the order of Project files was different from Window to non-Windows. (FB7582)
- Addressed inconsistent output of MP3 files on AIX. (FB7587)
- Addressed issue when some DWG documents may hang. (FB6982)
- Addressed the misidentification of uncompressed zip files. (FB7342)

- Archive processing has been moved into a separate module: 7z.dll, 7z.so, 7z.dylib. (FB7535)
- Extended PST reader to optionally extract embedded messages as EML instead of the default MSG. Enabled by passing "PSTMESSAGE TYPE=EML" as an option string. (FB6645)
- Extended the email headers that are emitted from a PST when EMAILMODE=AllHeaders is specified in the options string. (FB6937)
- Extended the Excel CSV reader to return all discovered metadata fields. (FB7289)
- Extended Word reader to extract the names of bookmarks, implemented for both Text and HTML modes. (FB6681)
- Extended Word, Excel and PowerPoint readers to extract both the text and URL of hyperlink items. Implemented in both Text and HTML modes. (FB6683)
- Extended Word, Excel and PowerPoint readers to extract values from Equations. Implemented for both Text and HTML modes. (FB6682)
- Implemented option string "PDFHIDETEXTLAYER=TRUE" that can be used to hide the OCR'd text of a document, but still presents the image view. (FB7310)
- Implemented option string "SHOWHIDDEN=HIDDEN" to have Office document formats output comments to the HTML as hidden <div>s. (FB7305)
- Implemented the ability to extract track-changes comments from Word documents. (FB6680)
- Improved handling of automatically generated emails (e.g. calendar acceptance) that may not have content. (FB7003)
- Improved handling of numbered paragraphs in Word documents. (FB6901)
- Improved image handling in PowerPoint reader that addresses some inconsistencies between platforms. (FB7398)
- Improved Linux distribution compatibility (e.g. SUSE 10) (FB7428)
- Improved the detection logic of DXF files that were previously identified as text. (FB7364)
- Improved the detection logic on EML when headers are incomplete. (FB7326)
- Improved the detection of CSV files that contain little or no white space characters. (FB6863)
- Improved the error checking when bad parameters are passed to the API. (FB7367)
- Improved the Excel HTML reader when processing fractions. (FB7365)
- Improved the font scaling of text boxes in Excel files. (FB7525)
- Improved the format detection logic of EML and MSG file types. (FB7280)
- Improved the handling non-message type MSG files. (FB7178)
- Improved the handling of auto size tables in PowerPoint 2007. (FB7304)
- Improved the handling of calendar MSG files. (FB7470)
- Improved the handling of column widths and alignment in Excel in Classic HTML mode. (FB7632)

- Improved the handling of contact MSG files. (FB7471)
- Improved the handling of corrupt Excel documents. (FB7437)
- Improved the handling of corrupt Excel files when processing in HTML or CSV mode. (FB7285)
- Improved the handling of corrupt PST files. (FB7372)
- Improved the handling of corrupt TIFF files. (FB7588)
- Improved the handling of corrupted RAR files. (FB6730)
- Improved the handling of DOCM files (Macro enabled Word documents). (FB7552)
- Improved the handling of embedded PDF documents in Office 2007 formats. (FB6731)
- Improved the handling of Excel documents that contain Chart worksheets. (FB7585)
- Improved the handling of Excel files saved on MacOS. (FB7079)
- Improved the handling of fields within the Word reader. (FB7370)
- Improved the handling of large HTML files. (FB7274)
- Improved the handling of metadata where content contains HTML tags. (FB6684)
- Improved the handling of MSG files that contain multiple "body" streams. (FB7179)
- Improved the handling of non-Latin characters in the Word 2007 reader. (FB7277)
- Improved the handling of numbering and Table of Contents in Word documents. (FB5972)
- Improved the handling of smart tag items in Word documents. (FB6883)
- Improved the handling of task MSG files. (FB7472)
- Improved the handling of vCard files. (FB7473)
- Improved the handling of Word documents that contain supplementary Unicode characters. (FB7452)
- Improved the identification of EML files when they do not contain recipient fields. (FB7451)
- Improved the memory utilization when processing large Excel files. (FB7504)
- Improved the performance of Excel reader when processing in Classic HTML mode. (FB6648)
- Improved the PST reader to capture emails that were previously not extracted. (FB7279)
- Improved the PST reader to capture emails that were previously not extracted. (FB7335)
- Improved the rendering of Autoshapes in Word documents. (FB7209)
- Improved the text alignment in PowerPoint 2007 in Classic HTML mode. (FB7553)
- Improved the Word Reader when processing form data in Classic HTML mode. (FB7298)
- Non-Windows binaries now include a version stamp, that can be displayed using: `strings -a "[Filename]" | grep -i "\$Revision"` (FB7453)
- Password protected ZIPs will now correctly return error code 5, instead of the generic error code 4. (FB6729)

- Resolved an issue when processing some non-Latin characters in the OneNote reader. (FB7583)
- Resolved an issue when sub-enumerating documents that have long filenames in Java. (FB7482)
- Resolved an issue where Shift-JIS text files were misidentified as XY Write. (FB6693)
- Resolved an issue where some binder files were not processed on non-Windows platforms. (FB7580)
- Resolved an issue where temporary files may be orphaned when processing some documents. (FB6624)
- The AVI reader was updated to insert paragraph breaks between each metadata item. (FB7189)
- The PowerPoint reader was extended to add option to show/hide hidden images from the output. Can be controlled by passing an option string in the form of: SHOWHIDDEN=OFF; SHOWHIDDEN=VISIBLE (FB7430)
- Updated the 7-zip reader to return dates in local time, to match other readers. (7351)
- Updated the Excel HTML reader to output slide numbers and dates in headers and footers. (FB6950)
- Updated the PowerPoint reader to output text in visual order, instead of logical/stored order. (FB7396)