

KeyView

Software Version 12.6

Viewing SDK Programming Guide



Document Release Date: June 2020
Software Release Date: June 2020

Legal notices

Copyright notice

© Copyright 2016-2020 Micro Focus or one of its affiliates.

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

Documentation updates

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

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

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

Support

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

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

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

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

Contents

Part I: Overview of Viewing SDK	13
Chapter 1: Introduction to Viewing SDK	14
Overview	14
Features	14
Viewing SDK and Visual Basic	15
Platforms, Compilers, and Dependencies	15
Supported Platforms	15
Supported Compilers	15
Software Dependencies	15
Windows Installation	16
Package Contents	17
License Information	17
Enable Advanced Document Readers	18
Update License Information	18
Directory Structure	19
Chapter 2: Getting Started	22
Before You Begin	22
View Initialization Information	22
Use an Initialization File	22
Viewing API	23
ActiveX Control	23
Use the Windows Registry File	23
Viewing API	24
ActiveX Control	24
Remove Functionality from an Application	25
Deploy Viewing API Applications	25
Deploy ActiveX Control Applications	26
Develop .NET Applications	26
Method and Property Naming Conventions	27
Sample Code	27
Deploy .NET Applications	28
Part II: Viewing API	30
Chapter 3: Use the Viewing API	31
Overview of the Viewing API	31
Create a Viewing API Window	32
Get the Viewer Window of the Document	33
Open and View a Document	33
Notification Messages	34
Save a Document	34

Convert a Document	35
Print a Document	35
Change the Print Job Name	36
Determine the Document Format	36
Extract Document Metadata	37
Change Document Options	37
Annotate, Highlight, or Index a Document	37
Draw a Page	38
Draw a Page into a Supplied Device Context	38
Edit a Document	38
Search for Text	38
Copy Text	39
Modify the Document View	39
Change the Layout of a Document	39
Change the Aspect Ratio of a Document	39
Invert, Rotate, or Magnify a Document	39
Display or Hide Gridlines in a Document	40
Play a Multimedia Document	40
Change the Current Object in a Document	41
View Deleted Items and Document Revision Marks	41
View Container Files	42
Microsoft Outlook Personal Folders (PST) Files	42
Choose the Reader to use for PST Files	43
Lotus Notes Database (NSF)	44
System Requirements	44
Installation and Configuration	45
Format Notes	45
View Mail Messages and Mail Stores	45
View Archive Files	47
Extract Subfiles to a Viewing Window or Disk	48
Display Subfiles in the Preview Pane	48
Set a Password for a Container File	49
View PDF Documents	49
Use the Acrobat ActiveX Control	49
Use the Microsoft WebBrowser ActiveX Control	49
Use a Graphic-Based PDF Reader	50
Use the kppdfldr Reader	50
Use the kppdf2ldr Reader	51
Specify the Graphic-based Reader	51
View or Extract PDF Portfolio Documents	52
View Microsoft Visio Files	53
Extract Microsoft Excel Formulas	53
Chapter 4: Viewing API Sample Programs	56
Overview	56
Compile the Sample Programs	56
Run the Sample Programs	56

Viewing SDK Initialization Information	56
hellovapi	57
Load kvvapi.dll	57
Create the VAPI Window	57
Open a Document	58
hellovapi.c	59
hellovapi.h	64
hellovapi.rc	64
vapidemo	64
mfckv	64
rfdemo	65
prntdemo	65
filetype	65
ihademo	65
drawdemo	66
uzipdemo	66
Chapter 5: Message Parameters	67
VAPIM_ANNOTATE	69
VAPIM_ENABLEINDEX	70
VAPIM_GETNEXTTEXTBUFFER	72
VAPIM_GETPAGEFROMLOGICAL	72
VAPIM_GETSUMMARYINFO	73
VAPIM_GETTEXT	74
VAPIM_GOTO_PAGE	74
VAPIM_HAVEHILITE	75
VAPIM_POSITION	75
VAPIM_POSITIONHILITE	76
VAPIM_SETCURSOR	77
VAPIM_SETHILITE	77
VAPIM_SETHILITEOPTIONS	78
VAPIM_SETINDEXBUFCHARSET	79
VAPIM_SHOWHITS	79
VAPIM_CONVERT	80
VAPIMWP_CANCONVERT	81
VAPIMWP_DRAW_DRAWPAGE	81
VAPIMWP_DRAW_DRAWTOFILE	83
VAPIMWP_DRAW_GETPAGECOUNT	84
VAPIMWP_DRAW_GETPAGESIZE	84
VAPIMWP_DRAW_GETWORKBOOKPAGECOUNT	85
VAPIMWP_DRAW_INIT	86
VAPIMWP_DRAW_SHUTDOWN	87
VAPIMWP_EDIT_CANCOPY	87
VAPIMWP_EDIT_CANFIND	88
VAPIMWP_EDIT_CANSELECTALL	89
VAPIMWP_EDIT_COPY	90
VAPIMWP_EDIT_FIND	90

VAPIMWP_EDIT_FIND_UNICODE	91
VAPIMWP_EDIT_GETFINDTEXT	91
VAPIMWP_EDIT_SELECTALL	92
VAPIMWP_FILE_CANSAVEAS	93
VAPIMWP_FILE_CANUNZIP	94
VAPIMWP_FILE_CLOSE	94
VAPIMWP_FILE_SAVEAS	95
VAPIMWP_FILE_UNZIP	95
VAPIMWP_INIT_GETCHARSET	96
VAPIMWP_INIT_GETDESCRIP	97
VAPIMWP_INIT_GETDOCCLASS	97
VAPIMWP_INIT_GETDOCFORMAT	98
VAPIMWP_INIT_GETFILENAME	99
VAPIMWP_INIT_GETHWNDVIEWER	100
VAPIMWP_INIT_JUMPTOFIRSTHILIITE	100
VAPIMWP_INIT_OPEN_DOCUMENT	101
VAPIMWP_INIT_SETPASSWORD	103
VAPIMWP_INIT_SETSRCCHARSET	104
VAPIMWP_INIT_SETTRGCHARSET	104
VAPIMWP_MULTIOBJ_CANMULTIOBJ	105
VAPIMWP_MULTIOBJ_CANNEXTOBJ	106
VAPIMWP_MULTIOBJ_CANPREVOBJ	107
VAPIMWP_MULTIOBJ_CANSETCURRENTOBJ	107
VAPIMWP_MULTIOBJ_GETOBJCOUNT	108
VAPIMWP_MULTIOBJ_NEXTOBJ	109
VAPIMWP_MULTIOBJ_OBJNAME	109
VAPIMWP_MULTIOBJ_PREVOBJ	110
VAPIMWP_MULTIOBJ_SETCURRENTOBJ	111
VAPIMWP_OPTIONS_GETOPTIONS_EX	111
VAPIMWP_OPTIONS_SETOPTIONS_EX	112
VAPIMWP_PRINT_ANNOTATIONS	113
VAPIMWP_PRINT_CANPRINT	114
VAPIMWP_PRINT_PAGESETUP	114
VAPIMWP_PRINT_PRINT	115
VAPIMWP_PRINT_PRINTHEADER	116
VAPIMWP_PRINT_PRINTSETUP	116
VAPIMWP_PRINT_PRINTTOPD	117
VAPIMWP_PRINT_PRINTTOPRINTER	117
VAPIMWP_PRINT_SETPRINTNAME	118
VAPIMWP_VIEW_CANASPECTRATIO	119
VAPIMWP_VIEW_CANDECREASEFONT	120
VAPIMWP_VIEW_CANFITWINDOW	120
VAPIMWP_VIEW_CANGOTO	121
VAPIMWP_VIEW_CANGRIDLINES	122
VAPIMWP_VIEW_CANINCREASEFONT	122
VAPIMWP_VIEW_CANINVERT	123

VAPIMWP_VIEW_CANLAYOUT	124
VAPIMWP_VIEW_CANMAGNIFY	125
VAPIMWP_VIEW_CANPAUSE	125
VAPIMWP_VIEW_CANPLAY	126
VAPIMWP_VIEW_CANPREVIEWPANE	127
VAPIMWP_VIEW_CANROTATE	128
VAPIMWP_VIEW_CANSTOP	128
VAPIMWP_VIEW_DECREASEFONT	129
VAPIMWP_VIEW_END	130
VAPIMWP_VIEW_GETASPECTRATIO	130
VAPIMWP_VIEW_GETGRIDLINES	131
VAPIMWP_VIEW_GETINVERT	132
VAPIMWP_VIEW_GETLAYOUT	133
VAPIMWP_VIEW_GETMAGNIFY	133
VAPIMWP_VIEW_GETPLAYMODE	134
VAPIMWP_VIEW_GETPREVIEWPANE	135
VAPIMWP_VIEW_GETROTATE	136
VAPIMWP_VIEW_GOTOPAGE	136
VAPIMWP_VIEW_INCREASEFONT	137
VAPIMWP_VIEW_LOOP	138
VAPIMWP_VIEW_PAUSE	138
VAPIMWP_VIEW_PLAY	139
VAPIMWP_VIEW_SETASPECTRATIO	139
VAPIMWP_VIEW_SETGRIDLINES	140
VAPIMWP_VIEW_SETINVERT	141
VAPIMWP_VIEW_SETLAYOUT	141
VAPIMWP_VIEW_SETMAGNIFY	142
VAPIMWP_VIEW_SETPREVIEWPANE	143
VAPIMWP_VIEW_SETROTATE	144
VAPIMWP_VIEW_STOP	144
Chapter 6: Notification Message Parameters	146
VAPINM_ANNOTATION_HIT	146
VAPINM_EXTENT	147
VAPINM_SELECTION	148
VAPINM_TEXTBUFFER	148
VAPINM_USERCLICK	150
VAPINM_VIEW_FILE	150
VAPINMWP_INIT_DISABLEUI	151
VAPINMWP_INIT_DOCTYPE	152
VAPINMWP_INIT_GETTEMPFILEPATH	152
VAPINMWP_INIT_OPENDOCDONE	153
VAPINMWP_INIT_PAGENUMBER	154
VAPINMWP_MULTIOBJ_OBJNAME	154
VAPINMWP_OPTIONS_GETOPTIONS_EX	155
VAPINMWP_PRINT_PRINTDONE	156
Chapter 7: Structures	157

ADDOCINFO	157
ALL_OPTIONS_EX	158
KPTPIOobj	159
KVSumInfoElemEx	160
KVSummaryInfoEx	160
TPVAPIAnnotation	161
TPVAPIConvert	162
TPVAPICreateParams	163
TPVAPIDrawFileInfo	164
TPVAPIDrawPageInfo	165
TPVAPIExtract	166
TPVAPIFindInfo	167
TPVAPIFirstLast	168
TPVAPIGetText	168
TPVAPIHiLiteColor	169
TPVAPIHiLiteOptions	170
TPVAPIOpenDocumentInfo	170
TPVAPIPageSize	174
TPVAPIPosition	175
TPVAPITextInfo	176
Part III: Viewing ActiveX Control	177
Chapter 8: Use the Viewing ActiveX Control	178
Overview of the Viewing ActiveX Control	178
Open and View a Document	179
Save a Document	179
Convert a Document	180
Print a Document	180
Determine the Document Format	181
Extract Document Metadata	181
Search for Text in a Document	181
Copy a Selected Area of Text	182
Copy all the Text in a Document	182
Create a Thumbnail Image of a Document Page	182
Filter a Document	183
Highlight Text in a Document	183
Annotate Text in a Document	183
Chapter 9: Control Sample Programs	184
Viewing SDK Initialization Information	184
fileview	184
Create a New Visual Basic Project 6.0	184
Draw the Controls	185
Set Objects and Properties	185
Create Event Procedures	186
dotnetview	187

Chapter 10: Control Methods	188
Annotate	189
ChangeObject	190
Close	190
Convert	191
Copy	192
DecreaseFont	192
DrawToFile	193
Find	194
GetNextTextBuffer	195
GetPageFromLogical	195
GetSelectedText	196
GetSummaryInfo	196
GetText	197
GoToPage	198
IncreaseFont	199
Open	199
Play	200
Position	201
PositionHiLite	201
PrintDlg	202
PrintOut	202
PrintOutEx	203
PrintPageSetup	204
SaveAs	204
SelectAll	205
SetCursor	205
SetFocusViewer	206
SetHiLite	206
SetHiLiteOptions	207
SetPassword	208
SetPrintName	208
ShowHits	209
UnZip	209
UnZipEx	210
Chapter 11: Control Properties	212
Introduction	214
Persistent Properties	214
Property Naming Conventions in .NET	214
"OPEN" Properties	214
ASCIICharSet	214
ASCIIFilterNonPrintable	215
ASCIIFontName	215
ASCIIFontSize	215
ASCIIFontStyle	216
ASCIIMarginBottom	216

ASCIIMarginLeft	216
ASCIIMarginRight	216
ASCIIMarginTop	217
ASCIIPrintLandscape	217
AspectRatio	217
CanCopy	218
CanDecreaseFont	218
CanFind	218
CanIncreaseFont	219
CanMultiObj	219
CanNextObj	219
CanPause	220
CanPlay	220
CanPrevObj	220
CanPrint	221
CanSaveAs	221
CanSelectAll	221
CanStop	222
CanUnZip	222
CanViewPane	222
CharSet	222
ContextMenu	223
DocumentClass	223
DocumentFormat	224
DocumentType	224
DrawPageCount	224
DrawPageHeight	225
DrawPageWidth	225
DrawWorkBookPageCount	225
FileName	226
HiLiteBackground	226
HiLiteForeground	226
HotKeys	226
ImageCustomSize	227
ImagePrintHorzAlign	227
ImagePrintMode	227
ImagePrintPercent	228
ImagePrintVertAlign	228
ImageScaling	228
IndexBufCharSet	229
Invert	229
JumpToFirstHiLite	229
MMPlayOption	229
MMScaleMovie	230
ObjName	230
OPENDisableUI	230

OPENHighLight	231
OPENMode	231
OPENWaitOnOpen	232
PrintAnnotations	232
PrintHeaders	232
RegIniMode	233
RegIniName	233
Rotate	233
SrcCharSet	234
SSDisplayGrid	234
SSDisplayHeaders	235
SSViewObjects	235
TrgCharSet	235
ViewPane	235
WPCustomSize	236
WPDisplayPict	236
WPPageLayout	236
WPScaleTable	237
WPViewMode	237
Chapter 12: Control Events	238
Annotation	238
KeyDown	238
MouseUp	239
OpenDocDone	239
PageNumber	240
PrintDone	240
PrintDoneEx	241
Selection	241
TextBuffer	241
UserClick	242
ViewExtent	242
ViewFile	243
Part IV: Appendixes	244
Appendix A: Supported Formats	245
Key to Supported Formats Table	245
Supported Formats	246
Appendix B: Document Readers	309
Key to Document Readers Table	309
Document Readers	311
Appendix C: Character Sets	338
Multibyte and Bidirectional Support	338
Coded Character Sets	346
Appendix D: File Format Detection	352

Introduction	352
Extract Format Information	352
Determine Format Support	352
Translate Format Information	353
Distinguish Between Formats	354
Determine a Document Reader	354
Category Values in the Initialization File and Registry	354
Appendix E: Extract and Format Lotus Notes Subfiles	358
Overview	358
Customize XML Templates	358
Use Demo Templates	359
Use Old Templates	359
Disable XML Templates	359
Template Elements and Attributes	360
Conditional Elements	360
Control Elements	361
Data Elements	362
Date and Time Formats	365
Lotus Notes Date and Time Formats	365
KeyView Date and Time Formats	366
Appendix F: List of Files Required for Redistribution	371
Core Files	371
Support Files	372
Document Readers and Writers	373
Archive Formats	373
Binary Formats	375
Computer-Aided Design Formats	375
Database Formats	375
Desktop Publishing Formats	375
Display Formats	375
Graphic Formats	376
Mail Formats	377
Presentation Formats	378
Spreadsheet Formats	379
Word Processor Formats	379
Miscellaneous Functionality	381
Viewing ActiveX Control	382
Appendix G: Configuration Options in kvsdk.ini	383
kvsdk.ini Options	383
Appendix H: Password Protected Files	385
Supported Password Protected File Types	385
View Password Protected Files	386
Send documentation feedback	387

Part I: Overview of Viewing SDK

This section provides a general overview of Micro Focus KeyView Viewing SDK and a description of the sample programs, and includes the following chapters:

- [Introduction to Viewing SDK](#)
- [Getting Started](#)

Chapter 1: Introduction to Viewing SDK

This guide is for developers who incorporate Micro Focus KeyView Viewing SDK components into their own applications. It is intended for readers who are familiar with Windows programming.

• Overview	14
• Features	14
• Viewing SDK and Visual Basic	15
• Platforms, Compilers, and Dependencies	15
• Windows Installation	16
• Package Contents	17
• License Information	17
• Directory Structure	19

Overview

The Viewing SDK is part of the Micro Focus KeyView suite of products. KeyView provides high-speed text extraction, conversion to web-ready HTML and well-formed XML, and high-fidelity document viewing.

The Viewing SDK enables you to build high-fidelity document viewing capabilities into your own applications. You can incorporate Viewing technology into your document management, web server, Internet or Intranet, groupware, information retrieval, email, or imaging applications. It enables your users to open, view, and print virtually any document, spreadsheet, presentation, graphic, or compression file, without having the native application or plug-in available.

Viewing SDK uses a standard Windows interface which integrates effectively using popular languages such as C++ (including Microsoft Foundation Classes), J#, and Visual Basic.

The SDK includes the following components:

- Viewing API (VAPI) – Windows messaging-based API
- Viewing ActiveX control (OCX) and .NET interface
- Sample programs

Features

With Viewing SDK, you can create an application by using the Viewing API or the Viewing ActiveX control to:

- View and print documents.
- Convert popular word processing and spreadsheet formats to text, Microsoft Rich Text Format (RTF), and HTML.

- Annotate, highlight, and filter documents.
- Generate thumbnail views of documents.
- Automatically recognize document types.

Viewing SDK and Visual Basic

The Viewing ActiveX control is ideally suited for developing Visual Basic® applications with viewing, conversion, and printing capabilities. The control provides most of the functionality of the Viewing Windows messaging API, but in the form of an ActiveX control that can be dropped onto your Visual Basic form.

Platforms, Compilers, and Dependencies

This section lists the supported platforms, supported compilers, and software dependencies for the KeyView software.

Supported Platforms

- Microsoft Windows Server 2019 x64
- Microsoft Windows Server 2016 x64
- Microsoft Windows Server 2012 x64
- Microsoft Windows 10 x64
- Microsoft Windows 8 x86 and x64
- Microsoft Windows 7 x86 and x64

Supported Compilers

Microsoft 32-bit C/C++ Optimizing Compiler Version 12.00.8804 for 80x86

Software Dependencies

Some KeyView components require that you have installed specific third-party software:

- Outlook 2002 or later is required to process Microsoft Outlook Personal Folders (PST) files using the MAPI-based reader (`psts.r`). The native PST reader (`pstxsr`) does not require Outlook.

NOTE: You must install an edition of Microsoft Outlook (32-bit or 64-bit) that matches the KeyView software. For example, if you use 32-bit KeyView, install 32-bit Outlook. If you use 64-bit KeyView, install 64-bit Outlook.

If the editions do not match, KeyView returns Error 32: KVErrror_PSTAccessFailed and an error message from Microsoft Office Outlook is displayed: Either there is a no default mail client or the current mail client cannot fulfill the messaging request. Please run Microsoft Outlook and set it as the default mail client.

- Lotus Notes or Lotus Domino (the minimum requirement is 6.5.1, but version 8.5 is recommended)—for Lotus Notes database (NSF) file viewing.
- Microsoft .NET Framework Version 2.0 Redistributable Package (if programming in .NET environment).
- Microsoft Visual J# .NET Version 2.0 Redistributable Package (if developing J# program in .NET environment).

Windows Installation

To install the SDK on Windows, use the following procedure.

To install the SDK

1. Run the installation program, `KeyViewProductNameSDK_VersionNumber_OS.exe`, where *ProductName* is the name of the product, *VersionNumber* is the product version number, and *OS* is the operating system.

For example:

`KeyViewViewingSDK_12.6_Windows_X86_64.exe`


The installation wizard opens.

2. Read the instructions and click **Next**.

The License Agreement page opens.

3. Read the agreement. If you agree to the terms, click **I accept the agreement**, and then click **Next**.

The Installation Directory page opens.

4. Select the directory in which to install the SDK. To specify a directory other than the default, click , and then specify another directory. After choosing where to install the SDK, click **Next**.

The License Key page opens.

5. Type the company name and license key that were provided when you purchased KeyView, and then click **Next**.
 - The company name is case sensitive.
 - The license key is a string that contains 31 characters.

NOTE: The installation program validates the company name and license key and generates

the file `install\OS\bin\kv.lic` (where `install` is your chosen installation folder and `OS` is the name of the operating system platform). The license information is validated when the KeyView API is used. If you do not enter a license key at this step, or if you enter invalid information, the KeyView SDK is installed, but the API does not function. When you obtain a valid license key, you can either re-install the KeyView SDK, or manually update the license key file (`kv.lic`) with the new information. For more information, see [License Information, below](#).

The Pre-Installation Summary dialog box opens.

6. Review the settings, and then click **Next**.

The SDK is installed.

7. Click **Finish**.

Package Contents

The Viewing SDK installation contains:

- Dynamic Link Library files and executable files necessary for viewing text from a wide variety of formats.
- Several sample programs that demonstrate Viewing SDK functionality. See [Viewing API Sample Programs, on page 56](#) and [Control Sample Programs, on page 184](#).
- The following files define the functions and structures used by your application to establish an interface with Viewing SDK:

<code>adAPI.h</code>	<code>kvvapi.h</code>
<code>adinfo.h</code>	<code>kwautdef.h</code>
<code>kv10obj.h</code>	<code>kwcmfio.h</code>
<code>kvcharset.h</code>	<code>kwcnmgr.h</code>
<code>kverrorcodes.h</code>	<code>kwkpif.h</code>
<code>kvoem.h</code>	<code>kwoption.h</code>
<code>kvtypes.h</code>	<code>language.h</code>

License Information

During installation, the installation program generates the `install/OS/bin/kv.lic` file, where `install` is the directory in which you installed KeyView, and `OS` is the operating system. This file is opened and validated when the KeyView API is used.

TIP: Where the API allows, Micro Focus recommends that you provide the license by using the API (`fpInitWithLicenseData()`), rather than using the license file.

The license key controls whether the following are enabled:

- the full version of the KeyView SDK
- the trial version of the KeyView SDK
- language detection and advanced document readers—The following components are considered advanced features, and are licensed separately:
 - Microsoft Outlook Personal Folders (PST) readers (`pstsr`, `pstnsr`, and `pstxsr`)
 - Lotus Notes database (NSF) reader (`nsfsr`)
 - Mailbox (MBX) reader (`mbxsr`)
 - Character set detection library (`kvlangdetect`)

If you change the license key at any time, you must update the licensing information. See [Update License Information](#).

Enable Advanced Document Readers

To enable advanced readers in one of the KeyView SDKs, you must obtain an appropriate license key from Micro Focus and update the installed license key with the new information as described in [Update License Information](#).

If you are enabling the MBX reader in an existing installation of Viewing SDK, in addition to updating the license key, you must also follow these steps:

If you are using the registry file:

1. Open the `install.reg.txt` in a text editor. The file is installed in the `install\redist` directory, where `install` is the directory in which you installed Viewing SDK.
2. Under the key `[HKEY_LOCAL_MACHINE\Software\Verity\Viewing SDK\KVMAILVE]`, change the parameter `"208=emlsr.dll"` to `"208=mbxsr.dll"`.
3. Save the file as `install.reg` and import the file into your Windows system registry.

If you are using the `kvsdk.ini` file:

1. Open the `kvsdk.ini` file with a text editor. The file is installed in the root of the Windows directory.
2. In the `[KVMAILVE]` section of the `kvsdk.ini` file, change the parameter `208=emlsr.dll` to `208=mbxsr.dll`.

Update License Information

If you currently have an evaluation version of KeyView and have purchased a full version of the SDK, or you are adding a document reader (for example, the PST reader), you must update the license information that was installed with the original version of the KeyView SDK.

If you installed a full version of KeyView, but did not enter licensing information at the time of installation, you must also update the license information.

To update the information, do one of the following:

- Pass the new license information to `fpInitWithLicenseData()`.
- Manually update the license information that is stored in the text file named `kv.lic`.
- Re-install the product and enter the new license information when prompted.

Where possible, Micro Focus recommends that you update the license information by using the API. This method allows you to update your license without including a license file in the software that you distribute. This method is available for the C APIs in the Filter SDK, HTML Export SDK, and XML Export SDK.

Alternatively, you can create a license file and include it in the bin folder with the KeyView DLLs. This file must be called `kv.lic`, and must have your organization name on the first line, and your license key on the second line, with no white-space surrounding either.

To update the license information in the API

- Pass the new license information to `fpInitWithLicenseData()`.

To update the kv.lic

1. Open the license key file, `kv.lic`, in a text editor. The file is in the `install\OS\bin` directory, where `install` is the directory in which you installed KeyView, and `OS` is the operating system. The file contains the following text:

```
COMPANY NAME  
XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
```

2. Replace the text `COMPANY NAME` with the company name that appears at the top of the License Key Sheet provided by Micro Focus. Enter the text exactly as it appears in the document.
3. Replace the characters `XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX` with the appropriate license key from the License Key Sheet provided by Micro Focus. The license key is listed in the **Key** column in the **Standalone Products** table. The key is a string that contains 31 characters, for example, `2TQD22D-2M6FV66-2KPF23S-2GEM5AB`. Enter the characters exactly as they appear in the document, including the dashes, but do not include a leading or trailing space.
4. The finished `kv.lic` file looks similar to the following:

```
Autonomy  
24QD22D-2M6FV66-2KPF23S-2G8M59B
```

5. Save the `kv.lic` file.

Directory Structure

Viewing SDK creates the following directory structure during installation. The variable `install` refers to the installation directory. By default, the installation directory is `C:\Program Files\Autonomy\KeyViewViewingSDK`.

The variable *OS* is the operating system for which the SDK is installed. For example, the *bin* directory on a standard 32-bit Windows installation would be located at *C:\Program Files\Autonomy\KeyViewViewingSDK\WINDOWS\bin*.

Viewing Installed Directory Structure

Directory	Description
<i>install</i> <i>\OS\bin</i>	Libraries, the <i>formats.ini</i> file, the <i>kv.lic</i> file, and a number of other supporting files.
<i>install\OS</i> <i>\bin\system</i>	Shared libraries used by Viewing SDK components.
<i>install</i> <i>\dotnetview</i>	A .NET workspace for Visual Studio. This is a J# sample program demonstrating basic Viewing functionality.
<i>install</i> <i>\drawdemo</i>	The Viewing API thumbnail sample program (draw into supplied DC).
<i>install</i> <i>\filetype</i>	The Viewing API sample program used to determine file type.
<i>install</i> <i>\fileview</i>	A Viewing OCX sample program.
<i>install</i> <i>\guide</i>	Contains the <i>KeyView Viewing SDK Programming Guide</i> in HTML and PDF format.
<i>install</i> <i>\hellovapi</i>	Sample code for a simple program that demonstrates how to use the Viewing API to display documents in a window. Micro Focus recommends that you review this sample first.
<i>install</i> <i>\ihademo</i>	A Viewing API sample program featuring indexing (filtering), highlighting, and annotating.
<i>install</i> <i>\include</i>	The header files required for Viewing SDK.
<i>install</i> <i>\mfckv</i>	A simple MFC (Microsoft Foundation Class) SDI application using Viewing API.
<i>install</i> <i>\prntdemo</i>	A sample program that uses the Viewing API to print documents.
<i>install</i> <i>\redist</i>	Contains the <i>install.reg</i> file, which contains initialization information used by Viewing SDK. See View Initialization Information, on page 22 .
<i>install</i> <i>\rel_notes</i>	Contains the <i>KeyViewViewing SDK Release Notes</i> in HTML and PDF format.
<i>install</i> <i>\rtfdemo</i>	A sample program that demonstrates the use of the Viewing API to convert documents to RTF.

Viewing Installed Directory Structure, continued

Directory	Description
<i>install</i> \uzipdemo	A sample program for unzipping source files to a selected directory.
<i>install</i> \vapidemo	A sample program that demonstrates most of the Viewing API functionality.
Windows system directory	Contains the <code>kvsdk.ini</code> file which contains initialization information used by Viewing SDK. See View Initialization Information, on page 22 .

Chapter 2: Getting Started

This section provides information on developing and deploying Viewing applications. It includes the following topics:

- [Before You Begin](#) 22
- [View Initialization Information](#) 22
- [Deploy Viewing API Applications](#) 25
- [Deploy ActiveX Control Applications](#) 26
- [Develop .NET Applications](#) 26

Before You Begin

Before you use Viewing SDK to build your own programs, review and run the sample programs provided with the product. Micro Focus recommends that you review the `hellovapi` sample program first. It is a simple program that demonstrates how to use the Viewing API to display documents within your application.

For information on the sample programs, see [Viewing API Sample Programs, on page 56](#) and [Control Sample Programs, on page 184](#).

View Initialization Information

Viewing uses initialization information for its internal operations, for example, to determine which components to load. You can store this information either in an initialization file or in the Windows registry.

The initialization file is called `kvsdk.ini` and is stored in the Windows system directory.

The file used to define registry settings is called `install.reg.txt` and is stored in the `install\redist` directory, where `install` is the directory in which you installed Viewing SDK.

You must customize the information in one of these files and specify in your application where the information is located.

Use an Initialization File

If you are using the initialization file (`kvsdk.ini`) to set initialization information, you must modify the file to reflect your company name and application name. The sample programs demonstrate how to use an initialization file.

NOTE: A copy of the original `kvsdk.ini` file (`install.ini`) is stored in the `install\redist` directory, where `install` is the directory in which you installed Viewing SDK. This file is not required for redistribution and is for reference only.

Viewing API

To specify an initialization file using the Viewing API

1. Create the [TPVAPICreateParams](#) structure. Set `uProfileType` to `PROFILEDF_USE_INI`, and `lpszIniFileName` to the location of the initialization file.

For example:

```
memset (&CreateParams, 0, sizeof(TPVAPICreateParams));
if (bUseIni)
{
    CreateParams.uProfileType    = PROFILEDF_USE_INI;
    CreateParams.lpszIniFileName = szIniFileName;
}
else
{
    CreateParams.uProfileType    = PROFILEDF_USE_REG;
    CreateParams.lpszRegistryName = REGISTRY_NAME_ASCII;
}
```

2. Create the VAPI window by using the standard Windows API functions `CreateWindow()` or `CreateWindowEx()`.

ActiveX Control

To specify an initialization file by using the ActiveX control

1. Set the [RegIniMode](#) property to 1.
2. Set the [RegIniName](#) property to the path and name of the initialization file. For example, `kvsdk.ini` or `c:\myprogram\myini.ini`. By default, Viewing looks for the initialization file in the Windows system directory.

For example:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    AxKEYview1.RegIniMode = 1
    AxKEYview1.RegIniName = "c:\windows\kvsdk.ini"
    AxKEYview1.Open("c:\test.doc")
End Sub
```

Use the Windows Registry File

If you are using the Windows registry to set initialization information, you must modify the registry file (`install.reg.txt`) to reflect your company name and application name. The file has a `.txt` extension for easy editing and viewing. After you have finished editing the file, remove the `.txt` extension. When your application is installed, import the `install.reg` file into the Windows Registry.

Viewing API

To specify the Windows registry by using the Viewing API

1. Create the [TPVAPICreateParams](#) structure. Set `uProfileType` to `PROFILEDF_USE_REG`, and `lpszRegistryName` to the location of the initialization file.

For example:

```
memset (&CreateParams, 0, sizeof(TPVAPICreateParams));
if (bUseRegistry)
{
    CreateParams.uProfileType    = PROFILEDF_USE_REG;
    CreateParams.lpszRegistryName = REGISTRY_NAME_ASCII;
}
else
{
    CreateParams.uProfileType    = PROFILEDF_USE_INI;
    CreateParams.lpszIniFileName = szIniFileName;
}
```

2. Create the VAPI window by using the standard Windows API functions `CreateWindow()` or `CreateWindowEx()`.

ActiveX Control

To specify the Windows registry by using the ActiveX control

1. Set the [RegIniMode](#) property to 2.
2. Set the [RegIniName](#) property to the registry key under `HKEY_LOCAL_MACHINE\Software` where the Viewing initialization information resides. For example, *YourCompany\YourProduct*.

For example:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    AxKEYview1.RegIniMode = 2
    AxKEYview1.RegIniName = "Autonomy\keyview"
    AxKEYview1.Open("c:\test.doc")
End Sub
```


Remove Functionality from an Application

To remove Viewing functionality from your application

1. Do not redistribute the Dynamic Link Library (DLL) associated with the component you want to remove.
[List of Files Required for Redistribution, on page 371](#) lists the files that you can redistribute with your application. It also shows the Dynamic Link Library (DLL) associated with each component.
2. In the `kvsdk.ini` file or the `install.reg.txt` file, remove references to the component that you want to remove.

For example, to remove support for Windows Animated Cursor, remove the line that references "Windows Animated Cursor" from the registry or initialization file, and do not redistribute the Windows Animated Cursor reader (`kpanindr.dll`).

NOTE: If you remove Viewing functionality for a graphic format, and you view a document that has an embedded graphic of that format, the graphic is not displayed.

The following is a summary of files required based on functionality:

- **Copy to clipboard**—The following files are required for copy to clipboard functionality:
 - `rtfcnv.dll`, `txtcnv.dll` (for word processor formats)
 - `rtfss.dll` (for spreadsheet formats)
 - `kpifutil.dll` (for picture formats)
- **SaveAs to RTF**—The following files are required for SaveAs to RTF functionality:
 - `kvcnv.dll`
 - `rtfcnv.dll` (for word processor formats)
 - `rtfss.dll` (for spreadsheet formats)
 - `kpifutil.dll` (for picture formats)

Deploy Viewing API Applications

After you have built an application with the Viewing API, you must do the following:

1. Install all required files to the `\bin` directory of your application's installation directory. [List of Files Required for Redistribution, on page 371](#) lists the components that must be redistributed with your application. It also shows the Dynamic Link Library (DLL) associated with each component.
2. Review the `kvsdk.ini` file or the `install.reg.txt` file to make sure that the appropriate files are referenced.
3. Update the HOME entry in the initialization file or registry file with the complete path to where you are installing Viewing components.

4. Specify whether you are using an initialization file or registry settings. See [View Initialization Information, on page 22](#).
5. If you are using an initialization file, install the file to the location specified by `lpszIniFileName` when the application is installed. See [TPVAPICreateParams , on page 163](#).
6. If you are using the registry file, import the `install.reg` file into the Windows Registry when the application is installed.

Deploy ActiveX Control Applications

After you have built an application with Viewing ActiveX control, you must do the following:

1. Install all required files to the `\bin` directory of your application's installation directory. [List of Files Required for Redistribution, on page 371](#) lists the components that must be redistributed with your application. It also shows the Dynamic Link Library (DLL) associated with each component.
2. Review the `kvsdk.ini` file or the `install.reg.txt` file to make sure that the appropriate files are referenced.
3. Update the `HOME` entry in the initialization file or registry file with the complete path to where you are installing Viewing components.
4. Specify whether you are using an initialization file or registry settings. See [View Initialization Information, on page 22](#).
5. If you are using an initialization file, install the file to the location specified by `RegIniName` when the application is installed.
6. If you are using the registry file, import the `install.reg` file into the Windows Registry when the application is installed.
7. Install the Viewing ActiveX control (`kvocx.ocx`) to the `\bin` directory of your application's installation directory and register the control in the system registry by running the following command:

```
regsvr32 C:\MyApp\bin\kvocx.ocx
```

Add `-s` to suppress any dialog boxes when registering the OCX.

To unregister the ActiveX control, run the command:

```
regsvr32 -u C:\MyApp\kvocx.ocx
```

Develop .NET Applications

This section describes how to create and deploy a .NET application by using the KeyView ActiveX Control. Although you can develop .NET applications in many different development environments, the instructions in this section refer to Microsoft Visual Studio 2005.

To create and deploy a .NET application

1. Install the KeyView Viewing SDK.

The installation automatically registers the Viewing ActiveX control, "KeyView OLE Control module (v1.0)" and installs the COM dynamic library (kvocx.ocx) to the Viewing \bin directory. You can also use the `regsvr32` command to register the ActiveX COM module. For example:

```
regsvr32.exe install\bin\kvocx.ocx
```

2. In Visual Studio 2005, select **Tools** from the main menu, and click **Choose Toolbox Items....**
3. In the **Choose Toolbox Items** dialog box, click the **COM Components** tab.
4. From the list of available COM components, select the **KeyView Control** check box, and then click **OK**.

A Windows control named **KeyView Control** appears in the Toolbox. You can use this KeyView control in the same way as other controls in the Toolbox.

When the .NET application is built, Visual Studio creates the following dynamic libraries:

- Interop.KEYVIEWLib.dll
- AxInterop.KEYVIEWLib.dll

These libraries are wrappers for the KeyView ActiveX control, and are required to use the control in a .NET environment.

Method and Property Naming Conventions

The .NET control class name for KeyView ActiveX control is `AxKEYVIEWLib.AxKEYview`, where the namespace `AxKEYVIEWLib` is the library name.

In J#, C#, and C++, all ActiveX control method names in the .NET class are the same as their COM counterparts. However, individual properties in .NET are defined using get and set methods of the following format:

`get_property_name`

`set_property name`

For example, `RegIniName` in COM has `get_RegIniName` and `set_RegIniName` methods in the .NET class.

IMPORTANT: In a Visual Basic .NET application, all properties and methods are used in the same way as in a Visual Basic COM application.

Sample Code

The following code demonstrates how to use the .NET class in a J# Windows Form program:

```
private void button1_Click(Object sender, System.EventArgs e)
{
```

```
this.axKEYview1.set_RegIniMode((short)1);  
this.axKEYview1.set_RegIniName("c:\windows\kvsdk.ini");  
this.axKEYview1.Open("c:\test.doc");  
}
```

The following code demonstrates how to use the .NET class in a Visual Basic Windows Form program:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click  
    AxKEYview1.RegIniMode = 1  
    AxKEYview1.RegIniName = "c:\windows\kvsdk.ini"  
    AxKEYview1.Open("c:\test.doc")  
End Sub
```

Deploy .NET Applications

After you have built a .NET application using Viewing ActiveX control, follow these steps to deploy the application:

1. Install all required files to the `\bin` directory of your application's installation directory. [List of Files Required for Redistribution, on page 371](#) lists the components that must be redistributed with your application. It also shows the Dynamic Link Library (DLL) associated with each component.
2. Review the `kvsdk.ini` file or the `install.reg.txt` file to make sure that the appropriate files are referenced.
3. Update the HOME entry in the initialization file or registry file with the complete path to where you are installing Viewing components.
4. Specify whether you are using an initialization file or registry settings. See [View Initialization Information, on page 22](#).
5. If you are using an initialization file, install the file to the location specified by `RegIniName` when the application is installed. See [RegIniName, on page 233](#).
6. If you are using the registry file, import the `install.reg` file into the Windows Registry when the application is installed.
7. Install the Viewing ActiveX control (`kvocx.ocx`) to the `\bin` directory of your application's installation directory. Register the control in the system registry by running the following command:

```
regsvr32 C:\MyApp\bin\kvocx.ocx
```

Add `-s` to suppress any dialog boxes when registering the OCX.

8. Install the wrapper files `Interop.KEYVIEWLib.dll` and `AxInterop.KEYVIEWLib.dll` to the `\bin` directory of your application's installation directory.
9. Install the Microsoft .NET Framework Version 2.0 Redistributable Package, which is available at <http://msdn.microsoft.com/netframework/downloads/updates/default.aspx>.
10. If the application is developed using J#, install the Microsoft Visual J# .NET Version 2.0

Redistributable Package, which is available at

<http://msdn.microsoft.com/netframework/downloads/updates/default.aspx>.

Part II: Viewing API

This section describes the Viewing API and provides detailed reference information and includes the following chapters:

- [Use the Viewing API](#)
- [Viewing API Sample Programs](#)
- [Message Parameters](#)
- [Notification Message Parameters](#)
- [Structures](#)

Chapter 3: Use the Viewing API

This section describes how to use the Viewing API to perform some basic viewing tasks.

• Overview of the Viewing API	31
• Create a Viewing API Window	32
• Open and View a Document	33
• Save a Document	34
• Convert a Document	35
• Print a Document	35
• Change the Print Job Name	36
• Determine the Document Format	36
• Extract Document Metadata	37
• Change Document Options	37
• Annotate, Highlight, or Index a Document	37
• Draw a Page	38
• Edit a Document	38
• Modify the Document View	39
• Change the Current Object in a Document	41
• View Deleted Items and Document Revision Marks	41
• View Container Files	42
• View PDF Documents	49
• View or Extract PDF Portfolio Documents	52
• View Microsoft Visio Files	53
• Extract Microsoft Excel Formulas	53

Overview of the Viewing API

The Viewing API (VAPI) enables you to build a Windows program that uses Viewing components to manage many types of document, including word processing, spreadsheet, presentation, and graphics. See [Document Readers, on page 309](#) for more information on supported formats.

You can use the Viewing API to create an application to:

- Open and view a document.
- Draw a page of a word processing document, spreadsheet, or a picture into a supplied Device Context (HDC). This is useful for generating *thumbnail* views of documents.
- Print a document (including the ability to print a document without viewing it) to a specified printer or to the default printer.

- Allow viewed word processing and spreadsheet documents to be saved as RTF, HTML, or text. Also, you can save image formats to other supported image formats.
- Convert word processing and spreadsheet documents to text, RTF, or HTML without viewing them.
- View or extract subfiles from a container file, such as ZIP, TAR, or PST.
- View and manipulate a graphic (including rotate and magnify).
- Annotate documents with a bitmap or selected text. The Viewing API includes annotation event notification for actions such as clicking and double-clicking, allowing for implementation of hyperlink and pop-up text.
- Highlight all occurrences of a word in a document.
- Filter spreadsheets, presentation graphics, and documents to text. A cross-platform C API that provides text filtering is also available. Contact Micro Focus for information on KeyView Filter SDK.
- Determine a document's format based on its contents, not its file extension.
- Obtain document metadata, such as a document's author or title.

Create a Viewing API Window

You must create a new VAPI window for each document that you open; each VAPI window manages only *one* document at a time.

You can create multiple VAPI windows to handle multiple documents simultaneously. After you create a VAPI window, you can use the Viewing API to manage the document by sending messages to the window and receiving notification messages from the window. When you are finished with the document, you destroy its VAPI window.

To create the VAPI window, use the standard Windows API functions `CreateWindow()` or `CreateWindowEx()`, with the following parameter values:

Parameter	Value/Description
LPCTSTR lpClassName	VAPIDF_VAPI_WINDOW_CLASS_NAME (defined in <code>kvvapi.h</code>).
LPCTSTR lpWindowName	NULL.
DWORD dwStyle	WS_CHILD or WS_DISABLED
int x, y	0, 0
int nWidth	The width of the application (parent) window.
int nHeight	The height of the application (parent) window.
HWND hWndParent	The handle of the application window. (See note below.)

Parameter	Value/Description
HMENU hMenu	NULL.
HINSTANCE hInstance	The handle of the VAPI library.
LPVOID lpParam	A pointer to a TPVAPICreateParams structure that specifies optional parameters. Through this structure, you specify whether you are using an initialization file or registry settings.

For example:

```
hWndVAPI = CreateWindow (VAPIDF_VAPI_WINDOW_CLASS_NAME,  
                        NULL,  
                        WS_CHILD | WS_DISABLED,  
                        rc.left, rc.top, rc.right, rc.bottom,  
                        hWnd,  
                        NULL,  
                        hLibVAPI,  
                        &CreateParams);
```

Get the Viewer Window of the Document

The Viewer window is a document-specific window that the VAPI window creates when you open a document. Because the Viewer window is controlled by the VAPI window, normally you should not need the handle of the Viewer window.

The Viewer window is subclassed by the VAPI window. That is, when the VAPI window creates the Viewer window, it subclasses the Viewer window so that the VAPI window intercepts all messages sent to the Viewer window. This allows the VAPI window to control the Viewer window and to handle the right-mouse context menu and common operations such as **SaveAs**.

Depending on the document type, the Viewer window might also create several child windows in order to handle the document. For example, when you open a spreadsheet document, the VAPI window creates a [Workbook] Viewer window, which in turn creates a [SpreadSheet] Viewer window for each spreadsheet page when it is accessed. All VAPI child windows are destroyed when the VAPI window is destroyed.

- To get the Viewer window handle of a document, use the [VAPIMWP_INIT_GETHWNDVIEWER](#) message.
- To disable the Viewer user interface for a document (that is, when the Viewer asks if the user interface is disabled before creating a dialog box), respond to the [VAPINMWP_INIT_DISABLEUI](#) notification message.

Open and View a Document

Because a document must be opened before it can be viewed, printed, saved, or can have any other operation performed on it, viewing a document in the Viewing API means to open *and* view a document.

It is possible, however, to open a document without viewing it, or in other words, to open a document with view mode disabled. In this mode, you can print or save the document without viewing it.

To open a document

1. Create a [TPVAPIOpenDocumentInfo](#) structure.

To open a document without viewing it (view mode disabled), set the `VAPIDF_FLAGS_OPEN_WITHOUT_VIEW` flag in the `nFlags` member of the `TPVAPIOpenDocumentInfo` structure.

2. Send VAPI a `VAPIM_INIT` message with the `wParam` set to `VAPIMWP_INIT_OPEN_DOCUMENT`, and the `lParam` set to the address of the `TPVAPIOpenDocumentInfo` structure. See [VAPIMWP_INIT_OPEN_DOCUMENT](#), on page 101.

For example:

```
memset (&OpenDocInfo, 0, sizeof(TPVAPIOpenDocumentInfo));
OpenDocInfo.lpszFilePath = szFileName;
lResult = SendMessage (hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_OPEN_DOCUMENT,
(LPARAM)OpenDocInfo);
```

Notification Messages

- To receive the status of the open document expressed as the percent done, wait for the [VAPINMWP_INIT_OPENDOCDONE](#), on page 153 notification message.
- To receive the current page number of the document, wait for the [VAPINMWP_INIT_PAGENUMBER](#) notification message.
- To receive the name of the current object of the document, wait for the [VAPINMWP_MULTIOBJ_OBJNAME](#) notification message.

Save a Document

To save a document

1. Open the document. See [Open and View a Document, on the previous page](#).

To make sure that the entire document is opened before the document is saved, open the document with the `bWait` member in the `TPVAPIOpenDocumentInfo` structure set to `TRUE`.

To save a document without viewing it, open the document with view mode disabled.

2. Use the [VAPIMWP_FILE_CANSAVEAS](#) message to determine whether the document is completely processed and can be saved.
3. Use the [VAPIMWP_FILE_SAVEAS](#) message to save the document through a **Save As** dialog box.

To override the file path that VAPI uses to convert an I/O object to a temporary file when saving an I/O object document, respond to the [VAPINMWP_INIT_GETTEMPFILEPATH](#) notification message.

Convert a Document

To convert a document to text, RTF, or HTML

1. Open the document. See [Open and View a Document, on page 33](#).

To make sure that the entire document is opened before the document is saved, open the document with the `wait` member in the [TPVAPIOpenDocumentInfo](#) structure set to `TRUE`.

To save a document without viewing it, open the document with view mode disabled.

2. Use the [VAPIMWP_CANCONVERT](#) message to determine whether the document is completely processed and can be converted.
3. Use either the [VAPIMWP_FILE_SAVEAS](#) message to convert the document through a **Save As** dialog box or the [VAPIM_CONVERT](#) message to convert the document without requiring the user to respond to the **Save As** dialog box.

To override the file path that VAPI uses to convert an I/O object to a temporary file when converting an I/O object document, respond to the [VAPINMWP_INIT_GETTEMPFILEPATH](#) notification message.

NOTE: Viewing SDK does not convert PDFs, presentations, container files, or graphics files to text, RTF, or HTML.

Print a Document

The [pmtdemo](#) sample program demonstrates how to print by using the Viewing API.

To print a document

1. Open the document. See [Open and View a Document, on page 33](#).

To make sure that the entire document is opened before the document is printed, open the document with the `wait` member in the [TPVAPIOpenDocumentInfo](#) structure set to `TRUE`.

To print a document without viewing it, open the document with view mode disabled.

2. Use the [VAPIMWP_PRINT_CANPRINT](#) message to determine whether a document is completely processed and ready for printing.
3. Optionally, use the [VAPIMWP_PRINT_PRINTHEADER](#) message to print the file name, page number, and page length at the top of each page of a printed output.

Used in conjunction with [VAPIMWP_PRINT_PRINTHEADER](#), the [VAPIMWP_PRINT_SETPRINTNAME](#) message replaces the default file name field of the header with another string.

4. Optionally, use the [VAPIMWP_PRINT_PRINTTOPD](#) message to set the standard Windows print options.
5. Optionally, use the [VAPIMWP_PRINT_PAGESETUP](#) message to set print page scaling for a spreadsheet.
6. Use either the [VAPIMWP_PRINT_PRINT](#) message to print by using a common **Print** dialog box or the [VAPIMWP_PRINT_PRINTTOPRINTER](#) message to print to a specific printer without a **Print** dialog box.

Change the Print Job Name

You can change the print job name in the `kvsdk.ini` file. The printer uses the print job name for all documents printed from KeyView Viewing SDK.

To change the print job name

1. Open the `kvsdk.ini` file with a text editor. The file is installed in the root of the Windows directory.
2. In the `[Settings]` section, set the `PrintJobName` parameter to the desired print job name. For example:

```
[Settings]
PrintJobName=MyPrintJob
```

3. Save the file.

Determine the Document Format

To determine a document format

1. Open the document. See [Open and View a Document, on page 33](#).

To get format information without viewing the document, set the `VAPIDF_FLAGS_OPEN_VAPI_ONLY` flag in the `nFlags` member of the `TPVAPIOpenDocumentInfo` structure.

To quickly determine a document's format, regardless of whether the document is supported for viewing, set the `VAPIDF_FLAGS_OPEN_FORMAT_ONLY` flag in the `nFlags` member of the `TPVAPIOpenDocumentInfo` structure. Only the `VAPIM_INIT` message with the `VAPIMWP_INIT_GETDOCFORMAT` parameter is supported when opening a document with the `VAPIDF_OPEN_FORMAT_ONLY` flag enabled.

2. Use the [VAPIMWP_INIT_GETDOCFORMAT](#) parameter of the `VAPIM_INIT` message to get the document format of the currently opened document.
3. Use the [VAPIMWP_INIT_GETDOCCLASS](#) parameter of the `VAPIM_INIT` message to get the general class to which the currently opened document belongs.

Extract Document Metadata

To extract metadata from a document, use the [VAPIM_GETSUMMARYINFO](#) message.

Change Document Options

Document options control display elements such as window size, zoom settings, margin size, and scaling. Options are defined for each file type category (for example, spreadsheets, multimedia, and word processing). The document options only apply to the current document and document type. In other words, it initializes the in-memory options of the current Viewer. The options are defined in `kwoption.h`.

To set options for a document

1. Create an [ALL_OPTIONS_EX](#) structure.
2. If you are using the [VAPIMWP_INIT_OPEN_DOCUMENT](#) message to set options, create a [TPVAPIOpenDocumentInfo](#) structure.
3. Use either the [VAPIMWP_OPTIONS_SETOPTIONS_EX](#) or [VAPIMWP_INIT_OPEN_DOCUMENT](#) message.

To get the options of a document, use the [VAPIMWP_OPTIONS_GETOPTIONS_EX](#) message.

Annotate, Highlight, or Index a Document

- Use the [VAPIM_ENABLEINDEX](#) message to enable index-only mode. This generates text buffer ([VAPIMM_TEXTBUFFER](#)) notification messages with document viewing disabled.
- To specify the character set for the returned text buffer, use [VAPIM_SETINDEXBUFCHARSET](#).
- To add and delete annotations, use the [VAPIM_ANNOTATE](#) message. The annotation is placed at a logical address.
- To add a highlight to a document, use the [VAPIM_SETHILITE](#) message.
- See the following messages and notification messages for more functionality related to annotating, highlighting, or indexing documents:
 - [VAPIM_GETNEXTTEXTBUFFER](#), on page 72
 - [VAPIM_GETPAGEFROMLOGICAL](#), on page 72
 - [VAPIM_GETTEXT](#), on page 74
 - [VAPIM_GOTO_PAGE](#), on page 74
 - [VAPIM_POSITION](#), on page 75
 - [VAPIM_SETCURSOR](#), on page 77

- [VAPIM_SHOWHITS](#) , on page 79
- [VAPINM_ANNOTATION_HIT](#) , on page 146
- [VAPINM_EXTENT](#) , on page 147
- [VAPINM_SELECTION](#) , on page 148
- [VAPINM_USERCLICK](#), on page 150
- The [VAPIMWP_PRINT_ANNOTATIONS](#), on page 113 parameter of [VAPIM_PRINT](#)

Draw a Page

Draw a Page into a Supplied Device Context

- Use the [VAPIMWP_DRAW_INIT](#) parameter of the [VAPIM_DRAW](#) message to initialize the drawing routine in VAPI. You must send this parameter before you open the document by using the [VAPIMWP_INIT_OPEN_DOCUMENT](#) message.
- To get the number of pages in a document, open the document with the `bwait` parameter in the `TPVAPIOpenDocumentInfo` structure set to `TRUE`, and use the [VAPIMWP_DRAW_GETPAGECOUNT](#) parameter of the [VAPIM_DRAW](#) message. See [VAPIMWP_INIT_OPEN_DOCUMENT](#) for more information. You can set `bwait` to `FALSE` if you do not want to wait for the whole file to be processed and just want to get the size of the first few pages and draw the first few pages, or if you want to draw pages in any order.

For spreadsheets, you must use the [VAPIMWP_DRAW_GETPAGECOUNT](#) parameter to draw the worksheet pages successfully. To change the worksheet, use the [VAPIM_MULTIOBJ](#) message. See [VAPIMWP_MULTIOBJ_CANMULTIOBJ](#), on page 105 for more information.

- To get the size of the specified page, use the [VAPIMWP_DRAW_GETPAGESIZE](#) parameter of the [VAPIM_DRAW](#) message.
- To draw the specified page into the supplied device context, use the [VAPIMWP_DRAW_DRAWPAGE](#) parameter of the [VAPIM_DRAW](#) message.
- To create a thumbnail image file of a document page, use the [VAPIMWP_DRAW_DRAWTOFILE](#) parameter of the [VAPIM_DRAW](#) message.

Edit a Document

Search for Text

- To determine whether a document can be searched, use the [VAPIMWP_EDIT_CANFIND](#) message. See [VAPIMWP_EDIT_CANFIND](#) , on page 88.
- To search a document for the specified text, use the [VAPIMWP_EDIT_FIND](#) message. See [VAPIMWP_EDIT_FIND](#) , on page 90.

- To get the currently selected text in a document, use the `VAPIMWP_EDIT_GETFINDTEXT` message. See [VAPIMWP_EDIT_GETFINDTEXT](#) , on page 91.

Copy Text

- To determine whether the selected text in a document can be copied, use the `VAPIMWP_EDIT_CANCOPY` message. See [VAPIMWP_EDIT_CANCOPY](#) , on page 87.
- To copy the selected text in a document, use the `VAPIMWP_EDIT_COPY` message. See [VAPIMWP_EDIT_COPY](#) , on page 90.
- To determine whether all the items in a document can be selected, use the `VAPIMWP_EDIT_CANSELECTALL` message. See [VAPIMWP_EDIT_CANSELECTALL](#) , on page 89.
- To select all the items in a document, use the `VAPIMWP_EDIT_SELECTALL` message. See [VAPIMWP_EDIT_SELECTALL](#), on page 92.

Modify the Document View

Change the Layout of a Document

- To determine whether the layout of a document can be changed, use the [VAPIMWP_VIEW_CANLAYOUT](#) message.
- To get the current layout of a document, use the [VAPIMWP_VIEW_GETLAYOUT](#) message.
- To set the layout of a document, use the [VAPIMWP_VIEW_SETLAYOUT](#) message.

Change the Aspect Ratio of a Document

- To determine whether the aspect ratio of a document can be changed, use the [VAPIMWP_VIEW_CANASPECTRATIO](#) message.
- To get the current aspect ratio of a document, use the [VAPIMWP_VIEW_GETASPECTRATIO](#) message.
- To set the aspect ratio of a document, use the [VAPIMWP_VIEW_SETASPECTRATIO](#) message.

Invert, Rotate, or Magnify a Document

- To determine whether a document can be inverted, use the [VAPIMWP_VIEW_CANINVERT](#) message.
- To get the current invert state of a document, use the [VAPIMWP_VIEW_GETINVERT](#) message.
- To set the invert state of a document, use the [VAPIMWP_VIEW_SETINVERT](#) message.

- To determine whether a document can be rotated, use the [VAPIMWP_VIEW_CANROTATE](#) message.
- To get the current rotation of a document, use the [VAPIMWP_VIEW_GETROTATE](#) message.
- To set the rotation of a document, use the [VAPIMWP_VIEW_SETROTATE](#) message.
- To determine whether a document can be magnified, use the [VAPIMWP_VIEW_CANMAGNIFY](#) message.
- To determine whether a document can be magnified to fit the document selection to the window, use the [VAPIMWP_VIEW_CANFITTOWINDOW](#) message.
- To get the current magnification of a document, use the [VAPIMWP_VIEW_GETMAGNIFY](#) message.
- To set the magnification of a document, use the [VAPIMWP_VIEW_SETMAGNIFY](#) message.

Display or Hide Gridlines in a Document

- To determine whether a document supports gridlines, use the [VAPIMWP_VIEW_CANGRIDLINES](#) message.
- To get the current gridline state of a document, use the [VAPIMWP_VIEW_GETGRIDLINES](#) message.
- To set the gridline state of a document, use the [VAPIMWP_VIEW_SETGRIDLINES](#) message.

Play a Multimedia Document

- To determine whether a multimedia document can be played, use the [VAPIMWP_VIEW_CANPLAY](#) message.
- To play a multimedia document, use the [VAPIMWP_VIEW_PLAY](#) message.
- To determine whether the playing of a multimedia document can be paused, use the [VAPIMWP_VIEW_CANPAUSE](#) message.
- To pause the playing of a multimedia document, use [VAPIMWP_VIEW_PAUSE](#) .
- To determine whether the playing of a multimedia document can be stopped, use the [VAPIMWP_VIEW_CANSTOP](#) message.
- To stop the playing of a multimedia document, use the [VAPIMWP_VIEW_STOP](#) message.
- To get the play mode (that is, to stop or loop at the end of a multimedia document after playing it), use the [VAPIMWP_VIEW_GETPLAYMODE](#) message.
- To set the play mode of a multimedia document to stop at the end after playing it, use the [VAPIMWP_VIEW_END](#) message.
- To set the play mode of a multimedia document to loop at the end after playing it, use the [VAPIMWP_VIEW_LOOP](#) message.

Change the Current Object in a Document

There are many Viewing parameters that control the objects in a multiple-object document. Examples of a multiple-object document include a Microsoft Excel spreadsheet with multiple worksheets, and a Microsoft PowerPoint presentation with multiple slides.

- To determine whether a document contains multiple objects, use the [VAPIMWP_MULTIOBJ_CANMULTIOBJ](#) message.
- To determine the number of objects in a multiple-object document, use the [VAPIMWP_MULTIOBJ_GETOBJCOUNT](#) message.
- To change the current object to the next object in a document, use the [VAPIMWP_MULTIOBJ_NEXTOBJ](#) message.
- To change the current object to the previous object in a document, use the [VAPIMWP_MULTIOBJ_PREVOBJ](#) message.
- To get the name of the current object in a document, use the [VAPIMWP_MULTIOBJ_OBJNAME](#) message.
- To change the current object to a target object in a document, use the [VAPIMWP_MULTIOBJ_SETCURRENTOBJ](#) message.
- To receive the name of the current object, which VAPI sends when the document is first opened or whenever the object changes, wait for the [VAPIMWP_MULTIOBJ_OBJNAME](#) notification message.

View Deleted Items and Document Revision Marks

The revision tracking feature in applications—such as Microsoft Word's **Track Changes**—marks changes to a document (typically, strikethrough for deleted text and underline for inserted text) and tracks each change by reviewer name and date.

If revision tracking was enabled when changes were made to a document, you can configure Viewing to display the deleted content, revision marks, and revision tracking information in the document. Content that was added to the document is underlined. Content that was deleted from the document is displayed with strikethrough formatting. The name of the reviewer who made the change and the date on which the change was made is displayed in a tooltip when you hover the cursor over the revised text.

To display revision tracking information

1. Create a [TPVAPIOpenDocumentInfo](#) structure.
2. Set the `VAPIDF_FLAGS_INCL_REVISION_MARK` flag in the `nFlags` member of the `TPVAPIOpenDocumentInfo` structure.
3. Send VAPI a `VAPIM_INIT` message with the `wParam` set to `VAPIMWP_INIT_OPEN_DOCUMENT`, and the `lParam` set to the address of the `TPVAPIOpenDocumentInfo` structure. See [VAPIMWP_INIT_OPEN_DOCUMENT](#), on page 101 for more information.

The View API Demo program demonstrates how to implement the revision mark feature.

View Container Files

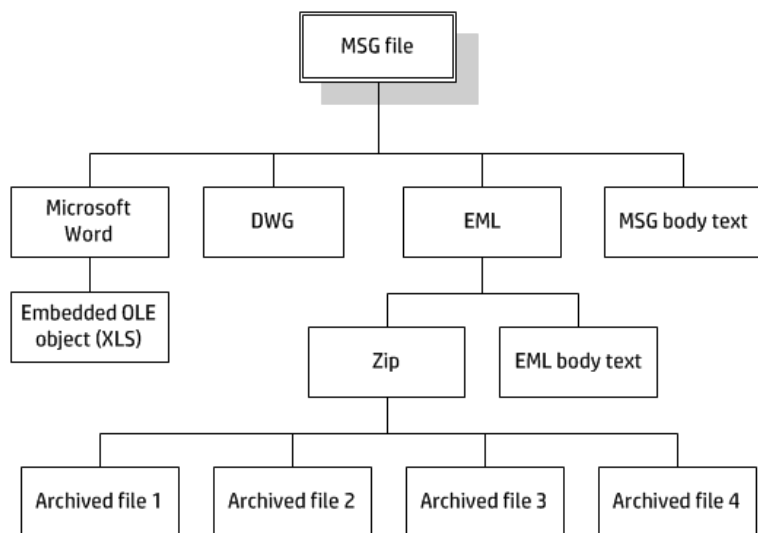
A *container* file has a main file (parent) and subfiles (children) embedded in the main file. The following are examples of container files:

- Compressed files such as ZIP, TAR, and RAR
- Mail messages such as Outlook (MSG) and Outlook Express (EML)
- Mail stores such as Microsoft Outlook Personal Folders (PST), Mailbox (MBX), and Lotus Notes database (NSF)

The subfiles might also be container files, creating a file hierarchy of multiple levels. For example, an MSG file (the root parent) might contain three attachments:

- a Microsoft Word document containing an embedded Microsoft Excel spreadsheet.
- an AutoCAD drawing file (DWG).
- an EML file with an attached ZIP file, which in turn contains four archived files.

Example Container File Tree Structure



NOTE: The parent MSG file contains four first-level children. The body text of a message file, although not a standalone file within the container, is considered a child of the parent file.

Microsoft Outlook Personal Folders (PST) Files

NOTE: The Microsoft Outlook Personal Folders (PST) readers are an advanced feature and are sold and licensed separately. To enable these readers in a KeyView SDK, you must obtain an appropriate license key from Micro Focus. For information about adding a new license key to an

existing installation, see [Update License Information, on page 18](#).

Choose the Reader to use for PST Files

KeyView provides the following ways of processing PST files:

- Indirectly, using the Microsoft Messaging Application Programming Interface (MAPI). MAPI is a Microsoft interface that enables different applications to exchange messages and attachments with each other. MAPI allows KeyView to open a PST file, traverse the folders and extract items. The `pstsr` reader uses MAPI, but requires that Microsoft Outlook is installed.
- Directly, without relying on the Microsoft interface to the PST format. Accessing the file directly does not require Microsoft Outlook. The `pstxsr` reader uses this approach.

The MAPI-based reader is used by default but you can choose `pstxsr` if you prefer.

The differences between the readers are summarized in the following table.

Feature	Native Reader (pstxsr)	MAPI-based Reader (pstsr)
Outlook required	No	Yes
MAPI properties supported	Yes. All properties defined in <code>mapitags.h</code> . Object properties are not supported.	
Password protection supported	Yes	Yes (using <code>KVCredential</code> structure)
Compressible encryption supported	Yes	Yes
High encryption supported	No	Yes

To use the native reader for PST files, change the PST entry in either the registry file or the initialization file as follows:

In the `kvsdk.ini` file

1. Open the `kvsdk.ini` file with a text editor. The file is installed in the root of the Windows directory.
2. In the `[KVMAILVE]` section of the `kvsdk.ini` file, change the parameter `297=pstsr.dll` to `297=pstxsr.dll`.

In the registry file

1. Open `install.reg.txt` in a text editor. The file is installed in the `install\redist` directory, where `install` is the directory in which you installed Viewing SDK.
2. Under the `[HKEY_LOCAL_MACHINE\Software\Autonomy\KeyviewViewingSDK\KVMAILVE]` key, change the parameter `"297"="pstsr.dll"` to `"297"="pstxsr.dll"`.

3. Save the file as `install.reg`.
4. Import the file into your Windows system registry.

System Requirements

MAPI is supported on Windows platforms only and relies on functionality in Outlook. If you want to use the MAPI-based reader, `pstsr`, Microsoft Outlook must be installed on the same machine as your application. Outlook must also be the default email application. KeyView supports the following PST formats and Outlook clients:

- Outlook 97 or later PST files

NOTE: The Outlook client must be the same version as, or newer than, the version of Outlook that generated the PST file.

- Outlook 2002 or later clients

NOTE: You must install an edition of Microsoft Outlook (32-bit or 64-bit) that matches the KeyView software. For example, if you use 32-bit KeyView, install 32-bit Outlook. If you use 64-bit KeyView, install 64-bit Outlook.

If the editions do not match, KeyView returns Error 32: `KVError_PSTAccessFailed` and an error message from Microsoft Office Outlook is displayed: Either there is a no default mail client or the current mail client cannot fulfill the messaging request. Please run Microsoft Outlook and set it as the default mail client.

Lotus Notes Database (NSF)

The NSF reader is an advanced feature and is sold and licensed separately. To enable this reader in a KeyView SDK, you must obtain the appropriate license key from Micro Focus. See [License Information, on page 17](#) for information on adding a new license key to an existing installation.

A Lotus Notes database is a single file that contains multiple documents called *notes*. Notes include design notes (such as forms, views, folders, navigators, outlines, pages, framesets, agents, and resources), data document notes, profile document notes, access control list notes, and collection (index) notes. KeyView can display text items, attachments, and OLE objects from data document notes only. Data document notes include emails, journal entries, discussion threads, documents (Microsoft Office and Lotus SmartSuite), and so on.

System Requirements

The NSF format is proprietary. Therefore, KeyView accesses NSF files indirectly by using the Lotus Notes API. Because the NSF reader relies on functionality in Lotus Notes, a Lotus Notes client or Lotus Domino server must be installed and configured on the same machine as the application that displays the NSF files.

KeyView supports Lotus Notes client version 6.5.1, Lotus Domino 6.5.1, and NSF files on the same platforms supported by Lotus Notes and Lotus Domino:

- Windows XP x86 (Service Pack 1 and 2)
- Windows 2000 x86 (Service Pack 2)

Installation and Configuration

Before KeyView can display NSF files, you must set up the Lotus Notes client or Lotus Domino server. Full configuration is not required. The following steps outline the minimal setup for NSF viewing:

1. Install the Lotus Notes client or Lotus Domino server. You do not need to configure the client or server.
2. Make sure that the file `notes.ini` is in the `install\lotus\notes` directory, where `install` is the directory where Lotus Notes is installed. If the file does not exist, create an ASCII file named `notes.ini`, and add the following text:

[Notes]
3. Add the `install\lotus\notes` and KeyView `bin` directories to the PATH environment variable. Micro Focus recommends that you add the KeyView `bin` directory because the Lotus Notes installation might contain older KeyView OEM libraries.

Format Notes

The KeyView NSF reader uses XML templates to format Lotus notes. You can customize the templates as required to approximate the look and feel of the original notes as closely as possible. For more information, see [Extract and Format Lotus Notes Subfiles, on page 358](#).

View Mail Messages and Mail Stores

You can display mail messages and mail stores in one of two ways:

- The Viewing window displays the file's hierarchy—showing all the children of the parent file—by using the archive format viewing engine, `kvarve.dll`. See [View Archive Files, on page 47](#) for more information.
- The Viewing windows displays the file as it would appear in a Microsoft Outlook Client. This display uses the mail format viewing engine, `kvmailve.dll`.

By default, mail messages and mail stores are displayed with the mail format viewing engine. To use the archive format viewing engine to display the complete file hierarchy, follow these steps:

In the `kvsdk.ini` file

1. Open the `kvsdk.ini` file with a text editor. The file is installed in the root of the Windows directory.
2. Remove the comments from the beginning of the following lines:

```
297=zip 0 kvarcve.dll; PST  
295=zip 0 kvarcve.dll; MSG MS Outlook
```

```
208=zip 0 kvarcve.dll; EML
299=zip 0 kvarcve.dll; Lotus Notes NSF
```

3. In the [VAPI] section of the kvsdk.ini file, insert comments at the beginning of the following lines:

```
; Mail formats
;kvmailve.dll=kvMAILVIEW;
;297=mail 0 kvmailve.dll; PST
;295=mail 0 kvmailve.dll; MSG MS Outlook
;208=mail 0 kvmailve.dll; EML
;299=mail 0 kvmailve.dll; Lotus Notes NSF
```

In the registry file

1. Open the install.reg.txt in a text editor. The file is installed in the *install\redist* directory, where *install* is the directory in which you installed Viewing SDK.
2. Remove the comments from the beginning of the following lines:

```
297=zip 0 kvarcve.dll; PST
295=zip 0 kvarcve.dll; MSG MS Outlook
208=zip 0 kvarcve.dll; EML
299=zip 0 kvarcve.dll; Lotus Notes NSF
```

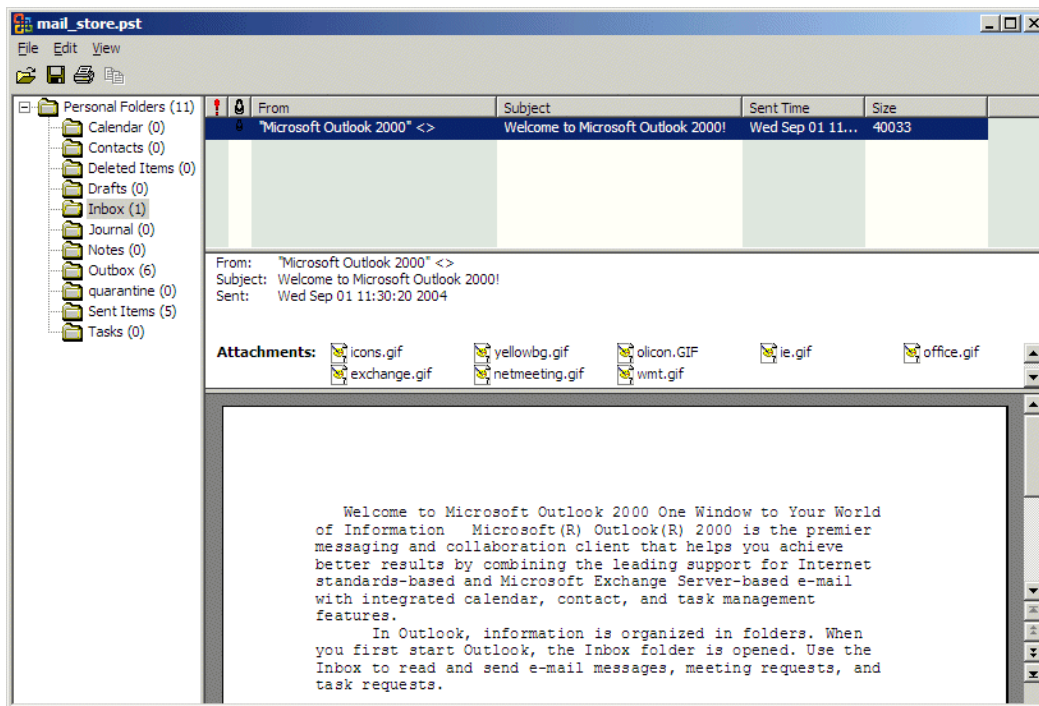
3. Under the [HKEY_LOCAL_MACHINE\Software\Autonomy\Viewing SDK\VAPI] key, insert comments at the beginning of the following lines:; Mail formats

```
; Mail formats
;kvmailve.dll=kvMAILVIEW;
;297=mail 0 kvmailve.dll; PST
;295=mail 0 kvmailve.dll; MSG MS Outlook
;208=mail 0 kvmailve.dll; EML
;299=mail 0 kvmailve.dll; Lotus Notes NSF
```

4. Save the file as *install.reg*.
5. Import the file into your Windows system registry.

The following figure shows a PST file displayed in the Viewing API sample program with the mail format viewing engine:

Display mail files with the mail format viewing engine



To extract the main message and its attachments to disk, select the main message and send an Unzip message or method (VAPIMWP_FILE_UNZIP or UnZip). See the implementation of the **Extract** menu in the View API Demo program (vapidemo).

To view an attachment, double-click the file in the **Attachments** field. The file is displayed in a separate window that can be closed.

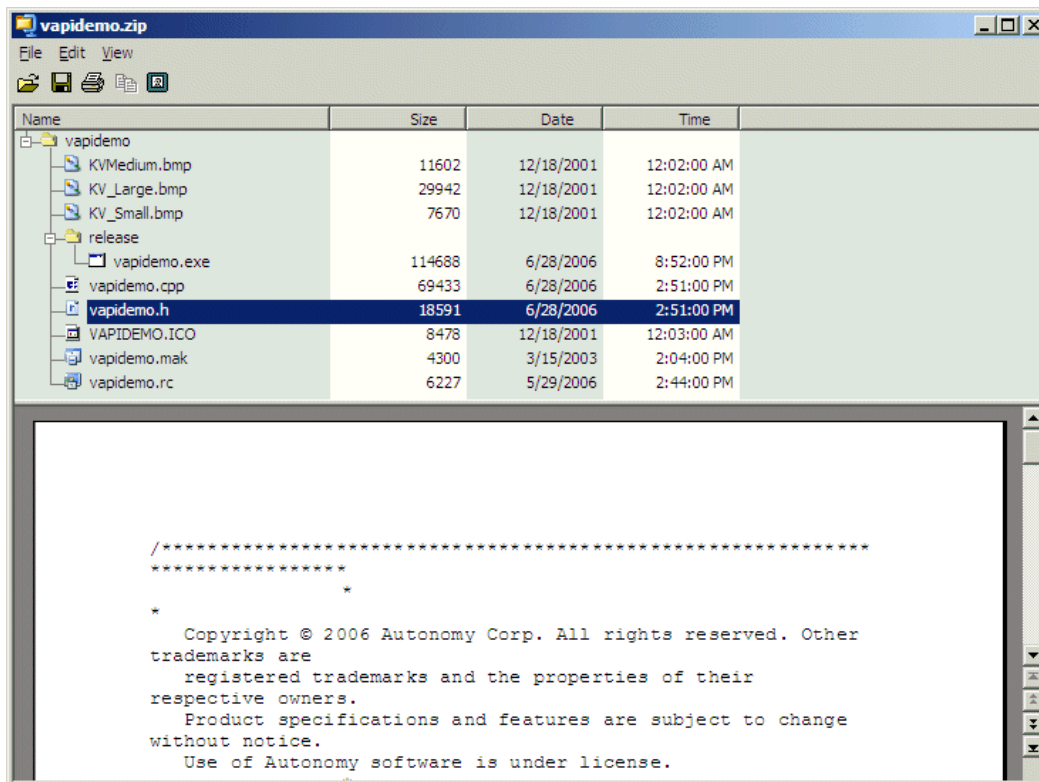
View Archive Files

The Viewing window displays an archive file's hierarchy—showing all the children of the parent file—by using the archive format viewing engine `kvarve.d11`. (You can also display mail files by using the archive format viewing engine. This is optional, and must be configured in the initialization file or registry file. See [View Mail Messages and Mail Stores, on page 45.](#))

When an archive file is opened for viewing, the archive's folders and subfiles are listed in one pane (the file list pane). When a user selects a subfile, the subfile is displayed in another pane (the preview pane). When a user double-clicks a selected subfile, the file's text is displayed in the entire application window. When a subfile is extracted to disk, the user is prompted for a target directory. If the file already exists on disk, a dialog box asks the user whether the file can be overwritten.

The following figure shows a ZIP file displayed in the Viewing API sample program.

Display an archive file with the archive viewing engine



Extract Subfiles to a Viewing Window or Disk

To extract a subfile or files from a container file to a Viewing window or disk

1. Open the container file. See [Open and View a Document, on page 33](#).
2. Use the `VAPIMWP_FILE_CANUNZIP` message to determine whether the selected subfile or files can be extracted.
3. Use the following `VAPIMWP_FILE_UNZIP` message to extract the selected subfile or files to a Viewing window or to disk. If you extract the files to disk, Viewing prompts you for the target directory and overwrite permission.

See the implementation of the **Extract** menu in the View API Demo program (`vapidemo`).

Display Subfiles in the Preview Pane

NOTE: The preview pane messages apply to the Archive Viewing Engine (`kvarcve`) only. The Mail Viewing Engine (`mailve`) does not use these settings.

To display a subfile in the preview pane

1. Use the [VAPIMWP_VIEW_CANPREVIEWPANE](#) message to determine whether a subfile can be displayed in a preview pane.
2. Use the [VAPIMWP_VIEW_GETPREVIEWPANE](#) message to determine whether the preview pane is currently being used.
3. If required, set the size and location of the preview pane using the `ARCHIVE_OPTIONS` structure. This structure is described in `kwoption.h`. See [Change Document Options, on page 37](#).
4. Use the [VAPIMWP_VIEW_SETPREVIEWPANE](#) message to specify the subfile is displayed in the preview pane.

The View API Demo program (`vapidemo`) demonstrates this functionality.

Set a Password for a Container File

For password-protected ZIP, PST, or NSF files, use the [VAPIMWP_INIT_SETPASSWORD](#) message to set the password before the file is opened.

View PDF Documents

You can view PDF files with Viewing SDK in one of three ways:

- View the PDF by using the Adobe Acrobat ActiveX control.
- View the PDF by using the Microsoft WebBrowser ActiveX control.
- View an image of each page of the PDF by using a graphic-based PDF reader (`kppdfldr` or `kppdf2ldr`).

By default, Viewing SDK uses the Acrobat ActiveX control to view PDF documents. If you do not want to redistribute the Acrobat Reader with your application, you can use a graphic-based reader instead.

Use the Acrobat ActiveX Control

The Acrobat control is automatically installed with Adobe® Reader® 4.0 or later. To download the Adobe Reader, go to www.adobe.com.

Use the Microsoft WebBrowser ActiveX Control

You can use the Microsoft WebBrowser ActiveX control to view PDF documents. The Microsoft WebBrowser ActiveX control is installed automatically with Microsoft Internet Explorer 3.0 or later. To use the WebBrowser ActiveX control to view PDF documents, follow one of these procedures:

In the `kvsdk.ini` file

1. Open the `kvsdk.ini` file with a text editor. The file is installed in the root of the Windows directory.
2. In the [General] section of the `kvsdk.ini` file, set the `UseHTMLPluginForPDF` parameter to **True**.
3. Pass the highlight or search term in by using `VAPIMWP_INIT_OPENDOCEX` (the extended version of `VAPIMWP_INIT_OPEN_DOCUMENT`. Refer to `ihademo.cpp` for details) with the `OpenDocInfo.lpszHighLight` structure. For example:

```
OpenDocInfo.lpszHighLight="search_term";
```

where `search_term` is the highlight or search term.

In the registry file

1. Open `install.reg.txt` in a text editor. The file is installed in the `install\redist` directory, where `install` is the directory in which you installed Viewing SDK.
2. Under the [HKEY_LOCAL_MACHINE\Software\Autonomy\Viewing SDK\General] key, set the following parameter:

```
"UseHTMLPluginForPDF"="true"
```

3. Save the file as `install.reg`.
4. Import the file into your Windows system registry.

Use a Graphic-Based PDF Reader

There are two graphic-based PDF readers available. The readers display PDFs by converting each page of the PDF to an image. If you do not want to redistribute the Acrobat Reader with your application, you can use a graphic-based reader instead.

The two readers support different features. Choose the appropriate reader depending on your requirements:

- The `kppdfldr` reader supports highlighting, annotation, and several other features, but also has several graphical limitations.
- The `kppdf2rdr` reader produces high-fidelity raster images but is a viewer only, and does not support highlighting or other features.

Use the `kppdfldr` Reader

The `kppdfldr` graphic-based reader has the following features:

- supports vector images
- supports rotation and scaling
- supports multibyte and bidirectional text

The `kppdfldr` reader has the following limitations:

- Embedded fonts in a PDF file are not translated correctly. They are usually displayed using the question mark (?) replacement character.
- If an unsupported font is encountered during conversion, the default font, Times New Roman, is substituted.
- Supports 180-degree rotation only for raster images.
- Supports the following color spaces: DeviceRGB, DeviceGray, DeviceCMYK, CalGray, and CalRGB color spaces. Indexed color spaces are supported as long as they are used with a supported basic color space.
- Does not support hyperlinks.
- Does not extract summary information (metadata).

Use the `kppdf2rdr` Reader

The `kppdf2rdr` graphic-based reader produces high-fidelity raster images. However, it has the following limitations:

- Does not support anything beyond viewing, such as highlighting or annotation.
- Does not support PDFs containing XFA forms content.

Specify the Graphic-based Reader

By default, the Acrobat control is used to view PDF documents. To use one of the graphic-based readers to view PDF documents, follow one of these procedures:

In the `kvsdk.ini` file

1. Open the `kvsdk.ini` file with a text editor. The file is installed in the root of the Windows directory.
2. In the `[VAPI]` section of the `kvsdk.ini` file, change the `200=doc 0 kvaxcc.dll` parameter to `200=pic 0 kpicve.dll`.
3. In the `[KVPICVE]` section, set the following parameter to the graphic-based reader you want to use. Set one of the following values:
 - For the `kppdfldr` reader:
`200=kppdfldr.dll`
This is the default setting.
 - For the `kppdf2rdr` reader:
`200=kppdf2rdr.dll`

In the registry file

1. Open `install.reg.txt` in a text editor. The file is installed in the `install\redist` directory, where `install` is the directory in which you installed Viewing SDK.
2. Under the `[HKEY_LOCAL_MACHINE\Software\Autonomy\Viewing SDK\VAPI]` key, change the `"200"="doc 0 kvaxcc.dll"` parameter to `"200"="pic 0 kvpicve.dll"`.
3. Under the `[HKEY_LOCAL_MACHINE\Software\Autonomy\Viewing SDK\KPICVE]` key, set the following parameter to the graphic-based reader you want to use. Set one of the following values:
 - For the `kppdfrdr` reader:
`"200"="kppdfrdr.dll"`
This is the default setting.
 - For the `kppdf2rdr` reader:
`"200"="kppdf2rdr.dll"`
4. Save the file as `install.reg`.
5. Import the file into your Windows system registry.

View or Extract PDF Portfolio Documents

A PDF Portfolio is a PDF file that contains multiple documents and an Adobe Flash interface that can provide information about them and allows you to navigate to them. The subfiles are not necessarily in PDF format; the portfolio might contain a PowerPoint presentation or a word processor file.

By default, the KeyView Viewing SDK opens PDF Portfolio files using the Adobe Acrobat ActiveX control. This requires that your users have Adobe Reader and Adobe Flash installed. Alternatively, you can configure KeyView to use the archive format viewing engine. This allows you to open the sub-files of the portfolio PDF without needing to have Adobe Acrobat or Adobe Flash installed, but does not allow you to view the Flash interface.

To choose how to view PDF Portfolio files

1. Open the `kvsdk.ini` file with a text editor. The file is installed in the root of the Windows directory.
2. In the `[VAPI]` section, change the value for the file category `1048`:
 - To use the Adobe ActiveX control:
`1048=doc 0 kvaxcc.dll`
 - To use the archive format viewing engine:
`1048=zip 0 kvarcve.dll`
3. Save and close the file.

View Microsoft Visio Files

Microsoft Visio files are supported by different readers or components depending on the file version:

- Version 2013 files are supported with ActiveX components included with the free Visio 2013 viewer provided by Microsoft. Image fidelity is supported, but additional features such as highlighting are not. Additional configuration steps are required.
- Version 2003–2010 files are supported with the `kpVSDrdr` reader by default. Image fidelity is supported. If desired, you can use the ActiveX components for these files instead of `kpVSDrdr`; this produces higher quality images but, as with 2013 files, does not support other Viewing features.
- Version 2002 and lower files are supported with the `vsdsr` reader. Image fidelity is not supported.

To enable support for Microsoft Visio 2013 files

1. Download and install the free Microsoft Visio 2013 Viewer from the following website:

<http://www.microsoft.com/en-us/download/details.aspx?id=35811>

2. Update the following entry in the `kvsdk.ini` file:

```
415=doc 0 kvaxcc.dll ; MS Visio 2013
```

To enable the ActiveX solution for Microsoft 2003–2010 files

1. Download and install the free Microsoft Visio 2013 Viewer from the following website:

<http://www.microsoft.com/en-us/download/details.aspx?id=35811>

2. Update the following entries in the `kvsdk.ini` file:

```
;294.6.0.11=prsgfx 0 kvpicve.dll ; MS Visio 2003/2007 (11.0)  
294.6.0.11=doc 0 kvaxcc.dll ; MS Visio 2003/2007 (11.0)
```

To disable the ActiveX functionality for 2003–2010 files and revert to the `kpVSDrdr` reader, update the entries as follows:

```
294.6.0.11=prsgfx 0 kvpicve.dll ; MS Visio 2003/2007 (11.0)  
;294.6.0.11=doc 0 kvaxcc.dll ; MS Visio 2003/2007 (11.0)
```

Extract Microsoft Excel Formulas

Normally, the actual value of a formula is extracted from an Excel spreadsheet; the formula from which the value is derived is not included in the output. However, KeyView enables you to include the value as well as the formula in the output. For example, if you configure Filter to extract the formula and the formula value, the output might look like this:

```
245 = SUM(B21:B26)
```

The calculated value from the cell is 245 and the formula from which the value is derived is SUM (B21:B26).

NOTE: Depending on the complexity of the formulas, enabling formula extraction might result in slightly slower performance.

To set the extraction option for formulas, add the following lines to the `formats.ini` file:

```
[Options]
getformulastring=option
```

where *option* is one of the following:

Option	Description
0	Extract the formula value only. This is the default. If formula extraction is enabled, and you want to return to the default, set this option.
1	Extract the formula only.
2	Extract the formula and the formula value.

If a function in a formula is not supported or is invalid, and option 1 or 2 is specified, only the calculated value is extracted. See the following table for a list of supported functions.

When you enable formula extraction, Filter can extract Microsoft Excel formulas containing the functions listed in the following table:

Supported Microsoft Excel functions

=ABS()	=ACOS()	=AND()	=AREAS()
=ASIN()	=ATAN2()	=ATAN2()	=AVERAGE()
=CELL()	=CHAR()	=CHOOSE()	=CLEAN()
=CODE()	=COLUMN()	=COLUMNS()	=CONCATENATE()
=COS()	=COUNT()	=COUNTA()	=DATE()
=DATEVALUE()	=DAVERAGE()	=DAY()	=DCOUNT()
=DDB()	=DMAX()	=DMIN()	=DOLLAR()
=DSTDEV()	=DSUM()	=DVAR()	=EXACT()
=EXP()	=FACT()	=FALSE()	=FIND()
=FIXED()	=FV()	=GROWTH()	=HLOOKUP()
=HOUR()	=ISBLANK()	=IF()	=INDEX()
=INDIRECT()	=INT()	=IPMT()	=IRR()
=ISERR()	=ISERROR()	=ISNA()	=ISNUMBER()
=ISREF()	=ISTEXT()	=LEFT()	=LEN()
=LINEST()	=LN()	=LOG()	=LOG10()

=LOGEST()	=LOOKUP()	=LOWER()	=MATCH()
=MAX()	=MDETERM()	=MID()	=MIN()
=MINUTE()	=MINVERSE()	=MIRR()	=MMULT()
=MOD()	=MONTH()	=N()	=NA()
=NOT()	=NOW()	=NPER()	=NPV()
=OFFSET()	=OR()	=PI()	=PMT()
=PPMT()	=PRODUCT()	=PROPER()	=PV()
=RATE()	=REPLACE()	=REPT()	=RIGHT()
=ROUND()	=ROUND()	=ROW()	=ROWS()
=SEARCH()	=SECOND()	=SIGN()	=SIN()
=SLN()	=SQRT()	=STDEV()	=SUBSTITUTE()
=SUM()	=SYD()	=T()	=TAN()
=TEXT()	=TIME()	=TIMEVALUE()	=TODAY()
=TRANSPOSE()	=TREND()	=TRIM()	=TRUE()
=TYPE()	=UPPER()	=VALUE()	=VAR()
=VLOOKUP()	=WEEKDAY()	=YEAR()	

Chapter 4: Viewing API Sample Programs

This section describes the sample programs that demonstrate how to use the API.

Overview

The following sample programs are provided for the Viewing API:

hellovapi, on the next page	vapidemo, on page 64
mfckv, on page 64	rtfdemo, on page 65
prntdemo, on page 65	filetype, on page 65
ihademo, on page 65	drawdemo, on page 66
uzipdemo, on page 66	

Micro Focus recommends that you review the `hellovapi` program first to help you get started. It is a simple program that demonstrates the basic functions of the Viewing API.

NOTE: The sample programs are Windows applications, not console applications. In other words, they contain a `WinMain` procedure instead of a `main` procedure.

Compile the Sample Programs

To compile the sample programs, use the makefile provided in each sample program directory. Make sure that the Viewing `include` directory is specified in the include path of the project.

After the executables are compiled and built, you must place them in the `release` subdirectory of each program.

Run the Sample Programs

To run a sample program

1. Install Viewing SDK.
2. Run the sample program from the `release` subdirectory of each sample program.

Viewing SDK Initialization Information

Viewing SDK uses initialization information for its internal operations; for example, to determine which components to load. You can store this information either in the Windows registry or in an initialization

file. When you use Viewing SDK you must tell it where to find this information and what form it is in. See [View Initialization Information, on page 22](#) for more information.

hellovapi demonstrates how to use the registry and the kvsdk.ini file.

hellovapi

hellovapi is a simple program that demonstrates how to use Viewing SDK to display documents in your application. The program creates a Windows application window, and then creates a child window that the Viewing SDK uses to display documents. The Viewing SDK is controlled by sending Windows-style messages to the child window. The set of messages that you send to the child window form the Viewing API (VAPI). The child window is known as a VAPI window. The VAPI messages are described in [Message Parameters, on page 67](#).

Load kvvapi.dll

Before you can create a VAPI window, you must load the VAPI library (kvvapi.dll) by using the LoadLibrary function. In hellovapi, the library is loaded in the InitializeVAPI function which is called during the processing of the WM_CREATE message.

The library is loaded in the following way:

1. InitializeVAPI calls either the GetPrivateProfileString Windows function or RegQueryValueEx to get the value of HOME in the General section of the kvsdk.ini file or Windows registry.

GetPrivateProfileString gets the value from kvsdk.ini, and RegQueryValueEx gets the value from the registry. The HOME setting specifies the location of the Viewing SDK bin directory. InitializeVAPI demonstrates how to get this setting from both the registry and kvsdk.ini file. By default, InitializeVAPI gets the value from kvsdk.ini.

2. InitializeVAPI then creates the path to the VAPI library and loads it:

```
wsprintf (szDLLPath, TEXT("%s\\%s"), szHome, VAPIDF_VAPI_DLL_NAME);  
hLibVAPI = LoadLibrary (szDLLPath);
```

Create the VAPI Window

After the VAPI library is loaded, the program creates the VAPI window by calling CreateWindow with a class name of VAPIDF_VAPI_WINDOW_CLASS_NAME:

```
hWndVAPI = CreateWindow (VAPIDF_VAPI_WINDOW_CLASS_NAME,  
                        NULL,  
                        WS_CHILD | WS_DISABLED,  
                        rc.left, rc.top, rc.right, rc.bottom,  
                        hWnd,  
                        NULL,  
                        hLibVAPI,  
                        &CreateParams);
```

The last parameter passes in a pointer to VAPI creation information. This creation information is in the structure of type `TPVAPICreateParams`, and tells VAPI where to locate the initialization settings, and whether they are in the registry or in `kvsdk.ini`. The `helloworld` program uses the `bUseRegistry` global variable to specify whether to get the settings from the registry or `kvsdk.ini`.

By default, `helloworld` tells VAPI to use the `kvsdk.ini` file that was located with the call to `InitializeVAPI`. The path to the `kvsdk.ini` file is stored in the `szIniFileName` global variable.

To specify the registry

1. Set `uProfileType` to `PROFILEDF_USE_REG`.
2. Set `lpszRegistryName` to the registry name of the Viewing SDK key under the main branch `HKEY_LOCAL_MACHINE\SOFTWARE`. The default is `Autonomy\Viewing SDK`.

To specify an initialization file

1. Set `uProfileType` to `PROFILEDF_USE_INI`.
2. Set `lpszIniFileName` to the location of the initialization file.

NOTE: The strings that you pass in for initialization information must be ASCII strings. If your application is in Unicode, you must convert the strings to ASCII before you pass them in.

```
// Initialize the parameters for the creation of the VAPIwindow.//
memset (&CreateParams, 0, sizeof(TPVAPICreateParams));
if (bUseRegistry)
{
    CreateParams.uProfileType    = PROFILEDF_USE_REG;
    CreateParams.lpszRegistryName = REGISTRY_NAME_ASCII;
}
else
{
    CreateParams.uProfileType    = PROFILEDF_USE_INI;
    CreateParams.lpszIniFileName = szIniFileName;
}
```

Open a Document

To tell VAPI to open a document

1. Create a `TPVAPIOpenDocInfoEx` structure.
2. Send VAPI a `VAPIM_INIT` message with the `wParam` set to `VAPIMWP_INIT_OPENDOCEX`, and the `lParam` set to the address of the `TPVAPIOpenDocInfoEx` structure.

This is demonstrated in the `OpenDoc` function. It uses a Windows Open dialog box to get the name of a file, and sets the `lpszFilePath` parameter of the `TPVAPIOpenDocInfoEx` structure to this file name. If it is successful, the `VAPIM_INIT` message returns `VAPI_RETURN_SUCCESS`.

NOTE: The file name must be an ASCII string. If your application is in Unicode, you must convert the string to ASCII before passing it in.

```
// Open the document.//
memset (&OpenDocInfo, 0, sizeof(TPVAPIOpenDocInfoEx));
OpenDocInfo.lpszFilePath = szFileName;
lResult = SendMessage (hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_OPENDOCEX,
                        (LPARAM)&OpenDocInfo);
```

hellovapi.c

```
// hellovapi.c
This program demonstrates how to use the Viewing API (VAPI) to display documents.//
#ifdef UNICODE
#ifndef _UNICODE
#define _UNICODE
#endif
#endif
#include <windows.h>
#include "kvoem.h"
#include "hellovapi.h"
#define REGISTRY_NAME TEXT("Autonomy\\Viewing SDK")
#define REGISTRY_NAME_ASCII "Autonomy\\Viewing SDK"
HWND      hWndVAPI      = NULL;    //VAPI window
HINSTANCE hLibVAPI      = NULL;    //VAPI library instance
BOOL      bUseRegistry = FALSE;    //Profile type (Set to FALSE for initialization
file)
TCHAR     szIniFileName[MAX_PATH];
LRESULT CALLBACK WndProc (HWND, UINT, WPARAM, LPARAM);
void OpenDoc (HWND);
BOOL InitializeVAPI (void);
int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance, PSTR szCmdLine,
int nCmdShow)
{
    static TCHAR szAppName[] = TEXT("Hello VAPI demo program");
    HWND        hWnd;
    MSG         msg;
    WNDCLASSEX wc;
    wc.cbSize      = sizeof (wc);
    wc.style       = 0;
    wc.lpfnWndProc = WndProc;
    wc.cbClsExtra  = 0;
    wc.cbWndExtra  = 0;
    wc.hInstance   = hInstance;
    wc.hIcon       = LoadIcon (NULL, IDI_APPLICATION);
    wc.hCursor     = LoadCursor (NULL, IDC_ARROW);
```

```
    wc.hbrBackground = (HBRUSH) (COLOR_WINDOW + 1);
    wc.lpszMenuName = MAKEINTRESOURCE(IDR_MENU);
    wc.lpszClassName = szAppName;
    wc.hIconSm = LoadIcon (NULL, IDI_APPLICATION);
    RegisterClassEx (&wc);
    hWnd = CreateWindow (szAppName, TEXT("Hello VAPI"),
        WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, CW_USEDEFAULT,
        CW_USEDEFAULT, CW_USEDEFAULT,
        NULL, NULL, hInstance, NULL);
    ShowWindow (hWnd, nCmdShow);
    UpdateWindow (hWnd);
    while (GetMessage (&msg, NULL, 0, 0))
    {
        TranslateMessage (&msg);
        DispatchMessage (&msg);
    }
    return msg.wParam;
}
LRESULT CALLBACK WndProc (HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    switch (uMsg)
    {
        case WM_CREATE:
        {
            TPVAPICreateParams CreateParams;
            RECT rc;
#ifdef UNICODE
            char szName[MAX_PATH];
#endif
            if (!InitializeVAPI())
            {
                return -1;
            }
            // Initialize parameters for creation of the VAPI window.//
            memset (&CreateParams, 0, sizeof(TPVAPICreateParams));
            if (bUseRegistry)
            {
                CreateParams.uProfileType = PROFILEDF_USE_REG;
                CreateParams.lpszRegistryName = REGISTRY_NAME_ASCII;
            }
            else
            {
                CreateParams.uProfileType = PROFILEDF_USE_INI;
#ifdef UNICODE
                WideCharToMultiByte (CP_ACP, 0, szIniFileName, -1, szName,
                    MAX_PATH, NULL, NULL);
                CreateParams.lpszIniFileName = szName;
#endif
            }
        }
    }
}
```

```
        CreateParams.lpszIniFileName = szIniFileName;
#endif
    }
    // Create the VAPI window. //
    GetClientRect (hWnd, &rc);
    hWndVAPI = CreateWindow (VAPIDF_VAPI_WINDOW_CLASS_NAME,
        NULL,
        WS_CHILD | WS_DISABLED,
        rc.left, rc.top, rc.right, rc.bottom,
        hWnd,
        NULL,
        hLibVAPI,
        &CreateParams);
    return 0;
}
case WM_DESTROY:
    // Destroy the VAPI window. //
    if (hWndVAPI != NULL)
    {
        DestroyWindow (hWndVAPI);
    }
    // Free the VAPI library. //
    if (hLibVAPI != NULL)
    {
        FreeLibrary (hLibVAPI);
    }
    PostQuitMessage (0);
    return 0;
case WM_CLOSE:
    DestroyWindow (hWnd);
    return 0;
case WM_SIZE:
    MoveWindow (hWndVAPI, 0, 0, LOWORD(lParam), HIWORD(lParam),
        TRUE);
    break;
case WM_COMMAND:
    switch (LOWORD(wParam))
    {
    case IDM_OPEN:
        OpenDoc (hWnd);
        return 0;
    case IDM_CLOSE:
        SendMessage (hWnd, WM_CLOSE, 0, 0);
        return 0;
    }
    break;
}
return DefWindowProc (hWnd, uMsg, wParam, lParam);
}
```

```
void OpenDoc (HWND hWnd)
{
    LRESULT          lResult;
    OPENFILENAME     ofn;
    TCHAR            szFileName[MAX_PATH];
    TPVAPIOpenDocInfoEx OpenDocInfo;
#ifdef UNICODE
    char szName[MAX_PATH];
#endif
    // Get a document name. //
    szFileName[0] = TEXT('\0');
    memset (&ofn, 0, sizeof(ofn));
    ofn.lStructSize = sizeof(OPENFILENAME);
    ofn.hwndOwner   = hWnd;
    ofn.lpstrFile   = szFileName;
    ofn.nMaxFile    = MAX_PATH;
    ofn.Flags       = OFN_PATHMUSTEXIST | OFN_HIDEREADONLY |
                    OFN_FILEMUSTEXIST;
    if (GetOpenFileName (&ofn) == 0)
    {
        return;
    }
    // Open the document. //
    memset (&OpenDocInfo, 0, sizeof(TPVAPIOpenDocInfoEx));
#ifdef UNICODE
    WideCharToMultiByte (CP_ACP, 0, szFileName, -1, szName, MAX_PATH,
                        NULL, NULL);
    OpenDocInfo.lpszFilePath = szName;
#else
    OpenDocInfo.lpszFilePath = szFileName;
#endif
    lResult = SendMessage (hWndVAPI, VAPIM_INIT,
                          VAPIMWP_INIT_OPENDOCEX, (LPARAM)&OpenDocInfo);
    if (lResult != VAPI_RETURN_SUCCESS)
    {
        MessageBox (hWnd, TEXT("Unable to view document."),
                    TEXT("Hello VAPI"), MB_OK);
        return;
    }
    return;
}
// Function:InitializeVAPI() //
// Summary: Load and initialize KVVAPI.dll for use with helloworld. //
BOOL InitializeVAPI (void)
{
    long lResult;
    TCHAR szDLLPath[MAX_PATH];
    TCHAR szHome[MAX_PATH];
    // Get the location of the VAPI DLL.//
```

```
if (bUseRegistry)
{
    HKEY hKey;
    TCHAR szSubKey[256];
    DWORD dwType;
    DWORD dwcbData;
    // Open the registry key. //
    wsprintf (szSubKey, TEXT("SOFTWARE\\%s\\General"), REGISTRY_NAME);
    lResult = RegOpenKeyEx (HKEY_LOCAL_MACHINE, szSubKey, 0, KEY_READ, &hKey);
    if (lResult != ERROR_SUCCESS)
    {
        return FALSE;
    }
    dwcbData = sizeof(szHome);
    lResult = RegQueryValueEx (hKey, TEXT("HOME"), NULL, &dwType,
        (PBYTE)szHome, &dwcbData);
    RegCloseKey (hKey);
    if (lResult != ERROR_SUCCESS)
    {
        return FALSE;
    }
}
else
{
    int nSize;
    // Get the location of the initialization file. //
    GetWindowsDirectory (szIniFileName, MAX_PATH);
    _tcscat (szIniFileName, TEXT("\\kvsdk.ini"));
    if (GetFileAttributes (szIniFileName) == 0xffffffff)
    {
        return FALSE;
    }
    nSize = GetPrivateProfileString (TEXT("General"), TEXT("HOME"),
        NULL, szHome, MAX_PATH, szIniFileName);
    if (nSize <= 0)
    {
        return FALSE;
    }
}
// Load the VAPI DLL. //
wsprintf (szDLLPath, TEXT("%s\\%s"), szHome, VAPIDF_VAPI_DLL_NAME);
hLibVAPI = LoadLibrary (szDLLPath);
if (hLibVAPI == NULL)
{
    return FALSE;
}
return TRUE;
}
```

helloworld.h

```
// helloworld.h
// This file is the main header file for helloworld.exe
// Resource definitions
#define IDR_MENU          100
#define IDM_OPEN         100
#define IDM_CLOSE        101
```

helloworld.rc

```
// helloworld.rc resource script
#include "helloworld.h"
// Menu
IDR_MENU MENU DISCARDABLE
BEGIN
    POPUP "&File"
        BEGIN
            MENUITEM "&Open",          IDM_OPEN
            MENUITEM "&Exit",         IDM_CLOSE
        END
END
```

vapidemo

The `vapidemo` sample program is a Windows application that demonstrates most of the functionality of the Viewing API. To start the `vapidemo` program, double-click the `vapidemo.exe` file, or type `vapidemo` at a DOS prompt or in the Run dialog box.

NOTE: Menu options in `vapidemo` do not adjust according to the format of the document you view; however, they appear shaded if they do not apply to the format. This is a limitation of the sample program only, not the Viewing SDK.

mfckv

The `mfckv` sample program is a Single Document Interface (SDI) application written with Microsoft Foundation Classes (MFC). To start the `mfckv` program, double-click the `mfckv.exe` file or enter the following command at a DOS prompt or in the Run dialog box:

```
mfckv
```

To open (view) a document, select **Open** from the **File** menu.

rtfdemo

The `rtfdemo` sample program demonstrates how to use the Viewing API to perform conversions of documents to RTF. To start the `rtfdemo` program, enter the following command at a DOS prompt or in the Run dialog box:

```
rtfdemo sourcefile targetfile
```

where *sourcefile* and *targetfile* include the complete path and file name.

prntdemo

The `prntdemo` sample program demonstrates how to use the Viewing API to print documents to a specified printer. To start the `prntdemo` program, enter the following command at a DOS prompt or in the Run dialog box:

```
prntdemo printername,printerdevice,printerport
```

For example:

```
prntdemo \\Calculus\HP Laserjet IIIsi,winspool,NE00:
```

The program displays the **Open** dialog box for you to select the file to print.

filetype

The `filetype` sample program demonstrates how to use the Viewing API to obtain a document's type. To start the `filetype` program, enter the following command at a DOS prompt or in the Run dialog box:

```
filetype file
```

where *file* includes the complete path and file name.

ihademo

The `ihademo` sample program is a simple Windows application that demonstrates how to index (filter), highlight, and annotate documents.

NOTE: The `ihademo` sample program is intended for word processing documents only. You might encounter limitations if you use it with other formats.

To start the `ihademo` program, double-click the `ihademo.exe` file, or enter the following command at a DOS prompt or in the Run dialog box:

```
ihademo optional_word_to_index
```

To open and view a document, select **Open** from the **File** menu. If you specified a word to index at the command line (the *optional_word_to_index* parameter), all occurrences of the indexed word in the

document (index hits) are highlighted when you open a document. To show or hide the index hits, select **Show Hits** from the **View** menu.

To index a document without viewing it, select **Index Only** before you select **Open**. In "Index Only" mode, the document is not displayed; rather, text buffers are returned. The first text buffer returns automatically. To get more text buffers, select **Drive Next Buffer**. The `baseAddress` returned in the text buffer is the starting logical address of the returned text.

To insert an annotation at the cursor position, select **Annotate** from the **View** menu. If no text is selected, the **Annotate** command inserts a bitmap. If any text is selected, the **Annotate** command turns green and the selected text is underlined. You can also annotate by using a double-click. This method annotates (with a green underline) five characters, starting at the cursor position. If you click the annotation, the annotation text returns in a message box.

To get the current view position (that is, the first and last logical address displayed), select **Get View Position** from the **View** menu.

To get the logical address and page number at the cursor position, select **Current Page Number** from the **View** menu.

To get the text at the cursor position, select **Get Text** from the **View** menu.

drawdemo

The `drawdemo` sample program demonstrates how to draw a page of a word processing, spreadsheet, presentation, or picture file into a supplied Device Context (thumbnail view). The program captures the first few pages (up to `MAX_PAGES_TO_DISPLAY`) of the document in metafiles and later uses these metafiles to draw the pages on the screen.

To start the `drawdemo` program, double-click the `drawdemo.exe` file, or enter the following command at a DOS prompt or in the Run dialog box:

```
drawdemo
```

To draw the first page of a document into a supplied Device Context, select **Open** from the **File** menu.

uzipdemo

The `uzipdemo` sample program demonstrates how to use the Viewing API to extract subfiles from a container file such as a ZIP archive.

To start the `uzipdemo` program, enter the following command at a DOS prompt or in the Run dialog box:

```
uzipdemo sourcefile targetdirectory
```

where *sourcefile* and *targetdirectory* includes the complete path and file name.

Chapter 5: Message Parameters

This section provides information on the message parameters in the Viewing API. It includes the following topics:

• VAPIM_ANNOTATE	69
• VAPIM_ENABLEINDEX	70
• VAPIM_GETNEXTTEXTBUFFER	72
• VAPIM_GETPAGEFROMLOGICAL	72
• VAPIM_GETSUMMARYINFO	73
• VAPIM_GETTEXT	74
• VAPIM_GOTO_PAGE	74
• VAPIM_HAVEHILITE	75
• VAPIM_POSITION	75
• VAPIM_POSITIONHILITE	76
• VAPIM_SETCURSOR	77
• VAPIM_SETHILITE	77
• VAPIM_SETHILITEOPTIONS	78
• VAPIM_SETINDEXBUFCHARSET	79
• VAPIM_SHOWHITS	79
• VAPIM_CONVERT	80
• VAPIMWP_CANCONVERT	81
• VAPIMWP_DRAW_DRAWPAGE	81
• VAPIMWP_DRAW_DRAWTOFILE	83
• VAPIMWP_DRAW_GETPAGECOUNT	84
• VAPIMWP_DRAW_GETPAGESIZE	84
• VAPIMWP_DRAW_GETWORKBOOKPAGECOUNT	85
• VAPIMWP_DRAW_INIT	86
• VAPIMWP_DRAW_SHUTDOWN	87
• VAPIMWP_EDIT_CANCOPY	87
• VAPIMWP_EDIT_CANFIND	88
• VAPIMWP_EDIT_CANSELECTALL	89
• VAPIMWP_EDIT_COPY	90
• VAPIMWP_EDIT_FIND	90
• VAPIMWP_EDIT_FIND_UNICODE	91
• VAPIMWP_EDIT_GETFINDTEXT	91
• VAPIMWP_EDIT_SELECTALL	92
• VAPIMWP_FILE_CANSAVEAS	93

• VAPIMWP_FILE_CANUNZIP	94
• VAPIMWP_FILE_CLOSE	94
• VAPIMWP_FILE_SAVEAS	95
• VAPIMWP_FILE_UNZIP	95
• VAPIMWP_INIT_GETCHARSET	96
• VAPIMWP_INIT_GETDESCRIP	97
• VAPIMWP_INIT_GETDOCCLASS	97
• VAPIMWP_INIT_GETDOCFORMAT	98
• VAPIMWP_INIT_GETFILENAME	99
• VAPIMWP_INIT_GETHWNDVIEWER	100
• VAPIMWP_INIT_JUMPTOFIRSTHILIITE	100
• VAPIMWP_INIT_OPEN_DOCUMENT	101
• VAPIMWP_INIT_SETPASSWORD	103
• VAPIMWP_INIT_SETSRCCHARSET	104
• VAPIMWP_INIT_SETTRGCHARSET	104
• VAPIMWP_MULTIOBJ_CANMULTIOBJ	105
• VAPIMWP_MULTIOBJ_CANNEXTOBJ	106
• VAPIMWP_MULTIOBJ_CANPREVOBJ	107
• VAPIMWP_MULTIOBJ_CANSETCURRENTOBJ	107
• VAPIMWP_MULTIOBJ_GETOBJCOUNT	108
• VAPIMWP_MULTIOBJ_NEXTOBJ	109
• VAPIMWP_MULTIOBJ_OBJNAME	109
• VAPIMWP_MULTIOBJ_PREVOBJ	110
• VAPIMWP_MULTIOBJ_SETCURRENTOBJ	111
• VAPIMWP_OPTIONS_GETOPTIONS_EX	111
• VAPIMWP_OPTIONS_SETOPTIONS_EX	112
• VAPIMWP_PRINT_ANNOTATIONS	113
• VAPIMWP_PRINT_CANPRINT	114
• VAPIMWP_PRINT_PAGESETUP	114
• VAPIMWP_PRINT_PRINT	115
• VAPIMWP_PRINT_PRINTHEADER	116
• VAPIMWP_PRINT_PRINTSETUP	116
• VAPIMWP_PRINT_PRINTTOPD	117
• VAPIMWP_PRINT_PRINTTOPRINTER	117
• VAPIMWP_PRINT_SETPRINTNAME	118
• VAPIMWP_VIEW_CANASPECTRATIO	119
• VAPIMWP_VIEW_CANDECREASEFONT	120
• VAPIMWP_VIEW_CANFITTOWINDOW	120
• VAPIMWP_VIEW_CANGOTO	121

• VAPIMWP_VIEW_CANGRIDLINES	122
• VAPIMWP_VIEW_CANINCREASEFONT	122
• VAPIMWP_VIEW_CANINVERT	123
• VAPIMWP_VIEW_CANLAYOUT	124
• VAPIMWP_VIEW_CANMAGNIFY	125
• VAPIMWP_VIEW_CANPAUSE	125
• VAPIMWP_VIEW_CANPLAY	126
• VAPIMWP_VIEW_CANPREVIEWPANE	127
• VAPIMWP_VIEW_CANTROTATE	128
• VAPIMWP_VIEW_CANSTOP	128
• VAPIMWP_VIEW_DECREASEFONT	129
• VAPIMWP_VIEW_END	130
• VAPIMWP_VIEW_GETASPECTRATIO	130
• VAPIMWP_VIEW_GETGRIDLINES	131
• VAPIMWP_VIEW_GETINVERT	132
• VAPIMWP_VIEW_GETLAYOUT	133
• VAPIMWP_VIEW_GETMAGNIFY	133
• VAPIMWP_VIEW_GETPLAYMODE	134
• VAPIMWP_VIEW_GETPREVIEWPANE	135
• VAPIMWP_VIEW_GETROTATE	136
• VAPIMWP_VIEW_GOTOPAGE	136
• VAPIMWP_VIEW_INCREASEFONT	137
• VAPIMWP_VIEW_LOOP	138
• VAPIMWP_VIEW_PAUSE	138
• VAPIMWP_VIEW_PLAY	139
• VAPIMWP_VIEW_SETASPECTRATIO	139
• VAPIMWP_VIEW_SETGRIDLINES	140
• VAPIMWP_VIEW_SETINVERT	141
• VAPIMWP_VIEW_SETLAYOUT	141
• VAPIMWP_VIEW_SETMAGNIFY	142
• VAPIMWP_VIEW_SETPREVIEWPANE	143
• VAPIMWP_VIEW_SETROTATE	144
• VAPIMWP_VIEW_STOP	144

VAPIM_ANNOTATE

Adds and deletes annotations, and determines whether annotations exist.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI,
            VAPIM_ANNOTATE,
            (WPARAM) wControl,
            (LPARAM) (TPVAPIAnnotation*) lpAnnotation );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
wControl	KV_DEL_ANNOTATION (0) – delete annotation KV_ADD_ANNOTATION (1) – add annotation KV_ANNOTATION_EXISTS (2) – query annotation
lpAnnotation	A pointer to a TPVAPIAnnotation structure that defines the annotation.

Returns

- For KV_DEL_ANNOTATION, SendMessage() returns TRUE if successful; FALSE if the annotation did not exist.
- For KV_ADD_ANNOTATION, SendMessage() returns 0 if successful; 1 if out of memory; 2 if the annotation could not be added because it would cause an overlap with an existing annotation; and 3 if the logical address was invalid.
- For KV_ANNOTATION_EXISTS, SendMessage() returns TRUE if the annotation exists; FALSE otherwise.

Discussion

The size of the bitmap is not relevant.

VAPIM_ENABLEINDEX

Enables index-only mode, also called document filtering. This generates text buffer (VAPIM_ TEXTBUFFER) notification messages with document viewing disabled. The first text buffer notification message is generated after a VAPIMWP_INIT_OPEN_DOCUMENT message is sent. To get additional text buffer notification messages in this mode, call the [VAPIM_GETNEXTTEXTBUFFER](#) message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI,
            VAPIM_ENABLEINDEX,
            (WPARAM) n_IndexMode,
            (LPARAM) (TPVAPIHiLiteColor*) lpHiLiteColor );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
n_IndexMode	0 – Combined mode (document view and text buffers) 1 – Text buffers only 2 – Disable index (document view only)
lpHiLiteColor	A pointer to a TPVAPIHiLiteColor structure that defines the highlight color used to signify an index hit.

Returns

SendMessage() returns TRUE if successful; FALSE otherwise.

Discussion

- This message is passed to the VAPI control window to notify the Viewing display engine that a document index is under way. This message *must* be sent before the [VAPIMWP_INIT_OPEN_DOCUMENT](#) message. Check the return value from VAPIMWP_INIT_OPEN_DOCUMENT to make sure that indexing was really supported.
- Calling this message produces a sequence of VAPINM_TEXTBUFFER notification messages to the calling window—that is, the parent of the VAPI window—as well as enabling the viewing engine to handle highlight and annotation requests. If you are using index-only mode, the VAPI window should be hidden and destroyed when the index is complete. No GDI output is generated and no information is stored to render the document. This results in a faster initial index of the document.

When Viewing is in index-only mode, VAPIM_GETNEXTTEXTBUFFER messages must be used to drive Viewing to obtain VAPINM_TEXTBUFFER notification messages, with the exception of the very first buffer. In other words, after you send a VAPIMWP_INIT_OPEN_DOCUMENT message, you either get back one VAPINM_TEXTBUFFER notification message automatically, or two when there is only one buffer in the file. When you need more, request it.

- When text buffers are no longer necessary, send VAPIM_ENABLEINDEX with n_IndexMode set to 2 and reopen the same document.

VAPIM_GETNEXTTEXTBUFFER

Gets text buffers in index-only mode.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_GETNEXTTEXTBUFFER, 0, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if a text buffer was returned, and FALSE if there are no more text buffers in the document, that is, the end of document is reached.

Discussion

This message is used to generate VAPINM_TEXTBUFFER notification messages when Viewing is in index-only mode (except for the very first text buffer, which comes automatically after a VAPIMWP_INIT_OPEN_DOCUMENT message is sent). The VAPINM_TEXTBUFFER notification messages is received before this message returns.

There might be two notification messages generated, one for the text buffer and one to indicate the end of the document.

VAPIM_GETPAGEFROMLOGICAL

Gets the page number for a logical address.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_GETPAGEFROMLOGICAL, 0L,  
            (LPARAM) (long) lLogicalAddress );
```


Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lLogicalAddress	A long integer that is the logical address for which to get the page number.

Returns

SendMessage() returns the page number the specified logical address resides on; or -1 on error.

Discussion

For spreadsheets, this message fails if the page containing the specified logical address has not been completely indexed yet.

VAPIM_GETSUMMARYINFO

Gets document metadata, also referred to as summary information.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, STAT wParam,  
            (LPARAM) (*KVSummaryInfoEx) pSummaryInfo );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
wParam	0 – get summary information 1 – free summary information
pSummaryInfo	A pointer to a KVSummaryInfoEx structure that contains summary information about the document.

Returns

SendMessage() returns TRUE if successful (pSummaryInfo filled in with valid information, if wParam is 0), FALSE if it fails.

VAPIM_GETTEXT

Gets a text buffer from a specified range.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_GETTEXT, 0L,  
            (LPARAM) (TPVAPIGetText*) lpGetText );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpGetText	A pointer to a TPVAPIGetText structure, which defines the text to get.

Returns

SendMessage() returns the number of BYTES it stored in the buffer; or -1 on error.

Discussion

- Send this message to obtain a buffer of text from a specified range. It is assumed that the text buffer is large enough to hold the required number of bytes. The data is not null terminated.
- This message does not wait for a logical address to become valid in the same way as VAPIM_POSITION (for non-spreadsheets). This message does not retrieve text across buffer boundaries.
- For spreadsheets, this message also fails if the page containing the entire text buffer—that is, containing the last address in the text buffer—is not completely indexed.

VAPIM_GOTO_PAGE

If indexing is enabled on the document by using the [VAPIM_ENABLEINDEX](#) message, the document is displayed at the specified page.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_GOTO_PAGE,  
            (WPARAM)(int) nPageNumber, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
nPageNumber	The page to display.

Returns

SendMessage() returns TRUE if the call succeeds.

Discussion

- This message can be used only with word processing files. To use similar functionality with PPT files or the graphic-based PDF reader, see [VAPIMWP_VIEW_GOTOPAGE](#) , on page 136.

VAPIM_HAVEHILITE

Determines whether there is a Previous or Next highlight relative to the current position.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_HAVEHILITE,  
            (WPARAM) (BOOL) bPrevious, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
bPrevious	TRUE checks for a previous highlight, FALSE checks for the next highlight.

Returns

Returns TRUE if there is a Previous or Next highlight (depending on the setting of bPrevious) relative to the current position.

VAPIM_POSITION

Positions the document in the viewing window.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_POSITION, 0L,  
            (LPARAM) (TPVAPIPosition*) lpPosition );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpPosition	A pointer to a TPVAPIPosition structure that defines the position.

Returns

SendMessage() returns TRUE on success; or FALSE on error (for example, invalid position).

Discussion

- You can use this message at any time to position the document within the viewing window. If successful, lpPosition->first and lpPosition->last is set on return.
- If lpPosition->position is set to -1, this message fills in only the first and last values, without changing the current view position.
- If the specified position is not processed when this message is called, Viewing takes exclusive control until the position is encountered. In other words, this message does not return until the desired position is set.
- For spreadsheets, if the page containing the specified position has not been indexed at the time this message is called, this message returns FALSE immediately, and does not wait until the page containing the specified position is indexed.

VAPIM_POSITIONHILITE

Changes focus from previous to next highlight.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_POSITIONHILITE,  
            (LPARAM) (BOOL) bPrev, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
bPrev	TRUE goes to previous highlight, FALSE goes to next highlight.

Returns

SendMessage() returns TRUE on success; or FALSE on error.

VAPIM_SETCURSOR

Sets the viewing engine cursor.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_SETCURSOR, 0L,  
            (LPARAM) (HCURSOR) hCursor );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
hCursor	The handle of the new cursor.

Returns

SendMessage() returns the handle of the active cursor (HCURSOR); or NULL on error.

Discussion

After sending this message to change the cursor, it is assumed you will send another message to change the cursor back to its original shape.

VAPIM_SETHILITE

Highlights a region of text.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_SETHILITE,
            (WPARAM) (int) cbTextToHilite,
            (LPARAM) (long) lLogicalAddress );
```

Arguments

Argument	Description
hWndVAPI	The handle of VAPI window.
cbTextToHilite	An integer that is the number of bytes to highlight.
lLogicalAddress	A long integer that is the logical address from which to start highlighting.

Returns

SendMessage() returns non-zero on success; or zero on error.

Discussion

- This message should be sent to Viewing when a VAPINM_TEXTBUFFER is received, and before processing for that message has returned control to Viewing. When the SendMessage(hWnd, VAPINM_TEXTBUFFER, . .) returns, it is assumed that any highlights have been added to the buffer. There is no limit to the number of highlights that can be added.
- For spreadsheets, this message can also fail if the page containing the entire text region—that is, containing the last address in the text region—is not completely indexed.

VAPIM_SETHILITEOPTIONS

Sets highlight options.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_SETHILITEOPTIONS,
            (LPARAM) (TPVAPIHiLiteOptions) pHiLiteOptions );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
pHiLiteOptions	A pointer to a TPVAPIHiLiteOptions structure that contains information for the highlight options.

Returns

SendMessage() returns TRUE on success; or FALSE on error.

VAPIM_SETINDEXBUFCHARSET

Sets the character set for the returned indexed text buffer.

Syntax

```
#include <kvvapi.h>  
SendMessage (g_hWndVAPI, VAPIM_SETINDEXBUFCHARSET, kvcharset, 0 );
```

Arguments

Argument	Description
g_hWndVAPI	The handle of the VAPI window.
kvcharset	A value from the KVCharSet enumerated type in kvcharset.h.

Returns

SendMessage() returns TRUE if the call succeeds.

Discussion

If kvcharset is KVCS_UNKNOWN, the character set of the returned buffer is the Windows native character set (for example, KVCS_1252 for an English machine). This is also the default character set on an English machine when VAPIM_SETINDEXBUFCHARSET message is not sent.

VAPIM_SHOWHITS

Shows or hides index hits.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_SHOWHITS,  
            (WPARAM) (BOOL) bShowHits, 0 );
```

Arguments

Argument	Description
hWndVAPI	The handle of VAPI window.
bShowHits	TRUE shows hits, FALSE hides hits.

Returns

SendMessage() returns TRUE on success; or FALSE on error.

Discussion

You can send this message at any time to control whether hits are shown.

VAPIM_CONVERT

Converts the currently open document to another format, without requiring the user to respond to the **SaveAs** dialog. To generate the **SaveAs** dialog box, use the VAPIMWP_FILE_SAVEAS message. The TPVAPIConvert structure includes two data members, one a target file and the other a format code, which you can set to RTF, text, or HTML.

Syntax

```
#include <kvvapi.h>  
SendMessage( hWndVAPI, VAPIM_CONVERT, 0,  
            (LPARAM) (TPVAPIConvert*) lpConvert );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpConvert	A pointer to a TPVAPIConvert structure.

Returns

SendMessage() returns TRUE if the conversion succeeds.

Discussion

To make sure that the entire document is opened before the document is converted, open the document with the `bwait` member in the [TPVAPIOpenDocumentInfo](#) structure set to `TRUE`. Use the [VAPIMWP_CANCONVERT](#) message to determine whether the document has been completely processed and is ready to be converted.

VAPIMWP_CANCONVERT

Determines whether a file can be converted to another format. This is a parameter of the `VAPIM_CONVERT` message.

Syntax

```
#include <kvvapi.h>
BOOL bSupported = FALSE;
SendMessage(hWndVAPI, VAPIM_CONVERT, VAPIMWP_CANCONVERT,
            (LPARAM)&bSupported);
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>bSupported</code>	A pointer to a Boolean variable.

Returns

If `bSupported` is `TRUE`, the file can be converted. If `bSupported` is `FALSE`, the file cannot be converted.

Discussion

Some file formats, such as PDF, presentation graphics files, and graphics, cannot be converted to RTF. In these cases, use `VAPIMWP_CANCONVERT` to check whether conversion is possible.

VAPIMWP_DRAW_DRAWPAGE

Draws a page in an area on a device context. This is a parameter of the `VAPIM_DRAW` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_DRAW, VAPIMWP_DRAW_DRAWPAGE,  
            (LPARAM) (TPVAPIDrawPageInfo*) pDrawInfo );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
pDrawInfo	A pointer to a TPVAPIDrawPageInfo structure that contains information used for drawing. To draw pages in any order, set the <code>bwait</code> parameter in the TPVAPIOpenDocumentInfo structure to FALSE .

Returns

SendMessage() returns:

- VAPI_RETURN_SUCCESS if the call succeeds.
- VAPI_RETURN_NOT_INITIALIZED if the drawing routines have not been initialized.
- VAPI_RETURN_NO_PAGE if the requested page does not exist, or is being displayed before all previous pages have been displayed.
- VAPI_RETURN_NOT_AVAILABLE if the document does not support this feature (for example, ZIP files, video, audio).
- VAPI_RETURN_ERROR if an error has occurred.

Discussion

- Before you send this message, initialize VAPI by sending the [VAPIMWP_DRAW_INIT](#) message, and then open the document by sending the [VAPIMWP_INIT_OPEN_DOCUMENT](#) message.
- Page numbers start at 0. For example, set `uPage` to 0 to draw page 1, and to 1 to draw page 2.
- By default, you must draw pages sequentially: to draw page 3, you must first draw pages 1 and 2. To draw pages in any order, set the `bwait` parameter in the [TPVAPIOpenDocumentInfo](#) structure to **FALSE**.

VAPIMWP_DRAW_DRAWTOFILE

Draws a page of a document to a graphic file (thumbnail). This is a parameter of the VAPIM_DRAW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_DRAW, VAPIMWP_DRAW_DRAWTOFILE,  
            (LPARAM) (TPVAPIDrawFileInfo*) pDrawInfo );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
pDrawInfo	A pointer to a TPVAPIDrawFileInfo structure that contains information used for drawing a page to a file. To draw pages in any order, set the <code>bwait</code> parameter in the TPVAPIOpenDocumentInfo structure to FALSE .

Returns

SendMessage() returns:

- VAPI_RETURN_SUCCESS if the call succeeds.
- VAPI_RETURN_NOT_INITIALIZED if the drawing routines have not been initialized.
- VAPI_RETURN_NO_PAGE if the requested page does not exist, or is being displayed before all previous pages have been displayed.
- VAPI_RETURN_NOT_AVAILABLE if the document does not support this feature (for example, ZIP files, video, audio).
- VAPI_RETURN_ERROR if an error has occurred.

Discussion

- Before you send this message, initialize VAPI by sending the [VAPIMWP_DRAW_INIT](#) message, and then open the document by sending the [VAPIMWP_INIT_OPEN_DOCUMENT](#) message.
- Page numbers start at 0. For example, set `uPageNumbers` to **0** to draw page 1, and to **1** to draw page 2. For word processing documents, pages must be drawn sequentially. For example, to draw page 3, you must first draw pages 0 and 1.

To draw pages in any order, set the `bwait` parameter in the [TPVAPIOpenDocumentInfo](#) structure to `FALSE`.

VAPIMWP_DRAW_GETPAGECOUNT

Gets the number of pages in a document. This is a parameter of the `VAPIM_DRAW` message.

Syntax

```
#include <kvvapi.h>
SendMessage( hWndVAPI, VAPIM_DRAW, VAPIMWP_DRAW_GETPAGECOUNT,
            (LPARAM) (unsigned int*) pPageCount );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>pPageCount</code>	A pointer to an <code>unsigned int</code> that returns the number of pages in the document.

Returns

`SendMessage()` returns `TRUE` if the call succeeds, in which case `pPageCount` returns the number of pages in the document. `SendMessage()` returns `FALSE` if the call fails.

Discussion

- Before you send this message, initialize VAPI by sending the [VAPIMWP_DRAW_INIT](#) message, and then open the document by sending the [VAPIMWP_INIT_OPEN_DOCUMENT](#) message.
- To make sure that the entire document is opened before the page count is retrieved, open the document with the `bwait` member in the [TPVAPIOpenDocumentInfo](#) structure set to `TRUE`. If you do not set `bwait` to `TRUE`, the returned page count might not be accurate.

VAPIMWP_DRAW_GETPAGESIZE

Gets the default size of a page. This is a parameter of the `VAPIM_DRAW` message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_DRAW, VAPIMWP_DRAW_GETPAGESIZE,
```

```
(LPARAM) (TPVAPIPageSize*) pPageSize );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
pPageSize	A pointer to a TPVAPIPageSize structure which contains information on the page size.

Returns

SendMessage() returns:

- VAPI_RETURN_SUCCESS if the call succeeds.
- VAPI_RETURN_NOT_INITIALIZED if the drawing routines have not been initialized.
- VAPI_RETURN_NO_PAGE if the requested page does not exist or is being displayed before all previous pages have been displayed.
- VAPI_RETURN_NOT_AVAILABLE if the document does not support this feature (for example, ZIP files, video, audio).
- VAPI_RETURN_ERROR if an error has occurred.

Discussion

Before you send this message:

- Initialize VAPI by sending the [VAPIMWP_DRAW_INIT](#) message.
- Open the document by sending the [VAPIMWP_INIT_OPEN_DOCUMENT](#) message.
- Send the [VAPIMWP_DRAW_DRAWPAGE](#) message to draw the page.

VAPIMWP_DRAW_GETWORKBOOKPAGECOUNT

Gets the number of workbook pages in a spreadsheet document. This is a parameter of the VAPIM_DRAW message.

Syntax

```
#include <kvvapi.h>;  
SendMessage( hWndVAPI, VAPIM_DRAW,  
             VAPIMWP_DRAW_GETWORKBOOKPAGECOUNT,  
             (LPARAM) (unsigned int*) pPageCount );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
pPageCount	A pointer to an unsigned int that returns the number of workbook pages in the spreadsheet document.

Returns

SendMessage() returns TRUE if the call succeeds, in which case pPageCount returns the number of workbook pages in the document. SendMessage() returns FALSE if the call fails.

Discussion

Before you send this message, initialize VAPI by sending the [VAPIMWP_DRAW_INIT](#) message and then open the document by sending the [VAPIMWP_INIT_OPEN_DOCUMENT](#) message.

VAPIMWP_DRAW_INIT

Initializes the drawing routines in VAPI. This is a parameter of the VAPIM_DRAW message.

Syntax

```
#include <kvvapi.h>  
SendMessage( hWndVAPI, VAPIM_DRAW, VAPIMWP_DRAW_INIT, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds and FALSE if the call fails.

Discussion

You must send this message prior to opening the document, that is, prior to sending the [VAPIMWP_INIT_OPEN_DOCUMENT](#) message.

VAPIMWP_DRAW_SHUTDOWN

Before a new document is opened, this frees up any data from a previously opened document. This is a parameter of the VAPIM_DRAW message.

Syntax

```
SendMessage(g_hWndVAPI, VAPIM_DRAW, DRAW_SHUTDOWN, 0L );
```

Arguments

Argument	Description
g_hWndVapi	The handle of the VAPI window
uMsg	VAPIM_DRAW
wParam	VAPIMWP_DRAW_SHUTDOWN
lParam	Not used, set to 0

Returns

SendMessage() returns:

- VAPI_RETURN_SUCCESS (or TRUE, value 1) if the call succeeds.
- VAPI_RETURN_NOT_INITIALIZED (value 5) if the drawing routines have not been initialized.
- VAPI_RETURN_ERROR (or FALSE, value 0) if the call fails.

Discussion

VAPIM_DRAW messages are used in the sample drawdemo program.

VAPIMWP_EDIT Cancopy

Determines whether content is selected in the currently opened document and can be copied to the clipboard. This is a parameter of the VAPIM_EDIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage( hWndVAPI, VAPIM_EDIT, VAPIMWP_EDIT_CANCOPY,  
            (LPARAM) (BOOL*) lpbCanCopy );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanCopy	A pointer to a flag that returns TRUE or FALSE, depending on whether the document contains a selection that you can copy.

Returns

- `SendMessage()` returns TRUE if the call succeeds, in which case `lpbCanCopy` returns TRUE or FALSE.
- `SendMessage()` returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanCopy` is undefined.

Discussion

Use this message to control the state of a **Copy** menu item or toolbar button.

VAPIMWP_EDIT_CANFIND

Determines whether the document contents can be searched. This is a parameter of the `VAPIM_EDIT` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_EDIT, VAPIMWP_EDIT_CANFIND,  
            (LPARAM) (BOOL*) lpbCanFind );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanFind	A pointer to a flag that returns TRUE or FALSE, depending on whether you can search the document.

Returns

- `SendMessage()` returns `TRUE` if the call succeeds, in which case `lpbCanFind` returns `TRUE` or `FALSE`.
- `SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanFind` is undefined.

Discussion

Use this message to control the state of a **Find** menu item or toolbar button.

VAPIMWP_EDIT_CANSELECTALL

Determines whether you can select all items in the document. If `lpbCanSelectAll` is `TRUE`, you can select all items in the document by using the [VAPIMWP_EDIT_SELECTALL](#) message. This is a parameter of the `VAPIM_EDIT` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_EDIT, VAPIMWP_EDIT_CANSELECTALL,  
            (LPARAM) (BOOL*) lpbCanSelectAll );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>lpbCanSelectAll</code>	A pointer to a flag that returns <code>TRUE</code> or <code>FALSE</code> , depending on whether you can select all items in the document.

Returns

`SendMessage()` returns `TRUE` if the call succeeds, in which case `lpbCanSelectAll` returns `TRUE` or `FALSE`.

`SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanSelectAll` is undefined.

Discussion

Use this message to control the state of a **Select All** menu item or toolbar button.

VAPIMWP_EDIT_COPY

Copies the current selection in the document to the clipboard. Use the [VAPIMWP_EDIT Cancopy](#) message to determine whether content is selected and can be copied to the clipboard.. This is a parameter of the VAPIM_EDIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_EDIT, VAPIMWP_EDIT_COPY, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the copy succeeded; FALSE otherwise.

Discussion

Use this message to implement a **Copy** menu item or toolbar button.

VAPIMWP_EDIT_FIND

Searches the document for the specified text. This is a parameter of the VAPIM_EDIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_EDIT, VAPIMWP_EDIT_FIND,  
            (LPARAM) (TPVAPIFindInfo*) lpFindInfo );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpFindInfo	A pointer to a TPVAPIFindInfo structure that contains information about the search text.

Returns

SendMessage() returns TRUE if the find succeeded; FALSE otherwise.

Discussion

Use this message to implement a **Find** menu item or toolbar button.

VAPIMWP_EDIT_FIND_UNICODE

Searches the document for the specified UNICODE text. This is a parameter of the VAPIM_EDIT message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_EDIT, VAPIMWP_EDIT_FIND_UNICODE,
            (LPARAM) (TPVAPIFindInfo*) lpFindInfo );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpFindInfo	A pointer to a TPVAPIFindInfo structure that contains information about the text to search for.

Returns

SendMessage() returns TRUE if the find succeeded; FALSE otherwise.

Discussion

Use this message to implement a **Find** UNICODE menu item or toolbar button.

VAPIMWP_EDIT_GETFINDTEXT

Gets the currently selected text in the document. This is a parameter of the VAPIM_EDIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_EDIT, VAPIMWP_EDIT_GETFINDTEXT,  
            (LPARAM) (HGLOBAL*) lphgFindText );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lphgFindText	A pointer to an HGLOBAL handle, which returns the currently selected text if any text is selected. You must GlobalLock() this handle before using it, and must GlobalFree() it afterwards. However, you should not GlobalAlloc() this handle, because the Viewer does this.

Returns

SendMessage() returns TRUE if the find text was returned successfully; FALSE otherwise.

Discussion

Use this message to set the default text in the **Find** dialog box for a **Find** menu item or toolbar button.

VAPIMWP_EDIT_SELECTALL

Selects all content in the currently opened document. Use the [VAPIMWP_EDIT_CANSELECTALL](#) message to determine whether you can select all content in the document. This is a parameter of the VAPIM_EDIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(g_hWndFocus, VAPIM_EDIT, VAPIMWP_EDIT_SELECTALL, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
uMsg	VAPIM_EDIT

Argument	Description
wParam	VAPIMWP_EDIT_SELECTALL
lParam	Not used, set to 0

Returns

SendMessage() returns TRUE if the selection succeeded; FALSE otherwise.

Discussion

Use this message to implement a **Select All** menu item or toolbar button.

VAPIMWP_FILE_CANSAVEAS

Determines whether the document can be saved and converted. This is a parameter of the VAPIM_FILE message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_FILE, VAPIMWP_FILE_CANSAVEAS,  
            (LPARAM) (BOOL*) lpbCanSaveAs );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanSaveAs	A pointer to a flag that returns TRUE or FALSE, depending on whether you can save and convert the document.

Returns

SendMessage() returns TRUE if the call succeeds, in which case lpbCanSaveAs returns TRUE or FALSE.

SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case lpbCanSaveAs is undefined.

Discussion

Use this message to control the state of a **Save As** menu item or toolbar button.

VAPIMWP_FILE_CANUNZIP

If a container file is open and there are subfiles selected in the file, this parameter determines whether the subfile or files can be extracted. This is a parameter of the VAPIM_FILE message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_FILE, VAPIMWP_FILE_CANUNZIP,  
            (LPARAM) (BOOL*) lpbCanUnZip );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanUnZip	A pointer to a flag that returns TRUE or FALSE depending on whether you can extract the subfiles.

Returns

SendMessage() returns TRUE if the call succeeds, in which case lpbCanUnZip returns TRUE or FALSE.

SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case lpbCanUnZip is undefined.

Discussion

Use this message to control the state of an **Extract** menu item or toolbar button.

VAPIMWP_FILE_CLOSE

Closes the currently opened document. It is not necessary to use this message, because issuing a second VAPIMWP_INIT_OPEN_DOCUMENT message automatically closes the currently opened document. This is a parameter of the VAPIM_FILE message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_FILE, VAPIMWP_FILE_CLOSE, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE on success; FALSE otherwise.

VAPIMWP_FILE_SAVEAS

Saves the current document in another format through a **Save As** dialog box. This is a parameter of the VAPIM_FILE message. See [Save a Document, on page 34](#) and [Convert a Document, on page 35](#).

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_FILE, VAPIMWP_FILE_SAVEAS, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE on success; FALSE otherwise.

Discussion

Use this message to implement a **Save As** menu item or toolbar button. You can save the document in its current format, or use Viewing conversions to convert it.

VAPIMWP_FILE_UNZIP

Use this message to extract selected subfiles in a container file either to disk or to a Viewing window. Container file types include ZIP, TAR, or PST. This prompts the user to specify a target directory and password (if required).

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_FILE, VAPIMWP_FILE_UNZIP,  
            (LPARAM) (BOOL) bUnzipToDisk );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
bUnzipToDisk	A flag that is TRUE to extract to disk, FALSE (the default) to extract and view.

Returns

SendMessage() returns TRUE on success; FALSE otherwise.

Discussion

Use this message to implement an **Extract** menu or toolbar button. You can extract the selected file to disk, or extract and view it.

VAPIMWP_INIT_GETCHARSET

Gets the character set of the VAPI window. This is a parameter of the VAPIM_INIT message.

Syntax

```
SendMessage(g_hWndFocus, VAPIM_INIT, VAPIMWP_INIT_GETCHARSET, 0)
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
uMsg	VAPIM_INIT
wParam	VAPIMWP_INIT_GETCHARSET
lParam	Not used, set to 0

Returns

This message returns a value from the `KVCharSet` type (see the `kvcharset.h` file for a description) corresponding to the character set that VAPI is using for viewing the document. You can use this message to control the state of a menu item or toolbar button that allows a user to select a character encoding for viewing.

Discussion

- `vapidemo.cpp` demonstrates how to use the message.
- A relevant message is `VAPIMWP_INIT_GETAUTOSELECT`.

VAPIMWP_INIT_GETDESCRIP

Gets a description of the format of the currently opened document. You must send this *after* the `VAPIM_INIT_OPEN_DOCUMENT` message returns. This is a parameter of the `VAPIM_INIT` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_GETDESCRIP,  
            (LPARAM) (char*) lpszDescription );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>lpszDescription</code>	A pointer to a Pascal string that returns a description of the format of the currently opened document.

Returns

`SendMessage()` returns `TRUE` on success; `FALSE` otherwise.

VAPIMWP_INIT_GETDOCCLASS

Indicates the general class to which the currently opened document belongs. This is a parameter of the `VAPIM_INIT` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_GETDOCCLASS,  
            (LPARAM) (int*) lpnClass );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpnClass	A pointer to an integer that returns the document class: <ul style="list-style-type: none">• 1 - Text document (ASCII)• 2 - Word processor document (WP)• 3 - Spreadsheet document (SS)• 4 - Image (Image)• 5 - Multimedia document (MM)• 6 - Fax document (FAX)• 7 - Presentation (PG)• 8 - Archive• 9 - Other

Returns

SendMessage() returns TRUE if successful (that is, if lpnClass contains valid information); otherwise it returns FALSE.

VAPIMWP_INIT_GETDOCFORMAT

Gets document format information. This is a parameter of the VAPIM_INIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_GETDOCFORMAT,  
            (LPARAM) (*ADDOCFORMAT) pDocInfo );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
pDocInfo	A pointer to the ADDOCINFO structure that contains information about the document format.

Returns

SendMessage() returns TRUE if successful (pDocInfo is filled in with valid information); otherwise it returns FALSE.

Discussion

If you want to get format information without viewing the document, set the VAPIDF_FLAGS_OPEN_VAPI_ONLY flag in the nFlags member of the [TPVAPIOpenDocumentInfo](#) structure. This structure is supplied when you open the document by using the VAPIMWP_INIT_OPEN_DOCUMENT message parameter.

VAPIMWP_INIT_GETFILENAME

Gets the file name of the current document. This is a parameter of the VAPIM_INIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_GETFILENAME,  
            (LPARAM) (char*) lpstzFileName );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpstzFileName	A pointer to a Pascal string that returns the file name string.

Returns

SendMessage() returns TRUE on success; otherwise it returns FALSE.

Discussion

This message returns the file name (not the full path), so the file name can be used in the window title. This message must be sent after the document is opened, otherwise the file name does not exist yet. If the file is a container file, such as a ZIP, TAR, or PST file, this file name is the name of an extracted subfile.

VAPIMWP_INIT_GETHWNDVIEWER

Gets the handle of the Viewer window. This is a parameter of the VAPIM_INIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_GETHWNDVIEWER,  
            (LPARAM) (HWND*) lphWndViewer );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lphWndViewer	A pointer to a handle that returns the Viewer window handle.

Returns

SendMessage() returns TRUE if the Viewer window exists; otherwise it returns FALSE.

Discussion

You must send this message *after* you send the VAPIMWP_INIT_OPEN_DOCUMENT message to open the document, otherwise the Viewer window does not exist yet.

VAPIMWP_INIT_JUMPTOFIRSTHILIITE

Jumps to the first highlight. You must send this message before you send the VAPIM_INIT_OPEN_DOCUMENT message. This applies only when you use XML documents created with the Verity Developer's Kit (VDK) to specify highlights. This is a parameter of the VAPIM_INIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_JUMPTOFIRSTHILITE,  
            0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE on success, or FALSE on error.

VAPIMWP_INIT_OPEN_DOCUMENT

Opens a document for one of the following operations:

- viewing
- determining a document format without viewing
- printing/converting/saving without viewing

This is a parameter of the VAPIM_INIT message. See [Open and View a Document, on page 33](#).

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_OPEN_DOCUMENT,  
            (LPARAM) (TPVAPIOpenDocumentInfo*) pOpenDocumentInfo );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
pOpenDocumentInfo	A pointer to a TPVAPIOpenDocumentInfo structure that contains information about the document to be opened.

Returns

- `SendMessage()` returns `VAPI_RETURN_SUCCESS` if the document opening is successfully initiated.
- If there is no viewer for the specified document, `SendMessage()` returns `VAPI_RETURN_NO_VIEWER`.
- If the format of the document could not be determined, `SendMessage()` returns `VAPI_RETURN_UNKNOWN_FORMAT`.
- If the document is password protected, `SendMessage()` returns `VAPI_RETURN_PASSWORD_PROTECTED`.
- If the drawing routines have not been initialized, `SendMessage()` returns `VAPI_RETURN_NOT_INITIALIZED`.
- If the requested page does not exist, or is being displayed before all previous pages have been displayed, `SendMessage()` returns `VAPI_RETURN_NO_PAGE`.
- If the document does not support this feature (for example, ZIP files, video, audio), `SendMessage()` returns `VAPI_RETURN_NOT_AVAILABLE`.
- If the KeyView license is invalid, `SendMessage()` returns `VAPI_RETURN_INVALID_LICENSE_KEY`.
- If the KeyView license is expired, `SendMessage()` returns `VAPI_RETURN_EXPIRED_LICENSE_KEY`.
- If the input file or stream is invalid or corrupt, `SendMessage()` returns `VAPI_RETURN_BAD_INPUT`.
- Any other error condition causes `VAPI_RETURN_ERROR` to be returned.

Discussion

- This message initiates the document opening and returns before the document opening is complete unless you set the `bwait` parameter in the `TPVAPIOpenDocumentInfo` structure to `TRUE`.
- This message produces several notification messages:
 - One or more `VAPINMWP_INIT_OPENDOCDONE` notification messages are received to report the status of the document opening.
 - In addition, if the document to be opened contains pages, a `VAPINMWP_INIT_PAGENUMBER` notification message is received to report the current page number.
 - Finally, if the document to be opened contains objects (for example, a spreadsheet document containing pages), a `VAPINMWP_MULTIOBJ_OBJNAME` notification message is received to report the current object name (for example, the name of the current spreadsheet page).
- To open a document to view as text, set the `bViewAsText` parameter in the `TPVAPIOpenDocumentInfo` structure to `TRUE`.
- To open and process a document (print, convert, and so on) without viewing, set the `nFlags` parameter in the `TPVAPIOpenDocumentInfo` structure to `VAPIDF_FLAGS_OPEN_WITHOUT_VIEW`. This flag tells VAPI to create a hidden Viewer window. You must also set the `bwait` parameter in

the `TPVAPIOpenDocumentInfo` structure to `TRUE`, except when you use APIs to draw documents without viewing. See [Draw a Page, on page 38](#) for more information.

For example, to print a document without viewing, open the document with the `bwait` parameter set to `TRUE` and the `nFlags` parameter set to `VAPIDF_FLAGS_OPEN_WITHOUT_VIEW`, and then send the `VAPIMWP_PRINT_PRINT` message.

- To open a document and return format information without viewing, set the `nFlags` parameter in the `TPVAPIOpenDocumentInfo` structure to `VAPIDF_FLAGS_OPEN_VAPI_ONLY`. This flag does not create a Viewer window.
- To make sure that a document is fully processed before an operation (such as printing, converting, or searching) is performed, set the `bwait` parameter in the `TPVAPIOpenDocumentInfo` structure to `TRUE`. This is useful when you want to use an operation immediately after opening the document.

VAPIMWP_INIT_SETPASSWORD

Sets the password to use to open a password-protected file before the file is opened. Currently, you can use this to set a password for ZIP and PST files. This is a parameter of the `VAPIM_INIT` message.

Syntax

```
SendMessage (hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_SETPASSWORD,  
            (LPARAM) (LPCTSTR) pPasswordInfo );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>pPasswordInfo</code>	A pointer to a password string.

Returns

`SendMessage()` returns `TRUE` if successful, or `FALSE` if it fails.

Discussion

- For password-protected PST files, you must call this message *before* the `VAPIMWP_INIT_OPEN_DOCUMENT` message.
- For password-protected ZIP files, you can call this message after the `VAPIMWP_INIT_OPEN_DOCUMENT`, but you must call it before the protected subfile is extracted or viewed.
- Unicode passwords are not supported.

VAPIMWP_INIT_SETSRCCHARSET

Sets the source character set of a document. This is a parameter of the VAPIM_INIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_SETSRCCHARSET,  
            (LPARAM) eCharset );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
eCharset	A value from the enumerated type KVCharSet. See the kvcharset.h file for a description.

Returns

SendMessage() returns TRUE if successful, or FALSE if it fails.

Discussion

This message is used to specify the character set for documents when the character set cannot be determined by Viewing, such as in the case of plain text documents.

VAPIMWP_INIT_SETTRGCHARSET

Sets the target character set of a document. This is a parameter of the VAPIM_INIT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_INIT, VAPIMWP_INIT_SETTRGCHARSET,  
            (LPARAM) eCharset );
```


Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
eCharset	A value from the enumerated type KVCharSet. See the kvcharset.h file for a description.

Returns

SendMessage() returns TRUE if successful, or FALSE if it fails.

Discussion

This message forces the character set Viewing uses to display a document. For example, this allows Japanese documents to be accurately displayed on an English Windows machine if the Japanese fonts are available.

VAPIMWP_MULTIOBJ_CANMULTIOBJ

Determines whether the document contains multiple objects. See [Change the Current Object in a Document, on page 41](#). This is a parameter of the VAPIM_MULTIOBJ message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_MULTIOBJ,
            VAPIMWP_MULTIOBJ_CANMULTIOBJ,
            (LPARAM) (BOOL*) lpbCanMultiObj );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanMultiObj	A pointer to a flag that returns TRUE or FALSE, depending on whether the document contains multiple objects.

Returns

- `SendMessage()` returns `TRUE` if the call succeeds, in which case `lpbCanMultiObj` returns `TRUE` or `FALSE`.
- `SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanMultiObj` is undefined.

Discussion

Use this message to control the state of a **Next Object** or **Previous Object** menu item or toolbar button.

VAPIMWP_MULTIOBJ_CANNEXTOBJ

Determines whether the next object can be selected in a multiple-object document. See [Change the Current Object in a Document, on page 41](#). This is a parameter of the `VAPIM_MULTIOBJ` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_MULTIOBJ, VAPIMWP_MULTIOBJ_CANNEXTOBJ,  
            (LPARAM) (BOOL*) &lpbCanNextObj );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>lpbCanNextObj</code>	A pointer to a flag that returns <code>TRUE</code> or <code>FALSE</code> , depending on whether the document can select the next objects.

Returns

`SendMessage()` returns `TRUE` if the current object can be changed to the next object; otherwise it returns `FALSE`.

Discussion

Use this message to control the state of a **Next Object** menu item or toolbar button.

VAPIMWP_MULTIOBJ_CANPREVOBJ

Determines whether the previous object can be selected in a multiple-object document. See [Change the Current Object in a Document, on page 41](#). This is a parameter of the VAPIM_MULTIOBJ message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_MULTIOBJ, VAPIMWP_MULTIOBJ_CANPREVOBJ,
           (LPARAM) (BOOL*) &lpcanPrevObj );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpcanPrevObj	A pointer to a flag that returns TRUE or FALSE, depending on whether the document can select the previous object.

Returns

SendMessage() returns TRUE if the current object can be changed to the previous object; otherwise it returns FALSE.

Discussion

Use this message to control the state of a **Previous Object** menu item or toolbar button.

VAPIMWP_MULTIOBJ_CANSETCURRENTOBJ

Determines whether the target object can be selected in a multiple-object document. See [Change the Current Object in a Document, on page 41](#). This is a parameter of the VAPIM_MULTIOBJ message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_MULTIOBJ,
           VAPIMWP_MULTIOBJ_CANSETCURRENTOBJ, (LPARAM) (int)
           nTargetObj);
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
nTargetObj	A zero-based integer that is the target object.

Returns

SendMessage() returns TRUE if the current object can be set to the target object; otherwise it returns FALSE.

Discussion

Use this message to control the state of a **Set Current Object** menu item or toolbar button.

VAPIMWP_MULTIOBJ_GETOBJCOUNT

Get the total number of objects in a multiple-object document. See [Change the Current Object in a Document, on page 41](#). This is a parameter of the VAPIM_MULTIOBJ message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_MULTIOBJ,
            VAPIMWP_MULTIOBJ_GETOBJCOUNT, (LPARAM) (int)
            lpbTotalObj);
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbTotalObj	A pointer to the total number of objects.

Returns

SendMessage() returns TRUE.

Discussion

You can use this message to implement an object count item.

VAPIMWP_MULTIOBJ_NEXTOBJ

Changes the current object to the next object in a multiple-object document. See [Change the Current Object in a Document, on page 41](#). This is a parameter of the VAPIM_MULTIOBJ message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_MULTIOBJ, VAPIMWP_MULTIOBJ_NEXTOBJ, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the current object was changed to the next object; otherwise it returns FALSE.

Discussion

You can use this message to implement a **Next Object** menu item or toolbar button. This message generates a VAPIMWP_MULTIOBJ_OBJNAME notification message that reports the new object name.

When viewing spreadsheets or presentations in index mode, this message fails if the next page is not completely indexed yet, or if the current page is the last page. In other words, the program does not allow the last page to wrap around to the first page, as it does in non-index mode.

VAPIMWP_MULTIOBJ_OBJNAME

Gets the current object name for a multiple-object document. See [Change the Current Object in a Document, on page 41](#). This is a parameter of the VAPIM_MULTIOBJ message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_MULTIOBJ, VAPIMWP_MULTIOBJ_OBJNAME,  
            (LPARAM) (char*) lpstzObjectName );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpstzObjectName	A pointer to a Pascal string that returns the object name string.

Returns

SendMessage() returns TRUE on success; otherwise it returns FALSE.

Discussion

You can use this message only with multiple-object documents.

VAPIMWP_MULTIOBJ_PREVOBJ

Changes the current object to the previous object in a multiple-object document. See [Change the Current Object in a Document, on page 41](#). This is a parameter of the VAPIM_MULTIOBJ message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_MULTIOBJ, VAPIMWP_MULTIOBJ_PREVOBJ, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the current object was changed to the previous object; otherwise it returns FALSE.

Discussion

- You can use this message to implement a **Previous Object** menu item or toolbar button. This message generates a VAPIMWP_MULTIOBJ_OBJNAME notification message that reports the new object name.

- When viewing spreadsheets or presentations in index mode, this message fails if the next page has not been completely indexed yet, or if the current page is the first page. It does not allow the first page to wrap around to the last page, as it does in non-index mode.

VAPIMWP_MULTIOBJ_SETCURRENTOBJ

Changes the current object to the target object in a multiple-object document. See [Change the Current Object in a Document, on page 41](#). This is a parameter of the VAPIM_MULTIOBJ message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_MULTIOBJ,
            VAPIMWP_MULTIOBJ_SETCURRENTOBJ, (LPARAM) (int)
            nTargetObj;)
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
nTargetObj	A zero-based integer that is the target object.

Returns

SendMessage() returns TRUE if the current object was changed to the target object; otherwise it returns FALSE.

Discussion

You can use this message to implement a **Set Current Object** menu item or toolbar button. This message generates a VAPIMWP_MULTIOBJ_OBJNAME notification message that reports the new object name.

VAPIMWP_OPTIONS_GETOPTIONS_EX

Gets the document options. Document options control display elements such as window size, zoom settings, margin size, scaling, and revision tracking information. Options are defined for each file type category (for example, spreadsheets, multimedia, and word processing). This is a parameter of the VAPIM_OPTIONS message. See [Change Document Options, on page 37](#).

Syntax

```
#include <kvvapi.h>
#include <kwoption.h>
SendMessage(hWndVAPI, VAPIM_OPTIONS,
             VAPIMWP_OPTIONS_GETOPTIONS_EX,
             (LPARAM) (ALL_OPTIONS_EX*) lpAllOptions );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpAllOptions	A pointer to an ALL_OPTIONS_EX structure that returns the document options.

Returns

SendMessage() returns TRUE if the call succeeds, in which case lpAllOptions returns the options.

SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case lpAllOptions is undefined.

Discussion

Use this message to get the options for the document. Only the options for the current document type are returned, not those for all document types.

VAPIMWP_OPTIONS_SETOPTIONS_EX

Sets the current document options. Document options control display elements such as window size, zoom settings, margin size, and scaling. Options are defined for each file type category (for example, spreadsheets, multimedia, and word processing). This is a parameter of the VAPIM_OPTIONS message. See [Change Document Options, on page 37](#).

Syntax

```
#include <kvvapi.h>
#include <kwoption.h>
SendMessage(hWndVAPI, VAPIM_OPTIONS,
             VAPIMWP_OPTIONS_SETOPTIONS_EX,
             (LPARAM) (ALL_OPTIONS_EX*) lpAllOptions );
```


Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpAllOptions	A pointer to an ALL_OPTIONS_EX structure that shows the document options.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to set the options for the document. This message does not save the document options in the registry. In addition, this message sets the document options for the current document and document type only. In other words, it initializes the in-memory options of the current Viewer.

VAPIMWP_PRINT_ANNOTATIONS

Specifies whether annotations are included in the printed output. To print a document, use either the [VAPIMWP_PRINT_PRINT](#), or [VAPIMWP_PRINT_PRINTTOPRINTER](#) message. This is a parameter of the VAPIM_PRINT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_PRINT, VAPIMWP_PRINT_ANNOTATIONS,  
            (LPARAM) (BOOL) bPrintAnnotations );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
bPrintAnnotations	If TRUE, the document is printed with annotations. If FALSE, the document is printed without annotations.

Returns

SendMessage() returns TRUE if the call succeeds.

VAPIMWP_PRINT_CANPRINT

Determines whether the document can be printed. This is a parameter of the VAPIM_PRINT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_PRINT, VAPIMWP_PRINT_CANPRINT,  
            (LPARAM) (BOOL*) lpbCanPrint );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanPrint	A pointer to a flag that returns TRUE or FALSE, depending on whether the document can be printed.

Returns

- SendMessage() returns TRUE if the call succeeds, in which case lpbCanPrint returns TRUE or FALSE.
- SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case lpbCanPrint is undefined.

Discussion

Use this message to control the state of a **Print** menu item or toolbar button.

VAPIMWP_PRINT_PAGESETUP

Sets up the print page scaling for a spreadsheet. This is a parameter of the VAPIM_PRINT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_PRINT, VAPIMWP_PRINT_PAGESETUP, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement a **Print Page Setup** menu item or toolbar button.

VAPIMWP_PRINT_PRINT

Prints the current document by calling the common **Print** dialog box. This is a parameter of the VAPIM_PRINT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_PRINT, VAPIMWP_PRINT_PRINT, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

- This message prints the current document by calling the common **Print** dialog box to set the printer parameters. You can use this message to implement a **Print** menu item or toolbar button.
- When printing in an application that is a Windows service, a default printer must be installed for the user account using the application.
- To make sure that the entire document is opened before the document is printed, open the

document with the `bwait` member in the [TPVAPIOpenDocumentInfo](#) structure set to `TRUE`. Use the [VAPIMWP_PRINT_CANPRINT](#) message to determine whether the document has been completely processed and is ready to be printed.

VAPIMWP_PRINT_PRINTHEADER

Specifies whether a print header appears at the top of the printed output. The print header consists of a left-justified file name and a right-justified page number followed by the page-length on the next line.

You can change the file name value by using the [VAPIMWP_PRINT_SETPRINTNAME](#) message. This is a parameter of the `VAPIM_PRINT` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_PRINT, VAPIMWP_PRINT_PRINTHEADER,  
            (LPARAM) (BOOL) bPrintHeaders );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>bPrintHeaders</code>	If <code>TRUE</code> , the document is printed with a header. If <code>FALSE</code> , the document is printed without a header.

Returns

`SendMessage()` returns `TRUE` if the call succeeds.

VAPIMWP_PRINT_PRINTSETUP

Opens a standard **Print Setup** dialog box. This is a parameter of the `VAPIM_PRINT` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_PRINT, VAPIMWP_PRINT_PRINTSETUP, 0L );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Allows the user to select general printing options, including the printer, page size, and page orientation.

VAPIMWP_PRINT_PRINTTOPD

Sets the standard Windows print options for printing files. This is a parameter of the VAPIM_PRINT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_PRINT, VAPIMWP_PRINT_PRINTTOPD,  
            (LPARAM) (PRINTDLG*) lpPD );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpPD	A pointer to a Windows PRINTDLG structure.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

VAPIMWP_PRINT_PRINTTOPRINTER

Prints the document to the specified printer. This is a parameter of the VAPIM_PRINT message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_PRINT, VAPIMWP_PRINT_PRINTTOPRINTER,  
            (LPARAM) (LPCSTR*) lpzPrinterDriver );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpszPrinterDriver	A string in the operating system code page that is the name of the printer driver, or NULL for the default printer. This string must be of the form <code>printername, printerdevice, printerport</code> . For example: <code>\\Calculus\HP LaserJet IIISi,winspool,NE00:</code>

Returns

`SendMessage()` returns `TRUE` if the call succeeds; otherwise it returns `FALSE`.

Discussion

- This message prints to the specified printer without calling the common **Print** dialog box. You can use this message to implement a **Print** menu item or toolbar button.
- When printing in an application that is a Windows service, a default printer must be installed for the user account using the application.
- To make sure that the entire document is opened before the document is printed, open the document with the `bwait` member in the [TPVAPIOpenDocumentInfo](#) structure set to `TRUE`. Use the [VAPIMWP_PRINT_CANPRINT](#) message to determine whether the document has been completely processed and is ready to be printed.
- The `printerdevice` in `lpszPrinterDriver` must be `WINSPOOL`, because `WINSPOOL` is the output device used by Microsoft Windows for all printers.

VAPIMWP_PRINT_SETPRINTNAME

Used in conjunction with [VAPIMWP_PRINT_PRINTHEADER](#), this message replaces the default file name field of the print header with a specified string. This is a parameter of the `VAPIM_PRINT` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_PRINT, VAPIMWP_PRINT_SETPRINTNAME,  
            (LPARAM) (char*) szPrintName );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
szPrintName	A string used to replace the file name field of the print header.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

VAPIMWP_VIEW_CANASPECTRATIO

Determines whether the document supports an aspect ratio. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANASPECTRATIO,  
            (LPARAM) (BOOL*) lpbCanAspectRatio );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanAspectRatio	A pointer to a flag that returns TRUE or FALSE, depending on whether the document supports an aspect ratio.

Returns

- SendMessage() returns TRUE if the call succeeds, in which case lpbCanAspectRatio returns TRUE or FALSE.
- SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case lpbCanAspectRatio is undefined.

Discussion

Use this message to control the state of an **Aspect Ratio** menu item or toolbar button.

VAPIMWP_VIEW_CANDECREASEFONT

Determines whether the document font can be decreased. If `lpbCanDecreaseFont` is `TRUE`, the font size can be decreased by using `VAPIMWP_VIEW_DECREASEFONT`. This is a parameter of the `VAPIM_VIEW` message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANDECREASEFONT,
            (LPARAM) (BOOL*) lpbCanDecreaseFont );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>lpbCanDecreaseFont</code>	A pointer to a flag that returns <code>TRUE</code> or <code>FALSE</code> , depending on whether the document font size can be decreased.

Returns

- `SendMessage()` returns `TRUE` if the call succeeds, in which case `lpbCanDecreaseFont` returns `TRUE` or `FALSE`.
- `SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanDecreaseFont` is undefined.

Discussion

Use this message to control the state of a **Decrease Font** menu item or toolbar button.

VAPIMWP_VIEW_CANFITTOWINDOW

Determines whether the document can be magnified to fit the document selection to the window. This is a parameter of the `VAPIM_VIEW` message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANFITTOWINDOW,
            (LPARAM) (BOOL*) lpbCanFitToWindow );
```


Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanFitToWindow	A pointer to a flag that returns TRUE or FALSE, depending on whether the document has a selection, and whether the document can be magnified to fit the selection to the window.

Returns

- `SendMessage()` returns TRUE if the call succeeds, in which case `lpbCanFitToWindow` returns TRUE or FALSE.
- `SendMessage()` returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanFitToWindow` is undefined.

Discussion

Use this message to control the state of a **Magnify** menu item or toolbar button.

VAPIMWP_VIEW_CANGOTO

Determines whether the document can go to a specified page or slide. This is a parameter of the `VAPIM_VIEW` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANGOTO,  
            (LPARAM) (BOOL*) lpbCanGoTo );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanGoTo	A pointer to a flag that returns TRUE or FALSE, depending on whether the document can go to a specified page or slide.

Returns

`SendMessage()` returns TRUE if the call succeeds, in which case `lpbCanGoTo` returns TRUE or FALSE.

Discussion

- Use this message to control the state of a **Go To Page** menu item or toolbar button.
- To implement a **Go To Page** menu item or toolbar button, use the [VAPIMWP_VIEW_GOTOPAGE](#) message.
- This message can be used only with word processing and presentation files.

VAPIMWP_VIEW_CANGRIDLINES

Determines whether the document supports gridlines. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANGRIDLINES,  
            (LPARAM) (BOOL*) lpbCanGridlines );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanGridlines	A pointer to a flag that returns TRUE or FALSE, depending on whether the document supports gridlines.

Returns

- `SendMessage()` returns TRUE if the call succeeds, in which case `lpbCanGridlines` returns TRUE or FALSE.
- `SendMessage()` returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanGridlines` is undefined.

Discussion

Use this message to control the state of a **Toggle Gridlines** menu item or toolbar button.

VAPIMWP_VIEW_CANINCREASEFONT

Determines whether the document font can be increased. If `lpbCanIncreaseFont` is TRUE, the font size can be increased by using the [VAPIMWP_VIEW_INCREASEFONT](#) message. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANINCREASEFONT,  
            (LPARAM) (BOOL*) lpbCanIncreaseFont );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanIncreaseFont	A pointer to a flag that returns TRUE or FALSE, depending on whether the document font size can be increased.

Returns

- SendMessage() returns TRUE if the call succeeds, in which case lpbCanIncreaseFont returns TRUE or FALSE.
- SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case lpbCanIncreaseFont is undefined.

Discussion

Use this message to control the state of an **Increase Font** menu item or toolbar button.

VAPIMWP_VIEW_CANINVERT

Determines whether the document colors can be inverted. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANINVERT,  
            (LPARAM) (BOOL*) lpbCanInvert );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Argument	Description
lpbCanInvert	A pointer to a flag that returns TRUE or FALSE, depending on whether the document colors can be inverted (for example, from black to white and white to black).

Returns

- `SendMessage()` returns TRUE if the call succeeds, in which case `lpbCanInvert` returns TRUE or FALSE.
- `SendMessage()` returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanInvert` is undefined.

Discussion

Use this message to control the state of an **Invert** menu item or toolbar button.

VAPIMWP_VIEW_CANLAYOUT

Determines whether the document layout can be changed. This is a parameter of the `VAPIM_VIEW` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANLAYOUT,  
            (LPARAM) (BOOL*) lpbCanLayout );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanLayout	A pointer to a flag that returns TRUE or FALSE, depending on whether the document layout can be changed.

Returns

- `SendMessage()` returns TRUE if the call succeeds, in which case `lpbCanLayout` returns TRUE or FALSE.
- `SendMessage()` returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanLayout` is undefined.

Discussion

Use this message to control the state of a **Wrap to Window**, **Page Layout**, or **Window Width** menu item or toolbar button.

VAPIMWP_VIEW_CANMAGNIFY

Determines whether the document can be magnified. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANMAGNIFY,  
            (LPARAM) (BOOL*) lpbCanMagnify );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanMagnify	A pointer to a flag that returns TRUE or FALSE, depending on whether the document can be magnified.

Returns

- SendMessage() returns TRUE if the call succeeds, in which case lpbCanMagnify returns TRUE or FALSE.
- SendMessage() returns FALSE if the call (for example, if there are invalid arguments or if no document is open), in which case lpbCanMagnify is undefined.

Discussion

Use this message to control the state of a **Magnify** menu item or toolbar button.

VAPIMWP_VIEW_CANPAUSE

Determines whether the multimedia document can be paused. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANPAUSE,  
            (LPARAM) (BOOL*) lpbCanPause );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanPause	A pointer to a flag that returns TRUE or FALSE, depending on whether the multimedia document can be paused.

Returns

- SendMessage() returns TRUE if the call succeeds, in which case lpbCanPause returns TRUE or FALSE.
- SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case lpbCanPause is undefined.

Discussion

Use this message to control the state of a **Pause** menu item or toolbar button.

VAPIMWP_VIEW_CANPLAY

Determines whether the multimedia document can be played. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANPLAY,  
            (LPARAM) (BOOL*) lpbCanPlay );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Argument	Description
lpbCanPlay	A pointer to a flag that returns TRUE or FALSE, depending on whether the multimedia document can be played.

Returns

- SendMessage() returns TRUE if the call succeeds, in which case lpbCanPlay returns TRUE or FALSE.
- SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case lpbCanPlay is undefined.

Discussion

Use this message to control the state of a **Play** menu item or toolbar button.

VAPIMWP_VIEW_CANPREVIEWPANE

Determines whether a file can be viewed in a preview pane. The message indicates TRUE only when container formats such as ZIP, TAR, or PST files are viewed.

To determine whether the preview pane is being used, use the VAPIMWP_VIEW_GETPREVIEWPANE message.

To specify if the preview pane should be used, use the VAPIMWP_VIEW_SETPREVIEWPANE message.

This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANPREVIEWPANE,
            (LPARAM) (BOOL*) lpbCanPreviewPane );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanPreviewPane	A pointer to a flag that returns TRUE or FALSE, depending on whether the document can be viewed in a preview pane. Only container files use the preview pane.

Returns

- `SendMessage()` returns `TRUE` if the call succeeds, in which case `lpbCanPreviewPane` returns `TRUE` or `FALSE`.
- `SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanPreviewPane` is undefined.

VAPIMWP_VIEW_CANROTATE

Determines whether the document can be rotated. This is a parameter of the `VAPIM_VIEW` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANROTATE,  
            (LPARAM) (BOOL*) lpbCanRotate );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>lpbCanRotate</code>	A pointer to a flag that returns <code>TRUE</code> or <code>FALSE</code> , depending on whether the document can be rotated.

Returns

- `SendMessage()` returns `TRUE` if the call succeeds, in which case `lpbCanRotate` returns `TRUE` or `FALSE`.
- `SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbCanRotate` is undefined.

Discussion

Use this message to control the state of a **Rotate** menu item or toolbar button.

VAPIMWP_VIEW_CANSTOP

Determines whether the playing of the multimedia document can be stopped. This is a parameter of the `VAPIM_VIEW` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_CANSTOP,  
            (LPARAM) (BOOL*) lpbCanStop );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbCanStop	A pointer to a flag that returns TRUE or FALSE, depending on whether the multimedia document can be stopped.

Returns

- SendMessage() returns TRUE if the call succeeds, in which case lpbCanStop returns TRUE or FALSE.
- SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case lpbCanStop is undefined.

Discussion

Use this message to control the state of a **Stop** menu item or toolbar button.

VAPIMWP_VIEW_DECREASEFONT

Decreases the document font size. Use the [VAPIMWP_VIEW_CANDECREASEFONT](#) message to determine whether the font size can be decreased. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_DECREASEFONT, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement a **Decrease Font** menu item or toolbar button.

VAPIMWP_VIEW_END

Sets the play mode of a multimedia document to stop at the end after playing. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_END, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement a **Stop At End** menu item or toolbar button.

VAPIMWP_VIEW_GETASPECTRATIO

Gets the aspect ratio of a document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_GETASPECTRATIO,  
            (LPARAM) (int*) lpnAspectRatio );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpnAspectRatio	A pointer to an integer that returns the aspect ratio: 0 - None 1 - Based on document 2 - Normal (use scanlines) 3 - Letter (times 2)

Returns

- `SendMessage()` returns `TRUE` if the call succeeds, in which case `lpnAspectRatio` returns the aspect ratio.
- `SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpnAspectRatio` is undefined.

Discussion

Use this message to set the state of an **Aspect Ratio** menu item or toolbar button.

VAPIMWP_VIEW_GETGRIDLINES

Gets the gridlines state of the document. This is a parameter of the `VAPIM_VIEW` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_GETGRIDLINES,  
            (LPARAM) (BOOL*) lpbGridlines );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpbGridlines	A pointer to a flag that returns <code>TRUE</code> or <code>FALSE</code> , depending on whether the document has gridlines set.

Returns

- `SendMessage()` returns `TRUE` if the call succeeds, in which case `lpbGridlines` returns `TRUE` or `FALSE`.
- `SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbGridlines` is undefined.

Discussion

Use this message to set the state of a **Toggle Gridlines** menu item or toolbar button.

VAPIMWP_VIEW_GETINVERT

Gets the invert state of the document. This is a parameter of the `VAPIM_VIEW` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_GETINVERT,  
            (LPARAM) (BOOL*) lpbInverted );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>lpbInverted</code>	A pointer to a flag that returns <code>TRUE</code> or <code>FALSE</code> , depending on whether the document colors are inverted (for example, from black to white and white to black).

Returns

`SendMessage()` returns `TRUE` if the call succeeds, in which case `lpbInverted` returns `TRUE` or `FALSE`.

`SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbInverted` is undefined.

Discussion

Use this message to set the state of an **Invert** menu item or toolbar button.

VAPIMWP_VIEW_GETLAYOUT

Gets the layout of the document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_GETLAYOUT,
            (LPARAM) (long*) lpLayout );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpLayout	A pointer to a long integer that returns the document layout: <ul style="list-style-type: none">• LOWORD(*lpLayout) 0 – Wrap to window.• LOWORD(*lpLayout) 1 – Page layout.• HIWORD(*lpLayout) 0 – Scale page to window width.• HIWORD(*lpLayout) <i>n</i> – Scale page to custom percentage.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to set the state of a **Wrap to Window**, **Page Layout**, or **Window Width** menu item or toolbar button.

VAPIMWP_VIEW_GETMAGNIFY

Gets the magnification of the document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_GETMAGNIFY,
            (LPARAM) (int*) lpnMagnify );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpnMagnify	A pointer to an integer that returns the document magnification: <ul style="list-style-type: none">• 0 – Custom• -1 – Page width• -2 – Page size• -3 – Fit selection to window

Returns

- `SendMessage()` returns `TRUE` if the call succeeds, in which case `lpnMagnify` returns the magnification.
- `SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpnMagnify` is undefined.

Discussion

Use this message to set the state of a **Magnify** menu item or toolbar button.

VAPIMWP_VIEW_GETPLAYMODE

Gets the play mode of a multimedia document. This is a parameter of the `VAPIM_VIEW` message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_GETPLAYMODE,
            (LPARAM) (int*) lpnPlayMode );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpnPlayMode	A pointer to an integer that returns the play mode of the multimedia document: 0 for stop at end; 1 for loop at end.

Returns

- `SendMessage()` returns `TRUE` if the call succeeds, in which case `lpnPlayMode` returns the play mode.
- `SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpnPlayMode` is undefined.

Discussion

Use this message to check or press a **Stop At End** or **Loop At End** menu item or toolbar button.

VAPIMWP_VIEW_GETPREVIEWPANE

Determines whether the preview pane is being used. The preview pane is only used to display a subfile in a container file. When it is enabled, the viewing area is divided into two panes: one pane displays the contents of the container file, the other displays the contents of the selected subfile. For more information, see [VAPIMWP_VIEW_SETPREVIEWPANE](#), on page 143. This is a parameter of the `VAPIM_VIEW` message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_GETPREVIEWPANE,  
            (LPARAM) (BOOL*) lpbPreviewPane );
```

Arguments

Argument	Description
<code>hWndVAPI</code>	The handle of the VAPI window.
<code>lpbPreviewPane</code>	A pointer to a flag that returns <code>TRUE</code> or <code>FALSE</code> , depending on whether the preview pane was returned. Only container files use the preview pane.

Returns

- `SendMessage()` returns `TRUE` if the call succeeds, in which case `lpbPreviewPane` returns `TRUE` or `FALSE`.
- `SendMessage()` returns `FALSE` if the call fails (for example, if there are invalid arguments or if no document is open), in which case `lpbPreviewPane` is undefined.

VAPIMWP_VIEW_GETROTATE

Gets the rotation of the document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_GETROTATE,
            (LPARAM) (int*) lpnRotate );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lpnRotate	A pointer to an integer that returns the rotation in degrees.

Returns

- SendMessage() returns TRUE if the call succeeds, in which case lpnRotate returns the rotation.
- SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case lpnRotate is undefined.

Discussion

Use this message to set the state of a **Rotate** menu item or toolbar button.

VAPIMWP_VIEW_GOTOPAGE

Goes to a specified page or slide in a document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_GOTOPAGE,
            (LPARAM) (int) nPage );
```


Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
nPage	A zero-based integer that is the page or slide number you want to go to.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

- Use this message to implement a **Go To Page** menu item or toolbar button.
- The [VAPIMWP_VIEW_CANGOTO](#) message determines whether you can go to specific pages or slides in a document.
- You can use this message only with PPT files or the graphic-based PDF reader (see [Use the kppdfldr Reader, on page 50](#)). To use similar functionality with word processing files, see [VAPIM_GOTO_PAGE](#) , on page 74.

VAPIMWP_VIEW_INCREASEFONT

Increases the document font size. Use [VAPIMWP_VIEW_CANINCREASEFONT](#) to determine whether the font size can be increased. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_INCREASEFONT, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement an **Increase Font** menu item or toolbar button.

VAPIMWP_VIEW_LOOP

Sets the play mode of a multimedia document to loop at the end after playing. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_LOOP, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement a **Loop At End** menu item or toolbar button.

VAPIMWP_VIEW_PAUSE

Pauses the playing of a multimedia document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_PAUSE, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement a **Pause** menu item or toolbar button.

VAPIMWP_VIEW_PLAY

Plays a multimedia document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_PLAY, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

You can use this message to implement a **Play** menu item or toolbar button.

VAPIMWP_VIEW_SETASPECTRATIO

Sets the aspect ratio of a document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_SETASPECTRATIO,  
            (LPARAM) (int) nAspectRatio );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
nAspectRatio	An integer that is the aspect ratio: <ul style="list-style-type: none">• 0 - None• 1 - Based on document• 2 - Normal (use scanlines)• 3 - Letter (times 2)

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement an **Aspect Ratio** menu item or toolbar button.

VAPIMWP_VIEW_SETGRIDLINES

Sets the gridlines state of the document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_SETGRIDLINES,  
            (LPARAM) (BOOL) bGridlines );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
bGridlines	A flag that is TRUE or FALSE to enable or disable the document gridlines.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement a **Toggle Gridlines** menu item or toolbar button.

VAPIMWP_VIEW_SETINVERT

Sets the invert state of the document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_SETINVERT, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

The message toggles the current invert state of the document. You can use this message to implement an **Invert** menu item or toolbar button.

VAPIMWP_VIEW_SETLAYOUT

Sets the layout of the document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_SETLAYOUT,  
            (LPARAM) (long) lLayout );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
lLayout	A long integer that is the document layout: <ul style="list-style-type: none">• LOWORD(lLayout) 0 Wrap to window.• LOWORD(lLayout) 1 Page layout.• HIWORD(lLayout) 0 Scale page to window width.• HIWORD(lLayout) n Scale page to custom percentage.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement a **Wrap to Window**, **Page Layout**, or **Window Width** menu item or toolbar button.

VAPIMWP_VIEW_SETMAGNIFY

Sets the magnification of the document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_SETMAGNIFY,  
            (LPARAM) (int) nMagnify );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
nMagnify	An integer that is the document magnification: <ul style="list-style-type: none">• n – Custom Percentage Value• -1 – Page width• -2 – Page size

Argument	Description
	<ul style="list-style-type: none">• -3 – Fit selection to window

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement a **Magnify** menu item or toolbar button.

The range of acceptable values is 10 to 400 percent.

VAPIMWP_VIEW_SETPREVIEWPANE

Specifies whether the preview pane is used to display a subfile in a container file. When the preview pane is enabled, the viewing area is divided into two panes: one pane displays the contents of the container file, the other displays the contents of the selected subfile. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_SETPREVIEWPANE,  
            (LPARAM) (BOOL) bNewValue );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
bNewValue	A flag that returns TRUE or FALSE, depending on whether the preview pane was set. Only container files use the preview pane.

Returns

- SendMessage() returns TRUE if the call succeeds, in which case bNewValue returns TRUE or FALSE.
- SendMessage() returns FALSE if the call fails (for example, if there are invalid arguments or if no document is open), in which case bNewValue is undefined.

VAPIMWP_VIEW_SETROTATE

Sets the rotation of the document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_SETROTATE,  
            (LPARAM) (int) nRotate );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.
nRotate	An integer that is the rotation in degrees.

Returns

SendMessage() returns TRUE if the call succeeds; otherwise it returns FALSE.

Discussion

Use this message to implement a **Rotate** menu item or toolbar button.

VAPIMWP_VIEW_STOP

Stops the playing of a multimedia document. This is a parameter of the VAPIM_VIEW message.

Syntax

```
#include <kvvapi.h>  
SendMessage(hWndVAPI, VAPIM_VIEW, VAPIMWP_VIEW_STOP, 0L );
```

Arguments

Argument	Description
hWndVAPI	The handle of the VAPI window.

Returns

`SendMessage()` returns `TRUE` if the call succeeds; otherwise it returns `FALSE`.

Discussion

Use this message to implement a **Stop** menu item or toolbar button.

Chapter 6: Notification Message Parameters

This section provides information on the notification message parameters in the Viewing API. It includes the following topics:

- [VAPINM_ANNOTATION_HIT](#) 146
- [VAPINM_EXTENT](#) 147
- [VAPINM_SELECTION](#) 148
- [VAPINM_TEXTBUFFER](#) 148
- [VAPINM_USERCLICK](#) 150
- [VAPINM_VIEW_FILE](#) 150
- [VAPINMWP_INIT_DISABLEUI](#) 151
- [VAPINMWP_INIT_DOCTYPE](#) 152
- [VAPINMWP_INIT_GETTEMPFILEPATH](#) 152
- [VAPINMWP_INIT_OPENDOCDONE](#) 153
- [VAPINMWP_INIT_PAGENUMBER](#) 154
- [VAPINMWP_MULTIOBJ_OBJNAME](#) 154
- [VAPINMWP_OPTIONS_GETOPTIONS_EX](#) 155
- [VAPINMWP_PRINT_PRINTDONE](#) 156

VAPINM_ANNOTATION_HIT

Description

Reports an annotation hit when the user clicks on an annotation.

Syntax

```
#include <kvvapi.h>
VAPINM_ANNOTATION_HIT = uMsg;
BOOL bDoubleClick = (BOOL) wParam;
long lLogicalAddress = (long) lParam;
```

Arguments

Argument	Description
bDoubleClick	A flag that is TRUE if the user double-clicked; FALSE if the user single-clicked.
lLogicalAddress	A long integer that is the logical address of the annotation.

Returns

None

Discussion

The logical address of the annotation is the same as that specified in the `VAPIM_ANNOTATE` message.

VAPINM_EXTENT

Description

Reports that the user changed the view extent.

Syntax

```
#include <kvvapi.h>
VAPINM_EXTENT = uMsg;
(wParam is not used)
TPVAPIFirstLast lpFirstLast = (TPVAPIFirstLast*) lParam;
```

Arguments

Argument	Description
<code>lpFirstLast</code>	A pointer to a TPVAPIFirstLast structure that defines the view extent.

Returns

None

Discussion

This message is received to report the first and last logical addresses that are currently displayed. This message is generated when the user scrolls the document or resizes the client area. This message is not generated until a `SB_ENDSCROLL` or `WM_KEYUP` message is received in the case of a scroll.

VAPINM_SELECTION

Description

Reports that the user changed the selection state.

Syntax

```
#include <kvvapi.h>  
VAPINM_SELECTION = uMsg;  
BOOL bHaveSelection = (BOOL) wParam;  
TPVAPIFirstLast lpFirstLast = (TPVAPIFirstLast*) lParam;
```

Arguments

Argument	Description
bHaveSelection	A flag that is TRUE if a selection exists; FALSE if a selection does not exist.
lpFirstLast	A pointer to a TPVAPIFirstLast structure that defines the selection, if a selection exists. It is undefined if a selection does not exist.

Returns

None

Discussion

The lpFirstLast parameter is not valid if the bHaveSelection parameter is FALSE.

VAPINM_TEXTBUFFER

Description

Returns a text buffer.

Syntax

```
#include <kvvapi.h>  
VAPINM_TEXTBUFFER = uMsg;  
(wParam is unused).  
TPVAPITextInfo lpTextInfo = (TPVAPITextInfo*) lParam;
```

Arguments

Argument	Description
lpTextInfo	Pointer to a TPVAPITextInfo structure that defines the text buffer.

Returns

None

Discussion

- The `lpTextInfo->cbText` parameter is the number of bytes of text in the buffer. Typically, the buffer is approximately 4 KB, but might be larger. It cannot exceed 10 KB.

Text buffers are usually created at an even boundary, such as the end of a paragraph, table row, or page column. However, if a table row or page column contains a large amount of text, it might be split across text buffers to make sure that `lpTextInfo->cbText` does not exceed 10 K. Individual words are never split across buffers. If `lpTextInfo` is NULL, the end of the document is reached.

- The `lpTextInfo->lpText` parameter is a pointer to the buffer of characters. Typically, the buffer is in the Windows ANSI character set; however, the Viewing API allows the user to select either the OEM or ANSI character set for text files. Depending on your integration of Viewing, this might or might not be an issue.

The buffer is zero-terminated. The terminator is not counted in the byte count. The text is an allocated buffer returned to the system upon return of this message. Therefore, you can write within this buffer if it is convenient.

Embedded control codes exist as follows:

`KV_EOP 0x01` End of paragraph.

`KV_EOC 0x02` End of cell.

`KV_PIC 0x03` Picture exists at this logical address.

- To form a logical address from a TEXTBUFFER message, take the base address and add to it the number of BYTES from the start of the text buffer (`lpText`) that the base address references.

For example, the following TEXTBUFFER messages might occur in a document:

```
{ 0, 1000, xxxx } { 1000, 2300, yyyy } { 3300, 1000, zzzz }
```

In this case, addresses 0 through 999 exist in the first TEXTBUFFER, addresses 1000 through 3299 in the second buffer, and 3300 through 4299 in the third buffer.

VAPINM_USERCLICK

Description

Reports that the user clicked the mouse on the document.

Syntax

```
#include <kvvapi.h>  
VAPINM_USERCLICK = uMsg;  
BOOL bDoubleClick = (BOOL) wParam;  
long lLogicalAddress = (long) lParam;
```

Arguments

Argument	Description
bDoubleClick	A flag that is TRUE if the user double-clicked; FALSE if the user single-clicked.
lLogicalAddress	A long integer that is the logical address of the mouse click.

Returns

None

Discussion

You can use the positional information to insert an annotation. This message is generated by a WM_LBUTTONDOWN or WM_LBUTTONDBLCLK.

NOTE: This message is not sent when the user has the Shift key depressed, because this indicates the selection is to be extended. In fact, this causes a VAPINM_SELECTION message to be sent.

VAPINM_VIEW_FILE

Description

Specifies a file that should be viewed. For example, this notification message is generated when the user double-clicks a subfile in a container file displayed in VAPI. This message is also generated when the user clicks a link to a local file from within an HTML file displayed in VAPI.

Syntax

```
#include <kvvapi.h>
VAPINM_VIEW_FILE = uMsg;
VAPINMWP_VIEW_KEEFILE or VAPINMWP_VIEW_DELETEFILE = wParam;
char* lpszFileName = (char*) lParam;
```

Arguments

Argument	Description
lpszFileName	A pointer to the complete file specification, including the path, of a file that should be viewed. If the wParam is VAPINMWP_VIEW_DELETEFILE, the file should be deleted after it is viewed (the file is a temporary file).

Returns

Returns TRUE if the message is processed.

VAPINMWP_INIT_DISABLEUI

Description

Determines whether the user interface should be disabled. This is a parameter of the VAPINM_INIT notification message.

Syntax

```
#include <kvvapi.h>
VAPINM_INIT = uMsg;
VAPINMWP_INIT_DISABLEUI = wParam;
(lParam is not used)
```

Returns

Returns TRUE to disable the user interface; otherwise it returns FALSE. The default is FALSE.

Discussion

This message is received when a Viewer asks VAPI if the user interface is disabled. The Viewer does this before a user interface action, such as creating a dialog box.

VAPINMWP_INIT_DOCTYPE

Description

This notification message is received during the opening of a file that cannot be opened. It indicates the document's format. If the document is successfully opened, use VAPIMWP_INIT_GETDESCRIP to obtain a description of the document's format. This is a parameter of the VAPINM_INIT notification message.

Syntax

```
#include <kvvapi.h>  
VAPINM_INIT = uMsg;  
VAPINMWP_INIT_DOCTYPE = wParam;  
char *lpzDescription = (char *) lParam;
```

Returns

The return value is ignored. The pointer to the lpzDescription becomes invalid after returning from this message.

VAPINMWP_INIT_GETTEMPFILEPATH

Description

Asks for a temporary file path for VAPI to use. This is a parameter of the VAPINM_INIT notification message.

Syntax

```
#include <kvvapi.h>  
VAPINM_INIT = uMsg;  
VAPINMWP_INIT_GETTEMPFILEPATH = wParam;  
char* lpstzTempFilePath = (char*) lParam;
```

Arguments

Argument	Description
lpstzTempFilePath	A pointer to a Pascal string that returns the temporary file path string as a C string.

Returns

Returns TRUE if the `lpstzTempFilePath` string was set; otherwise it returns FALSE.

Discussion

This message is received when VAPI converts an I/O object to a file during a **Save As** operation. If the temporary file path is not set, VAPI creates one.

VAPINMWP_INIT_OPENDOCDONE

Description

Reports the status of the document open process. This is a parameter of the VAPINM_INIT notification message.

Syntax

```
#include <kvvapi.h>
VAPINM_INIT = uMsg;
VAPINMWP_INIT_OPENDOCDONE = wParam;
int nPercentDone = (int) lParam;
```

Arguments

Argument	Description
nPercentDone	The percentage done of the document open process.

Returns

None

Discussion

- This message is received during and after the processing of the VAPIMWP_INIT_OPEN_DOCUMENT message. Multiple messages might be received, with increasing values of percentage done. The document is open when the percentage done is $\geq 100\%$.
- A negative value of nPercentDone indicates that an error occurred during the processing of the document.

VAPINMWP_INIT_PAGENUMBER

Description

Reports the current page number of the document. This is a parameter of the VAPINM_INIT notification message.

Syntax

```
#include <kvvapi.h>
VAPINM_INIT = uMsg;
VAPINMWP_INIT_PAGENUMBER = wParam;
int nCurrentPage = (int) LOWORD(lParam);
int cTotalPages = (int) HIWORD(lParam);
```

Arguments

Argument	Description
nCurrentPage	The current page number of the document.
cTotalPages	The total number of pages in the document.

Returns

None

Discussion

This message is received after the document is opened and whenever the document page is changed.

VAPINMWP_MULTIOBJ_OBJNAME

Description

Reports the current object name of the document. This is a parameter of the VAPINM_MULTIOBJ notification message.

Syntax

```
#include <kvvapi.h>
VAPINM_MULTIOBJ = uMsg;
```

```
VAPINMWP_MULTIOBJ_OBJNAME = wParam;  
LPCSTR lpzObjectName = (LPCSTR) lParam;
```

Arguments

Argument	Description
lpzObjectName	The current object name of the document.

Returns

None

Discussion

This message is received after the document is opened and whenever the document object is changed.

VAPINMWP_OPTIONS_GETOPTIONS_EX

Description

Asks for the options for the current document. This is a parameter of the VAPINM_OPTIONS notification message.

Syntax

```
#include <kvvapi.h>  
VAPINM_OPTIONS = uMsg;  
VAPINMWP_OPTIONS_GETOPTIONS_EX = wParam;  
ALL_OPTIONS_EX* lpAllOptions = (ALL_OPTIONS_EX*) lParam;
```

Arguments

Argument	Description
lpAllOptions	A pointer to an ALL_OPTIONS_EX structure to get the document options.

Returns

Returns TRUE if the lpAllOptions structure was processed and initialized; otherwise it returns FALSE.

Discussion

This message is received during the document open process. VAPI initializes the Viewer before the Viewer opens the document. Therefore, this message is received before the VAPIMWP_INIT_OPEN_DOCUMENT or VAPIMWP_INIT_OPENDOCWAIT message returns.

VAPINMWP_PRINT_PRINTDONE

Description

Reports the status of a document that is being printed. This is a parameter of the VAPINM_PRINT notification message.

Syntax

```
#include <kvvapi.h>  
VAPINM_PRINT = uMsg;  
VAPINMWP_PRINT_PRINTDONE = wParam;  
long lStatus = (long) lParam;
```

Arguments

Argument	Description
lStatus	<ul style="list-style-type: none">• 1 - print successful• 2 - user cancelled operation

Returns

None

Chapter 7: Structures

This section describes the structures of the Viewing API. It includes the following topics:

- [ADDOCINFO](#)157
- [ALL_OPTIONS_EX](#) 158
- [KPTPIOobj](#)159
- [KVSumInfoElemEx](#) 160
- [KVSummaryInfoEx](#) 160
- [TPVAPIAnnotation](#)161
- [TPVAPIConvert](#) 162
- [TPVAPICreateParams](#) 163
- [TPVAPIDrawFileInfo](#) 164
- [TPVAPIDrawPageInfo](#)165
- [TPVAPIExtract](#) 166
- [TPVAPIFindInfo](#) 167
- [TPVAPIFirstLast](#) 168
- [TPVAPIGetText](#) 168
- [TPVAPIHiLiteColor](#) 169
- [TPVAPIHiLiteOptions](#) 170
- [TPVAPIOpenDocumentInfo](#)170
- [TPVAPIPageSize](#) 174
- [TPVAPIPosition](#) 175
- [TPVAPITextInfo](#) 176

ADDOCINFO

Description

This structure defines the parameters used by the [VAPIMWP_INIT_GETDOCFORMAT](#) message. It provides the format, file class, and version number of the source document. It is defined in `adinfo.h`.

Syntax

```
#include <adinfo.h>
typedef struct
{
    ENdocClass      eClass;
    ENdocFmt        eFormat;
```

```
    long          lVersion;  
    unsigned long ulAttributes;  
}  
ADDOCINFO, *ADDOCINFOPTR;
```

Members

eClass	The file class of the source document (for example, spreadsheet, word processor, or encapsulation format) as defined by the <code>ENDocClass</code> enumerated type.
eFormat	The major format of the source document (for example, Microsoft Word XML format, or Corel Presentation) as defined by the <code>ENDocFmt</code> enumerated type in <code>adinfo.h</code> . The <code>ENDocFmt</code> type provides a unique ID for each major format.
lVersion	The version number of the document format. The number is multiplied by 1,000, so, for example, 1.02 is represented by 1020.
ulAttributes	Other attributes of the document as defined by the <code>ENDocAttributes</code> enumerated type.

ALL_OPTIONS_EX

Description

This structure defines the document options. Document options control display elements such as window size, zoom settings, margin size, scaling, and revision tracking information. Options are defined for each file type category (for example, spreadsheets, multimedia, and word processing). See [Change Document Options, on page 37](#).

Syntax

```
#include <kwoption.h>  
typedef struct ALL_OPTIONS_EX_TAG  
{  
    int          size;  
    MMD_OPTIONS MMDOptions;  
    WPD_OPTIONS WPDOptions;  
    SSD_OPTIONS SSDOptions;  
    ASCII_OPTIONS ASCIIOptions;  
    IMAGE_OPTIONS IMGOptions;  
    GX_OPTIONS   GFXOptions;  
    FX_OPTIONS   FAXOptions;  
    GL_OPTIONS   GeneralOptions;  
    ARCHIVE_OPTIONS ArchiveOptions;  
    BOOL         SaveOptions;  
    char         szSectionName[ 16 ];
```

```
    char          szSectionTitle[ 16 ];  
    HTML_OPTIONS HTMLOptions;  
    PG_OPTIONS   PGOptions;  
}  
ALL_OPTIONS_EX;
```

See the `kwoption.h` file for a description of this structure.

Members

<code>size</code>	The size of the structure.
<code>MMOptions</code>	A pointer to the document options for multimedia files.
<code>WPOptions</code>	A pointer to the document options for word processing files.
<code>SSOptions</code>	A pointer to the document options for spreadsheet files.
<code>ASCIIOptions</code>	A pointer to the document options for ASCII files.
<code>IMGOptions</code>	A pointer to the document options for graphic files.
<code>GFXOptions</code>	A pointer to the document options for GFX files
<code>FAXOptions</code>	A pointer to the document options for FAX files.
<code>GeneralOptions</code>	A pointer to general options.
<code>ArchiveOptions</code>	A pointer to options that affect archive files.
<code>SaveOptions</code>	Currently not used.
<code>szSectionName[16]</code>	Currently not used.
<code>szSectionTitle[16]</code>	Currently not used.
<code>HTMLOptions</code>	A pointer to options that affect HTML files.
<code>PGOptions</code>	A pointer to options affecting presentation files.

KPTPIOobj

Description

This structure defines the I/O object.

Syntax

```
#include <kwkpiif.h>
```

See the `kvioobj.h` file for a description of this structure.

KVSumInfoElemEx

Description

This structure defines the individual metadata elements, and is defined in `kvtypes.h`.

Syntax

```
typedef struct tag_KVSumInfoElemEx
{
    int             isValid;
    KVSumInfoType  type;
    void           *data;
    char           *pcType;
}
KVSumInfoElemEx;
```

Members

- isValid** Specifies whether the data value is present in the document. The setting 1 specifies that the value is valid and exists.
- type** The data type of the metadata element. The types are defined in `KVSumInfoType` in `kvtypes.h`.
- data** The content of the metadata field.
- If the `type` member is `KV_Int4`, or `KV_Bool`, this member contains the actual value. Otherwise, this member is a pointer to the actual value.
- `KV_DateTime` and `KV_IEEE8` point to an 8-byte value.
- `KV_String` and `KV_Unicode` point to the beginning of the string containing the text. `KV_Unicode` is replaced with `KV_String` when the UNICODE value has been character mapped to the desired output character set.
- pcType** A pointer to the name (text description) of the metadata field.

KVSummaryInfoEx

Description

This structure defines the parameters used by the [VAPIM_GETSUMMARYINFO](#) message. It provides a count of the number of metadata elements, and a pointer to the first element of the array of individual elements. (Metadata is also referred to as document summary information.)

Syntax

```
#include <kvtypes.h>
typedef struct tag_KVSummaryInfoEx
{
    int                nElem;
    KVSUMINFOELEMEX   *pElem;
}
KVSummaryInfoEx;
```

Members

nElem The number of metadata elements contained in the array. This value is derived from the enumerated type **KVSUMTYPE**.

pElem Points to the first element of the array of document metadata elements defined by the [KVSUMINFOELEMEX](#) structure.

Discussion

- **nElem** might be zero. This indicates that the document did not contain metadata, such as an ASCII text document. If **nElem** is not zero, **nElem** is at least 42, and possibly more. This value is derived from the **KVSUMTYPE** enumerated type in **kvtypes.h**. The first 42 members of **pElem** are ordered according to the sort order of **KVSUMTYPE**. For example, **pElem[0]** is the code page of the document, and **pElem[10]** is the date the document was last printed.
- If **nElem** is equal or greater than 42, the returned value is a non-standard metadata field.

TPVAPIAnnotation

Description

This structure defines the parameters used by the [VAPIM_ANNOTATE](#) message.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIAnnotation
{
    long                position;
    ENVAPIAnnotationType type;
    HBITMAP             hBitmap;
    Int                 cbText;
    COLORREF            color;
```

```
        HCURSOR          hCursor;  
    }  
    TPVAPIAnnotation;
```

Members

Member	Description
position	A long integer. The position where the annotation applies. This is required.
type	The annotation type as defined in <code>ENVAPIAnnotationType</code> in <code>kvvapi.h</code> . This is required. The following options are available: <ul style="list-style-type: none">• <code>kvBitmap</code> – use a bitmap for the annotation.• <code>kvUnderline</code> – use an underline for the annotation.• <code>kvDottedUnderline</code> – use a dotted underline for the annotation. Currently not implemented.• <code>kvStrikeout</code> – use strikethrough as the annotation. Currently not implemented.
hBitmap	If the annotation type is bitmap, this is the handle of the bitmap.
cbText	If the annotation type is underline, this is the length of the underlined text.
color	If the annotation type is underlined, this is the <code>COLORREF</code> value of the underlined text.
hCursor	The handle of the cursor when the mouse hovers over the annotation.

TPVAPIConvert

Description

This structure defines the parameters used by the [VAPIM_CONVERT](#) message.

Syntax

```
#include <kvvapi.h>  
typedef struct tag_TPVAPIConvert  
{  
    LPSTR          lpszCode;  
    LPSTR          lpszTarget;  
}  
TPVAPIConvert;
```

Members

`lpszCode` The format code of the format to which to convert the document. The following options are available:

- `txt` — conversion to text format
- `rtf` — conversion to RTF format
- `htm` — conversion to HTML format

`lpszTarget` The target path and file name for the converted file.

TPVAPICreateParams

Description

This structure defines the parameters used to create the VAPI window. A VAPI window is created using the standard Windows API functions `CreateWindow()` or `CreateWindowEx()`. See [Create a Viewing API Window, on page 32](#).

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPICreateParams
{
    UINT            uProfileType;
    LPSTR           lpszRegistryName;
    LPSTR           lpszIniFileName;
    LPVOID          lpvViewerCreateParams;
    BOOL            bSendErrorNM;
    BOOL            bToolBar;
    LPVOID*         lp1pvKeyView;
}
TPVAPICreateParams;
```

Members

`uProfileType` Specifies whether initialization information is located in an initialization file or registry. This is optional. The following options are available:

- | | | |
|--------------------------------------|----------------|-------------------------|
| <code>PROFILEDF_USE_UNDEFINED</code> | <code>0</code> | Use default (Registry) |
| <code>PROFILEDF_USE_INI</code> | <code>1</code> | Use initialization file |
| <code>PROFILEDF_USE_REG</code> | <code>2</code> | Use Registry |

The default is PROFILEDF_USE_REG.

See [View Initialization Information, on page 22](#).

<code>lpszRegistryName</code>	If you are using the registry file to specify initialization information, this is the registry name of key under HKEY_LOCAL_MACHINE\SOFTWARE. This is optional. The default is VAPIDF_REGISTRY_NAME = [Autonomy\Keyview].
<code>lpszIniFileName</code>	If you are using an initialization file to specify initialization information, this is the file name (not the full path) of the initialization file. This is optional. The default is VAPIDF_INI_FILE_NAME = [KeyView.ini].
<code>lpvViewerCreateParams</code>	Viewer window create. This is optional. Use for custom Viewer.
<code>bSendErrorNM</code>	A flag to tell VAPI to send VAPINM_ERROR notification messages. This is optional. The default is TRUE.
<code>bToolBar</code>	Reserved. This must be FALSE.
<code>lpvpKeyView</code>	Reserved. This must be NULL.

TPVAPIDrawFileInfo

Description

This structure defines the parameters used by the [VAPIMWP_DRAW_DRAWTOFILE](#) message to draw a page to a graphic file.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIDrawFileInfo
{
    LPSTR    lpszTarget;
    LPSTR    lpszGfxOutput;
    UINT     uPageNumber;
    int      iWidth;
    int      iHeight;
    int      iPicXRes;
    int      iPicYRes;
    int      iCompressionQuality;
}
TPVAPIDrawFileInfo;
```

Members

`lpszTarget` The target path and file name to which the page is written. The file should

use one of the following four extensions:

.bmp
.jpg
.png
.tif

<code>lpszGfxOutput</code>	<p>The output graphics format. The following options are available:</p> <p><code>KVGFX_OUTPUT_BMP</code> <code>KVGFX_OUTPUT_JPEG</code> <code>KVGFX_OUTPUT_PNG</code> <code>KVGFX_OUTPUT_TIFF</code> (uncompressed TIFFs)</p>
<code>uPageNumber</code>	<p>The number of the page to be rasterized into a thumbnail.</p> <p>Page numbers start at 0. For example, set <code>uPageNumber</code> to 0 to draw page 1, and to 1 to draw page 2. For word processing documents, pages must be drawn sequentially. For example, to draw page 3, you must first draw pages 0 and 1.</p> <p>If the <code>bWait</code> member of <code>TPVAPIOpenDocumentInfo</code> is set to <code>FALSE</code>, you can draw any page. See TPVAPIOpenDocumentInfo, on page 170. If the page you request is beyond the last page, a <code>VAPI_RETURN_NO_PAGE</code> error is returned.</p>
<code>iWidth</code>	<p>The maximum picture width (in TWIPS).</p>
<code>iHeight</code>	<p>The maximum picture height (in TWIPS).</p>
<code>iPicXRes</code>	<p>The desired horizontal resolution (0 for default).</p>
<code>iPicYRes</code>	<p>The desired vertical resolution (0 for default).</p>
<code>iCompressionQuality</code>	<p>This parameter controls the output quality of graphics that support compression quality (for example, JPEG). The valid range is 0 to 100. A value of 0 means default quality (85 compression); 1 is the lowest quality (highest compression and therefore the smallest file size); 100 is the highest quality (no compression and therefore the largest file size).</p> <p>The default is 0.</p>

TPVAPIDrawPageInfo

Description

This structure defines the parameters used by the `VAPIMWP_DRAW_DRAWPAGE` message to draw a page into a device context.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIOpenDocumentInfo
{
    HDC                hdc;
    unsigned int       nPage;
    RECT               Rect;
    RECT               RectUsed;
    BOOL               bCentre;
    HBRUSH              hBrush;
}
TPVAPIOpenDocumentInfo;
```

Members

- hdc** The device context in which to draw the page.
- nPage** The number of the page to draw. Page numbers start at 0. For example, set `nPage` to 0 to draw page 1, and to 1 to draw page 2. For word processing documents, pages must be drawn sequentially. For example, to draw page 3, you must first draw pages 0 and 1.
- If the `bWait` member of `TPVAPIOpenDocumentInfo` is set to `FALSE`, you can draw any page. See [TPVAPIOpenDocumentInfo, on page 170](#). If the page you request is beyond the last page, a `VAPI_RETURN_NO_PAGE` error is returned.
- Rect** A rectangle in the device coordinates specifying where to draw the page.
- RectUsed** Returns the rectangle that the page was actually drawn in. `RectUsed` might be different than `Rect` because Viewing maintains the aspect ratio of the document.
- bCentre** Because `RectUsed` might be different than `Rect`, this flag specifies whether the page should be centered in `Rect`.
- hBrush** The handle of a brush used to paint the background when `RectUsed` is different than `Rect`. If this is `NULL`, the background is not filled.

TPVAPIOpenDocumentInfo

Description

This structure defines the parameters used by the `VAPIMWP_FILE_EXTRACT` message to extract subfiles from a container file.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIExtract
{
    LPSTR    szTargetDirectory;
    LPSTR    szPassWord;
    BOOL     bPreserveDirStructure;
    BOOL     bFailIfExists;
}
TPVAPIExtract;
```

Members

<code>szTargetDirectory</code>	The target directory to which to extract the subfiles.
<code>szPassWord</code>	Password required to open a password-protected subfile.
	NOTE: This member is obsolete.
<code>bPreserveDirStructure</code>	Specifies whether to preserve the directory structure.
<code>bFailIfExists</code>	Specifies whether to fail if the file already exists.

TPVAPIFindInfo

Description

This structure defines the parameters used by [VAPIMWP_EDIT_FIND](#) for a text search request.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIFindInfo
{
    LPSTR    lpszFindText;
    BOOL     bFindDown;
    BOOL     bMatchCase;
    BOOL     bWholeWordOnly;
}
TPVAPIFindInfo;
```

Members

<code>lpszFindText</code>	A string containing the text to find.
<code>bFindDown</code>	Specifies whether to search from the selected point in the document to the beginning or to the end of the document.
<code>bMatchCase</code>	Specifies whether to match the case of the search term.
<code>bWholeWordOnly</code>	Specifies whether to search for whole words or partial words. Currently not implemented.

TPVAPIFirstLast

Description

This structure defines a text extent or selection.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIFirstLast
{
    long        first;
    long        last;
}
TPVAPIFirstLast;
```

Members

<code>first</code>	A long integer that is the first logical address of the text.
<code>last</code>	A long integer that is the last logical address of the text.

TPVAPIGetText

Description

This structure defines the parameters used by the [VAPIM_GETTEXT](#) message to get a text buffer from a specified range.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIOGetText
{
    long        start;
    int         cbText;
    BYTE*      lpText;
}
TPVAPIOGetText;
```

Members

- start A long integer that is the starting logical address of the text.
- cbText An integer that is the number of bytes of text to copy.
- lpText A pointer to the byte buffer in which to return the text.

TPVAPIHiLiteColor

Description

This structure defines the highlight color used by the [VAPIM_ENABLEINDEX](#) message to mark an index hit.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIHiLiteColor
{
    COLORREF   foreground;
    COLORREF   background;
}
TPVAPIHiLiteColor;
```

Members

- foreground The COLORREF value that is the highlight foreground color.
- background The COLORREF value that is the highlight background color.

TPVAPIHiLiteOptions

Description

This structure defines the highlight color options used by the [VAPIM_SETHILITEOPTIONS](#) message.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIHiLiteOptions
{
    TPVAPIHiliteColor    hlColorRec;
    int                  TextFontSize;
    char                  TextFontName[LF_FACESIZE];
}
TPVAPIHiLiteOptions;
```

Members

HlColorRec The handle of a highlight color as described in the [TPVAPIHiliteColor](#) structure.

TextFontSize The font size in points.

TextFontName The font name.

TPVAPIOpenDocumentInfo

Description

This structure defines the parameters used by the [VAPIMWP_INIT_OPEN_DOCUMENT](#) message to open a document.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIOpenDocumentInfo
{
    int                size;
    LPSTR              lpszFilePath;
    KPTPIOobj*         lpIOobj;
    adDocDesc*         lpadDocDesc;
    BOOL               bADInstallViewer;
    int                nViewAsMode;
```

```
    LPSTR        lpszOrigFilePath;  
    Int         nFlags;  
    LPSTR        lpszResID;  
    LPSTR        lpszVMLFilePath;  
    Void*       lpOptions;  
    HGLOBAL      hGlobalMem;  
    DWORD       dwcbGlobalMem;  
    BOOL        bWait;  
    int         nGeneralTab;  
    BOOL        bAutoViewAsText;  
    LPSTR        lpszHighLight;  
    BOOL        bMatchHighLightCase;  
}  
TPVAPIOpenDocumentInfo;
```

Members

size	This parameter must be initialized to <code>sizeof(TPVAPIOpenDocumentInfo)</code> before calling <code>VAPIMWP_INIT_OPEN_DOCUMENT</code> .
lpszFilePath	A string containing the full file path of the document to open. This is not required if you are using <code>lpIOobj</code> or <code>hGlobalMem</code> .
lpIOobj	A pointer to a KPTPIOobj structure that contains the I/O object of the document to open. This is not required if you are using <code>lpszFilePath</code> or <code>hGlobalMem</code> .
lpadDocDesc	A pointer to an <code>adDocDesc</code> structure that contains the document format information. Set this to <code>NULL</code> .
bADInstallViewer	A flag to install Viewer if the document is not supported (optional).
nViewAsMode	A flag to display the document as formatted text, text, or hexadecimal (optional). The options are described in Options for nViewAsMode , on the next page .
lpszOrigFilePath	If <code>lpszFilePath</code> is not the original file path, this optional string contains the full path to the original file.
nFlags	A bit field that contains additional options for opening a document (optional). The options are described in Options for nFlags , on page 174 .
lpszResID	A string that contains the resource ID to use (optional).
lpszVMLFilePath	Reserved. Set this to <code>NULL</code> .
lpOptions	A pointer to an ALL_OPTIONS_EX structure to change the default options for the document (optional). See Change Document Options , on page 37 for more information.
hGlobalMem	A block of memory that contains the input file data (optional). If you use this

	member, <code>lpIOobj</code> and <code>lpzFilePath</code> must be <code>NULL</code> .
<code>dwcbGlobalMem</code>	The size in bytes of the memory specified by <code>hGlobalMem</code> . This is ignored if <code>hGlobalMem</code> is not used.
<code>bWait</code>	<p>Set to <code>TRUE</code> to make <code>SendMessage()</code> on open <i>not</i> return until the document is fully processed.</p> <p>This ensures that the document is fully processed before an operation (such as printing, converting, or searching) is performed, and is useful when you want to use an operation immediately after opening the document.</p> <p>If you are opening a document for viewing only, set this to <code>FALSE</code> so that the first page of the document can be viewed as soon as it is processed.</p> <p>If you are drawing a word processing document and only want to process it up to the specified page, set <code>bWait</code> to <code>FALSE</code>. When the specified page is reached, processing is paused. This setting minimizes delays.</p>
<code>nGeneralTab</code>	This parameter must be set to <code>0</code> .
<code>bAutoViewAsText</code>	<p>If you set this flag to <code>TRUE</code>, the document is automatically displayed as unformatted text when the document format cannot be determined or is not supported (optional).</p> <p>This member will be deprecated in a future release. To display an unknown or unsupported format as text or hexadecimal, set <code>nViewAsMode</code>, on the previous page to <code>VIEW_MODE_AUTO_TEXT</code> or <code>VIEW_MODE_AUTO_HEX</code>.</p>
<code>lpzHighlight</code>	A string that contains text to be highlighted if it is found in the document (optional). You can either search for an intact string or for individual words that might or might not be adjacent. To search for individual words, separate each word with <code>\t</code> , which indicates a tab.
<code>bMatchHighlightCase</code>	A flag to indicate if the text in <code>lpzHighlight</code> is to be matched case-sensitively (<code>TRUE</code> if it is case sensitive).

Discussion

- If the `bWait` parameter is set to `TRUE`, you can determine whether the document has been completely processed and is ready for an operation by using the appropriate "Can" messages, such as `VAPIMWP_CANCONVERT`, `VAPIMWP_FILE_CANSAVEAS`, and `VAPIMWP_PRINT_CANPRINT`.
- The `nViewAsMode` member can be one of the following options:

Options for `nViewAsMode`

Option	Description
<code>VIEW_MODE_NORMAL</code>	Displays the document as formatted text when the format can be determined and is supported. This is the default.

Options for nViewAsMode , continued

Option	Description
VIEW_ MODE_ TEXT	Displays each byte as ASCII when the format can be determined.
VIEW_ MODE_ HEX	Displays each byte as hexadecimal when the format can be determined.
VIEW_ MODE_ AUTO_ TEXT	Automatically displays each byte as ASCII when the format cannot be determined or is not supported. This option overrides the setting in bAutoViewAsText, on the previous page .
VIEW_ MODE_ AUTO_ HEX	Automatically displays each byte as hexadecimal when the format cannot be determined or is not supported. This option overrides the setting in bAutoViewAsText, on the previous page .

- For the nViewAsMode member, the VIEW_MODE_NORMAL, VIEW_MODE_TEXT, and VIEW_MODE_HEX options are mutually exclusive; and the VIEW_MODE_AUTO_TEXT and VIEW_MODE_AUTO_HEX options are mutually exclusive. This means you can set a maximum of two options at one time. For example, you can set nViewAsMode as:

```
nViewAsMode= VIEW_MODE_TEXT | VIEW_MODE_AUTO_HEX
```

This configuration results in the following behavior:

File characteristic	Behavior
The file format cannot be determined	The VAPI_RETURN_UNKNOWN_FORMAT message is returned, and the file is displayed as hexadecimal.
The file format can be determined, but is not supported	The VAPI_RETURN_NO_VIEWER message is returned, and the file is displayed as ASCII text.
The file format can be determined and is supported	The VAPI_RETURN_SUCCESS message is returned, and the file is displayed as ASCII text.

- The `nFlags` member can be one of the following options:

Options for `nFlags`

Option	Description
VAPIDF_FLAGS_OPEN_FORMAT_ONLY	<p>Opens a document to determine the document format, regardless of whether the document is supported for viewing. After the document is opened, you can then call the VAPIMWP_INIT_GETDOCFORMAT message to get the format information.</p> <p>This flag does not create a Viewer window.</p>
VAPIDF_FLAGS_OPEN_WITHOUT_VIEW	<p>Opens a document in a hidden viewer window. Use this flag to process a document (print, convert, and so on) without viewing. (You must also set the <code>bWait</code> member to <code>TRUE</code>.)</p> <p>For example, to print a document, set the <code>VAPIDF_FLAGS_OPEN_WITHOUT_VIEW</code> flag, and then send the <code>VAPIMWP_PRINT_PRINT</code> message.</p>
VAPIDF_FLAGS_OPEN_VAPI_ONLY	<p>Opens a document without viewing and returns format information with the notification message <code>VAPIMWP_INIT_DOCTYPE</code>. This flag does not create a Viewer window.</p>
VAPIDF_FLAGS_NO_UI	<p>Suppresses GUI elements that are not called explicitly.</p> <p>For example, if you set this flag and the document format is not supported, the "unsupported format" dialog box does not display. However, if you set this flag and send the VAPI message to request the SaveAs dialog box, the SaveAs dialog box displays.</p>
VAPIDF_FLAGS_INCL_REVISION_MARK	<p>Displays the deleted content, revision marks, and revision tracking information in a document. See View Deleted Items and Document Revision Marks, on page 41.</p>

TPVAPIPageSize

Description

This structure defines the parameters used by the [VAPIMWP_DRAW_GETPAGESIZE](#) message to get the size of a page.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIPageSize
```

```
{
    unsigned int    nPage;
    unsigned int    nWidth;
    unsigned int    nHeight;
}
TPVAPIPageSize;
```

Members

nPage	The page number.
nWidth	The default width of the page.
nHeight	The default height of the page.

TPVAPIPosition

Description

This structure defines the position of the viewing window, and is used by the [VAPIM_POSITION](#) message to position the document within the viewing window.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPIPosition
{
    long    first;
    long    last;
    long    position;
}
TPVAPIPosition;
```

Members

first	A long integer that returns the first visible logical address.
last	A long integer that returns the last visible logical address.
position	A long integer that is the position (logical address) to make visible.

TPVAPITextInfo

Description

This structure defines a text buffer. See [VAPINM_TEXTBUFFER](#), on page 148.

Syntax

```
#include <kvvapi.h>
typedef struct tag_TPVAPITextInfo
{
    long        lBaseAddress;
    int         cbText;
    BYTE*       lpText;
}
TPVAPITextInfo;
```

Members

- lBaseAddress** A long integer that returns the base logical address at the start of the buffer.
- cbText** An integer that returns the number of bytes in the buffer.
- lpText** A pointer to the byte buffer, which returns the pointer to the text buffer. The text buffer can contain text or commands. Commands are ANSI strings with the following extensions (embedded control codes):
- **KV_EOP** 0x01 End of paragraph.
 - **KV_EOC** 0x02 End of cell.
 - **KV_PIC** 0x03 Picture exists at this logical address.

Part III: Viewing ActiveX Control

This section provides procedural and reference information for the Viewing ActiveX control and includes the following chapters:

- [Use the Viewing ActiveX Control](#)
- [Control Sample Programs](#)
- [Control Methods](#)
- [Control Properties](#)
- [Control Events](#)

Chapter 8: Use the Viewing ActiveX Control

This section describes how to use the Viewing ActiveX control. It includes the following topics:

• Overview of the Viewing ActiveX Control	178
• Open and View a Document	179
• Save a Document	179
• Convert a Document	180
• Print a Document	180
• Determine the Document Format	181
• Extract Document Metadata	181
• Search for Text in a Document	181
• Copy a Selected Area of Text	182
• Copy all the Text in a Document	182
• Create a Thumbnail Image of a Document Page	182
• Filter a Document	183
• Highlight Text in a Document	183
• Annotate Text in a Document	183

Overview of the Viewing ActiveX Control

Viewing includes an ActiveX control that provides the same functionality of the Viewing API. This control is ideally suited to Visual Basic developers, although it can be used in other development environments that support controls.

NOTE: For information on using the Viewing ActiveX control in a .NET application, see [Develop .NET Applications](#), on page 26.

You can use the Viewing controls to create an application (or HTML page) to:

- Open and view a document.
- Draw a page of a word processing document, spreadsheet, or a picture into a supplied Device Context (HDC). This is useful for generating *thumbnail* views of documents.
- Print a document (or print a document without viewing it) to a specified printer or to the default printer.
- Allow viewed word processing and spreadsheet documents to be saved as RTF, HTML, or text. Also, you can save image formats to other supported image formats.
- Convert word processing and spreadsheet documents to text, RTF, or HTML without viewing them.

- View or extract subfiles from a container file, such as ZIP, TAR, or PST.
- View and manipulate a graphic (including rotate and magnify).
- Annotate documents with a bitmap or selected text. The View API includes annotation event notification for actions such as clicking and double-clicking, allowing for implementation of hyperlinks and pop-up text.
- Highlight all occurrences of a word in a document.
- Filter spreadsheets, presentation graphics, and documents to text. A cross-platform C API that provides text filtering is also available. Contact Micro Focus for information on KeyView Filter SDK.
- Determine the format of a document based on its contents rather than its file extension.

Open and View a Document

Because you must open a document before you can view, print, or save it, or perform any other operation on it, viewing a document means to open *and* view a document. It is possible, however, to open a document without viewing it, or in other words, to open a document with view mode disabled. In this mode, you can print or save the document without viewing it.

To open and view a document, call the `Open` method. See [Open, on page 199](#) for a description of the possible return codes.

For example:

```
nRet = KeyView1.Open("c:\docs\bigtree.jpg")
```

Using the default settings, this example results in a view of the specified document. However, a number of properties are available to change the default behavior of the `Open` method. See [Control Properties, on page 212](#) for a list of relevant properties. The properties that begin with "OPEN" (for example, the `OPENMode` property and `OPENWaitOnOpen` property) apply to the `Open` method.

When a document is opened successfully, the `OpenDocDone` and `PageNumber` events are generated. These events indicate the progress of the document processing.

Save a Document

To save a document

1. Open the document. See [Open, on page 199](#).

Set the `OPENWaitOnOpen` property to `TRUE` to prevent the `Open` method from returning before the document is completely processed.

To save a document without viewing it, set the `OPENMode` property to 2 (open without generating a view).

2. Use the `CanSaveAs` property to determine whether the document is completely processed and can be saved.

3. Call the [SaveAs](#) method. To display the **Save As** dialog box and allow the user to save to a target file name, set the `FileName` parameter to an empty string.

Convert a Document

To convert a document to text, RTF, or HTML

1. Open the document. See [Open, on page 199](#) for more information.

Set the `OPENWaitOnOpen` property to `TRUE` to prevent the `Open` method from returning before the document is completely processed.

To convert a document without viewing it, set the `OPENMode` property to 2 (open without generating a view).
2. Use the `CanSaveAs` property to determine whether the document is completely processed and can be saved.
3. Call the [Convert, on page 191](#) method.

NOTE: Viewing SDK does not convert PDF, presentations, container files, or graphics files to text, RTF, or HTML.

Print a Document

To print a document

1. Open the document. See [Open, on page 199](#) for more information.

Set the `OPENWaitOnOpen` property to `TRUE` to make sure that the entire document is opened before the document is printed.

To print a document without viewing it, set the `OPENMode` property to 2 (open without generating a view). You can also set the `Visible` property of the Viewing control object to `FALSE`.
2. Use the `CanPrint` property to determine whether the document is completely processed and ready for printing.
3. Optionally, use the `PrintHeaders` property to specify whether the file name and page number header are printed at the top of each page of the printed output.

Used in conjunction with the `PrintHeaders` property, the `SetPrintName` method replaces the default file name field of the print header with another string.
4. Use either the `PrintDlg` method to print using a common **Print** dialog box, or the `PrintOut` method to print to a specific printer without a **Print** dialog box.

NOTE: When printing in an application that is an NT service, a default printer must be installed for the user account using the application.

Determine the Document Format

To determine the document format

1. Open the document whose format you want to determine. See [Open](#) for more information.
2. Use the [DocumentType](#) or [DocumentFormat](#) property to determine the format.

The [DocumentType](#) property returns the format as a text description, such as "Microsoft Word for Windows". The [DocumentFormat](#) property returns the format as a numeric value. These properties are set to the document format regardless of whether the document can be viewed.

3. To get the general class to which the currently opened document belongs, use the [DocumentClass](#) property.

Extract Document Metadata

To extract document metadata

1. Open the document whose metadata you want to extract. See [Open](#) for more information.
2. Set `nItem` to 0 (zero).
3. Use the [GetSummaryInfo](#) method to get the total number of summary information items available in the document (parameter `nTotalItem`).
4. Set `nItem` to a number between zero and `nTotalItem` ($0 \leq nItem < nTotalItem$).
5. Use the [GetSummaryInfo](#) method again to get the specific information for the `nItem` item. After the call, the other parameters (such as `nValid`, `nType`, `lVal`, `szVal`, and `szUserVal`) hold the values for the summary information item.

Search for Text in a Document

To search for specified text in a document

1. Open the document that you want to search. See [Open](#) for more information.
2. Use the [CanFind](#) property to determine whether the document is completely processed and can be searched.
3. Call the [Find](#) method without a search string specified to open a **Find** dialog box.
4. To perform a **Find Next**, call the `Find` method repeatedly with the original search term.

Copy a Selected Area of Text

To select and copy text

1. Open the document from which you want to copy the text. See [Open](#) for more information.
2. Highlight a region of text to copy.
3. Use the [CanCopy](#) property to determine whether the selected text can be copied.
4. Use the [GetSelectedText](#) method to get the selected text.
5. Use the [Copy](#) method.
6. Paste the text in another document, such as a text file.

Copy all the Text in a Document

To copy all the text in a document

1. Open the document from which you want to copy all the text. See [Open](#) for more information.
2. Use the [CanSelectAll](#) method to determine whether all contents in the document can be selected.
3. Use the [SelectAll](#) method.
4. Use the [CanCopy](#) property to determine whether the text can be copied.
5. Use the [Copy](#) method.
6. Paste the text in another document, such as a text file.

Create a Thumbnail Image of a Document Page

To create thumbnails of a document page

1. Set the [OPENMode](#) property to 7 (drawing mode enabled).
2. Open the document for which you want to create a thumbnail image. See [Open](#) for more information.
3. Use the [DrawToFile](#) method.

Filter a Document

To filter a document

1. Set the [OPENMode](#) property to 6 (index-only mode).
2. Open the document that you want to filter. See [Open](#) for more information.
3. Use the [GetNextTextBuffer](#) method to get the text buffers.

Highlight Text in a Document

To highlight text in a document

1. Set the [OPENMode](#) property to 5 (enable index mode).
2. Open the document in which you want to place highlights. See [Open](#) for more information.
3. Use the [SetHiLiteOptions](#) method to set the highlight color and font.
4. Use the [SetHiLite](#) method to specify the text to be highlighted.

Annotate Text in a Document

To add annotations to or remove annotations from a document

1. Set the [OPENMode](#) property to 5 (enable index mode).
2. Open the document you want to annotate. See [Open](#) for more information.
3. Use the [Annotate](#) method to add and delete annotations.

Chapter 9: Control Sample Programs

This section describes the sample programs that demonstrate how to use the ActiveX control.

- [Viewing SDK Initialization Information](#) 184
- [fileview](#) 184
- [dotnetview](#) 187

Viewing SDK Initialization Information

Viewing SDK uses initialization information for its internal operations; for example, to determine which components to load. You can store this information either in the Windows registry or in an initialization file. When you use Viewing SDK you must tell it where to find this information and what form it is in. See [View Initialization Information, on page 22](#) for more information.

fileview

FILEVIEW is a sample program that demonstrates how to insert the Viewing ActiveX control into a Visual Basic application and use it to display documents.

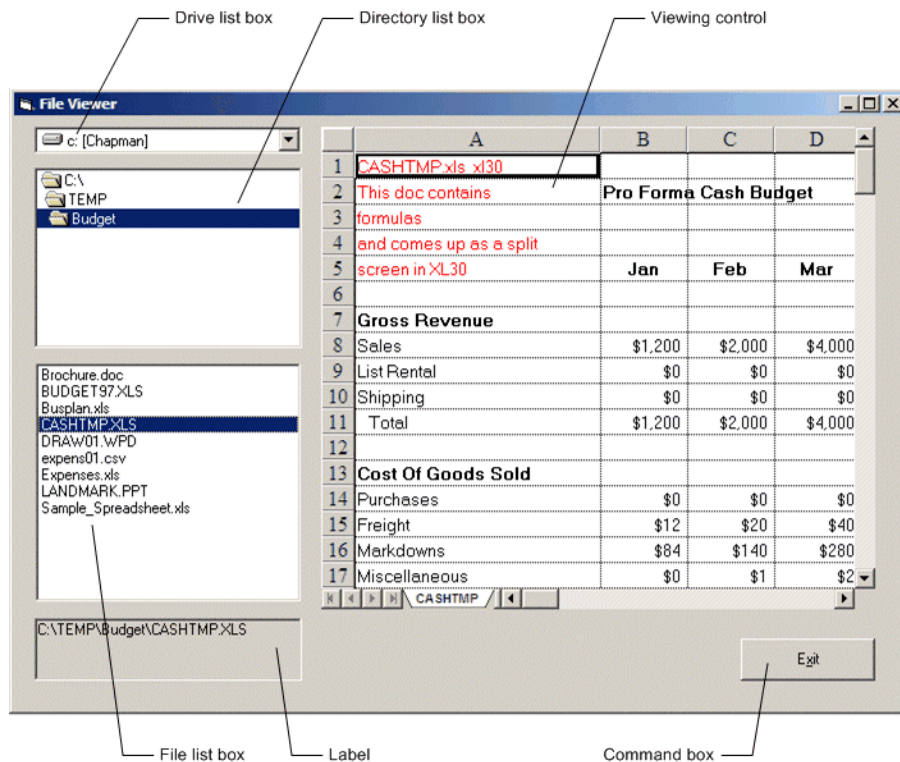
Create a New Visual Basic Project 6.0

1. Start a new project by choosing **New Project** from the **File** menu.
2. Add the Viewing Control to the project. From the **Project** menu, select **Components**.
3. In the **Components** dialog box, select **KeyView OLE Control module** from the **Controls** list box. Click **OK**.

If the KeyView OLE Control module is not listed in the **Controls** list box, click **Browse** to locate and register the control.

The Viewing icon  appears in the Toolbox.

Draw the Controls



Draw the controls on the form according to the diagram above. Use the following controls:

Button	Control
	Command button
	Label
	Drive list box
	Directory list box
	File list box
	Viewing control

Set Objects and Properties

After you design the form, you need to set the following properties:

Object	Property	Setting
Form	Caption	File Viewer
Label	BorderStyle	1-Fixed Single
	Caption	(Empty)
Command button	Caption	Exit

NOTE: Use the default settings for all other properties and objects.

Create Event Procedures

In the File Viewer application, create the following event procedures for five different controls. After you have completed these event procedures, you can compile and run the application.

Form Load event: The `Form_Load` event sets the drive and path to the drive and directory where the sample application is loaded, and specifies whether initialization information is stored in the registry or the `kvsdk.ini` file. Add the following code to the `Form_Load` event procedure:

```
Private Sub Form_Load()  
    Drive1.Drive = App.Path  
    Dir1.Path = App.Path  
    KEYview1.RegIniMode = 1  
    KEYview1.RegIniName = "kvsdk.ini"  
End Sub
```

Command button Click event: The command button's `Click` event ends the application. Add the following code to the `Command1_Click` event procedure:

```
Private Sub Command1_Click ()  
    Unload Me  
    End          ' Ends the application.  
End Sub
```

Drive list and directory list boxes Change events: To make the drive, directory, and file list boxes work together, add the following code to the `Drive1_Change` and `Dir1_Change` event procedures:

```
Private Sub Drive1_Change ()  
    Dir1.Path = Drive1.Drive          ' Update directory path.  
End Sub  
Private Sub Dir1_Change ()  
    File1.Path = Dir1.Path          ' Update files.  
End Sub
```

File list box Double-click event: The `File1_DblClick` event procedure for the file list box displays the full path name of the selected file in the label control, and displays the picture itself in the image control. You need the following code:

```
Private Sub File1_DblClick ()  
    If Right(File1.Path, 1) <> "\" Then
```

```
        Label1.Caption = File1.Path & "\" & File1.FileName  
    Else  
        ' If root directory  
        Label1.Caption = File1.Path & File1.FileName  
    End If  
    nRet = KEYview1.Open(Label1.Caption)  
End Sub
```

Notice how the full path name is constructed:

- `File1.Path` returns the drive and directory path, "\" adds a backslash separator, and `File1.FileName` returns the file name.
- The `Right` function checks to see whether the path name is the root directory (\). If it is not, the full path name is assigned to the label's caption. If the path name is the root directory, the backslash is omitted.
- The `Open` method loads the file specified by the path name in the label caption.

dotnetview

Viewing SDK includes a .NET workspace for Visual Studio called `dotnetview`. This is a J# application that demonstrates basic Viewing functionality. To use the .NET sample, open the workspace file `dotnetview.sn1` in Visual Studio 2005 and follow the instructions in [Develop .NET Applications, on page 26](#). The source file is `form1.js1`.

Chapter 10: Control Methods

This section describes the Viewing ActiveX control methods.

• Annotate	189
• ChangeObject	190
• Close	190
• Convert	191
• Copy	192
• DecreaseFont	192
• DrawToFile	193
• Find	194
• GetNextTextBuffer	195
• GetPageFromLogical	195
• GetSelectedText	196
• GetSummaryInfo	196
• GetText	197
• GoToPage	198
• IncreaseFont	199
• Open	199
• Play	200
• Position	201
• PositionHiLite	201
• PrintDlg	202
• PrintOut	202
• PrintOutEx	203
• PrintPageSetup	204
• SaveAs	204
• SelectAll	205
• SetCursor	205
• SetFocusViewer	206
• SetHiLite	206
• SetHiLiteOptions	207
• SetPassword	208
• SetPrintName	208
• ShowHits	209
• UnZip	209
• UnZipEx	210

NOTE: The Viewing control's methods in this chapter show the syntax in Visual Basic.

Annotate

Adds and deletes annotations, and determines whether annotations exist. Use this method with the [OPENMode](#) property set to 5 (indexing enabled).

Syntax

```
bRet = KeyView.Annotate(action, position, type, bitmapFile, cbText, colorref, cursorFile)
```

Parameters

Parameter	Type	Description
action	Integer	<ul style="list-style-type: none">• 0 – delete annotation.• 1 – add annotation.• 2 – query annotation.
position	Long	Logical address to apply the annotation.
type	Integer	<ul style="list-style-type: none">• 0 – use a bitmap for the annotation.• 1 – use an underline as the annotation.• 2 – use a dotted underline as the annotation. Currently not implemented.• 3 – use a strikethrough as the annotation. Currently not implemented.
bitmapFile	String	If the annotation type is bitmap, this is the bitmap file name (*.bmp).
cbText	Integer	If the annotation type is underline, this is the length of the underlined text.
colorref	Long	If the annotation type is underline, this is the color of the underlined text.
cursorFile	String	The cursor file (*.cur) to use when mouse hovers over the annotation.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

ChangeObject

Changes the current object in a multiple-object document. Examples of a multiple-object document include:

- A Microsoft Excel spreadsheet with multiple worksheets where each worksheet is considered an object.
- A Microsoft PowerPoint presentation with multiple slides where each slide is considered an object.

Syntax

```
bRet = KeyView.ChangeObject(direction)
```

Parameters

Parameter	Type	Description
<code>direction</code>	Integer	0 – select the next object. 1 – select the previous object.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

Close

Closes the currently opened document. The `Close` method is not required because calling the `Open` method automatically closes the currently opened document.

Syntax

```
bRet = KeyView.Close
```

Parameters

None

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

Convert

Converts a document to text, RTF, or HTML format.

Before you convert a document, you must open it by using the [Open](#) method. To make sure that the entire document is opened before the document is converted, set the [OPENWaitOnOpen](#) property to **TRUE**. Use the [CanSaveAs](#) property to check whether the document is completely processed and is ready for conversion.

To convert a document without viewing it, set the [OPENMode](#) property to 2 (open without generating a view).

Syntax

```
bRet = KeyView.Convert(FormatCode, TargetFile)
```

Parameters

Parameter	Type	Description
FormatCode	String	The format code of the format to which to convert the document. The following options are available: txt — conversion to text format rtf — conversion to RTF format htm — conversion to HTML format
TargetFile	String	The target path and file name for the converted file.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

Example

The following code converts a file to RTF:

```
Function ConvertToRTF(FileName As String, TargetFile As String) As Boolean
    KeyView.OPENMode = 2
    nRet = KeyView.Open(FileName)
    If (nRet = 1) Then
        bRet = KeyView.Convert("rtf",TargetFile)
    End If
    bRet2 = KeyView.Close
    ConvertToRTF = bRet
End Function
```

In this example, OPENMode is set to 2 to prevent a view of the document.

Copy

Copies the current selection to the clipboard. Use the [CanCopy](#) property to determine whether the content is selected and can be copied by using the Copy method.

Syntax

```
bRet = KeyView.Copy
```

Parameters

None

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

DecreaseFont

Decreases the document font size. Use the [CanDecreaseFont](#) property to determine whether the font size can be decreased, and therefore, whether the DecreaseFont method succeeds.

Syntax

```
bRet = KeyView.DecreaseFont
```

Parameters

None

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

DrawToFile

Draws a page of a document to a graphic file (thumbnail). Use this method with the [OPENMode](#) property set to 7 (drawing mode enabled).

Syntax

```
bRet = KeyView.DrawToFile(szTargetFile, szGfxOutput, uPageNumber, nWidth, nHeight,  
nPicXRes, nPicYRes, nQuality)
```

Parameters

Parameter	Type	Description
szTargetFile	String	The target path and file name to which the page is written. The file should use one of the following four extensions: <ul style="list-style-type: none">• .bmp• .jpg• .png• .tif
SzGfxOutput	String	The output graphics format. The following options are available: <ul style="list-style-type: none">• bmp• jpg

Parameter	Type	Description
		<ul style="list-style-type: none">• png• tif (uncompressed TIFFs) <p>You must specify the parameter in lower case.</p>
uPageNumber	UINT	The number of the page to be rasterized into a thumbnail. Page numbers start at 0. For example, set uPageNumber to 0 to draw page 1, and to 1 to draw page 2. For word processing documents, pages must be drawn sequentially. For example, to draw page 3, you must first draw pages 0 and 1.
nWidth	Integer	The maximum picture width (in TWIPS).
nHeight	Integer	The maximum picture height (in TWIPS).
nPicXRes	Integer	The desired horizontal resolution (0 for default).
nPicYRes	Integer	The desired vertical resolution (0 for default).
nQuality	Integer	This parameter controls the output quality of graphics that support compression quality (for example, JPEG). The valid range is 0 to 100. A value of 0 means default quality (85 compression); 1 is the lowest quality (highest compression and therefore the smallest file size); 100 is the highest quality (no compression and therefore the largest file size). The default is 0.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

Find

Searches the currently opened document for the specified text. To perform a **Find Next**, call this method repeatedly with the original search term. If you specify an empty string for the FindText, a **Find** dialog box appears.

Syntax

```
bRet = KeyView.Find(FindText, FindDown, MatchCase)
```

Parameters

Parameter	Type	Description
FindText	String	A string that contains the text to find.
FindDown	Boolean	Specifies whether to search from the selected point in the document to the beginning (FALSE) or to the end (TRUE) of the document.
MatchCase	Boolean	Specifies whether to match the case of the search term.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

GetNextTextBuffer

Gets text buffers. Use this method with the [OPENMode](#) property set to 6 (index only mode).

Syntax

```
bRet = KeyView.GetNextTextBuffer
```

Parameters

None

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

GetPageFromLogical

Gets the page number for a logical address.

Syntax

```
lRet = KeyView.GetPageFromLogical(lLogicalAddress)
```

Parameters

Parameter	Type	Description
<code>lLogicalAddress</code>	Long	The logical address for which to get the page number.

Returns

Type	Description
Long	<i>n</i> – Page number the specified logical address resides on -1 – Error

GetSelectedText

Returns any currently selected text. The [CanCopy](#) property returns **TRUE** if any text is currently selected.

Syntax

```
SelectedText = KeyView.GetSelectedText
```

Parameters

None

Returns

Type	Description
String	A description of the specified format.

GetSummaryInfo

Gets the document metadata, also referred to as summary information. See [Extract Document Metadata, on page 181](#).

Syntax

```
bRet = KeyView.GetSummaryInfo(nItem, nTotalItem, nValid, nType, lVal, szVal,  
szUserVal)
```

Parameters

Parameter	Type	Description
nItem	Integer	The summary information item number.
nTotalItem	Integer	The total number of summary information items available in the source document.
nValid	Integer	Specifies whether the data value is present in the document. 0 – The summary information field is not available. 1 – The summary information field is available.
nType	Integer	The data type of the metadata element. 1 – szVal contains the field contents. 2 – lVal contains the field contents. The types are defined in KVSumInfoType in kvtypes.h.
lVal	Integer	Contains the contents of the summary information field if the contents are numeric.
szVal	String	Contains the contents of the summary information field if the contents are non-numeric.
szUserVal	String	The summary information field name.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

GetText

Gets a text buffer from a specified range. Use this method with the [OPENMode](#) property set to 5 (indexing enabled).

Syntax

```
bRet = KeyView.GetText(start, cbText, pszText)
```

Parameters

Parameter	Type	Description
start	Long	The starting logical address of the text.
cbText	Integer	The number of bytes of text to copy.
pszText	String	A pointer to the logical address where the text will be stored.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

GoToPage

Displays the document at the specified page. Use this method with the [OPENMode](#) property set to 5 (indexing enabled).

Syntax

```
nRet = KeyView.GoToPage(nPageNumber)
```

Parameters

Parameter	Type	Description
nPageNumber	Integer	The page to display.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

IncreaseFont

Increases the document font size. Use the [CanIncreaseFont](#) property to determine whether the font size can be increased, and therefore, whether the `IncreaseFont` method succeeds.

Syntax

```
bRet = KeyView.IncreaseFont
```

Parameters

None

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

Open

Opens a document.

The properties that begin with "OPEN" (for example, the `OPENMode` property and `OPENWaitOnOpen` property) apply to the `Open` method. However, they do not apply to the initial call to the `Open` method; that is, they do not affect the currently opened document and apply only to documents opened with subsequent calls to the `Open` method. See [Control Properties, on page 212](#).

Syntax

```
nRet = KeyView.Open(fileName)
```

Parameters

Parameter	Type	Description
filename	String	A string containing the file name of the document to open.

Returns

Type	Description
Short	0 – if an error occurs during open. 1 – if the document opened successfully. 2 – if the document is of an unknown format. 3 – if no viewer is available for the document format. 4 – if the document is password-protected. 5 – if the drawing routines have not been initialized. 6 – if the requested page does not exist, or is being displayed before all previous pages have been displayed. 7 – if the document does not support this feature (for example, ZIP files, video, or audio). 8 – if the KeyView license is invalid. 9 – if the KeyView license is expired. 10 – if the input file or stream is invalid or corrupt.

Play

Plays, pauses, or stops a multimedia (digital video or sound) document.

Syntax

```
bRet = KeyView.Play(Mode)
```

Parameters

Parameter	Type	Description
Mode	Integer	0 – Play 1 – Pause 2 – Stop

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

Position

Positions the document in the viewing window.

Syntax

```
bRet = KeyView.Position(first, last, position)
```

Parameters

Parameter	Type	Description
first	Long	The first visible logical address.
last	Long	The last visible logical address.
position	Long	The position (logical address) to make visible.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

PositionHiLite

Changes focus from the previous to next highlight. This only applies when using XML files with the Verity Developer's Kit (VDK) to specify highlights.

Syntax

```
bRet = KeyView.PositionHiLite(bPrev)
```

Parameters

Parameter	Type	Description
bPrev	Boolean	TRUE – go to previous highlight FALSE – go to next highlight.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

PrintDlg

Prints a document by calling the common **Print** dialog box. [Print a Document, on page 180.](#)

Syntax

```
bRet = KeyView.PrintDlg
```

Parameters

None

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

PrintOut

Prints a document to a specified printer without calling the common **Print** dialog box, or to the default printer if none is specified. See [Print a Document, on page 180.](#)

Syntax

```
bRet = KeyView.PrintOut(Printer)
```

Parameters

Parameter	Type	Description
Printer	String	The name of the printer to use. If the string is empty, the default printer is used. An example of a valid string value is \\Calculus\HP LaserJet IIISi

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

PrintOutEx

Prints a document to a specified printer without calling the common **Print** dialog box, or to the default printer if none is specified. The method can specify the page range and number of copies to print. See [Print a Document, on page 180](#).

Syntax

```
bRet = KeyView.PrintOutEx(Printer, FromPage, ToPage, Copies, Flags)
```

Parameters

Parameter	Type	Description
Printer	String	The name of the printer to use. If the string is empty, the default printer is used. An example of a valid string value is <code>\\Calculus\HP LaserJet IIISi</code>
FromPage	Integer	The number of the first page to print. For spreadsheet documents, this is the first sheet number.
ToPage	Integer	The number of the last page to print. For spreadsheet documents, this is the last sheet number.
Copies	Integer	The number of copies to print.
Flags	Integer	Reserved. This parameter should be set to 0.

NOTE: If `FromPage`, `ToPage`, and `Copies` are all set to 0, this method behaves the same way as [PrintOut](#)

If `FromPage` and `ToPage` are both set to 0, all pages are printed.

If `Copies` is set to 0, one copy is printed.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

PrintPageSetup

Displays a dialog box that allows the user to set up print page scaling for a spreadsheet.

Syntax

```
bRet = KeyView.PrintPageSetup
```

Parameters

None

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

SaveAs

Saves the current document in another format.

If the `FileName` parameter is an empty string, this method displays the **Save As** dialog box, which allows the user to specify conversion, compression, and encodings, as well as the target file name. Otherwise, the `FileName` parameter should be the complete path and file name of the desired target file. A copy operation from the currently opened file to the specified target file occurs. Use the [CanSaveAs](#) property to check whether the document has been completely processed and can be saved.

Syntax

```
bRet = KeyView.SaveAs(FileName)
```

Parameters

Parameter	Type	Description
fileName	String	The complete path and file name of the target file. This parameter can be an empty string that displays the Save As dialog box.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

SelectAll

Selects all items in the currently opened document. Use the [CanSelectAll](#) method to determine whether all contents in the document can be selected.

Syntax

```
bRet = KeyView.SelectAll
```

Parameters

None

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

SetCursor

Sets the viewing engine cursor.

Syntax

```
bRet = KeyView.SetCursor(cursorFile, bRestore)
```

Parameters

Parameter	Type	Description
CursorFile	String	The cursor file (*.cur) to use.
bRestore	Boolean	TRUE restores the default cursor. FALSE uses the cursor file.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

SetFocusViewer

Sets the current focus to the Viewer window.

Syntax

```
bRet = KeyView.SetFocusViewer
```

Parameters

None

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

SetHiLite

Highlights a region of text. Use this method with the [OPENMode](#) property set to 5 (indexing enabled).

Syntax

```
bRet = KeyView.SetHiLite(cbTextToHiLite, lLogicalAddress)
```

Parameters

Parameter	Type	Description
<code>cbTextToHiLite</code>	Integer	The number of bytes to highlight.
<code>lLogicalAddress</code>	Long	The logical address from which to start highlighting.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

SetHiLiteOptions

Set the highlight options. Use this method with the [OPENMode](#) property set to 5 (indexing enabled).

Syntax

```
bRet = KeyView.SetHiLiteOptions(BackColor, ForeColor, FontSize, FontName)
```

Parameters

Parameter	Type	Description
<code>BackColor</code>	Integer	The COLORREF value that is the highlight background color.
<code>ForeColor</code>	Integer	The COLORREF value that is the highlight foreground color.
<code>FontSize</code>	Integer	The font size in points.
<code>FontName</code>	String	The font name.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

SetPassword

Sets a password to use to open a password-protected file before the file is opened. Currently, you can use this to set a password for ZIP, PST, and NSF files.

NOTE: Unicode passwords are not supported.

Syntax

```
bRet = KeyView.SetPassword (szPassword);
```

Parameters

Parameter	Type	Description
szPassword	String	The password string.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

SetPrintName

Used in conjunction with the `PrintHeaders` property, this method replaces the default file name field of the print header with a specified string.

Syntax

```
bRet = KeyView.SetPrintName(szPrintName)
```

Parameters

Parameter	Type	Description
szPrintName	String	A string used to replace the file name field of the <code>PrintHeader</code> property. For example, you could replace the file name with "Copyright 2003".

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

ShowHits

Shows or hides index hits. Use this method with the [OPENMode](#) property set to 5 (indexing enabled).

Syntax

```
bRet = KeyView.ShowHits(bShow)
```

Parameters

Parameter	Type	Description
bShow	Boolean	TRUE shows hits. FALSE hides hits.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

UnZip

Extracts selected subfiles in the currently opened container file to disk or to a Viewer window.

Syntax

```
bRet = KeyView.UnZip(UnZipToDisk)
```

Parameters

Parameter	Type	Description
UnZipToDisk	Boolean	If TRUE, a dialog box prompts the user to specify a path where the subfiles are extracted. If FALSE, the subfiles are extracted and displayed in the preview pane. This is the default.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

UnZipEx

Extracts the subfiles in the currently opened container file to disk without requiring the user to respond to dialog boxes.

Syntax

```
bRet = KeyView.UnZipEx(TargetDirectory, Password, bFailIfExists,  
bPreserveDirectory)
```

Parameters

Parameter	Type	Description
TargetDirectory	String	The directory to which to extract the subfiles.
PassWord	String	The password to use to open a password-protected container file.
bFailIfExists	Boolean	If TRUE and a subfile is extracted and a file of the same name already exists in the target directory, the existing file is not overwritten. If FALSE, the existing subfile is overwritten.
bPreserveDirectory	Boolean	If TRUE, the directory structure is preserved. If FALSE, the directory structure is not preserved.

Returns

Type	Description
Boolean	TRUE if the method succeeds. FALSE if the method fails.

Chapter 11: Control Properties

This section describes the Viewing ActiveX control properties.

• Introduction	214
• ASCIICharSet	214
• ASCIIFilterNonPrintable	215
• ASCIIFontName	215
• ASCIIFontSize	215
• ASCIIFontStyle	216
• ASCIIMarginBottom	216
• ASCIIMarginLeft	216
• ASCIIMarginRight	216
• ASCIIMarginTop	217
• ASCIIPrintLandscape	217
• AspectRatio	217
• CanCopy	218
• CanDecreaseFont	218
• CanFind	218
• CanIncreaseFont	219
• CanMultiObj	219
• CanNextObj	219
• CanPause	220
• CanPlay	220
• CanPrevObj	220
• CanPrint	221
• CanSaveAs	221
• CanSelectAll	221
• CanStop	222
• CanUnZip	222
• CanViewPane	222
• CharSet	222
• ContextMenu	223
• DocumentClass	223
• DocumentFormat	224
• DocumentType	224
• DrawPageCount	224
• DrawPageHeight	225

- [DrawPageWidth](#) 225
- [DrawWorkBookPageCount](#) 225
- [FileName](#) 226
- [HiLiteBackground](#) 226
- [HiLiteForeground](#) 226
- [HotKeys](#) 226
- [ImageCustomSize](#) 227
- [ImagePrintHorzAlign](#) 227
- [ImagePrintMode](#) 227
- [ImagePrintPercent](#) 228
- [ImagePrintVertAlign](#) 228
- [ImageScaling](#) 228
- [IndexBufCharSet](#) 229
- [Invert](#) 229
- [JumpToFirstHiLite](#) 229
- [MMPlayOption](#) 229
- [MMScaleMovie](#) 230
- [ObjName](#) 230
- [OPENDisableUI](#) 230
- [OPENHighLight](#) 231
- [OPENMode](#) 231
- [OPENWaitOnOpen](#) 232
- [PrintAnnotations](#) 232
- [PrintHeaders](#) 232
- [RegIniMode](#) 233
- [RegIniName](#) 233
- [Rotate](#) 233
- [SrcCharSet](#) 234
- [SSDisplayGrid](#) 234
- [SSDisplayHeaders](#) 235
- [SSViewObjects](#) 235
- [TrgCharSet](#) 235
- [ViewPane](#) 235
- [WPCustomSize](#) 236
- [WPDisplayPict](#) 236
- [WPPageLayout](#) 236
- [WPScaleTable](#) 237
- [WPViewMode](#) 237

Introduction

Persistent Properties

Persistent properties are those that can be set in the **Properties** page at design time. The selected value is stored as a part of your project and used at runtime.

Property Naming Conventions in .NET

In J#, C#, and C++, all ActiveX control method names in the .NET class are the same as their COM counterparts. However, individual properties in .NET are defined using `get` and `set` methods of the following format:

`get_property_name`

`set_property name`

For example:

```
private void button1_Click(Object sender, System.EventArgs e)
{
    this.axKEYview1.set_RegIniMode((short)1);
    this.axKEYview1.set_RegIniName("c:\\windows\\kvsdk.ini");
    this.axKEYview1.Open("c:\\test.doc");
}
```

IMPORTANT: In a Visual Basic .NET application, all properties and methods are used in the same way as in a Visual Basic COM application.

See [Develop .NET Applications, on page 26](#) for more information.

"OPEN" Properties

The "Open" properties (such as `OPENWaitUponOpen` and `OPENMode`) apply to the `Open` method. However, they do not apply to the initial call to the `Open` method; that is, they do not affect the currently opened document, and apply only to documents opened with subsequent calls to the `Open` method. These properties are persistent and are available through the **Properties** page.

ASCIICharSet

Specifies the character set used to display an ASCII text file. The ANSI character set is the Microsoft Windows default. The DOS code page is useful if the text file was created with a DOS editor. This is a persistent property.

Returns

Type	Description
Integer	0 – Displays the document using the ANSI character set. 1 – Displays the document using the DOS code page.

ASCIIFilterNonPrintable

Specifies whether non-printable characters can be viewed in an ASCII text file. This is a persistent property.

Returns

Type	Description
Boolean	TRUE – non-printable characters can be viewed. FALSE – non-printable characters cannot be viewed.

ASCIIFontName

Specifies the font name to be used in an ASCII text file. This is a persistent property.

Returns

Type	Description
String	A description of the desired font name.

ASCIIFontSize

Specifies the font size to be used in an ASCII text file. This is a persistent property.

Returns

Type	Description
Integer	<i>n</i> – The font size in points.

ASCIIFontStyle

Specifies the font style used to display an ASCII text file.

Returns

Type	Description
Integer	0 – Displays text in normal font.
	1 – Displays text in bold font.
	2 – Displays text in italic font.
	3 – Displays text in bold and italic font.

ASCIIMarginBottom

Specifies the bottom page margin of an ASCII text file (between 0 - 4 inches or 100 mm). This is a persistent property.

Returns

Type	Description
Integer	<i>n</i> – The bottom page margin in TWIPS (the default is 1440).

ASCIIMarginLeft

Specifies the left page margin of an ASCII text file. This is between zero and 4 inches, or zero and 100 mm. This is a persistent property.

Returns

Type	Description
Integer	<i>n</i> – The left page margin in TWIPS (the default is 1440).

ASCIIMarginRight

Specifies the right page margin of an ASCII text file. This is between zero and 4 inches, or zero and 100 mm. This is a persistent property.

Returns

Type	Description
------	-------------

Integer *n* – The right page margin in TWIPS (the default is 1440).

ASCIIMarginTop

Specifies the top page margin of an ASCII text file (between 0 and 4 inches or 100 mm). This is a persistent property.

Returns

Type	Description
------	-------------

Integer *n* – The top page margin in TWIPS (the default is 1440).

ASCIIPrintLandscape

Specifies whether the ASCII text file is printed in portrait or landscape mode. This is a persistent property.

Returns

Type	Description
------	-------------

Integer 0 – Prints the document in portrait mode.

1 – Prints the document in landscape mode.

AspectRatio

Specifies the current aspect ratio of the document (the pictures). **This property is read-only.**

Returns

Type	Description
------	-------------

Integer -1 – Not applicable.

0 – None.

1 – Based on the document.

Type	Description
	2 – Normal (use scanlines).
	3 – Letter (times 2).

CanCopy

Specifies whether content is selected in the currently opened document. If this returns TRUE, the selection can be copied to the clipboard by using the [Copy](#) method. Any selected text can be obtained by using the [GetSelectedText](#) method. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE if the selected text can be copied.
	FALSE if the selected text cannot be copied.

CanDecreaseFont

Specifies whether the document font size can be decreased. If this returns TRUE, the font size can be decreased by using the [DecreaseFont](#) method. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE if the font size can be decreased.
	FALSE if the font size cannot be decreased.

CanFind

Specifies whether the currently opened document can be searched. If this returns TRUE, the document can be searched by using the [Find](#) method. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE if the document can be searched.
	FALSE if the document cannot be searched.

CanIncreaseFont

Specifies whether the document font size can be increased. If this returns TRUE, the font size can be increased by using the [IncreaseFont](#) method. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE if the font size can be increased. FALSE if the font size cannot be increased.

CanMultiObj

Specifies whether a document contains multiple objects. If this property is TRUE, the `ObjName` property and `ChangeObject` method are applicable to the opened document. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE if the document contains multiple objects. FALSE if the document does not contain multiple objects.

CanNextObj

Specifies whether the next object is available in a multiple-object document. **This property is read-only.**

Examples of a multiple-object document include:

- A Microsoft Excel spreadsheet with multiple worksheets, where each worksheet is considered an object.
- A Microsoft PowerPoint presentation with multiple slides, where each slide is considered an object.

Returns

Type	Description
Boolean	TRUE – the next object is available. FALSE – the next object is not available.

CanPause

Specifies whether the playing of a multimedia document can be paused. This returns TRUE only if you have a multimedia document opened and playing. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE if the document can be paused. FALSE if the document cannot be paused.

CanPlay

Specifies whether a multimedia document can be played. This property is TRUE if the document is a multimedia (sound or digital video) document, and if the document is not currently playing. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE if the document can be played. FALSE if the document cannot be played.

CanPrevObj

Specifies whether the previous object is available in a multiple-object document. **This property is read-only.**

Examples of a multiple-object document include:

- A Microsoft Excel spreadsheet with multiple worksheets, where each worksheet is considered an object.
- A Microsoft PowerPoint presentation with multiple slides, where each slide is considered an object.

Returns

Type	Description
Boolean	TRUE– the previous object is available. FALSE – the previous object is not available.

CanPrint

Specifies whether the currently opened document is completely open and ready for printing. A document must be fully processed before it can be printed. Use the [OPENWaitOnOpen](#) property to make sure that the `Open` method does not return until the entire document is open. **This property is read-only.**

If this property is `TRUE`, the document can be printed by using the [PrintDlg](#) or [PrintOut](#) method.

Returns

Type	Description
Boolean	<code>TRUE</code> if the document can be printed. <code>FALSE</code> if the document cannot be printed.

CanSaveAs

Specifies whether the currently opened document can be saved or converted to a different file. If this is `TRUE`, the document can be saved or converted by using the [SaveAs](#) and [Convert](#) methods. **This property is read-only.**

Returns

Type	Description
Boolean	<code>TRUE</code> if the document can be saved.

CanSelectAll

Specifies whether all items in the currently opened document can be selected. If this property is `TRUE`, all items in the document can be selected by using the [SelectAll](#) method. **This property is read-only.**

Returns

Type	Description
Boolean	<code>TRUE</code> if the document can be selected.

CanStop

Specifies whether the playing of a multimedia document can be stopped. This property returns TRUE only if you have a multimedia document opened and playing. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE if the document can be stopped.

CanUnZip

Returns TRUE if the currently opened document is a container file and there are subfiles selected in the file. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE if the document can be unzipped.

CanViewPane

Specifies whether a preview pane can be viewed. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE – can show preview pane. This applies only to container files. FALSE – cannot show a preview pane.

CharSet

Enables you to get and set the character set of a document that is open in the Viewing window.

Returns

Type	Description
Integer	A value from the enumerated type <code>KVCharSet</code> . See the <code>kvcharset.h</code> file for a description.

ContextMenu

Turns the context menu on or off. The context menu is the menu that appears when you right mouse click in the client area. This is a persistent property.

Returns

Type	Description
Boolean	<code>TRUE</code> if the context menu is enabled.

DocumentClass

Indicates which general class the currently opened document belongs to. **This property is read-only.**

The classes are as follows:

- | | |
|----------------------------------|------------------------|
| 1 – Text document (ASCII) | 6 – Fax document (FAX) |
| 2 – Word processor document (WP) | 7 – Presentation (PG) |
| 3 – Spreadsheet document (SS) | 8 – Archive document |
| 4 – Image (Image) | 9 – Other |
| 5 – Multimedia document (MM) | |

The document class is useful to determine whether a specific set of properties is applicable to the currently opened document. The codes specified in the brackets above are prefixes used for all persistent properties specific to that document type.

For example, a multimedia document has a `DocumentClass` of 5, and the properties `MMPlayOption` and `MMScaleMovie` are relevant only to multimedia documents. The value of these properties is ignored for any other kind of document.

The one exception to the above rule is the `WPPageLayout` property, which is applicable to both text and word processor documents.

Returns

Type	Description
------	-------------

Short	The document class (a value of 1 through 9, as described above).
-------	--

DocumentFormat

Specifies the document format value of a document. This value corresponds to the numerical equivalent of the `DocumentType` property value. For example, the "Microsoft Word for Windows" `DocumentType` corresponds to "44" for the `DocumentFormat` property. **This property is read-only.**

The document formats are defined in the `adinfo.h` header file.

Returns

Type	Description
------	-------------

Integer	n – The document format value.
---------	----------------------------------

DocumentType

Specifies the type of the currently opened document, such as "Microsoft Word for Windows". If an `Open` fails because the document type is not supported for viewing by Viewing SDK, this property still contains the document type until another call to the `Open` method is made, even though no document is currently opened. **This property is read-only.**

The document formats are defined in the `adinfo.h` header file.

Returns

Type	Description
------	-------------

String	A description of the type of the currently opened document.
--------	---

DrawPageCount

Specifies the total number of pages in the document. Use this property with the `OPENMode` property set to 7 (drawing mode enabled). **This property is read-only.**

To make sure that the entire document is opened before the page count is retrieved, set the `OPENWaitOnOpen` property to `TRUE`. If `OPENWaitOnOpen` is not set to `TRUE`, the returned page count might not be accurate.

Returns

Type	Description
------	-------------

Long	n – The number of pages.
------	----------------------------

DrawPageHeight

Specifies the page height. Use this property with the [OPENMode](#) property set to 7 (drawing mode enabled). **This property is read-only.**

Returns

Type	Description
------	-------------

Long	n – The page height.
------	------------------------

DrawPageWidth

Specifies the page width. Use this property with the [OPENMode](#) property set to 7 (drawing mode enabled). **This property is read-only.**

Returns

Type	Description
------	-------------

Long	n – The page width.
------	-----------------------

DrawWorkBookPageCount

Specifies the number of workbook pages in spreadsheet documents. Use this property with the [OPENMode](#) property set to 7 (drawing mode enabled). **This property is read-only.**

Returns

Type	Description
------	-------------

Long	n – Number of pages.
------	------------------------

FileName

Specifies the file name of the currently opened document, for example `file1.doc`. **This property is read-only.**

Returns

Type	Description
String	The file name of the currently opened document.

HiLiteBackground

Specifies the background highlight color. Use this property with the [OPENMode](#) property set to 5 (indexing enabled).

Returns

Type	Description
Long	n – The color value.

HiLiteForeground

Specifies the foreground highlight color. Use this property with the [OPENMode](#) property set to 5 (indexing enabled).

Returns

Type	Description
Long	n – The color value.

HotKeys

Specifies whether or not hotkeys are enabled for KeyView and the parent application. The default is TRUE.

Returns

Type	Description
Boolean	TRUE – Hotkeys are enabled in KeyView and disabled in parent applications. FALSE – Hotkeys are disabled in KeyView and enabled in parent applications.

ImageCustomSize

Specifies the image document view in a custom size, but only if the `ImageScaling` property is set to `10`. The range of acceptable values is 10 to 400 percent. This is a persistent property.

Returns

Type	Description
Integer	<i>n</i> – Specify a custom view size.

ImagePrintHorzAlign

Specifies the horizontal print alignment of the image document. This is a persistent property.

Returns

Type	Description
Integer	0 – Print the image aligned to the left. 1 – Print the image aligned to horizontal center. 2 – Print the image aligned to the right.

ImagePrintMode

Specifies the image document print mode. This is a persistent property.

Returns

Type	Description
Integer	0 – Print the image in its original size. 1 – Print the image at full-page size.

Type	Description
2	Print the image at a customized scaling factor based on the value of the <code>ImagePrintPercent</code> property (see ImagePrintPercent , below).

ImagePrintPercent

Specifies the image print mode in a custom size, but only if the `ImagePrintMode` property is set to 2 (customized scaling). The range of acceptable values is 10 to 400 percent. This is a persistent property.

Returns

Type	Description
Integer	<i>n</i> – Specify a custom print size.

ImagePrintVertAlign

Specifies the vertical print alignment of the image document. This is a persistent property.

Returns

Type	Description
Integer	0 – Print the image aligned to the top of the page.
2	Print the image aligned to the bottom of the page.

ImageScaling

Specifies the image document view. This is a persistent property.

Returns

Type	Description
Integer	1 – Display the image to fit the selected portion of the image to the window, while maintaining the image's aspect ratio.
2	Display the image to fit to the window, while maintaining the image's aspect ratio.
3	Display the image in its original size.
10	Display the image at a customized scaling factor based on the value of the <code>ImageCustomSize</code> property.

IndexBufCharSet

Sets the character set for the returned indexed text buffer of an open document. See [GetNextTextBuffer](#), on page 195 for more information.

Returns

Type	Description
Integer	A value from the KVCharSet enumerated type. See the kvcharset.h file for a description.

Invert

Indicates and allows the inversion status of a document to be specified, for example, turning black to white and white to black.

Returns

Type	Description
Short	-1 – Not applicable to the opened document. 0 – The opened document is not inverted. 1 – The opened document is inverted.

JumpToFirstHiLite

Specifies whether to jump to the first highlight. This applies only when using XML files with the Verity Developer's Kit (VDK) to specify highlights.

Returns

Type	Description
Boolean	TRUE – jump to the first highlight. FALSE – do not jump to the first highlight.

MMPlayOption

Specifies whether a multimedia file plays continuously (loops). This is a persistent property.

Returns

Type	Description
Integer	0 – Play the multimedia file once and stop. 1 – The multimedia file loops.

MMScaleMovie

Specifies whether the movie should be played at original size or if it should fit to window.

Returns

Type	Description
Boolean	TRUE – fit the movie to the window size. FALSE – play the movie at the original size.

ObjName

Specifies the name of the currently active object. This applies only to multiple-object documents, such as spreadsheets with multiple worksheets. **This property is read-only.**

Returns

Type	Description
Boolean	TRUE if the context menu is enabled.

OPENDisableUI

This property disables the generation of dialog boxes or message boxes by the KeyView control. Dialog boxes such as **Save As** still appear if you specifically call the SaveAs method.

Returns

Type	Description
Boolean	TRUE – do not generate message boxes or dialog boxes.

OPENHighLight

Specify the text to appear highlighted in subsequently opened documents. For example, `OPENHighLight = 1 dog and mouse` highlights all occurrences of `dog` and `mouse` (case sensitive) in documents subsequently opened by using the `Open` method.

You can also search for non-adjacent words by separating them with a tab character, indicated by `(tab)`. For example, `OPENHighLight = 1 dog(tab)mouse` highlights all occurrences of `dog` or `mouse` (case sensitive).

Returns

Type	Description
String	The first character should be a 1 (case sensitive) or a 0 (case insensitive), followed by the text to be highlighted.

OPENMode

The `OPENMode` property specifies the behavior of the `Open` method.

Returns

Type	Description
Short	<ul style="list-style-type: none">0 – Default open, generates a formatted view of the document if the document format is supported.1 – Open with a text view of the document.2 – Open without generating a view. This is useful when, for example, you want to print a document without viewing it.3 – Open with a text view of the document automatically when the document is an unsupported format.5 – Open the file with indexing enabled. This generates text buffer events.6 – Open the file for indexing only. This generates text buffer events with document viewing disabled. To get text buffer events in this mode, you need to call the <code>GetNextTextBuffer</code> method—except for the first text buffer).7 – Open the file with drawing methods and properties enabled.8 – Open the files with revision marks enabled.

OPENWaitOnOpen

If this property is set to **TRUE**, the `Open` method does not return until the entire document is open. This ensures that the document is fully processed before an operation (such as printing, converting, or searching) is performed, and is useful when you want to use an operation immediately after opening the document. Use the various "Can" properties, such as `CanPrint`, `CanSaveAs`, and `CanFind`, to determine whether the document has been completely processed and is ready for the operation.

If you are opening the document for viewing only, set this property to **FALSE** so that the first page of the document can be viewed as soon as it is processed.

Returns

Type	Description
------	-------------

Boolean	If TRUE , the <code>Open</code> method does not return until the document is completely processed.
---------	---

PrintAnnotations

Specifies whether annotations should be printed in the printed output of a document.

Returns

Type	Description
------	-------------

Boolean	TRUE – print annotations.
---------	----------------------------------

	FALSE – do not print annotations.
--	--

PrintHeaders

Specifies whether a file name and page number header should be printed at the top of each page of the printed output of a document.

Returns

Type	Description
------	-------------

Boolean	TRUE – print the header.
---------	---------------------------------

	FALSE – do not print the header.
--	---

RegIniMode

The `RegIniMode` property specifies whether Viewing gets initialization information from an initialization file (.ini) or the registry. The `RegIniName` property must also be set. See [View Initialization Information, on page 22](#) for more information. This is a persistent property.

NOTE: By default, Viewing SDK looks for the initialization file in the Windows system directory; however, you can specify a file in another location by using the `RegIniName` property.

Returns

Type	Description
------	-------------

Short	1 – use the initialization file specified in the <code>RegIniName</code> property. 2 – use the registry entry specified in the <code>RegIniName</code> property.
-------	---

RegIniName

The initialization file (if `RegIniMode` is 1) or the registry key (if `RegIniMode` is 2) where the Viewing initialization information resides. The `RegIniMode` property must also be set. See [View Initialization Information, on page 22](#) for more information. This is a persistent property.

Returns

Type	Description
------	-------------

String	When <code>RegIniMode</code> is 1, this is the path and name of the initialization file. For example, <code>kvsdk.ini</code> or <code>C:\myprogram\myini.ini</code> . By default, Viewing looks for the initialization file in the Windows system directory. When <code>RegIniMode</code> is 2, this is the registry key under <code>HKEY_LOCAL_MACHINE\Software</code> . For example, <code>YourCompany\YourProduct</code> . Do not specify the complete registry name, only the <code>Company\Product</code> portion.
--------	--

Rotate

Rotates the currently opened document (fax or picture) and returns the current rotation angle.

Returns

Type	Description
Short	If the value is -1, the file cannot be rotated. To rotate the file, specify one of the following angles: 0, 90, 180, 270, or 360.

Sample Code

```
If KEYview1.Rotate > -1 Then ' Can the file be rotated?  
    KEYview1.Rotate = 180    ' Rotate 180 degrees.  
End If
```

SrcCharSet

Sets the source character set of a document to be opened. This property is used to specify the character set for documents when the character set cannot be determined by Viewing, such as in the case of plain text documents.

Returns

Type	Description
Integer	A value from the KVCharSet enumerated type. See the kvcharset.h file for a description.

SSDisplayGrid

Specifies whether the spreadsheet document displays gridlines. This is a persistent property.

Returns

Type	Description
Integer	0 – Do not display gridlines. 1 – Display gridlines. 2 – Display gridlines based on the setting in the original document.

SSDisplayHeaders

Specifies whether the spreadsheet document displays and prints headings, such as row numbers and column letters. This is a persistent property.

Returns

Type	Description
Integer	0 – Do not display headers.
	1 – Display headers.
	2 – Display headers based on the setting in the original document.

SSViewObjects

Specifies whether the spreadsheet document displays embedded graphic objects. This is a persistent property.

Returns

Type	Description
Integer	0 – Do not display graphic objects.
	1 – Display graphic objects.

TrgCharSet

Sets the target character set of a document to be opened. This property forces the character set Viewing uses to display a document. For example, this allows Japanese documents to be accurately displayed on an English Windows machine if the Japanese fonts are available.

Returns

Type	Description
Integer	A value from the KVCharSet enumerated type. See the kvcharset.h file for a description.

ViewPane

Specifies whether the preview pane of a container file is enabled. When the preview pane is enabled, the viewing area is divided into two panes: one pane displays the contents of the container file, the

other displays the contents of the selected subfile.

Returns

Type	Description
Boolean	TRUE – the preview pane is enabled. FALSE – the preview pane is not enabled.

WPCustomSize

Specifies the word processing document view in a custom size, but only if `WPViewMode` is set to **1** (page layout mode) and the `WPPageLayout` property is set to **10** (customized scaling factor). The range of acceptable values is 10 to 400 percent. This is a persistent property.

Returns

Type	Description
Integer	<i>n</i> – Specify a custom view size.

WPDisplayPict

Specifies whether pictures in a word processing document are displayed. This is a persistent property.

Returns

Type	Description
Integer	0 – Do not display pictures. 1 – Display pictures.

WPPageLayout

Specifies the page layout mode of the word processing document. This property applies only if `WPViewMode` is set to **1** (page layout mode). This is a persistent property.

Returns

Type	Description
Integer	0 – Display the document at full size (100%).

Type	Description
1	– Display the document at the current page width.
10	– Display the document at a customized scaling factor based on the value of the WPCustomSize property.

WPScaleTable

Specifies whether tables in the word processing document are scaled. This property applies only if [WPViewMode](#) is set to 0 (fit to window mode). In this mode, setting the property to TRUE scales tables within documents. This is a persistent property.

Returns

Type	Description
Boolean	TRUE – scale tables to fit window. FALSE – tables retain their original size.

WPViewMode

Specifies whether the word processing document fits to the window size or appears in page layout mode. This is a persistent property.

Returns

Type	Description
Integer	0 – Display the document in fit to window mode. 1 – Display the document in page layout mode.

Chapter 12: Control Events

This section describes the Viewing ActiveX control events. It includes the following topics:

- [Annotation](#) 238
- [KeyDown](#) 238
- [MouseUp](#) 239
- [OpenDocDone](#) 239
- [PageNumber](#) 240
- [PrintDone](#) 240
- [PrintDoneEx](#) 241
- [Selection](#) 241
- [TextBuffer](#) 241
- [UserClick](#) 242
- [ViewExtent](#) 242
- [ViewFile](#) 243

NOTE: The Viewing control's events in this chapter show the syntax in Visual Basic.

Annotation

These events are generated to report that the user clicked an annotation.

Syntax

`Annotation(bDoubleClick, lLogicalAddress)`

Parameters

Parameter	Type	Description
<code>bDoubleClick</code>	Boolean	This flag is TRUE if the user double-clicked, and FALSE if the user single-clicked.
<code>lLogicalAddress</code>	Long	The logical address of the mouse click.

KeyDown

These events are generated to indicate that a key is pressed. The `HotKeys` property must be set to **FALSE** before the file is opened for these events to be generated.

For more information about `HotKeys`, see [HotKeys](#), on page 226.

Syntax

`KeyDown(KeyCode, Shift)`

Parameters

Parameter	Type	Description
<code>KeyCode</code>	Integer	The virtual key code of the key that was pressed.
<code>Shift</code>	Integer	This parameter is always 0.

MouseUp

These events are generated to indicate that the right mouse button is released. The `ContextMenu` property must be set to `FALSE` before the file is opened for these events to be generated. Users can combine `MouseUp` and `ContextMenu` to disable `KeyView`'s context menu and implement their own.

For more information about `ContextMenu`, see [ContextMenu, on page 223](#).

Syntax

`MouseUp(Button, Shift, X, Y)`

Parameters

Parameter	Type	Description
<code>Button</code>	Short	This parameter should always be set to 2.
<code>Shift</code>	Short	This parameter should always be set to 0.
<code>X</code>	Long	The x-coordinate of the cursor when a mouse button was released. The coordinate is relative to the upper-left corner of the control window.
<code>Y</code>	Long	The y-coordinate of a cursor when a mouse button was released. The coordinate is relative to the upper-left corner of the control window.

OpenDocDone

These events are generated to indicate the percentage (0-100) of the document processed. These events are generated in response to the `Open` method call.

Syntax

`OpenDocDone(PercentageDone)`

Parameters

Parameter	Type	Description
PercentageDone	Short	The percentage of the document processed.

PageNumber

This event is generated in response to an `Open` method call. It indicates the total number of pages in the document and the page number at which the document is currently being viewed.

Syntax

`PageNumber(CurrentPage, TotalPages)`

Parameters

Parameter	Type	Description
CurrentPage	Short	The current page number of the document.
TotalPages	Short	The total number of pages in the document. This number reflects the total number of processed pages, until the <code>OpenDocDone</code> event reaches 100.

PrintDone

This event indicates that the printing of the document is complete.

NOTE: This event does not return any status reports. If you want to receive a status report about your print job, use [PrintDoneEx](#).

Syntax

`PrintDone()`

Parameters

None

PrintDoneEx

This event indicates that the printing of the document is complete, and reports the status of the print job.

NOTE: If you do not want to receive status reports, use [PrintDone](#).

Syntax

```
PrintDoneEx(printStatus)
```

Parameters

Parameter	Type	Description
printStatus	Short	Returns one of four status reports: 0: General error 1: Success 2: Printing aborted by user 3: Printing aborted due to Windows GDI call failure

Selection

These events are generated to report that the user changed the selection state.

Syntax

```
Selection(bHaveSelection, lStart, lEnd)
```

Parameters

Parameter	Type	Description
BHaveSelection	Boolean	The flag is TRUE if a selection exists, or FALSE if a selection does not exist.
lStart	Long	The logical address of the start of the view.
lEnd	Long	The logical address of the end of the view.

TextBuffer

This event returns a text buffer.

Syntax

```
TextBuffer(lBaseLogicalAddress, szBuffer)
```

Parameters

Parameter	Type	Description
lBaseLogicalAddress	Long	The logical address of the text buffer.
szBuffer	String	A pointer to the text buffer.

UserClick

These events are generated to report that the user clicked the mouse on the document. To generate the events, set the [OPENMode](#) property to 5 before the document is opened. This indicates that the file should be opened with indexing enabled.

Syntax

```
UserClick(bDoubleClick, lLogicalAddress)
```

Parameters

Parameter	Type	Description
bDoubleClick	Boolean	The flag is TRUE if the user double-clicked, and FALSE if the user single-clicked.
lLogicalAddress	Long	The logical address of the mouse click.

ViewExtent

These events are generated to report that the user changed the view extent.

Syntax

```
ViewExtent(lStart, lEnd)
```

Parameters

Parameter	Type	Description
lStart	Long	The logical address of the start of the view.
lEnd	Long	The logical address of the end of the view.

ViewFile

This event is generated in response to a user double-clicking on the contents of a displayed container file or using the UnZip method. You can then respond to this event to Open the file.

Syntax

```
ViewFile(FileName, DeleteFile)
```

Parameters

Parameter	Type	Description
FileName	String	The path and file name of the file that the user double-clicked within the container file.
DeleteFile	Boolean	If this is TRUE, delete the file when you are finished with it (it is a temporary file).

Part IV: Appendixes

This section provides information on files required for redistribution, character sets, and supported and detected formats, and includes the following appendixes:

- [Supported Formats](#)
- [Document Readers](#)
- [Character Sets](#)
- [File Format Detection](#)
- [Extract and Format Lotus Notes Subfiles](#)
- [List of Files Required for Redistribution](#)
- [Configuration Options in kvsdk.ini](#)
- [Password Protected Files](#)

Appendix A: Supported Formats

This section lists the file formats that KeyView can detect.

- [Key to Supported Formats Table](#)245
- [Supported Formats](#)246

Key to Supported Formats Table

The supported formats table includes the following information:

Column	Description
Format Name	The format name that is returned by KeyView format detection. <ul style="list-style-type: none">• In the C API, these values are defined in the <code>ENdocFmt</code> enumeration in <code>adDocFmt.h</code>.
Number	The format number that is returned by KeyView format detection. This is the value associated with the Format Name in the relevant enumeration.
Category	This value is used in the KeyView configuration file <code>formats.ini</code> to specify the reader to use to filter, export, or view the format. Several formats might have the same category value.
Description	A short description of the file format.
MIME Type	The MIME type (if any).
Extension	A list of common file extensions for the file format. NOTE: This is not a complete list of file extensions. KeyView does not distinguish between file types based on their extension. Instead, it detects the file format based on the file content. This is more reliable because content cannot always be predicted from the file extension, and because some file extensions are associated with multiple formats.
File Class	The KeyView file class. <ul style="list-style-type: none">• In the C API, these values are defined in the <code>ENdocClass</code> enumeration in <code>adinfo.h</code>.

Supported Formats

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Reserved__Fmt	-1	-1				AutoDetNoFormat	
Unknown_Fmt	0	0				AutoDetNoFormat	
AES_Multiplus_Comm_Fmt	1	1	Multiplus (AES)		PTF	adWORDPROCESSOR	
ASCII_Text_Fmt	2	2	Plain Text file	text/plain	TXT	adWORDPROCESSOR	afsr
MSDOS_Batch_File_Fmt	3	2	MS-DOS Batch File	application/x-bat	BAT	adEXECUTABLE	afsr
Applix_Alis_Fmt	4	3	Applix Asterix		AX	adWORDPROCESSOR	axsr
BMP_Fmt	5	4	Windows Bitmap Image (BMP)	image/bmp	BMP	adRASTERIMAGE	bmpr , kpbmprdr
CT_DEF_Fmt	6	5	Convergent Technologies DEF Comm. Format			adWORDPROCESSOR	cdsr
Corel_Draw_Fmt	7	6	Corel Draw (up to version 13/X3)	application/coreldraw	CDR	adVECTORGRAPHIC	kpcdrdr
CGM_ClearText_Fmt	8	8	Computer Graphics Metafile (CGM)		CGM	adVECTORGRAPHIC	kpcgmrdr
CGM_Binary_Fmt	9	8	Computer Graphics Metafile (CGM)	image/cgm	CGM	adVECTORGRAPHIC	kpcgmrdr
CGM_Character_Fmt	10	8	Computer Graphics Metafile (CGM)		CGM	adVECTORGRAPHIC	kpcgmrdr
Word_Connection_Fmt	11	9	Word Connection		CN	adWORDPROCESSOR	stringssr
COMET_TOP_Word_Fmt	12	10	Nixdorf COMET TOP Financial Accounting software			adWORDPROCESSOR	
CEOwrite_Fmt	13	11	CEOwrite		CW	adWORDPROCESSOR	stringssr
DSA101_Fmt	14	12	DSA101 (Honeywell Bull)			adWORDPROCESSOR	stringssr
DCA_RFT_Fmt	15	13	DCA-RFT (IBM Revisable Form)	application/dca-rft	RFT, DC	adWORDPROCESSOR	dcasr
CDA_DDIF_Fmt	16	14	CDA / DDIF		DDIF	adWORDPROCESSOR	
DG_CDS_Fmt	17	16	DG Common Data Stream		CDS	adWORDPROCESSOR	stringssr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
			(CDS)				
Micrografx_Draw_Fmt	18	18	Windows Draw (Micrografx)		DRW	adVECTORGRAPHIC	
Data_Point_VistaWord_Fmt	19	19	Vistaword		DV	adWORDPROCESSOR	stringssr
DECdx_Fmt	20	20	DEC WPS Plus DX format		DX	adWORDPROCESSOR	
Enable_WP_Fmt	21	21	Enable Word Processing		WPF	adWORDPROCESSOR	stringssr
EPSF_Fmt	22	22	Encapsulated PostScript	application/postscript	EPS	AutoDetNoFormat	kpepsrdr
Preview_EPSF_Fmt	23	22	Encapsulated PostScript	application/postscript		AutoDetNoFormat	kpepsrdr
MS_Executable_Fmt	24	23	MSDOS/Windows Program	application/x-msdownload	EXE	adEXECUTABLE	exesr
G31D_Fmt	25	24	CCITT G3 1D			adRASTERIMAGE	
GIF_87a_Fmt	26	25	Graphics Interchange Format (GIF87a)	image/gif	GIF	adRASTERIMAGE	gifsr , kpgifdr
GIF_89a_Fmt	27	25	Graphics Interchange Format (GIF89a)	image/gif	GIF	adRASTERIMAGE	gifsr , kpgifdr
HP_Word_PC_Fmt	28	26	HP Word PC		HW	adWORDPROCESSOR	stringssr
IBM_1403_LinePrinter_Fmt	29	27	IBM 1403 Line Printer		I4	adWORDPROCESSOR	
IBM_DCF_Script_Fmt	30	28	DCF Script		IC	adWORDPROCESSOR	stringssr
IBM_DCA_FFT_Fmt	31	29	DCA-FFT (IBM Final Form)		IF, FFT	adWORDPROCESSOR	
Interleaf_Fmt	32	30	Interleaf			adWORDPROCESSOR	
GEM_Image_Fmt	33	31	GEM Bit Image		IMG	adRASTERIMAGE	
IBM_Display_Write_Fmt	34	32	Display Write		IP	adWORDPROCESSOR	dw4sr
Sun_Raster_Fmt	35	33	Sun Raster	image/x-cmu-raster	RAS, RS, SUN	adRASTERIMAGE	kpsunrdr
Ami_Pro_Fmt	36	35	Lotus Ami Pro	application/x-lotus-amipro	SAM	adWORDPROCESSOR	lasr
Ami_Pro_StyleSheet_Fmt	37	35	Lotus Ami Pro Style Sheet			adWORDPROCESSOR	lasr
MORE_Fmt	38	36	MORE Database MAC			adOUTLINE	
Lyrix_Fmt	39	37	Lyrix Word Processing			adWORDPROCESSOR	stringssr
MASS_11_Fmt	40	38	MASS-11		M1	adWORDPROCESSOR	stringssr
MacPaint_Fmt	41	39	MacPaint		PNTG	adRASTERIMAGE	kpmacrdr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
MS_Word_Mac_Fmt	42	40	Microsoft Word for Macintosh (up to version 3)	application/msword	DOC	adWORDPROCESSOR	mbsr
SmartWare_II_Comm_Fmt	43	41	SmartWare II			adCOMMUNICATION	
MS_Word_Win_Fmt	44	42	Microsoft Word for Windows (up to version 6)	application/msword	DOC, WPS	adWORDPROCESSOR	misr
Multimate_Fmt	45	43	MultiMate		MM	adWORDPROCESSOR	stringssr
Multimate_Fnote_Fmt	46	43	MultiMate Footnote File		MMFN	adWORDPROCESSOR	stringssr
Multimate_Adv_Fmt	47	43	MultiMate Advantage			adWORDPROCESSOR	stringssr
Multimate_Adv_Fnote_Fmt	48	43	MultiMate Advantage Footnote File			adWORDPROCESSOR	stringssr
Multimate_Adv_II_Fmt	49	43	MultiMate Advantage II			adWORDPROCESSOR	stringssr
Multimate_Adv_II_Fnote_Fmt	50	43	MultiMate Advantage II Footnote File		FBX, FNX	adWORDPROCESSOR	stringssr
Multiplan_PC_Fmt	51	44	Multiplan (PC)			adSPREADSHEET	
Multiplan_Mac_Fmt	52	44	Multiplan (Mac)			adSPREADSHEET	
MS_RTF_Fmt	53	45	Rich Text Format (RTF)	application/rtf	RTF	adWORDPROCESSOR	rtfsr
MS_Word_PC_Fmt	54	46	Microsoft Word for PC (up to version 6)	application/x-ms-wordpc	MW	adWORDPROCESSOR	mwsr
MS_Word_PC_StyleSheet_Fmt	55	46	Microsoft Word for PC (up to version 6) Style Sheet			adWORDPROCESSOR	mwsr
MS_Word_PC_Glossary_Fmt	56	46	Microsoft Word for PC (up to version 6) Glossary			adWORDPROCESSOR	mwsr
MS_Word_PC_Driver_Fmt	57	46	Microsoft Word for PC (up to version 6) Driver			adWORDPROCESSOR	mwsr
MS_Word_PC_Misc_Fmt	58	46	Microsoft Word for PC (up to version 6) Miscellaneous File			adWORDPROCESSOR	mwsr
NBI_Async_Archive_Fmt	59	47	NBI Async Archive Format			adWORDPROCESSOR	
Navy_DIF_Fmt	60	48	Navy DIF (document interchange format)		ND	adWORDPROCESSOR	stringssr
NBI_Net_Archive_Fmt	61	49	NBI OASys Net Archive Format		NN	adWORDPROCESSOR	nnsr
NIOS_TOP_Fmt	62	50	NIOS TOP			adWORDPROCESSOR	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
FileMaker_Mac_Fmt	63	51	Filemaker MAC		FP5, FP7	adDATABASE	
ODA_Q1_11_Fmt	64	52	ODA / ODIF Q1 11		OD	adWORDPROCESSOR	stringssr
ODA_Q1_12_Fmt	65	52	ODA / ODIF Q1 12		OD	adWORDPROCESSOR	stringssr
OLIDIF_Fmt	66	53	OLIDIF (Olivetti)			adWORDPROCESSOR	
Office_Writer_Fmt	67	55	Office Writer		OW	adWORDPROCESSOR	stringssr
PC_Paintbrush_Fmt	68	56	PC Paintbrush Graphics (PCX)	image/vnd.zbrush.pcx	PCX	adRASTERIMAGE	kppcxldr
CPT_Comm_Fmt	69	57	CPT Corporation word processor		PF	adWORDPROCESSOR	stringssr
Lotus_PIC_Fmt	70	58	Lotus PIC	image/x-pict	PIC	adVECTORGRAPHIC	kppicldr
Mac_PICT_Fmt	71	59	QuickDraw Picture	image/x-pict	PCT	AutoDetNoFormat	kppctrdr
Philips_Script_Word_Fmt	72	60	Philips Script			adWORDPROCESSOR	
PostScript_Fmt	73	61	PostScript	application/postscript	PS	adVECTORGRAPHIC	
PRIMEWORD_Fmt	74	62	PRIMEWORD			adWORDPROCESSOR	pwsr
Quadratron_Q_One_v1_Fmt	75	63	Q-One V1.93J		Q1, QX	adWORDPROCESSOR	stringssr
Quadratron_Q_One_v2_Fmt	76	64	Q-One V2.0		Q1, QX	adWORDPROCESSOR	stringssr
SAMNA_Word_IV_Fmt	77	65	SAMNA Word		SAM	adWORDPROCESSOR	stringssr
Ami_Pro_Draw_Fmt	78	66	Lotus Ami Pro Draw		SDW	adVECTORGRAPHIC	kpsdwrdr
SYLK_Spreadsheet_Fmt	79	67	SYmbolic LinK (SYLK) format		SLK	adSPREADSHEET	
SmartWare_II_WP_Fmt	80	68	Informix SmartWare II word processor		DOC, SMT	adWORDPROCESSOR	swsr
Symphony_Fmt	81	69	Lotus Symphony spreadsheet		WR1	adSPREADSHEET	
Targa_Fmt	82	70	Targa image	image/x-tga	TGA	adRASTERIMAGE	kpTGArdr
TIFF_Fmt	83	71	Tag Image File Format (TIFF)	image/tiff	TIF, TIFF	adRASTERIMAGE	kptifdr , tifsr
Targon_Word_Fmt	84	72	Targon Word		TW	adWORDPROCESSOR	stringssr
Uniplex_Ucalc_Fmt	85	73	Uniplex Ucalc		SS	adSPREADSHEET	
Uniplex_WP_Fmt	86	74	Uniplex word processor		UP	adWORDPROCESSOR	stringssr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
MS_Word_UNIX_Fmt	87	75	Microsoft Word UNIX	application/msword		adWORDPROCESSOR	
WANG_PC_Fmt	88	76	WANG PC			adWORDPROCESSOR	
WordERA_Fmt	89	77	WordERA		DC, GL, FR	adWORDPROCESSOR	stringssr
WANG_WPS_Comm_Fmt	90	78	WANG WPS		WF	adWORDPROCESSOR	stringssr
WordPerfect_Mac_Fmt	91	79	WordPerfect MAC	application/x-corel-wordperfect		adWORDPROCESSOR	wpmsr
WordPerfect_Fmt	92	86	WordPerfect version 4	application/x-corel-wordperfect	WP, WP4	adWORDPROCESSOR	stringssr
WordPerfect_VAX_Fmt	93	139	WordPerfect VAX	application/x-corel-wordperfect		adWORDPROCESSOR	
WordPerfect_Macro_Fmt	94	139	WordPerfect Macro	application/vnd.wordperfect	MRS	adWORDPROCESSOR	
WordPerfect_Dictionary_Fmt	95	139	WordPerfect Spelling Dictionary	application/vnd.wordperfect	SPW	adWORDPROCESSOR	
WordPerfect_Thesaurus_Fmt	96	139	WordPerfect Thesaurus	application/vnd.wordperfect		adWORDPROCESSOR	
WordPerfect_Resource_Fmt	97	139	WordPerfect Resource File	application/vnd.wordperfect	WWK, PRS	adWORDPROCESSOR	
WordPerfect_Driver_Fmt	98	139	WordPerfect Driver	application/vnd.wordperfect	IRS, VRS	adWORDPROCESSOR	
WordPerfect_Cfg_Fmt	99	139	WordPerfect Configuration File	application/vnd.wordperfect	PFX	adWORDPROCESSOR	
WordPerfect_Hyphenation_Fmt	100	139	WordPerfect Hyphenation Dictionary	application/vnd.wordperfect	HYC	adWORDPROCESSOR	
WordPerfect_Misc_Fmt	101	139	WordPerfect Miscellaneous File	application/vnd.wordperfect		adWORDPROCESSOR	
WordMARC_Fmt	102	82	WordMARC Composer	video/x-ms-wm	WM, PW	adWORDPROCESSOR	stringssr
Windows_Metatile_Fmt	103	83	Windows Metafile	image/wmf	WMF	adVECTORGRAPHIC	kpwmfrdr
Windows_Metatile_NoHdr_Fmt	104	83	Windows Metafile (no header)	image/wmf	WMF	adVECTORGRAPHIC	kpwmfrdr
SmartWare_II_DB_Fmt	105	84	Informix SmartWare II database			adDATABASE	
WordPerfect_Graphics_Fmt	106	195	WordPerfect Graphics (version 2 and higher)	application/vnd.wordperfect	WPG, QPG	AutoDetNoFormat	kpwg2rdr , kpwpgdrdr
WordStar_Fmt	107	87	WordStar		WS, WSD	adWORDPROCESSOR	stringssr
WANG_WITA_Fmt	108	88	WANG WITA		WT	adWORDPROCESSOR	stringssr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Xerox_860_Comm_Fmt	109	89	Xerox 860			adWORDPROCESSOR	stringssr
Xerox_Writer_Fmt	110	91	Xerox Writer			adWORDPROCESSOR	stringssr
DIF_SpreadSheet_Fmt	111	92	Data Interchange Format (DIF)	application/dif+xml	DIF	adSPREADSHEET	difsr
Enable_Spreadsheet_Fmt	112	93	Enable Spreadsheet	application/vnd.epson.ssf	SSF	adSPREADSHEET	
SuperCalc_Fmt	113	94	Sorcim SuperCalc spreadsheet		CAL	adSPREADSHEET	
UltraCalc_Fmt	114	95	UltraCalc spreadsheet			adSPREADSHEET	
SmartWare_II_SS_Fmt	115	96	Informix SmartWare II spreadsheet			adSPREADSHEET	
SOF_Encapsulation_Fmt	116	97	Serialized Object Format (SOF)	application/java-serialized-object	SOF	adENCAPSULATION	
PowerPoint_Win_Fmt	117	98	Microsoft PowerPoint PC (up to version 4)	application/x-ms-powerpoint	PPT	adPRESENTATION	kpp40rdr
PowerPoint_Mac_Fmt	118	99	Microsoft PowerPoint MAC (up to version 4)	application/x-ms-powerpoint	PPT	adPRESENTATION	
PowerPoint_95_Fmt	119	212	Microsoft PowerPoint 95	application/x-ms-powerpoint	PPT	adPRESENTATION	kpp95rdr
PowerPoint_97_Fmt	120	272	Microsoft PowerPoint 97	application/x-ms-powerpoint	PPT	adPRESENTATION	kpp97rdr
PageMaker_Mac_Fmt	121	100	PageMaker for Macintosh			adDESKTOPPUBLSH	
PageMaker_Win_Fmt	122	101	PageMaker for Windows			adDESKTOPPUBLSH	
MS_Works_Mac_WP_Fmt	123	103	Microsoft Works Word Processor for MAC	application/x-msworks	MWK	adWORDPROCESSOR	stringssr
MS_Works_Mac_DB_Fmt	124	104	Microsoft Works Database for MAC	application/x-msworks		adDATABASE	
MS_Works_Mac_SS_Fmt	125	105	Microsoft Works Spreadsheet for MAC	application/x-msworks		adSPREADSHEET	mwssr
MS_Works_Mac_Comm_Fmt	126	106	Microsoft Works Communication for MAC	application/x-msworks		adCOMMUNICATION	
MS_Works_DOS_WP_Fmt	127	107	Microsoft Works Word Processor for DOS	application/x-msworks	WPS	adWORDPROCESSOR	stringssr
MS_Works_DOS_DB_Fmt	128	108	Microsoft Works Database for DOS	application/x-msworks	WDB	adDATABASE	
MS_Works_DOS_SS_Fmt	129	109	Microsoft Works	application/x-msworks		adSPREADSHEET	mwssr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Fmt			Spreadsheet for DOS				
MS_Works_Win_WP_Fmt	130	227	Microsoft Works Word Processor for Windows	application/x-msworks	WPS, W40	adWORDPROCESSOR	msw6sr , mswsr
MS_Works_Win_DB_Fmt	131	231	Microsoft Works Database for Windows	application/x-msworks		adDATABASE	
MS_Works_Win_SS_Fmt	132	228	Microsoft Works Spreadsheet for Windows	application/x-msworks	WKS, S30, S40	adSPREADSHEET	mwssr
PC_Library_Fmt	133	111	DOS/Windows Object Library	application/x-archive	LIB, A	adLIBRARY	
MacWrite_Fmt	134	112	MacWrite	application/macwriteii		adWORDPROCESSOR	stringssr
MacWrite_II_Fmt	135	113	MacWrite II	application/macwriteii		adWORDPROCESSOR	stringssr
Freehand_Fmt	136	114	Freehand MAC	image/x-freehand		adVECTORGRAPHIC	
Disk_Doubler_Fmt	137	115	Disk Doubler			adENCAPSULATION	
HP_GL_Fmt	138	116	HP Graphics Language	vector/x-hpgl	HPGL, HPG	adVECTORGRAPHIC	
FrameMaker_Fmt	139	136	FrameMaker	application/vnd.frameMaker	FM, FRM	adDESKTOPPUBLSH	
FrameMaker_Book_Fmt	140	136	FrameMaker Book	application/vnd.frameMaker	BOOK	adDESKTOPPUBLSH	
Maker_Markup_Language_Fmt	141	174	Maker Markup Language	application/vnd.mif		adDESKTOPPUBLSH	
Maker_Interchange_Fmt	142	117	Maker Interchange Format (MIF)	application/x-mif	MIF	adWORDPROCESSOR	mifsr
JPEG_File_Interchange_Fmt	143	118	JPEG Interchange Format	image/jpeg	JPG, JPEG	adRASTERIMAGE	jpgsr , kpjpgdr
Reflex_Fmt	144	119	Borland Reflex database			adDATABASE	
Framework_Fmt	145	276	Framework office suite			adMIXED	
Framework_II_Fmt	146	120	Framework II office suite		FW3	adMIXED	
Paradox_Fmt	147	121	Borland Paradox database		DB	adDATABASE	
MS_Windows_Write_Fmt	148	123	Microsoft Windows Write	application/x-ms-write	WRI	adWORDPROCESSOR	mwsr
Quattro_Pro_DOS_Fmt	149	124	Quattro Pro for DOS	application/x-quattropro	WQ1	adSPREADSHEET	
Quattro_Pro_Win_Fmt	150	184	Quattro Pro for Windows	application/x-quattro-win	WB1, WB2, WB3	adSPREADSHEET	qpssr
Persuasion_Fmt	151	126	Adobe Persuasion			adPRESENTATION	
Windows_Icon_Fmt	152	128	Windows Icon Format	image/ico	ICO	adRASTERIMAGE	kpicordr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Windows_Cursor_Fmt	153	133	Windows Cursor	image/x-win-bitmap	CUR	adRASTERIMAGE	
MS_Project_Activity_Fmt	154	129	Microsoft Project (up to version 3) activity file			adSCHEDULE	
MS_Project_Resource_Fmt	155	129	Microsoft Project (up to version 3) resource file			adSCHEDULE	
MS_Project_Calc_Fmt	156	129	Microsoft Project (up to version 3) calc file			adSCHEDULE	
PKZIP_Fmt	157	132	ZIP Archive	application/zip	ZIP, ZIPX	adENCAPSULATION	unzip
Quark_Xpress_Fmt	158	134	Quark Xpress MAC			adDESKTOPPUBLISH	
ARC_PAK_Archive_Fmt	159	135	PAK/ARC Archive		ARC, PAK	adENCAPSULATION	
MS_Publisher_Fmt	160	137	Microsoft Publisher (up to version 3)	application/x-mspublisher	PUB	adDESKTOPPUBLISH	mspubsr
PlanPerfect_Fmt	161	138	PlanPerfect			adSCHEDULE	
WordPerfect_Auxiliary_Fmt	162	139	WordPerfect auxiliary file		WPW	adMISC	
MS_WAVE_Audio_Fmt	163	141	Microsoft Wave	audio/wav	WAV	adSOUND	MCI , niffsr
MIDI_Audio_Fmt	164	142	MIDI audio	audio/mid	MID, MIDI	adSOUND	MCI
AutoCAD_DXF_Binary_Fmt	165	143	AutoCAD DXF	image/x-dxf	DXF	adVECTORGRAPHIC	kpDXFrdr , kpODArdr
AutoCAD_DXF_Text_Fmt	166	143	AutoCAD DXF	image/x-dxf	DXF	adVECTORGRAPHIC	kpDXFrdr , kpODArdr
dBase_Fmt	167	144	dBase	application/x-dbf	DBF, VCX	adDATABASE	dbfsr
OS_2_PM_Metfile_Fmt	168	145	OS/2 PM Metfile		MET	adVECTORGRAPHIC	
Lasergraphics_Language_Fmt	169	146	Lasergraphics Language			adVECTORGRAPHIC	
AutoShade_Rendering_Fmt	170	147	AutoShade Rendering			adVECTORGRAPHIC	
GEM_VDI_Fmt	171	148	GEM VDI Metfile image		GEM, GDI	adVECTORGRAPHIC	
Windows_Help_Fmt	172	149	Windows Help File	application/winhelp	HLP	adMISC	
Volkswriter_Fmt	173	150	Volkswriter word processor		VW4	adWORDPROCESSOR	stringsr
Ability_WP_Fmt	174	151	Ability Word Processor			adWORDPROCESSOR	
Ability_DB_Fmt	175	151	Ability Database			adDATABASE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Ability_SS_Fmt	176	151	Ability Spreadsheet			adSPREADSHEET	
Ability_Comm_Fmt	177	151	Ability Presentation			adCOMMUNICATION	
Ability_Image_Fmt	178	151	Ability Image			adRASTERIMAGE	
XyWrite_Fmt	179	152	XYWrite / Nota Bene		XY4	adWORDPROCESSOR	xywsr
CSV_Fmt	180	153	CSV (Comma Separated Values)	text/csv	CSV	adSPREADSHEET	csvsr
IBM_Writing_Assistant_Fmt	181	154	IBM Writing Assistant		IWA	adWORDPROCESSOR	stringsr
WordStar_2000_Fmt	182	155	WordStar 2000		WS2	adWORDPROCESSOR	stringsr
HP_PCL_Fmt	183	157	HP Printer Control Language	application/pcl	PCL	adVECTORGRAPHIC	
UNIX_Exe_PreSysV_VAX_Fmt	184	158	Unix Executable (PDP-11/pre-System V VAX)	application/octet-stream		adEXECUTABLE	
UNIX_Exe_Basic_16_Fmt	185	158	Unix Executable (Basic-16)	application/octet-stream		adEXECUTABLE	
UNIX_Exe_x86_Fmt	186	158	Unix Executable (x86)	application/octet-stream		adEXECUTABLE	
UNIX_Exe_iAPX_286_Fmt	187	158	Unix Executable (iAPX 286)	application/octet-stream		adEXECUTABLE	
UNIX_Exe_MC68k_Fmt	188	158	Unix Executable (MC680x0)	application/octet-stream		adEXECUTABLE	
UNIX_Exe_3B20_Fmt	189	158	Unix Executable (3B20)	application/octet-stream		adEXECUTABLE	
UNIX_Exe_WE32000_Fmt	190	158	Unix Executable (WE32000)	application/octet-stream		adEXECUTABLE	
UNIX_Exe_VAX_Fmt	191	158	Unix Executable (VAX)	application/octet-stream		adEXECUTABLE	
UNIX_Exe_Bell_5_Fmt	192	158	Unix Executable (Bell 5.0)	application/octet-stream		adEXECUTABLE	
UNIX_Obj_VAX_Demand_Fmt	193	159	Unix Object Module (VAX Demand)			adOBJECTMODULE	
UNIX_Obj_MS8086_Fmt	194	159	Unix Object Module (old MS 8086)			adOBJECTMODULE	
UNIX_Obj_Z8000_Fmt	195	159	Unix Object Module (Z8000)			adOBJECTMODULE	
AU_Audio_Fmt	196	161	NeXT/Sun Audio Data	audio/basic	AU, SND	adSOUND	MCI
NeWS_Font_Fmt	197	162	NeWS bitmap font			adFONT	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
cpio_Archive_CRChdr_Fmt	198	163	cpio archive (CRC Header)	application/x-cpio		adENCAPSULATION	
cpio_Archive_CHRhdr_Fmt	199	163	cpio archive (CHR Header)	application/x-cpio		adENCAPSULATION	
PEX_Binary_Archive_Fmt	200	164	SUN PEX Binary Archive			adENCAPSULATION	
Sun_vfont_Fmt	201	165	SUN vfont Definition			adFONT	
Curses_Screen_Fmt	202	166	Curses Screen Image			adRASTERIMAGE	
UUEncoded_Fmt	203	167	UU encoded	text/x-uencode	UUE	adENCAPSULATION	uuds
WriteNow_Fmt	204	168	WriteNow MAC			adWORDPROCESSOR	strings
PC_Obj_Fmt	205	169	DOS/Windows Object Module	application/octet-stream	OBJ	adOBJECTMODULE	
Windows_Group_Fmt	206	170	Windows Group			adMISC	
TrueType_Font_Fmt	207	171	TrueType Font	application/x-font-ttf	TTF	adFONT	
Windows_PIF_Fmt	208	172	Program Information File (PIF)	application/octet-stream	PIF	adMISC	
MS_COM_Executable_Fmt	209	173	PC (.COM)	application/octet-stream	COM	adEXECUTABLE	
Stuftit_Fmt	210	175	Stuftit (MAC)	application/x-stuftit	HQX	adENCAPSULATION	
PeachCalc_Fmt	211	176	PeachCalc		CAL	adSPREADSHEET	
Wang_GDL_Fmt	212	177	WANG Office GDL Header			adENCAPSULATION	
Q_A_DOS_Fmt	213	179	Q & A for DOS			adWORDPROCESSOR	strings
Q_A_Win_Fmt	214	180	Q & A for Windows		JW	adWORDPROCESSOR	strings
WPS_PLUS_Fmt	215	181	WPS-PLUS	application/vnd.ms-wpl	WPL	adWORDPROCESSOR	strings
DCX_Fmt	216	182	DCX FAX Format(PCX images)	image/dcx	DCX	adFAXFORMAT	kpdcxrdr
OLE_Fmt	217	183	OLE Compound Document		OLE	adENCAPSULATION	oles
EBCDIC_Fmt	218	186	EBCDIC Text			adWORDPROCESSOR	
DCS_Fmt	219	187	DCS			adWORDPROCESSOR	
UNIX_SHAR_Fmt	220	190	SHAR shell archive format	application/x-shar	SHAR	adENCAPSULATION	
Lotus_Notes_BitMap_Fmt	221	191	Lotus Notes Bitmap			adRASTERIMAGE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Lotus_Notes_CDF_Fmt	222	193	Lotus Notes CDF	application/cdf	CDF	adWORDPROCESSOR	stringssr
Compress_Fmt	223	192	Unix Compress	application/x-compress	Z	adENCAPSULATION	kvzee , kvzeesr
GZ_Compress_Fmt	224	198	GZ Compress	application/gzip	GZ	adENCAPSULATION	kvgz , kvgzsr
TAR_Fmt	225	194	TAR archive	application/tar	TAR	adENCAPSULATION	tarsr
ODIF_FOD26_Fmt	226	196	Open Document Architecture (ODA / ODIF) FOD26	application/oda	F26	adWORDPROCESSOR	
ODIF_FOD36_Fmt	227	196	Open Document Architecture (ODA / ODIF) FOD36	application/oda	F36	adWORDPROCESSOR	
ALIS_Fmt	228	197	ALIS			adWORDPROCESSOR	
Envoy_Fmt	229	199	WordPerfect Envoy	application/envoy	EVY	adWORDPROCESSOR	
PDF_Fmt	230	200	Portable Document Format	application/pdf	PDF	adWORDPROCESSOR	kppdf2rdr , kppdfdrdr , pdf2sr , pdfsr
BinHex_Fmt	231	206	BinHex	application/mac-binhex40	HQX	adENCAPSULATION	kvhqsr
SMTP_Fmt	232	207	SMTP	message/rfc822	SMTP	adENCAPSULATION	emlsr
MIME_Fmt	233	208	MIME (EML, MBX email) ¹	message/rfc822	EML, MBX	adENCAPSULATION	mbxsr
USENET_Fmt	234	264	USENET	message/news		adWORDPROCESSOR	
SGML_Fmt	235	209	SGML	text/sgml	SGML	adWORDPROCESSOR	afsr
HTML_Fmt	236	210	HTML	text/html	HTM, HTML	adWORDPROCESSOR	htmsr
ACT_Fmt	237	211	ACT! CRM software		ACT	adWORDPROCESSOR	
PNG_Fmt	238	213	Portable Network Graphics (PNG)	image/png	PNG	adRASTERIMAGE	kppngrdr , pngsr
MS_Video_Fmt	239	214	Video for Windows (AVI)	video/avi	AVI	adMOVIE	MCI
Windows_Animated_Cursor_Fmt	240	215	Windows Animated Cursor		ANI	adRASTERIMAGE	kpanirdr
Windows_CPP_Obj_Storage_Fmt	241	216	Windows C++ Object Storage			adMIXED	
Windows_Palette_Fmt	242	217	Windows Palette		PAL	adRASTERIMAGE	
RIFF_DIB_Fmt	243	218	RIFF Device Independent Bitmap			adRASTERIMAGE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
RIFF_MIDI_Fmt	244	219	RIFF MIDI	audio/midi	RMI	adSOUND	
RIFF_Multimedia_Movie_Fmt	245	220	RIFF Multimedia Movie		MMM	adMOVIE	
MPEG_Fmt	246	221	MPEG Movie	video/mpeg		adMOVIE	
QuickTime_Fmt	247	222	QuickTime Movie, MPEG-4 audio	video/quicktime	MOV, QT, MP4	adMOVIE	MCI , mpeg4sr
AIFF_Fmt	248	223	Audio Interchange File Format (AIFF)	audio/aiff	AIF, AIFF, AIFC	adSOUND	MCI , aiffr
Amiga_MOD_Fmt	249	224	Amiga MOD		MOD	adSOUND	
Amiga_IFF_8SVX_Fmt	250	225	Amiga IFF (8SVX) Sound	audio/x-8svx	IFF	adSOUND	
Creative_Voice_Audio_Fmt	251	226	Creative Voice (VOC)		VOC	adSOUND	
AutoDesk_Animator_FLI_Fmt	252	229	AutoDesk Animator FLIC	video/x-fli	FLI	adANIMATION	
AutoDesk_AnimatorPro_FLC_Fmt	253	230	AutoDesk Animator Pro FLIC	video/x-flc	FLC	adANIMATION	
Compactor_Archive_Fmt	254	233	Compactor / Compact Pro	application/mac-compactpro		adENCAPSULATION	
VRML_Fmt	255	234	VRML	model/vrml	WRL	adVECTORGRAPHIC	
QuickDraw_3D_Metafile_Fmt	256	235	QuickDraw 3D Metafile			adVECTORGRAPHIC	
PGP_Secret_Keyring_Fmt	257	236	PGP Secret Keyring	application/pgp		adENCAPSULATION	
PGP_Public_Keyring_Fmt	258	237	PGP Public Keyring	application/pgp		adENCAPSULATION	
PGP_Encrypted_Data_Fmt	259	238	PGP Encrypted Data	application/pgp		adENCAPSULATION	
PGP_Signed_Data_Fmt	260	239	PGP Signed Data	application/pgp		adENCAPSULATION	
PGP_SignedEncrypted_Data_Fmt	261	240	PGP Signed and Encrypted Data	application/pgp		adENCAPSULATION	
PGP_Sign_Certificate_Fmt	262	241	PGP Signature Certificate	application/pgp-signature	SIG	adENCAPSULATION	
PGP_Compressed_Data_Fmt	263	246	PGP Compressed Data	application/pgp		adENCAPSULATION	
PGP_ASCII_Public_	264	242	ASCII-armored PGP Public	application/pgp	PGP	adENCAPSULATION	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Keyring_Fmt			Keyring				
PGP_ASCII_Encoded_Fmt	265	243	ASCII-armored PGP encoded	application/pgp		adENCAPSULATION	
PGP_ASCII_Signed_Fmt	266	244	ASCII-armored PGP signed	application/pgp		adENCAPSULATION	
OLE_DIB_Fmt	267	245	OLE DIB object			adRASTERIMAGE	
SGI_Image_Fmt	268	247	SGI Image	image/sgi	RGB	adRASTERIMAGE	kpsgirdr
Lotus_ScreenCam_Fmt	269	248	Lotus ScreenCam	application/vnd.lotus-screencam	SCM	adANIMATION	
MPEG_Audio_Fmt	270	249	MPEG Audio	audio/mpeg	MPEGA, MPG, MP3	adSOUND	MCI , mp3sr
FTP_Software_Session_Fmt	271	250	FTP Session Data		STE	adCOMMUNICATION	
Netscape_Bookmark_File_Fmt	272	210	Netscape Bookmark File	text/html		adWORDPROCESSOR	htmsr
Corel_Draw_CMX_Fmt	273	252	Corel CMX	application/cmx	CMX	adVECTORGRAPHIC	
AutoDesk_DWG_Fmt	274	253	AutoDesk Drawing (DWG)	image/x-dwg	DWG	adVECTORGRAPHIC	kpDWGrdr , kpODArdr
AutoDesk_WHIP_Fmt	275	254	AutoDesk WHIP		WHP	adVECTORGRAPHIC	
Macromedia_Director_Fmt	276	255	Macromedia Director	application/x-director	DCR	adANIMATION	
Real_Audio_Fmt	277	256	Real Audio	audio/x-pn-realaudio	RM, RA	adSOUND	
MSDOS_Device_Driver_Fmt	278	257	MSDOS Device Driver	application/octet-stream	SYS	adEXECUTABLE	
Micrografx_Designer_Fmt	279	258	Micrografx Designer		DSF	adVECTORGRAPHIC	
SVF_Fmt	280	259	Simple Vector Format (SVF)	image/x-svf	SVF	adVECTORGRAPHIC	
Applix_Words_Fmt	281	261	Applix Words	application/x-applix-word	AW	adWORDPROCESSOR	awsr
Applix_Graphics_Fmt	282	262	Applix Graphics		AG	adPRESENTATION	kpagrdr
MS_Access_Fmt	283	263	Microsoft Access (versions 1 and 2)	application/x-msaccess	MDB	adDATABASE	mdbsr
MS_Access_95_Fmt	284	263	Microsoft Access 95	application/msaccess	MDB	adDATABASE	mdbsr
MS_Access_97_Fmt	285	263	Microsoft Access 97	application/msaccess	MDB	adDATABASE	mdbsr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
MacBinary_Fmt	286	265	MacBinary	application/x-macbinary	BIN	adENCAPSULATION	macbinsr
Apple_Single_Fmt	287	266	Apple Single			adENCAPSULATION	
Apple_Double_Fmt	288	267	Apple Double	multipart/appledouble	AD	adENCAPSULATION	
Enhanced_Metafile_Fmt	289	270	Enhanced Metafile	image/x-emf	EMF	adVECTORGRAPHIC	kpemfrdr
MS_Office_Drawing_Fmt	290	271	Microsoft Office Drawing			adVECTORGRAPHIC	kpmsordr
XML_Fmt	291	285	XML	text/xml	XML	adWORDPROCESSOR	xmlsr
DeVice_Independent_Fmt	292	274	DeVice Independent file (DVI)	application/x-dvi	DVI	adVECTORGRAPHIC	
Unicode_Fmt	293	275	Unicode text file	text/plain	UNI	adWORDPROCESSOR	unisir
Lotus_123_Worksheet_Fmt	294	81	Lotus 1-2-3	application/x-lotus-123	WKS, WK1, WK3, WK4	adSPREADSHEET	wkssr
Lotus_123_Format_Fmt	295	81	Lotus 1-2-3 Formatting	application/x-123	FM3	adSPREADSHEET	l123sr
Lotus_123_97_Fmt	296	81	Lotus 1-2-3 97	application/x-lotus-123	123	adSPREADSHEET	l123sr
Lotus_Word_Pro_96_Fmt	297	268	Lotus Word Pro 96	application/vnd.lotus-wordpro	LWP, MWP	adWORDPROCESSOR	lwpsr
Lotus_Word_Pro_97_Fmt	298	268	Lotus Word Pro 97	application/vnd.lotus-wordpro	LWP, MWP	adWORDPROCESSOR	lwpsr
Freelance_DOS_Fmt	299	140	Lotus Freelance for DOS	application/x-freelance	PRZ	adPRESENTATION	kppzrdr
Freelance_Win_Fmt	300	140	Lotus Freelance for Windows	application/x-freelance	PRE, FLW	adPRESENTATION	kpprerdr
Freelance_OS2_Fmt	301	140	Lotus Freelance for OS/2	application/x-freelance	PRS	adPRESENTATION	kpprerdr
Freelance_96_Fmt	302	140	Lotus Freelance 96	application/x-freelance	PRZ	adPRESENTATION	kppzrdr
Freelance_97_Fmt	303	140	Lotus Freelance 97	application/x-freelance	PRZ	adPRESENTATION	kppzrdr
MS_Word_95_Fmt	304	189	Microsoft Word 95	application/msword	DOC	adWORDPROCESSOR	mw6sr
MS_Word_97_Fmt	305	269	Microsoft Word 97	application/msword	DOC, WPS, WBK	adWORDPROCESSOR	mw8sr
Excel_Fmt	306	90	Microsoft Excel (up to version 5)	application/x-ms-excel	XLS	adSPREADSHEET	xlssr
Excel_Chart_Fmt	307	90	Microsoft Excel (up to version 5) chart	application/x-ms-excel	XLC	adSPREADSHEET	xlssr
Excel_Macro_Fmt	308	90	Microsoft Excel (up to version 5) macro	application/vnd.ms-excel	XLM	adSPREADSHEET	xlssr
Excel_95_Fmt	309	188	Microsoft Excel 95	application/x-ms-excel	XLS	adSPREADSHEET	xlssr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Excel_97_Fmt	310	188	Microsoft Excel 97	application/x-ms-excel	XLS, XLR	adSPREADSHEET	xlssr
Corel_Presentations_Fmt	311	127	Corel Presentations	application/x-corelpresentations	XFD, XFDL	adPRESENTATION	kpswhrdr
Harvard_Graphics_Fmt	312	131	Harvard Graphics		PR4	adPRESENTATION	
Harvard_Graphics_Chart_Fmt	313	131	Harvard Graphics Chart		CH3, CHT	adVECTORGRAPHIC	
Harvard_Graphics_Symbol_Fmt	314	131	Harvard Graphics Symbol File		SY3	adVECTORGRAPHIC	
Harvard_Graphics_Cfg_Fmt	315	131	Harvard Graphics Configuration File			adVECTORGRAPHIC	
Harvard_Graphics_Palette_Fmt	316	131	Harvard Graphics Palette			adVECTORGRAPHIC	
Lotus_123_R9_Fmt	317	81	Lotus 1-2-3 Release 9	application/x-lotus-123	123	adSPREADSHEET	l123sr
Applix_Spreadsheets_Fmt	318	278	Applix Spreadsheets	application/x-applix-spreadsheet	AS	adSPREADSHEET	assr
MS_Pocket_Word_Fmt	319	45	Microsoft Pocket Word		PWD	adWORDPROCESSOR	rtfsr
MS_DIB_Fmt	320	279	Microsoft Device Independent Bitmap	image/bmp	DIB	adRASTERIMAGE	
MS_Word_2000_Fmt	321	269	Microsoft Word 2000	application/msword	DOC	adWORDPROCESSOR	mw8sr
Excel_2000_Fmt	322	188	Microsoft Excel 2000	application/x-ms-excel	XLS	adSPREADSHEET	xlssr
PowerPoint_2000_Fmt	323	272	Microsoft PowerPoint 2000	application/x-ms-powerpoint	PPT	adPRESENTATION	kpp97rdr
MS_Access_2000_Fmt	324	263	Microsoft Access 2000	application/x-msaccess	MDB	adDATABASE	mdbsr
MS_Project_4_Fmt	325	281	Microsoft Project 4		MPP	adSCHEDULE	mpps
MS_Project_41_Fmt	326	281	Microsoft Project 4.1		MPP	adSCHEDULE	mpps
MS_Project_98_Fmt	327	281	Microsoft Project 98	application/vnd.ms-project	MPP	adSCHEDULE	mpps
Folio_Flat_Fmt	328	282	Folio Flat File		FFF	adWORDPROCESSOR	foliosr
HWP_Fmt	329	283	HWP (Arae-Ah Hangul)	application/x-hwp	HWP	adWORDPROCESSOR	hwpsr , hwpsr
ICHITARO_Fmt	330	284	ICHITARO (v4-10)		JTD	adWORDPROCESSOR	jtdsr
IS_XML_Fmt	331	273	Extended or Custom XML	text/xml	XML	adWORDPROCESSOR	
Oasys_Fmt	332	286	Oasys	application/vnd.fujitsu.oasys	OAS, OA2, OA3	adWORDPROCESSOR	oa2sr
PBM_ASC_Fmt	333	287	Portable Bitmap Utilities	image/pbm	PBM	adRASTERIMAGE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
			ASCII format (PBM)				
PBM_BIN_Fmt	334	287	Portable Bitmap Utilities BINARY format (PBM)	image/pbm	PBM	adRASTERIMAGE	
PGM_ASC_Fmt	335	288	Portable Greymap Utilities ASCII format (PGM)	image/x-pgm	PGM	adRASTERIMAGE	
PGM_BIN_Fmt	336	288	Portable Greymap Utilities BINARY format (PGM)	image/x-pgm	PGM	adRASTERIMAGE	
PPM_ASC_Fmt	337	289	Portable Pixmap Utilities ASCII format (PPM)	image/x-portable-pixmap	PPM	adRASTERIMAGE	
PPM_BIN_Fmt	338	289	Portable Pixmap Utilities BINARY format (PPM)	image/x-portable-pixmap	PPM	adRASTERIMAGE	
XBM_Fmt	339	290	X Bitmap format (XBM)	image/x-xbitmap	XBM	adRASTERIMAGE	
XPM_Fmt	340	291	X Pixmap format (XPM)	image/xpm	XPM	adRASTERIMAGE	
FPX_Fmt	341	292	Kodak FlashPix FPX Image format	image/fpx	FPX	adRASTERIMAGE	
PCD_Fmt	342	293	PCD Image format	image/pcd	PCD	adRASTERIMAGE	
MS_Visio_Fmt	343	294	Microsoft Visio (up to version 11)	image/x-vsd	VSD	adPRESENTATION	kpVSD2rdr , vsdsr
MS_Project_2000_Fmt	344	281	Microsoft Project 2000	application/vnd.ms-project	MPP	adSCHEDULE	mppsrsr
MS_Outlook_Fmt	345	295	Microsoft Outlook message	application/vnd.ms-outlook	MSG, OFT	adENCAPSULATION	msgsr
ELF_Relocatable_Fmt	346	159	ELF Relocatable	application/octet-stream	O	adOBJECTMODULE	
ELF_Executable_Fmt	347	158	ELF Executable	application/octet-stream		adEXECUTABLE	
ELF_Dynamic_Lib_Fmt	348	160	ELF Dynamic Library	application/octet-stream	SO	adLIBRARY	
MS_Word_XML_Fmt	349	285	Microsoft Word 2003 XML	text/xml	XML	adWORDPROCESSOR	xmlsr
MS_Excel_XML_Fmt	350	285	Microsoft Excel 2003 XML	text/xml	XML	adWORDPROCESSOR	xmlsr
MS_Visio_XML_Fmt	351	285	Microsoft Visio 2003 XML	text/xml	VDX	adWORDPROCESSOR	xmlsr
SO_Text_XML_Fmt	352	314	OpenDocument format (OpenOffice 1/StarOffice 6,7) Text XML	application/vnd.sun.xml.writer	SXW	adWORDPROCESSOR	odfwpsr
SO_Spreadsheet_XML_Fmt	353	315	OpenDocument format (OpenOffice 1/StarOffice 6,7) Spreadsheet XML	application/vnd.sun.xml.calc	SXC, STC	adSPREADSHEET	sosr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
SO_Presentation_XML_Fmt	354	316	OpenDocument format (OpenOffice 1/StarOffice 6,7) Presentation XML	application/vnd.sun.xml.impress	SXD, SXI	adPRESENTATION	kpodfrdr
XHTML_Fmt	355	296	XHTML	text/xhtml	XML, XHTML, XHT	adWORDPROCESSOR	
MS_OutlookPST_Fmt	356	297	Microsoft Outlook Personal Folders File (.pst)	application/vnd.ms-outlook-pst	PST	adENCAPSULATION	pstnsr , pstrs , pstxsr
RAR_Fmt	357	298	RAR archive format	application/x-rar-compressed	RAR, REV, R00, R01	adENCAPSULATION	rarsr
Lotus_Notes_NSF_Fmt	358	299	IBM Lotus Notes Database NSF/NTF	application/x-lotus-notes	NSF	adENCAPSULATION	nsfsr
Macromedia_Flash_Fmt	359	300	Macromedia Flash (.swf)	application/x-shockwave-flash	SWF, SWD	adWORDPROCESSOR	swfsr
MS_Word_2007_Fmt	360	301	Microsoft Word 2007 XML - Docx	application/x-ms-word07	DOCX, DOTX	adWORDPROCESSOR	mwxsr
MS_Excel_2007_Fmt	361	302	Microsoft Excel 2007 XML	application/x-ms-excel07	XLSX, XLTX	adSPREADSHEET	xlsxsr
MS_PPT_2007_Fmt	362	303	Microsoft PowerPoint 2007 XML	application/x-ms-powerpoint07	PPTX, POTX, PPSX	adPRESENTATION	kpppxrdr
OpenPGP_Fmt	363	304	OpenPGP Message Format (with new packet format)	application/pgp-encrypted	PGP	adENCAPSULATION	
Intergraph_V7_DGN_Fmt	364	305	Intergraph Standard File Format (ISFF) V7 DGN (non-OLE)		DGN	adVECTORGRAPHIC	
MicroStation_V8_DGN_Fmt	365	306	MicroStation V8 DGN (OLE)		DGN	adVECTORGRAPHIC	
MS_Word_Macro_2007_Fmt	366	307	Microsoft Word Macro 2007 XML	application/x-ms-word07m	DOCM, DOTM	adWORDPROCESSOR	mwxsr
MS_Excel_Macro_2007_Fmt	367	308	Microsoft Excel Macro 2007 XML	application/x-ms-excel07m	XLSM, XLTM, XLAM	adSPREADSHEET	xlsxsr
MS_PPT_Macro_2007_Fmt	368	309	Microsoft PPT Macro 2007 XML	application/x-ms-powerpoint07m	PPTM, POTM, PPSM, PPAM	adPRESENTATION	kpppxrdr
LZH_Fmt	369	310	LZH Archive	application/x-lzh-compressed	LZH, LHA	adENCAPSULATION	lzhsr
Office_2007_Fmt	370	311	Office 2007 document		XLSB	adMISC	
MS_XPS_Fmt	371	312	Microsoft XML Paper Specification (XPS)	application/vnd.ms-xpsdocument	XPS	adWORDPROCESSOR	xpssr
Lotus_Domino_DXL_Fmt	372	313	IBM Domino Data in XML	text/xml	DXL	adENCAPSULATION	dxlsr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
			format (.dxf)				
ODF_Text_Fmt	373	314	ODF Text	application/vnd.oasis.opendocument.text	ODT	adWORDPROCESSOR	odfwpsr
ODF_Spreadsheet_Fmt	374	315	ODF Spreadsheet	application/vnd.oasis.opendocument.spreadsheet	ODS	adSPREADSHEET	odfsssr
ODF_Presentation_Fmt	375	316	ODF Presentation	application/vnd.oasis.opendocument.presentation	ODP	adPRESENTATION	kpodfrdr
Legato_Extender_ONM_Fmt	376	317	Legato Extender Native Message ONM	application/x-lotus-notes	ONM	adENCAPSULATION	onmsr
bin_Unknown_Fmt	377	318	Bin unknown format (.xxx)			adWORDPROCESSOR	
TNEF_Fmt	378	319	Transport Neutral Encapsulation Format (TNEF)	application/vnd.ms-tnef		adENCAPSULATION	tnefsr
CADAM_Drawing_Fmt	379	320	CADAM Drawing		CDD	adVECTORGRAPHIC	
CADAM_Drawing_Overlay_Fmt	380	321	CADAM Drawing Overlay		CDO	adVECTORGRAPHIC	
NURSTOR_Drawing_Fmt	381	322	NURSTOR Drawing		NUR	adVECTORGRAPHIC	
HP_GLP_Fmt	382	323	HP Graphics Language (Plotter)	vector/x-hpgl2	HPG	adVECTORGRAPHIC	
ASF_Fmt	383	324	Advanced Systems Format (ASF)	application/x-ms-asf	ASF	adMISC	asfsr
WMA_Fmt	384	325	Windows Media Audio Format (WMA)	audio/x-ms-wma	WMA	adSOUND	asfsr
WMV_Fmt	385	326	Windows Media Video Format (WMV)	video/x-ms-wmv	WMV	adMOVIE	asfsr
EMX_Fmt	386	327	Legato EMailXtender Archives Format (EMX)		EMX	adENCAPSULATION	emxsr
Z7Z_Fmt	387	328	7 Zip Format (7z)	application/7z	7Z	adENCAPSULATION	z7zsr
MS_Excel_Binary_2007_Fmt	388	329	Microsoft Excel Binary 2007	application/vnd.ms-excel.sheet.binary.macroenabled.12	XLSB	adSPREADSHEET	xlsbsr
CAB_Fmt	389	330	Microsoft Cabinet File (CAB)	application/vnd.ms-cab-compressed	CAB	adENCAPSULATION	cabsr
CATIA_Fmt	390	331	CATIA Formats (CAT*)		CATPART, CATPRODUCT ²	adVECTORGRAPHIC	kpCATrdr
YIM_Fmt	391	332	Yahoo Instant Messenger History		DAT	adWORDPROCESSOR	yimsr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
ODF_Drawing_Fmt	392	316	ODF Drawing/Graphics	application/vnd.oasis.opendocument.graphics	ODG	adVECTORGRAPHIC	kpodfrdr
Founder_CEB_Fmt	393	333	Founder Chinese E-paper Basic (ceb)	application/ceb	CEB	adWORDPROCESSOR	cebsr
QPW_Fmt	394	334	Corel Quattro Pro 9+ for Windows	application/quattro-pro	QPW	adSPREADSHEET	qpwsr
MHT_Fmt	395	335	MHTML format (MHT) ¹	multipart/related	MHT, MHTML	adWORDPROCESSOR	mhtr
MDI_Fmt	396	336	Microsoft Document Imaging Format	image/vnd.ms-modi	MDI	adRASTERIMAGE	
GRV_Fmt	397	337	Microsoft Office Groove Format	application/vnd.groove-injector	GRV	adWORDPROCESSOR	
IWWP_Fmt	398	338	Apple iWork Pages format	application/vnd.apple.pages	PAGES	adWORDPROCESSOR	iwpsr
IWSS_Fmt	399	339	Apple iWork Numbers format	application/vnd.apple.numbers	NUMBERS	adSPREADSHEET	iwssr
IWPG_Fmt	400	340	Apple iWork Keynote format	application/vnd.apple.keynote	KEY	adPRESENTATION	kplWPGrdr
BKF_Fmt	401	341	Windows Backup File		BKF	adENCAPSULATION	bkfsr
MS_Access_2007_Fmt	402	342	Microsoft Access 2007	application/msaccess	ACCDB	adDATABASE	mdbsr
ENT_Fmt	403	343	Microsoft Entourage Database Format			adENCAPSULATION	entsr
DMG_Fmt	404	344	Mac Disk Copy Disk Image File	application/x-apple-diskimage	DMG	adENCAPSULATION	dmgsr
CWK_Fmt	405	345	AppleWorks (Claris Works) File	application/appleworks	CWK	adWORDPROCESSOR	stringsr
OO3_Fmt	406	346	Omni Outliner V3 File		OO3	adWORDPROCESSOR	oo3sr
OPML_Fmt	407	347	Omni Outliner OPML File		OPML	adWORDPROCESSOR	oo3sr
Omni_Graffle_XML_Fmt	408	348	Omni Graffle XML File		GRAFFLE	adVECTORGRAPHIC	kpGFLrdr
PSD_Fmt	409	349	Photoshop Document	image/vnd.adobe.photoshop	PSD, PSB	adRASTERIMAGE	psdsr
Apple_Binary_PLList_Fmt	410	350	Apple Binary Property List format		PLIST	adMISC	
Apple_iChat_Fmt	411	351	Apple iChat format		ICHAT	adWORDPROCESSOR	ichatsr
OOUTLINE_Fmt	412	352	OOutliner File		OOUTLINE	adWORDPROCESSOR	oo3sr
BZIP2_Fmt	413	353	Bzip 2 Compressed File	application/x-bzip2	BZ2	adENCAPSULATION	bzip2sr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
ISO_Fmt	414	354	ISO-9660 CD Disc Image Format	application/x-iso9660-image	ISO	adENCAPSULATION	isosr
DocuWorks_Fmt	415	355	DocuWorks Format	application/vnd.fujixerox.docuworks	XDW	adWORDPROCESSOR	
RealMedia_Fmt	416	356	RealMedia Streaming Media	application/vnd.rm-realmedia	RM, RA	adMOVIE	
AC3Audio_Fmt	417	357	AC3 Audio File Format	audio/ac3	AC3	adSOUND	
NEF_Fmt	418	358	Nero Encrypted File		NEF	adENCAPSULATION	
SolidWorks_Fmt	419	359	SolidWorks Format Files		SLDASM, SLDPRT, SLDDRW, SLDDRT	adVECTORGRAPHIC	
XFDL_Fmt	420	366	Extensible Forms Description Language	application/x-xfdl	XFDL, XFD	adPRESENTATION	kpXFDLrdr
Apple_XML_PLIST_Fmt	421	367	Apple XML Property List format		PLIST	adMISC	
OneNote_Fmt	422	368	OneNote Note Format	application/onenote	ONE	adWORDPROCESSOR	kpONErdr
IFilter_Fmt	423	369	iFilter			adWORDPROCESSOR	
Dicom_Fmt	424	370	Digital Imaging and Communications in Medicine (Dicom)	application/dicom	DCM	adRASTERIMAGE	dcmsr
EnCase_Fmt	425	371	Expert Witness Compression Format (EnCase)		E01, L01, Lx01	adENCAPSULATION	encase2sr , encasesr
Scrap_Fmt	426	372	Shell Scrap Object File		SHS	adENCAPSULATION	olesr
MS_Project_2007_Fmt	427	373	Microsoft Project 2007	application/vnd.ms-project	MPP	adSCHEDULE	mppsrsr
MS_Publisher_98_Fmt	428	374	Microsoft Publisher from version 98	application/x-mspublisher	PUB	adDESKTOPPUBLISH	mspubsr
Skype_Fmt	429	375	Skype Log File		DBB	adWORDPROCESSOR	skypesr
HL7_Fmt	430	377	Health level7 message		HL7	adWORDPROCESSOR	hl7sr
MS_OutlookOST_Fmt	431	378	Microsoft Outlook Offline Folders File (OST)	application/vnd.ms-outlook-pst	OST	adENCAPSULATION	pffsr
Epub_Fmt	432	379	Electronic Publication	application/epub+zip	EPUB	adWORDPROCESSOR	epubsr
MS_OEDBX_Fmt	433	380	Microsoft Outlook Express DBX Message Database		DBX	adENCAPSULATION	dbxsr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
BB_Activ_Fmt	434	381	BlackBerry Activation File		DAT	adWORDPROCESSOR	
DiskImage_Fmt	435	382	Disk Image		DMG	adENCAPSULATION	
Milestone_Fmt	436	383	Milestone Document		MLS, ML3, ML4, ML5, ML6, ML7, ML8, ML9, MLA	adRASTERIMAGE	
E_Transcript_Fmt	437	384	RealLegal E-Transcript File		PTX	adWORDPROCESSOR	
PostScript_Font_Fmt	438	385	PostScript Type 1 Font	application/x-font	PFB	adFONT	
Ghost_DiskImage_Fmt	439	386	Ghost Disk Image File		GHO, GHS	adENCAPSULATION	
JPEG_2000_JP2_File_Fmt	440	387	JPEG-2000 JP2 File Format Syntax (ISO/IEC 15444-1)	image/jp2	JP2, JPF, J2K, JPWL, JPX, PGX	adRASTERIMAGE	jp2000sr , kjpj2000rdr
Unicode_HTML_Fmt	441	388	Unicode HTML	text/html	HTM, HTML	adWORDPROCESSOR	unihtmsr
CHM_Fmt	442	389	Microsoft Compiled HTML Help	application/x-chm	CHM	adENCAPSULATION	chmsr
EMCMF_Fmt	443	390	Documentum EMCMF format		EMCMF	adENCAPSULATION	msgsr
MS_Access_2007_Tmpl_Fmt	444	391	Microsoft Access 2007 Template		ACCDT	adDATABASE	
Jungum_Fmt	445	392	Samsung Electronics Jungum Global document		GUL	adWORDPROCESSOR	
JBIG2_Fmt	446	393	JBIG2 File Format	image/jbig2	JB2, JBIG2	adRASTERIMAGE	kpJBIG2rdr
EFax_Fmt	447	394	eFax file		EFX	adRASTERIMAGE	
AD1_Fmt	448	395	AD1 Evidence file		AD1	adENCAPSULATION	ad1sr
SketchUp_Fmt	449	396	Google SketchUp		SKP	adVECTORGRAPHIC	
GWFS_Email_Fmt	450	397	Group Wise File Surf email		GWFS	adENCAPSULATION	gwfssr
JNT_Fmt	451	398	Windows Journal format		JNT	adWORDPROCESSOR	
Yahoo_yChat_Fmt	452	399	Yahoo! Messenger chat log		YCHAT	adWORDPROCESSOR	
PaperPort_MAX_File_Fmt	453	400	PaperPort MAX image file	image/max	MAX	adRASTERIMAGE	
ARJ_Fmt	454	402	ARJ (Archive by Robert Jung) file format	application/arj	ARJ	adENCAPSULATION	multiarcsr
RPMSG_Fmt	455	403	Microsoft Outlook Restricted Permission	application/x-microsoft-rpmsg-message	RPMSG	adENCAPSULATION	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
			Message				
MAT_Fmt	456	404	MATLAB file format	application/x-matlab-data	MAT, FIG	adWORDPROCESSOR	
SGY_Fmt	457	405	SEG-Y Seismic Data format		SGY, SEGY	adWORDPROCESSOR	
CDXA_MPEG_PS_Fmt	458	406	MPEG-PS container with CDXA stream	video/mpeg	MPG	adMOVIE	
EVT_Fmt	459	407	Microsoft Windows NT Event Log		EVT	adMISC	
EVTX_Fmt	460	408	Microsoft Windows Vista Event Log		EVTX	adMISC	
MS_OutlookOLM_Fmt	461	409	Microsoft Outlook for Macintosh format		OLM	adENCAPSULATION	olmsr
WARC_Fmt	462	410	Web ARChive	application/warc	WARC	adENCAPSULATION	
JAVACLASS_Fmt	463	411	Java Class format	application/x-java-class	CLASS	adWORDPROCESSOR	
VCF_Fmt	464	412	Microsoft Outlook vCard file format	text/vcard	VCF	adWORDPROCESSOR	vcfsr
EDB_Fmt	465	413	Microsoft Exchange Server Database file format		EDB	adENCAPSULATION	
ICS_Fmt	466	414	Microsoft Outlook iCalendar file format	text/calendar	ICS, VCS	adENCAPSULATION	icssr
MS_Visio_2013_Fmt	467	415	Microsoft Visio 2013	application/vnd.visio	VSDX, VSTX, VSSX	adPRESENTATION	ActiveX components, kpVSDXrdr
MS_Visio_2013_Macro_Fmt	468	415	Microsoft Visio 2013 macro	application/vnd.visio	VSDM, VSTM, VSSM	adPRESENTATION	kpVSDXrdr
ICHITARO_Compr_Fmt	469	417	ICHITARO Compressed format	application/x-js-taro	JTDC	adWORDPROCESSOR	jtdsr
IWWP13_Fmt	470	418	Apple iWork 2013 Pages format		IWA, PAGES	adWORDPROCESSOR	iwwp13sr
IWSS13_Fmt	471	419	Apple iWork 2013 Numbers format		IWA, NUMBERS	adSPREADSHEET	iwss13sr
IWPG13_Fmt	472	420	Apple iWork 2013 Keynote format		IWA, KEY	adPRESENTATION	kplWPG13rdr, kplWPGrdr
XZ_Fmt	473	421	XZ archive format	application/x-xz	XZ	adENCAPSULATION	multiarcsr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Sony_WAVE64_Fmt	474	422	Sony Wave64 format	audio/wav64	W64	adSOUND	
Conifer_WAVPACK_Fmt	475	423	Conifer Wavpack format	audio/x-wavpack	WV	adSOUND	
Xiph_OGG_VORBIS_Fmt	476	424	Xiph Ogg Vorbis format	audio/ogg	OGG	adSOUND	
MS_Visio_2013_Stencil_Fmt	477	415	MS Visio 2013 stencil format	application/vnd.visio	VSSX	adPRESENTATION	kpVSDXrdr
MS_Visio_2013_Stencil_Macro_Fmt	478	415	MS Visio 2013 stencil Macro format	application/vnd.visio	VSSM	adPRESENTATION	kpVSDXrdr
MS_Visio_2013_Template_Fmt	479	415	MS Visio 2013 template format	application/vnd.visio	VSTX	adPRESENTATION	kpVSDXrdr
MS_Visio_2013_Template_Macro_Fmt	480	415	MS Visio 2013 template Macro format	application/vnd.visio	VSTM	adPRESENTATION	kpVSDXrdr
Borland_Reflex_2_Fmt	481	425	Borland Reflex 2 format		R2D	adDATABASE	
PKCS_12_Fmt	482	426	PKCS #12 (p12) format	application/x-pkcs12	P12, PFX	adWORDPROCESSOR	
B1_Fmt	483	427	B1 format	application/x-b1	B1	adENCAPSULATION	b1sr
ISO_IEC_MPEG_4_Fmt	484	428	ISO/IEC MPEG-4 (ISO 14496) format	video/mp4	MP4	adMOVIE	mpeg4sr
RAR5_Fmt	485	429	RAR5 Format	application/x-rar-compressed	RAR	adENCAPSULATION	multiarcsr
Unigraphics_NX_Fmt	486	362	Unigraphics (UG) NX CAD Format		PRT	adVECTORGRAPHIC	kpUGrdr
PTC_Creo_Fmt	487	430	PTC Creo CAD Format		ASM, PRT	adVECTORGRAPHIC	
KML_Fmt	488	431	Keyhole Markup Language	application/vnd.google-earth.kml+xml	KML	adWORDPROCESSOR	xmlsr
KMZ_Fmt	489	432	Zipped Keyhole Markup Language	application/vnd.google-earth.kmz	KMZ	adWORDPROCESSOR	unzip
WML_Fmt	490	433	Wireless Markup Language	text/vnd.wap.wml	WML	adWORDPROCESSOR	xmlsr
ODF_Formula_Fmt	491	434	ODF Formula	application/vnd.oasis.opendocument.formula	ODF	adWORDPROCESSOR	unzip
SO_Text_Fmt	492	435	Star Office 4,5 Writer Text	application/vnd.stardivision.writer	SDW, SGL, VOR	adWORDPROCESSOR	kpsdwrdr , stawsr
SO_Spreadsheet_Fmt	493	436	Star Office 4,5 Calc Spreadsheet	application/vnd.stardivision.calc	SDC	adSPREADSHEET	starcsr
SO_Presentation_Fmt	494	437	Star Office 4,5 Impress Presentation	application/vnd.stardivision.draw	SDD, SDA	adPRESENTATION	kpsddrdr
SO_Math_Fmt	495	438	Star Office 4,5 Math	application/vnd.stardivision.math	SMF	adMISC	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
STEP_Fmt	496	439	ISO 10303-21 STEP format			adMISC	
STL_Fmt	497	364	3D Systems STL ASCII format			adMISC	
Wiki_Fmt	545	488	MediaWiki File	text/x-mediawiki		adWORDPROCESSOR	afsr
MS_Word_2007_Flat_XML_Fmt	546	301	Microsoft Word 2007 XML - Flat xml	text/xml	XML	adWORDPROCESSOR	mwxsr
Matroska_Fmt	547	489	Matroska video File	video/x-matroska	MKV	adMOVIE	
SVG_Fmt	548	490	Scalable Vector Graphics image	image/svg+xml	SVG	adVECTORGRAPHIC	xmslr
Shapefile_Fmt	549	491	Shapefile	application/x-shapefile	SHP, SHX	adGIS	
Flash_Video_Fmt	550	492	Flash video File	video/x-flv	FLV	adMOVIE	
Embedded_OpenType_Fmt	551	493	Embedded OpenType font	application/vnd.ms-fontobject	EOT	adFONT	
Web_Open_Font_Fmt	552	494	Web Open Font Format	font/woff	WOFF, WOFF2	adFONT	
OpenType_Fmt	553	495	OpenType Font	font/otf	OTF	adFONT	
MNG_Fmt	554	496	Multiple-image Network Graphics	video/x-mng	MNG	adANIMATION	
JNG_Fmt	555	497	JPEG Network Graphics	image/x-jng	JNG	adRASTERIMAGE	
AppleScript_Binary_Fmt	556	498	AppleScript Binary Source Code		SCPT	adSOURCECODE	
Maya_Binary_Fmt	557	499	Autodesk Maya binary file		MB	adCAD	
Jupiter_Tessellation_Fmt	558	363	UGS Jupiter Tessellation file		JT	adCAD	
OGV_Fmt	559	500	Ogg Theora Video format	video/ogg	OGV	adMOVIE	
OGG_Container_Fmt	560	501	General Ogg Container format	application/ogg	OGG	adMISC	
GNU_Message_Catalog_Fmt	561	502	GNU Message Catalog format		MO	adMISC	
Windows_Shortcut_Fmt	562	503	Windows shortcut file	application/x-ms-shortcut	LNK	adMISC	
Apple_Typedstream_Fmt	563	504	Apple/NeXT typedstream data format			adMISC	
XCF_Fmt	564	505	GIMP XCF image	image/x-xcf	XCF	adRASTERIMAGE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
PaintShop_Pro_Fmt	565	506	PaintShop Pro image		PSP, PSPIMAGE	adRASTERIMAGE	
SQLite_Database_Fmt	566	507	SQLite database format	application/x-sqlite3	QHC	adDATABASE	
MySQL_Table_Fmt	567	508	MySQL table definition file		FRM	adDATABASE	
Microsoft_Program_DB_Fmt	568	509	Microsoft Program Database format		PDB	adDATABASE	
OpenEXR_Fmt	569	510	OpenEXR image format		EXR	adRASTERIMAGE	
XMV_Fmt	570	511	4X Movie File		4XM	adMOVIE	
AMV_Fmt	571	512	AMV video file		AMV	adMOVIE	
NIFF_Fmt	572	513	Notation Interchange File Format		NIF	adSOUND	
CuBase_Fmt	573	514	Steinberg CuBase file			adSOUND	
SoundFont_Fmt	574	515	SoundFont file			adSOUND	
WebP_Fmt	575	516	WebP image	image/webp	WEBP	adRASTERIMAGE	
ICC_Fmt	576	517	International Color Consortium files	application/vnd.iccprofile	ICC, ICM	adMISC	
PCF_Fmt	577	518	X11 Portable Compiled Font file	application/x-font-pcf	PCF	adFONT	
WebM_Fmt	578	519	WebM video file	video/webm	WEBM	adMOVIE	
AMFF_Fmt	579	520	Amiga Metafile		AMF	adVECTORGRAPHIC	
ANBM_Fmt	580	521	IFF Animated Bitmap			adRASTERIMAGE	
ANIM_Fmt	581	522	IFF Amiga animated raster graphics format			adRASTERIMAGE	
DEEP_Fmt	582	523	IFF-DEEP TVPaint image		DEEP	adRASTERIMAGE	
FAXX_Fmt	583	524	IFF-FAXX Facsimile image			adRASTERIMAGE	
ICON_Fmt	584	525	IFF Glow Icon image			adRASTERIMAGE	
ILBM_Fmt	585	526	Interleaved BitMap image		IFF	adRASTERIMAGE	
LWOB_Fmt	586	527	LightWave Object format		LWOB	adMISC	
MAUD_Fmt	587	528	IFF-MAUD MacroSystem audio format			adSOUND	
PBM_Fmt	588	529	IFF Planar BitMap			adRASTERIMAGE	
TDDD_Fmt	589	530	IFF TDDD and Imagine		TDD	adRASTERIMAGE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
			Object animation format				
DjVu_Fmt	590	531	AT&T DjVu format	image/vnd.djvu	DJVU	adWORDPROCESSOR	
InDesign_Fmt	591	532	Adobe InDesign document	application/x-indesign	INDD	adDESKTOPPUBLSH	
Calamus_Fmt	592	533	Calamus Desktop Publishing			adDESKTOPPUBLSH	
Adaptive_MultiRate_Fmt	593	534	Adaptive Multi-Rate audio format	audio/amr	AMR	adSOUND	
FLAC_Fmt	594	535	Free Lossless Audio Codec format	audio/flac	FLAC	adSOUND	
Ogg_FLAC_Fmt	595	536	Ogg Container FLAC audio format		OGG	adSOUND	
SAS7BDAT_Fmt	596	537	SAS7BDAT database storage format		SAS7BDAT	adDATABASE	
Design_Web_Format_Fmt	597	538	Autodesk Design Web Format	model/vnd.dwf	DWF	adCAD	
Adobe_Flash_Audio_Book_Fmt	598	539	Adobe Flash Player audio book	audio/mp4	F4B	adSOUND	mpeg4sr
Adobe_Flash_Audio_Fmt	599	540	Adobe Flash Player audio	audio/mp4	F4A	adSOUND	mpeg4sr
Adobe_Flash_Protected_Video_Fmt	600	541	Adobe Flash Player protected video	video/mp4	F4P	adMOVIE	mpeg4sr
Adobe_Flash_Video_Fmt	601	542	Adobe Flash Player video	video/x-f4v	F4V	adMOVIE	mpeg4sr
Audible_Audiobook_Fmt	602	543	Audible Enhanced Audiobook	audio/vnd.audible.aax	AAX	adSOUND	mpeg4sr
Canon_Camera_Fmt	603	544	Canon Digital Camera image			adRASTERIMAGE	
Canon_Raw_Fmt	604	545	Canon Raw image		CR3	adRASTERIMAGE	
Casio_Camera_Fmt	605	546	Casio Digital Camera image			adRASTERIMAGE	
Convergent_Design_Fmt	606	547	Convergent Design file			adRASTERIMAGE	
DMB_MAF_Audio_Fmt	607	548	DMB MAF audio			adSOUND	
DMB_MAF_Video_Fmt	608	549	DMB MAF video			adMOVIE	
DMP_Content_Fmt	609	550	Digital Media Project Content Format			adMISC	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
DVB_Fmt	610	551	Digital Video Broadcast format	video/vnd.dvb.file	DVB	adMOVIE	
Dirac_Wavelet_Compression_Fmt	611	552	ISO-BMFF Dirac Wavelet compression			adMISC	
HEICS_Image_Sequence_Fmt	612	553	High Efficiency Image Format HEVC image sequence	image/heic-sequence	HEICS	adRASTERIMAGE	
HEIC_Image_Fmt	613	554	High Efficiency Image Format HEVC image	image/heic	HEIC	adRASTERIMAGE	
HEIFS_Image_Sequence_Fmt	614	555	High Efficiency Image Format image sequence	image/heif-sequence	HEIFS	adRASTERIMAGE	
HEIF_Image_Fmt	615	556	High Efficiency Image Format image	image/heif	HEIF	adRASTERIMAGE	
ISMACryp_Fmt	616	557	ISMACryp 2.0 Encrypted format			adENCAPSULATION	
ISO_3GPP2_Fmt	617	558	3GPP2 video file	video/3gpp2	3G2	adMOVIE	mpeg4sr
ISO_3GPP_Fmt	618	559	3GPP video file	video/3gpp	3GP	adMOVIE	mpeg4sr
ISO_JPEG2000_JP2_Fmt	619	560	ISO-BMFF JPEG 2000 image	image/jp2	JP2	adRASTERIMAGE	jp2000sr , kpjp2000rdr
ISO_JPEG2000_JPM_Fmt	620	561	ISO-BMFF JPEG 2000 compound image	image/jpm	JPM	adRASTERIMAGE	jp2000sr , kpjp2000rdr
ISO_JPEG2000_JPX_Fmt	621	562	ISO-BMFF JPEG 2000 with extensions	image/jpx	JPX	adRASTERIMAGE	jp2000sr , kpjp2000rdr
ISO_QuickTime_Fmt	622	563	Apple ISO-BMFF QuickTime video	video/quicktime	QT, MOV	adMOVIE	MCI
KDDI_Video_Fmt	623	564	KDDI Video file	video/3gpp2		adMOVIE	mpeg4sr
MAF_Photo_Player_Fmt	624	565	MAF Photo Player			adMISC	
MPEG4_AVC_Fmt	625	566	ISO-BMFF MPEG-4 with AVC extension	video/mp4		adMOVIE	mpeg4sr
MPEG4_M4A_Fmt	626	567	Apple MPEG-4 Part 14 audio	audio/x-m4a	M4A	adSOUND	mpeg4sr
MPEG4_M4B_Fmt	627	568	Apple MPEG-4 Part 14 audio book	audio/mp4	M4B	adSOUND	mpeg4sr
MPEG4_M4P_Fmt	628	569	Apple MPEG-4 Part 14 protected audio	audio/mp4	M4P	adSOUND	mpeg4sr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
MPEG4_M4V_Fmt	629	570	Apple MPEG-4 Part 14 video	video/x-m4v	M4V	adMOVIE	mpeg4sr
MPEG4_Sony_PSP_Fmt	630	571	Sony PSP MPEG-4	audio/mp4	MP4	adSOUND	mpeg4sr
MPEG_21_Fmt	631	572	MPEG-21	audio/mp4		adMISC	mpeg4sr
Mobile_QuickTime_Fmt	632	573	Mobile QuickTime video	video/quicktime	MQV	adMOVIE	MCI
Motion_JPEG_2000_Fmt	633	574	Motion JPEG 2000	video/mj2	MJ2, MJP2	adMOVIE	jp2000sr , kpjp2000rdr
NTT_MPEG4_Fmt	634	575	NTT MPEG-4	video/mp4		adMOVIE	mpeg4sr
Nero_MPEG4_AVC_Profile	635	576	Nero MPEG-4 profile with AVC extension	video/mp4		adMOVIE	
Nero_MPEG4_Audio_Fmt	636	577	Nero AAC audio	audio/mp4		adSOUND	mpeg4sr
Nero_MPEG4_Profile	637	578	Nero MPEG-4 profile	video/mp4		adMOVIE	
OMA_DRM_Fmt	638	579	OMA DRM (ISOBMFF) Format			adMISC	
Panasonic_Camera_Fmt	639	580	Panasonic Digital Camera image			adRASTERIMAGE	
Ross_Video_Fmt	640	581	Ross video			adMOVIE	
SDA_Video_Fmt	641	582	SDA SD Memory Card video			adMOVIE	
Samsung_Stereoscopic_Fmt	642	583	Samsung stereoscopic stream			adMISC	
Sony_XAVC_Fmt	643	584	Sony XAVC video			adMOVIE	mpeg4sr
JPEG_2000_PGX_Fmt	644	585	JPEG 2000 PGX Verification Model image		PGX	adRASTERIMAGE	jp2000sr , kpjp2000rdr
Apple_Desktop_Services_Store_Fmt	645	586	Apple Desktop Services Store file		DS_Store	adMISC	
Core_Audio_Fmt	646	587	Apple Core Audio Format	audio/x-caf	CAF	adSOUND	
VICAR_Fmt	647	588	VICAR image format		IMG	adRASTERIMAGE	
FITS_Fmt	648	589	Flexible Image Transport System FITS image	image/fits	FIT	adRASTERIMAGE	
DIF_Fmt	649	590	Digital Interface Format (DIF) DV video		DV	adMOVIE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
MPEG_Transport_Stream_Fmt	650	591	MPEG Transport Stream data	video/MP2T	TS	adMISC	
MPEG_Sequence_Fmt	651	592	MPEG Sequence format	video/mpeg		adMISC	
Ogg_OGM_Fmt	652	593	Ogg OGM video format	video/ogg	OGM	adMOVIE	
Ogg_Speex_Fmt	653	594	Ogg Speex audio format	audio/ogg	SPX	adSOUND	
Ogg_Opus_Fmt	654	595	Ogg Opus audio format	audio/ogg	OGG	adSOUND	
Musepack_Audio_Fmt	655	596	Musepack audio format	audio/x-musepack	MPC	adSOUND	
ART_Image_Fmt	656	597	ART image format		ART	adRASTERIMAGE	
Vivo_Fmt	657	598	Vivo audio-video format	video/vnd.vivo	VIV	adMOVIE	
QCP_Fmt	658	599	Qualcomm QCP audio	audio/qcelp	QCP	adSOUND	
CSP_Codec_Fmt	659	600	Creative Signal Processor codec		CSP	adMISC	
TwinVQ_Fmt	660	601	NTT TwinVQ audio format		VQF	adSOUND	
Interplay_MVE_Fmt	661	602	Interplay MVE video format		MVE	adMOVIE	
IRIX_Moviemaker_Fmt	662	603	IRIX Silicon Graphics moviemaker video file	video/x-sgi-movie	MV, MOVIE	adMOVIE	
Sega_FILM_Fmt	663	604	Sega FILM video format		CPK, CAK	adMOVIE	
SMAF_Fmt	664	605	Synthetic music Mobile Application Format	application/vnd.smaf	MMF	adSOUND	
NIST_SPHERE_Fmt	665	606	NIST SPeech HEader REsources format		NIST	adSOUND	
Chinese_AVS_Fmt	666	607	Chinese AVS video format			adMOVIE	
VQA_Fmt	667	608	Westwood Studios Vector Quantized Animation video file		VQA	adANIMATION	
YAFA_Fmt	668	609	Wildfire YAFA animation		YAFA	adANIMATION	
Origin_MVE_Fmt	669	610	Origin Wing Commander III MVE movie format		MVE	adMOVIE	
BBC_Dirac_Fmt	670	611	BBC Dirac video format	video/x-dirac	DRC	adMOVIE	
Maya_ASCII_Fmt	671	612	Autodesk Maya ASCII file format		MA	adCAD	
RenderMan_Fmt	672	613	Pixar RenderMan Interface		RIB	adVECTORGRAPHIC	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
			Bytestream file				
NOFF_Binary_Fmt	673	614	NOFF 3D Object File Format		NOFF	adVECTORGRAPHIC	
VTK_ASCII_Fmt	674	615	Visualization Toolkit VTK ASCII format		VTK	adVECTORGRAPHIC	
VTK_Binary_Fmt	675	616	Visualization Toolkit VTK Binary format		VTK	adVECTORGRAPHIC	
Wolfram_CDF_Fmt	676	617	Wolfram Mathematica Computable Document Format	application/cdf	CDF	adMISC	
Wolfram_Notebook_Fmt	677	618	Wolfram Mathematica Notebook Format		NB	adMISC	
HDF4_Fmt	678	619	Hierarchical Data Format HDF4	application/x-hdf	HDF, H4	adMISC	
HDF5_Fmt	679	620	Hierarchical Data Format HDF5	application/x-hdf	HDF, H5	adMISC	
ARMovie_Fmt	680	621	Acorn RISC ARMovie video format		RPL	adMOVIE	
Windows_TV_DVR_Fmt	681	622	Windows Television DVR format		WTV	adMOVIE	
InstallShield_Z_Fmt	682	623	InstallShield Z archive format	application/x-compress	Z	adENCAPSULATION	
MS_DirectDraw_Surface_Fmt	683	624	Microsoft DirectDraw Surface container format		DDS	adENCAPSULATION	
Bink_Fmt	684	625	Bink audio-video container format		BIK, BK2	adMOVIE	
LZMA_Fmt	685	626	LZMA compressed data format	application/x-lzma	LZMA	adENCAPSULATION	
True_Audio_Fmt	686	627	True Audio format	audio/x-tta	TTA	adSOUND	
Keepass_Fmt	687	628	Keepass Password file		KDB, KDBX	adMISC	
RPM_Fmt	688	629	RPM Package Manager file	application/x-rpm	RPM	adENCAPSULATION	
Printer_Font_Metrics_Fmt	689	630	Adobe Printer Font Metrics format	application/x-font-printer-metric	PFM	adFONT	
Adobe_Font_Metrics_Fmt	690	631	Adobe Font Metrics ASCII format	application/x-font-adobe-metric	AFM	adFONT	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Printer_Font_ASCII_Fmt	691	632	Adobe Printer Font ASCII format	application/x-font-type1	PFA	adFONT	
Netware_Loadable_Module_Fmt	692	633	Netware Loadable Module format		NLM	adMISC	
TCPdump_pcap_Fmt	693	634	TCPdump packet stream capture savefile format	application/vnd.tcpdump.pcap	PCAP	adMISC	
Multiple_Master_Font_Fmt	694	635	Adobe Multiple master font format		MMM	adFONT	
TrueType_Font_Collection_Fmt	695	636	TrueType font collection format	application/x-font-ttf	TTC	adFONT	
Shapefile_Spatial_Index_Fmt	696	637	Shapefile binary spatial index format	application/x-shapefile	SBX, SBN	adGIS	
Java_Key_Store_Fmt	697	638	Java Key Store format	application/x-java-keystore	KS	adMISC	
Java_JCE_Key_Store_Fmt	698	639	Java JCE Key Store format	application/x-java-jce-keystore		adMISC	
Quark_Xpress_Intel_Fmt	699	640	QuarkXPress Intel format	application/vnd.quark.quarkxpress	QXB	adDESKTOPPUBLSH	
Windows_Imaging_Fmt	700	641	Microsoft Windows Imaging Format WIM		WIM	adMISC	
VMware_Virtual_Disk_Fmt	701	642	VMware Virtual Disk Format 5.0	application/x-vmrk	VMDK	adMISC	
XPCconnect_Typelib_Fmt	702	643	XPCconnect Typelib Format		XPT	adMISC	
MS_DOS_Compression_Fmt	703	644	Microsoft MS-DOS installation compression (SZDD, KWAJ)	application/x-ms-compress	EX_	adENCAPSULATION	
DLS_Fmt	704	645	DLS Downloadable Sounds format		DLS	adSOUND	
MS_Windows_Registry_Fmt	705	646	Microsoft Windows Registry format			adMISC	
Microsoft_Help_2_Fmt	706	647	Microsoft Help 2.0 format	application/x-ms-reader	HXD, HXW, HXH	adENCAPSULATION	
Qt_Translation_Fmt	707	648	Qt binary translation file format		QM	adMISC	
PEM_SSL_Certificate_Fmt	708	649	PEM-encoded SSL certificate	application/pkix-cert	CRT, PEM, CER, KEY	adENCAPSULATION	
PostScript_Printer_Description_Fmt	709	650	Adobe PostScript Printer Description file	application/vnd.cups-ppd	PPD	adMISC	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Speedo_Font_Fmt	710	651	Speedo Font format		SPD	adFONT	
InstallShield_Cabinet_Fmt	711	652	InstallShield Cabinet Archive format		CAB, HDR	adENCAPSULATION	
InstallShield_Uninstall_Fmt	712	653	InstallShield Uninstall format		ISU	adENCAPSULATION	
MS_OEDBX_Folder_Fmt	713	654	Outlook Express DBX folder database format		DBX	adENCAPSULATION	
LabVIEW_Fmt	714	655	National Instruments LabVIEW file format		VI	adMISC	
SAP_Archive_SAR_Fmt	715	656	SAP compression archive SAR format		SAR	adENCAPSULATION	
Netscape_Address_Book_Fmt	716	657	Netscape Address Book format		NAB	adMISC	
Universal_3D_Fmt	717	658	Universal 3D file format		U3D	adVECTORGRAPHIC	
Open_Inventor_ASCII_Fmt	718	659	Open Inventor ASCII format		IV	adVECTORGRAPHIC	
Open_Inventor_Binary_Fmt	719	660	Open Inventor Binary format		IV	adVECTORGRAPHIC	
X_Window_Dump_Fmt	720	661	X Window Dump image	image/x-xwindowdump	XWD	adRASTERIMAGE	
Git_Packfile_Fmt	721	662	Git Packfile format		PACK	adENCAPSULATION	
Xara_Xar_Fmt	722	663	Xara X Xar image format	application/vnd.xara	XAR	adVECTORGRAPHIC	
Internet_Archive_ARC_Fmt	723	664	Internet Archive ARC format	application/x-ia-arc	ARC	adENCAPSULATION	
Applix_Builder_Fmt	724	665	Applix Builder format		AB	adMISC	
Applix_Bitmap_Fmt	725	666	Applix Bitmap image format		IM	adRASTERIMAGE	
PEM_RSA_Private_Key_Fmt	726	667	PEM-encoded RSA private key		PEM	adENCAPSULATION	
MIFF_Fmt	727	668	Magick Image File Format		MIFF	adRASTERIMAGE	
Subversion_Dump_Fmt	728	669	Subversion Dump format			adENCAPSULATION	
Virtual_Hard_Disk_Fmt	729	670	Microsoft Virtual Hard Disk format	application/x-vhd	VHD	adENCAPSULATION	
Direct_Access_Archive_Fmt	730	671	PowerISO Direct Access Archive format		DAA	adENCAPSULATION	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Debian_Binary_Fmt	731	672	Debian binary package format	application/x-debian-package	DEB	adENCAPSULATION	
XUL_Fastload_Fmt	732	673	Mozilla XUL Fastload format		MFL	adMISC	
Nastran_OP2_Fmt	733	674	Nastran OP2 format		OP2	adCAD	
Binary_Logging_Fmt	734	675	CAD Binary Logging Format		BLF	adCAD	
Measurement_Data_Fmt	735	676	CAD Measurement Data Format		MDF	adCAD	
Abaqus_ODB_Fmt	736	677	Abaqus ODB Format		ODB	adCAD	
Open_Diagnostic_Data_Exchange_Fmt	737	678	Vector Open Diagnostic Data Exchange format		ODX	adCAD	xmlsr
Vector_ASCII_Fmt	738	679	Vector CAD ASCII ASC format		ASC	adCAD	
LSDYNA_State_Database_Fmt	739	680	LS-DYNA State Database format			adCAD	
LSDYNA_Binary_Output_Fmt	740	681	LS-DYNA binary output (binout) format			adCAD	
MS_Power_BI_Fmt	741	682	Microsoft Power BI Desktop format		PBIX	adANALYTICS	pbixsr
Tableau_Workbook_Fmt	742	683	Tableau Workbook format		TWB	adANALYTICS	xmlsr
Tableau_Packaged_Workbook_Fmt	743	684	Tableau Packaged Workbook format		TWBX	adANALYTICS	unzip
Tableau_Extract_Fmt	744	685	Tableau Extract format		TDE	adANALYTICS	
Tableau_Data_Source_Fmt	745	686	Tableau Data Source format		TDS	adANALYTICS	xmlsr
Tableau_Packaged_Data_Source_Fmt	746	687	Tableau Packaged Data Source format		TDSX	adANALYTICS	unzip
Tableau_Preferences_Fmt	747	688	Tableau Preferences format		TPS	adANALYTICS	xmlsr
Tableau_Map_Source_Fmt	748	689	Tableau Map Source format		TMS	adANALYTICS	xmlsr
Windows_Installer_Fmt	908	848	MSI Windows Installer format	application/x-ole-storage	MSI	adENCAPSULATION	olesr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Autodesk_3ds_Max_Fmt	909	849	Autodesk 3ds Max format		MAX	adCAD	
PhotoDraw_Mix_Fmt	910	850	PhotoDraw MIX image	image/vnd.mix	MIX	adRASTERIMAGE	
Softimage_SCN_Fmt	911	851	Softimage Scene SCN format		SCN	adCAD	
Parasolid_XT_Fmt	912	852	Parasolid ascii XT format		X_T	adCAD	
Parasolid_XB_Fmt	913	853	Parasolid binary XB format		X_B	adCAD	
IGES_Fmt	914	854	Initial Graphics Exchange Specification format	model/iges	IGS	adCAD	
ACE_Archive_Fmt	915	855	ACE archive format	application/x-ace-compressed	ACE	adENCAPSULATION	
Grasshopper_GHX_Fmt	916	856	Grasshopper GHX format		GHX	adCAD	xmlsr
MS_FrontPage_Macro_Fmt	917	857	Microsoft FrontPage macro file format		FPM	adWORDPROCESSOR	
MS_AtWork_Fax_Fmt	918	858	Microsoft AtWork Fax format		AWD	adFAXFORMAT	
MS_Image_Composer_Fmt	919	859	Microsoft Image Composer format		MIC	adRASTERIMAGE	
MS_Visual_InterDev_Fmt	920	860	Microsoft Visual InterDev web project items file		WDM	adMISC	
Macromedia_Flash_FLA_OLE_Fmt	921	861	Macromedia Flash FLA Project File OLE format		FLA	adWORDPROCESSOR	
Corel_Draw_X4_Fmt	922	862	CorelDRAW version X4 onwards	application/x-vnd.corel.zcf.draw.document+zip	CDRX	adVECTORGRAPHIC	
Ogg_Daala_Fmt	923	863	Ogg Daala video format	video/daala	OGV	adMOVIE	
Ogg_BBC_Dirac_Fmt	924	864	Ogg BBC Dirac video format	video/x-dirac	OGV	adMOVIE	
PKCS_7_Fmt	925	865	PKCS #7 cryptographic format	application/pkcs7-signature	P7S	adWORDPROCESSOR	
Time_Stamped_Data_Fmt	926	866	Time-stamped data format	application/timestamped-data	TSD	adENCAPSULATION	
Sereal_Fmt	927	867	Sereal data serialization format	application/sereal	SRL	adMISC	
Associated_Signature_Simple_Fmt	928	868	Associated Signature Container Simple format	application/vnd.etsi.asic-s+zip	ASICS	adENCAPSULATION	
Associated_Signature_	929	869	Associated Signature	application/vnd.etsi.asic-e+zip	ASICE	adENCAPSULATION	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Extended_Fmt			Container Extended format				
iBooks_Fmt	930	870	Apple iBooks format	application/x-ibooks+zip	IBOOKS	adWORDPROCESSOR	
PDF_Forms_Data_Fmt	931	871	PDF Forms Data Format	application/vnd.fdf	FDF	adWORDPROCESSOR	
PDF_XML_Forms_Data_Fmt	932	872	PDF XML Forms Data Format	application/vnd.adobe.xfdf	XPDF	adWORDPROCESSOR	xmlsr
AxCrypt_Fmt	933	873	AxCrypt encrypted document	application/x-axcrypt	AXX	adENCAPSULATION	
Unix_Archive_Fmt	934	874	Unix Archive ar format	application/x-archive	AR	adENCAPSULATION	
Berkeley_Btree_Database_Fmt	935	875	Berkeley DB btree database format	application/x-berkeley-db	DB	adDATABASE	
Berkeley_Hash_Database_Fmt	936	876	Berkeley DB hash database format	application/x-berkeley-db	DB	adDATABASE	
Berkeley_Log_Database_Fmt	937	877	Berkeley DB log database format	application/x-berkeley-db		adDATABASE	
Berkeley_Queue_Database_Fmt	938	878	Berkeley DB queue database format	application/x-berkeley-db		adDATABASE	
BitTorrent_Fmt	939	879	BitTorrent file format	application/x-bittorrent	TORRENT	adMISC	
Chrome_Extension_Fmt	940	880	Google Chrome Extension format	application/x-chrome-package	CRX	adENCAPSULATION	
Dalvik_Executable_Fmt	941	881	Dalvik Executable dex format	application/x-dex	DEX	adEXECUTABLE	
Foxmail_Fmt	942	882	Foxmail email format	application/x-foxmail	BOX	adWORDPROCESSOR	
GRIB_Fmt	943	883	General Regularly-distributed Information in Binary form GRIB format	application/x-grib	GRB, GRIB2	adMISC	
Zstandard_Fmt	944	884	Zstandard compression format	application/zstd	ZSTD	adENCAPSULATION	
LZ4_Fmt	945	885	LZ4 compressed file	application/x-lz4	LZ4	adENCAPSULATION	
MS_Money_Fmt	946	886	Microsoft Money format	application/x-msmoney	MNY	adSPREADSHEET	
NetCDF_Fmt	947	887	Network Common Data Form NetCDF format	application/x-netcdf	NC	adMISC	
SAS6_Data_Fmt	948	888	SAS 6 Data storage format	application/x-sas-data-v6	SD2	adDATABASE	
SAS_Transport_Fmt	949	889	SAS Transport File XPORT format	application/x-sas-xport	XPT, XPORT	adDATABASE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Snappy_Framed_Fmt	950	890	Snappy Framed compression format	application/x-snappy-framed	SZ	adENCAPSULATION	
Stata_Data_Fmt	951	891	Stata Data Format	application/x-stata-dta	DTA	adDATABASE	
SPSS_SAV_Fmt	952	892	SPSS Statistics Data File Format		SAV	adDATABASE	
Zoo_Archive_Fmt	953	893	Zoo Compressed Archive Format	application/x-zoo	ZOO	adENCAPSULATION	
CDX_Fmt	954	894	ChemDraw CDX format	chemical/x-cdx	CDX	adSCIENTIFIC	
CDXML_Fmt	955	895	ChemDraw CDXML format	application/vnd.chemdraw+xml	CDXML	adSCIENTIFIC	xmlsr
BPG_Fmt	956	896	Better Portable Graphics BPG format	image/x-bpg	BPG	adRASTERIMAGE	
Apple_Icon_Fmt	957	897	Apple Icon image format	image/icns	ICNS	adRASTERIMAGE	
NITF_Fmt	958	898	National Imagery Transmission Format NITF image	image/nitf	NTF, NITF	adRASTERIMAGE	
ERDAS_Imagine_Fmt	959	899	ERDAS Imagine image format	application/x-erdas-hfa	HFA, RRD, AUX	adRASTERIMAGE	
MS_Office_Temporary_Owner_Fmt	960	900	Microsoft Office temporary owner file	application/x-ms-owner		adMISC	
EAC3_Audio_Fmt	961	901	Enhanced-AC3 (EAC3) Audio File format	audio/eac3	AC3	adSOUND	
COFF_Relocatable_Fmt	962	902	Common Object File Format (COFF) relocatable object	application/x-object-file	O	adOBJECTMODULE	
COFF_Executable_Fmt	963	903	Common Object File Format (COFF) executable	application/x-executable-file		adEXECUTABLE	
COFF_Dynamic_Lib_Fmt	964	904	Common Object File Format (COFF) dynamic library	application/x-library-file		adLIBRARY	
ELF_Core_Fmt	965	905	ELF Core file	application/x-coreDump		adMISC	
Purify_Fmt	966	906	Rational Purify data file		PFY	adMISC	
Kryptel_Fmt	967	907	Kryptel encrypted file		EDC	adENCAPSULATION	
Windows_Core_Dump_Fmt	968	908	Windows heap or mini core dump file	application/x-dmp	DMP	adMISC	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Qt_Prerendered_Font_Fmt	969	909	Qt Prerendered Font format		QPF2	adFONT	
AIX_Relocatable_Fmt	970	910	AIX/RISC COFF relocatable object	application/x-object-file		adOBJECTMODULE	
AIX_Executable_Fmt	971	911	AIX/RISC COFF executable	application/x-executable-file		adEXECUTABLE	
AIX_Dynamic_Lib_Fmt	972	912	AIX/RISC COFF dynamic library	application/x-library-file	A	adLIBRARY	
HPUX_Relocatable_Fmt	973	913	HPUX/PA-RISC COFF relocatable object	application/x-object-file		adOBJECTMODULE	
HPUX_Executable_Fmt	974	914	HPUX/PA-RISC COFF executable	application/x-executable-file		adEXECUTABLE	
HPUX_Dynamic_Lib_Fmt	975	915	HPUX/PA-RISC COFF dynamic library	application/x-library-file	SL	adLIBRARY	
XML_EBCDIC_Fmt	976	916	EBCDIC-encoded XML file	application/xml	XML	adWORDPROCESSOR	
MPEG_JVT_H264_Fmt	977	917	MPEG JVT-NAL sequence H264 video	video/h264	264	adMOVIE	
Material_Exchange_Fmt	978	918	Material Exchange Format audio-video container format	application/mxf	MXF	adMOVIE	
MS_Agent_Character_Fmt	979	919	Microsoft Agent Character file		ACS	adMOVIE	
Quicken_Fmt	980	920	Quicken data file		QDF	adMISC	
MS_Outlook_Address_Fmt	981	921	Microsoft Outlook address file		WAB	adMISC	
MS_Answer_Wizard_Fmt	982	922	Microsoft Answer Wizard file			adMISC	
ADX_Fmt	983	923	ADX audio file		ADX	adSOUND	
System_Deployment_Image_Fmt	984	924	Microsoft System Deployment Image SDI format		SDI	adMISC	
Free_Lossless_Image_Fmt	985	925	Free Lossless Image Format (FLIF)	image/flif	FLIF	adRASTERIMAGE	
DPX_Fmt	986	926	Digital Picture Exchange (DPX) image format	image/dpx	DPX	adRASTERIMAGE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Avro_Fmt	987	927	Apache Avro binary format		AVRO	adMISC	
InstallShield_Archive_Fmt	988	928	InstallShield archive (early versions) format		EX_	adENCAPSULATION	
Mac_Executable_Fmt	989	929	Mac OS-X (Mach-O) executable format			adEXECUTABLE	
GDSII_Fmt	990	930	GDSII data format		GDS, GDS2	adCAD	gdsiisr
ActiveMime_Fmt	991	931	Microsoft ActiveMime (mso) documents	application/x-mso	MSO	adMISC	
SmartCharts_Fmt	992	932	BizInt SmartCharts data format		CHP, CHRR	adMISC	
Webex_ARF_Fmt	993	933	Webex advanced network ARF recordings		ARF	adMOVIE	
Webex_WRF_Fmt	994	934	Webex local WRF recordings		WRF	adMOVIE	
PGP_NetShare_Fmt	995	935	Symantec PGP NetShare encrypted file			adENCAPSULATION	
Ability_WP_OLE_Fmt	996	936	Ability Write later versions format		AWW	adWORDPROCESSOR	
Ability_SS_OLE_Fmt	997	937	Ability Spreadsheet later versions format		AWS	adSPREADSHEET	
InDesign_IDML_Fmt	998	938	Adobe InDesign IDML format	application/vnd.adobe.indesign-idml-package	IDML	adDESKTOPPUBLSH	
Executable_JAR_Fmt	999	939	Executable Java Archive (jar) file	application/java-archive	JAR	adENCAPSULATION	unzip
IDOL_IDX_Fmt	1000	940	IDOL Server IDX file		IDX	adENCAPSULATION	
Android_Package_Kit_Fmt	1001	941	Android Package Kit (APK) format	application/vnd.android.package-archive	APK	adEXECUTABLE	
Android_Binary_XML_Fmt	1002	942	Android Binary XML (compressed by aapt) format	application/xml	XML	adWORDPROCESSOR	
Java_WAR_Fmt	1003	943	Java WAR file format		WAR	adENCAPSULATION	
Java_EAR_Fmt	1004	944	Java EAR file format		EAR	adENCAPSULATION	
Atom_Syndication_Fmt	1005	945	Atom Syndication Format	application/atom+xml	ATOM	adWORDPROCESSOR	xmlsr
RSS_Fmt	1006	946	RSS syndication XML	application/rss+xml	RSS	adWORDPROCESSOR	xmlsr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
			format				
SMIL_Fmt	1007	947	Synchronized Multimedia Integration Language (SMIL) XML format	application/smil+xml	SMIL	adWORDPROCESSOR	xmlsr
XSLT_Fmt	1008	948	Extensible Stylesheet Language Transformations (XSLT) format	application/xslt+xml	XSL, XSLT	adWORDPROCESSOR	xmlsr
XML_Shareable_Playlist_Fmt	1009	949	XML Shareable Playlist Format (XSPF)	application/xspf+xml	XSPF	adWORDPROCESSOR	xmlsr
FictionBook_Fmt	1010	950	FictionBook e-book XML format	application/x-fictionbook+xml	FB2	adWORDPROCESSOR	xmlsr
Adobe_Premiere_Project_Fmt	1011	951	Adobe Premiere project format	image/vnd.adobe.premiere	PPJ	adMISC	
RDF_XML_Fmt	1012	952	RDF/XML format	application/rdf+xml	RDF	adWORDPROCESSOR	xmlsr
Really_Simple_Discovery_Fmt	1013	953	Really Simple Discovery (RSD) XML format	application/rsd+xml	RSD	adWORDPROCESSOR	xmlsr
SBML_Fmt	1014	954	Systems Biology Markup Language (SBML) XML format	application/sbml+xml	SBML	adWORDPROCESSOR	xmlsr
SRU_Fmt	1015	955	Search/Retrieve via URL (SRU) XML format	application/sru+xml	SRU	adWORDPROCESSOR	xmlsr
SSML_Fmt	1016	956	Speech Synthesis Markup Language (SSML) XML format	application/ssml+xml	SSML	adWORDPROCESSOR	xmlsr
PLS_Fmt	1017	957	Pronunciation Lexicon Specification (PLS) XML format	application/pls+xml	PLS	adWORDPROCESSOR	xmlsr
TEI_Fmt	1018	958	Text Encoding Initiative (TEI) XML format	application/tei+xml	TEI	adWORDPROCESSOR	xmlsr
METS_Fmt	1019	959	Metadata Encoding and Transmission Standard (METS) XML format	application/mets+xml	METS	adWORDPROCESSOR	xmlsr
MODS_Fmt	1020	960	Metadata Object Description Schema (MODS) XML format	application/mods+xml	MODS	adWORDPROCESSOR	xmlsr
Metalink_Fmt	1021	961	Metalink XML format	application/metalink4+xml	METALINK	adWORDPROCESSOR	xmlsr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Open_eBook_Fmt	1022	962	Open eBook (OEBPS) XML format	application/oebps-package+xml	OPF	adWORDPROCESSOR	xmlsr
SRGS_Fmt	1023	963	Speech Recognition Grammar Specification (SRGS) XML format	application/srgs+xml	SRGS	adWORDPROCESSOR	xmlsr
SPARQL_Results_Fmt	1024	964	SPARQL Query Results XML format	application/sparql-results+xml	SRX	adWORDPROCESSOR	xmlsr
Adobe_XML_Data_Package_Fmt	1025	965	Adobe XML Data Package format	application/vnd.adobe.xdp+xml	XDP	adWORDPROCESSOR	xmlsr
ESzigno_Fmt	1026	966	e-Szigno signed xml document	application/vnd.eszigno3+xml	ES3	adWORDPROCESSOR	xmlsr
Mozilla_XUL_Fmt	1027	967	Mozilla XML User Interface Language (XUL) XML format	application/vnd.mozilla.xul+xml	XUL	adWORDPROCESSOR	xmlsr
SyncML_Fmt	1028	968	Synchronization Markup Language (SyncML) XML format	application/vnd.syncml+xml	XML	adWORDPROCESSOR	xmlsr
VoiceXML_Fmt	1029	969	VoiceXML (VXML) XML format	application/voicexml+xml	VXML	adWORDPROCESSOR	xmlsr
TI_Target_Configuration_Fmt	1030	970	Texas Instruments CCXML target configuration XML format		CCXML	adWORDPROCESSOR	
LZFSE_Fmt	1031	971	Lempel-Ziv Finite State Entropy (LZFSE) compression format		LZFSE	adENCAPSULATION	
Kindle_eBook_Fmt	1032	972	Amazon Kindle or Mobipocket eBook format	application/vnd.amazon.ebook	AZW, PRC	adWORDPROCESSOR	
Oasis_Stream_Fmt	1033	973	Open Artwork System Interchange Standard (OASIS) format		OAS	adMISC	
Amazon_KFX_Fmt	1034	974	Amazon KFX eBook format		KFX	adWORDPROCESSOR	
KTX_Fmt	1035	975	KTX image format	image/ktx	KTX	adRASTERIMAGE	
GMSH_Mesh_Fmt	1036	976	GMSH Mesh polygon format	model/mesh	MSH	adCAD	
Collada_DAE_Fmt	1037	977	Collada Digital Asset Exchange (DAE) format	model/vnd.collada+xml	DAE	adCAD	xmlsr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
YIN_Fmt	1038	978	YIN XML format	application/yin+xml	YIN	adWORDPROCESSOR	xmlsr
MPEG_Playlist_Fmt	1039	979	MPEG audio playlist format	audio/mpegurl	M3U	adSOUND	
Windows_Audio_Playlist_Fmt	1040	980	Windows Audio playlist format	audio/x-ms-wax	WAX	adSOUND	xmlsr
DTS_Audio_Fmt	1041	981	DTS Coherent Acoustics audio format	audio/vnd.dts	DTS	adSOUND	
Chemical_Markup_Language_Fmt	1042	982	Chemical Markup Language (CML) XML format	chemical/x-cml	CML	adWORDPROCESSOR	xmlsr
CrystalMaker_Fmt	1043	983	CrystalMaker chemical format	chemical/x-cmdf	CMDF	adSCIENTIFIC	
VTK_XML_Fmt	1044	984	Visualization Toolkit VTK XML format	model/vnd.vtu	VTU	adVECTORGRAPHIC	xmlsr
IPFIX_Fmt	1045	985	IP Flow Information Export (IPFIX) format	application/ipfix	IPFIX	adMISC	
Portable_Font_Resource_Fmt	1046	986	Portable Font Resource font format	application/font-tdpfr	PFR	adFONT	
MARC_Fmt	1047	987	Machine-Readable Cataloging (MARC21) format	application/marc	MARC	adDATABASE	
MARC_XML_Fmt	1048	988	Machine-Readable Cataloging (MARC) XML format	application/marcxml+xml	XML	adWORDPROCESSOR	xmlsr
XAR_Fmt	1049	989	Extensible Archive (XAR) format			adENCAPSULATION	
Symbian_Installer_Fmt	1050	990	Symbian installer format	application/vnd.symbian.install	SIS	adENCAPSULATION	
SO_Drawing_XML_Fmt	1051	316	OpenDocument format (OpenOffice 1/StarOffice 6.7) Drawing XML	application/vnd.sun.xml.draw	SXD	adVECTORGRAPHIC	kpodfrdr
SO_Text_Global_XML_Fmt	1052	991	OpenDocument format (OpenOffice 1/StarOffice 6.7) Writer Master document XML	application/vnd.sun.xml.writer.global	SXG	adWORDPROCESSOR	
ODF_Chart_Fmt	1053	992	ODF Chart	application/vnd.oasis.opendocument.chart	ODC	adVECTORGRAPHIC	
ODF_Database_Fmt	1054	993	ODF Database	application/vnd.sun.xml.base	ODB	adDATABASE	
ODF_Image_Fmt	1055	994	ODF Image	application/vnd.oasis.opendocument.image	ODI	adRASTERIMAGE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
ODF_Text_Master_Fmt	1056	995	ODF Text Master	application/vnd.oasis.opendocument.text-master	ODM	adWORDPROCESSOR	
ODF_Text_Web_Fmt	1057	996	ODF Text Web	application/vnd.oasis.opendocument.text-web	OTH	adWORDPROCESSOR	
ODF_Chart_Template_Fmt	1058	997	ODF Chart Template	application/vnd.oasis.opendocument.chart-template	OTC	adVECTORGRAPHIC	
ODF_Formula_Template_Fmt	1059	998	ODF Formula Template	application/vnd.oasis.opendocument.formula-template	OTF	adWORDPROCESSOR	unzip
ODF_Drawing_Template_Fmt	1060	316	ODF Drawing/Graphics Template	application/vnd.oasis.opendocument.graphics-template	OTG	adVECTORGRAPHIC	kpodfrdr
ODF_Image_Template_Fmt	1061	999	ODF Image Template	application/vnd.oasis.opendocument.image-template	OTI	adRASTERIMAGE	
ODF_Presentation_Template_Fmt	1062	316	ODF Presentation Template	application/vnd.oasis.opendocument.presentation-template	OTP	adPRESENTATION	kpodfrdr
ODF_Spreadsheet_Template_Fmt	1063	315	ODF Spreadsheet Template	application/vnd.oasis.opendocument.spreadsheet-template	OTS	adSPREADSHEET	odfsssr
ODF_Text_Template_Fmt	1064	314	ODF Text Template	application/vnd.oasis.opendocument.text-template	OTT	adWORDPROCESSOR	odfwpsr
ODF_Chart_XML_Fmt	1065	1000	ODF Chart flat XML format	application/vnd.oasis.opendocument.chart.xml	FODC	adVECTORGRAPHIC	
ODF_Drawing_XML_Fmt	1066	1001	ODF Drawing/Graphics flat XML format	application/vnd.oasis.opendocument.formula.xml	FODG	adWORDPROCESSOR	
ODF_Formula_XML_Fmt	1067	1002	ODF Formula flat XML format	application/vnd.oasis.opendocument.graphics.xml	FODF	adVECTORGRAPHIC	
ODF_Image_XML_Fmt	1068	1003	ODF Image flat XML format	application/vnd.oasis.opendocument.image.xml	FODI	adRASTERIMAGE	
ODF_Presentation_XML_Fmt	1069	1004	ODF Presentation flat XML format	application/vnd.oasis.opendocument.presentation.xml	FODP	adPRESENTATION	
ODF_Spreadsheet_XML_Fmt	1070	1005	ODF Spreadsheet flat XML format	application/vnd.oasis.opendocument.spreadsheet.xml	FODS	adSPREADSHEET	
ODF_Text_XML_Fmt	1071	1006	ODF Text flat XML format	application/vnd.oasis.opendocument.text.xml	FODT	adWORDPROCESSOR	
ODF_Extension_Fmt	1072	1007	ODF Extension format	application/vnd.openofficeorg.extension	OXT	adMISC	
StarView_Metafile_Fmt	1073	1008	OpenOffice StarView MetaFile format	image/x-svm	SVM	adRASTERIMAGE	
BBeB_LRF_eBook_Fmt	1074	1009	Broad Band eBook (BBeB) in LRF format	application/x-ext-lrf	LRF	adWORDPROCESSOR	
GPG_Trust_DB_Fmt	1075	1010	GPG trust database format		GPG	adMISC	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
VICE_Emulator_Fmt	1076	1011	VICE (Versatile Commodore Emulator) format		VSF	adMISC	
Portable_Game_Notation_Fmt	1077	1012	Portable Game Notation chess format	application/vnd.chess-pgn	PGN	adWORDPROCESSOR	
Doom_WAD_Fmt	1078	1013	Doom IWAD/PWAD format	application/x-doom	WAD	adMISC	
Device_Tree_Blob_Fmt	1079	1014	Linux Device Tree Blob format		DTB	adMISC	
BDF_Font_Fmt	1080	1015	Glyph Bitmap Distribution Format	application/x-font-bdf	BDF	adFONT	
PC_Screen_Font_Fmt	1081	1016	PC Screen Font format	application/x-font-psf	PSF	adFONT	
JNLP_Fmt	1082	1017	Java Network Launching Protocol	application/x-java-jnlp-file	JNLP	adWORDPROCESSOR	xmlsr
XAML_Browser_Application_Fmt	1083	1018	XAML Browser Application (XBAP) format	application/x-ms-xbap	XBAP	adWORDPROCESSOR	xmlsr
MS_Binder_Fmt	1084	1019	Microsoft Office Binder format	application/x-msbinder	OBP	adENCAPSULATION	
XAP_Fmt	1085	1020	Microsoft Silverlight application (XAP) format	application/x-silverlight-app	XAP	adENCAPSULATION	
Stuffit_X_Fmt	1086	1021	Stuffit X (SITX) archive format	application/x-stuffitx	SITX	adENCAPSULATION	
FIG_Fmt	1087	1022	Facility for Interactive Generation of figures (FIG) image format	application/x-fig	FIG	adVECTORGRAPHIC	
XPIInstall_Fmt	1088	1023	XPIInstall Cross-Platform Installer Module (XPI) format	application/x-xpinstall	XPI	adENCAPSULATION	
XDF_Fmt	1089	1024	Extensible Data Format (XDF) XML format		XDF	adWORDPROCESSOR	xmlsr
MXML_Fmt	1090	1025	MXML UI markup language XML format		MXML	adWORDPROCESSOR	xmlsr
MusicXML_Fmt	1091	1026	MusicXML format	application/vnd.recordare.musicxml	MXL	adENCAPSULATION	xmlsr
Finale_Fmt	1092	1027	Finale audio format		MUS	adSOUND	
Spotfire_DXP_Fmt	1093	1028	TIBCO Spotfire DXP data format	application/vnd.spotfire.dxp	DXP	adANALYTICS	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
MS_Office_Theme_2007_Fmt	1094	1029	Microsoft Office theme format	application/vnd.ms-officetheme	THMX	adMISC	
Adobe_AIR_Installer_Fmt	1095	1030	Adobe AIR application installer package	application/vnd.adobe.air-application-installer-package+zip	AIR	adENCAPSULATION	
Flex_Project_Fmt	1096	1031	Adobe Flash Flex project file format	application/vnd.adobe.fxp	FXP	adENCAPSULATION	
FoxPro_Fmt	1097	1032	FoxPro compiled source format		FXP	adLIBRARY	
VST_Preset_Fmt	1098	1033	Virtual Studio Technology (VST) preset format		FXP	adSOUND	
Mischief_Image_Fmt	1099	1034	Mischief vector graphics image format		ART	adVECTORGRAPHIC	
FreeArc_Fmt	1100	1035	FreeArc archive format	application/x-freearc	ARC	adENCAPSULATION	
Autodesk_3ds_Fmt	1101	1036	Autodesk 3ds format	application/x-3ds	3DS	adCAD	
Monkeys_Audio_Fmt	1102	1037	Monkey's Audio format		APE	adSOUND	
CALS_Fmt	1103	1038	CALS raster image format		CAL	adRASTERIMAGE	
Dr_Halo_PAL_Fmt	1104	1039	Dr Halo raster image PAL file format		PAL	adRASTERIMAGE	
DPG_Fmt	1105	1040	Nintendo DS DPG video format		DPG	adMOVIE	
JPEG_XR_Fmt	1106	1041	JPEG XR (extended range) image format	image/vnd.ms-photo	JXR, HDP	adRASTERIMAGE	
TCR_eBook_Fmt	1107	1042	TCR/ZVR (Text Compression for Reader) eBook format		TCR, ZVR	adWORDPROCESSOR	
IHEX_Fmt	1108	1043	Intel Hex format		IHEX	adENCAPSULATION	
QCOW_Fmt	1109	1044	QEMU Copy On Write		QCOW	adENCAPSULATION	
VDI_Fmt	1110	1045	VirtualBox Disk Image		VDI	adENCAPSULATION	
OneNote_Alternate_Fmt	1111	1046	OneNote Alternative Packaging Format			adWORDPROCESSOR	onealtsr
RMS_Protected_Fmt	1112	1047	Rights Management Services (RMS)-protected format		PFILE, PPDF, PJPG, PTXT	adWORDPROCESSOR	pfilesr
Portfolio_PDF_Fmt	1113	1048	Portfolio PDF File	application/pdf	PDF	adWORDPROCESSOR	pdfsr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Crystal_Reports_Fmt	1114	1049	SAP Crystal Reports format	application/x-rpt	RPT	adANALYTICS	
Thumbs_db_Fmt	1115	1050	Microsoft Windows thumbs.db format		DB	adENCAPSULATION	
PagePlus_Fmt	1116	1051	Serif PagePlus format		PPP	adDESKTOPPUBLISH	
MS_Project_Exchange_Fmt	1117	1052	Microsoft Project Exchange format		MPX	adSCHEDULE	
MS_Management_Pack_MPX_Fmt	1118	1053	Microsoft Systems Center Operation Manager (SCOM) management pack MPX format		MPX	adMISC	xmlsr
AutoCAD_VBA_Project_Fmt	1119	1054	AutoCAD VBA project format		DVB	adMISC	
PLY_ASCII_Fmt	1120	1055	Polygon File Format (PLY) ASCII format		PLY	adCAD	
PLY_Binary_Fmt	1121	1056	Polygon File Format (PLY) binary format		PLY	adCAD	
JavaView_JVX_Fmt	1122	1057	JavaView XML (JVX) format		JVX	adCAD	xmlsr
X3D_Fmt	1123	1058	Extensible 3d Graphics (X3D) XML format	model/x3d+xml	X3D	adCAD	
ZBrush_Project_Fmt	1124	1059	ZBrush ZProject (ZPR) format		ZPR	adCAD	
ZBrush_Tool_Fmt	1125	1060	ZBrush ZTtool (ZTL) format		ZTL	adCAD	
Windows_Installer_Patch_Fmt	1126	1061	Microsoft Windows Installer Patch Package (MSP) format		MSP	adENCAPSULATION	
Windows_Installer_Transform_Fmt	1127	1062	Microsoft Windows Installer Transform (MST) format		MST	adENCAPSULATION	
Lotus_Approach_Fmt	1128	1063	Lotus Approach format	application/vnd.lotus-approach	APR, MPR	adDATABASE	
Outlook_SendRcv_Settings_Fmt	1129	1064	Microsoft Outlook 2002 Send-Receive Settings		SRS	adMISC	
MS_Publisher_Scheme_Fmt	1130	1065	Microsoft Publisher colour scheme		SCM	adMISC	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
SO_Chart_Fmt	1131	1066	Star Office 4,5 Chart	application/vnd.stardivision.chart	SDS	adVECTORGRAPHIC	
SO_Database_Fmt	1132	1067	Star Office 4,5 Database	application/vnd.stardivision.base	SDB	adDATABASE	
SO_Library_Fmt	1133	1068	Star Office 4,5 Library		SBL	adLIBRARY	
PageMaker_Document_Fmt	1134	1069	Adobe PageMaker document	application/pagemaker	PMD	adDESKTOPPUBLSH	
MS_DTS_Fmt	1135	1070	Microsoft Data Transformation Services (DTS) package file		DTS	adMISC	
Cognos_PowerPlay_PPR_Fmt	1136	1071	Cognos PowerPlay up to version 7 (PPR) format		PPR	adANALYTICS	
Visual_Studio_SUO_Fmt	1137	1072	Microsoft Visual Studio solution user options (suo) file		SUO	adMISC	
MS_GraphEdit_Fmt	1138	1073	Microsoft GraphEdit File format		GRF	adMISC	
ArcGIS_Graph_Fmt	1139	1074	ArcGIS Graph format		GRF	adGIS	
SID_Audio_Fmt	1140	1075	SID Audio format	audio/prs.sid	SID	adSOUND	
MrSID_Fmt	1141	1076	LizardTech MrSID image format	image/x-mrsid	SID	adRASTERIMAGE	
Cardfile_Fmt	1142	1077	Microsoft Windows Cardfile address book format	application/x-mscardfile	CRD	adWORDPROCESSOR	
MS_Word_Mac_4_Fmt	1143	205	Microsoft Word for Macintosh (version 4,5)	application/msword	DOC	adWORDPROCESSOR	mbsr
WordPerfect_5_Fmt	1144	80	WordPerfect (version 5)	application/x-corel-wordperfect	WOP, DOC	adWORDPROCESSOR	wosr
WordPerfect_6_Fmt	1145	178	WordPerfect (version 6 and higher)	application/x-corel-wordperfect	WPD	adWORDPROCESSOR	wp6sr
WordPerfect_Graphics_1_Fmt	1146	85	WordPerfect Graphics (version 1)	application/vnd.wordperfect	WPG, QPG	AutoDetNoFormat	
Organization_Chart_Fmt	1147	1078	OrgPlus Organization Chart	application/orgplus	OPX	adDATABASE	
Lotus_Organizer_Fmt	1148	1079	Lotus Organizer documents	application/vnd.lotus-organizer	OR2, OR3, OR4, OR5, OR6	adSCHEDULE	
MS_DBML_Fmt	1149	1080	Microsoft Database Markup Language XML document		DBML	adWORDPROCESSOR	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
XMind_Fmt	1150	1081	XMind document	application/xmind	XMIND	adPRESENTATION	
MSI_Cerius_Fmt	1151	1082	MSI Cerius chemical formula document	chemical/x-cerius	MSI	adSCIENTIFIC	
GenBank_Fmt	1152	1083	GenBank DNA character sequence document	chemical/x-genbank	GB	adSCIENTIFIC	
GIS_World_File_Fmt	1153	1084	ESRI GIS World file		BPW, GFW, JGW, J2W, PGW, SDW, TFW, WLD	adGIS	afsr
GIS_Projection_Metadata_Fmt	1154	1085	ESRI Projection Metadata (PRJ) file		PRJ	adGIS	
PowerWorld_Binary_Fmt	1155	1086	PowerWorld Binary (PWB) file		PWB	adCAD	
PowerWorld_Display_Fmt	1156	1087	PowerWorld Display (PWD) file		PWD	adCAD	
ArcXML_Fmt	1157	1088	ESRI ArcIMS project XML file (ArcXML)		AXL	adGIS	
GAMS_GDX_Fmt	1158	1089	General Algebraic Modeling System (GAMS) Data Exchange (GDX) format		GDX	adSCIENTIFIC	
ArcMap_MXD_Fmt	1159	1090	ArcMap Map Exchange Document project (MXD)		MXD	adGIS	
RRDtool_Fmt	1160	1091	RRDtool (Round Robin Database) data file		RRD	adDATABASE	
HWPX_Fmt	1161	1092	Hangul HWPX document	application/hwp+zip	HWPX	adWORDPROCESSOR	
SolidWorks_2015_Fmt	1162	1093	SolidWorks (2015 onwards) file		SLDPRT, SLDDRW, SLDASM	adCAD	
MS_Photo_Editor_Fmt	1163	1094	Microsoft Photo Editor 'embedded GIF' file	application/vnd.ms-photo-editor		adRASTERIMAGE	
MS_Word_HTML_Fmt	1164	1095	Microsoft Word HTML format		DOC, HTM	adWORDPROCESSOR	
MS_Excel_HTML_Fmt	1165	1096	Microsoft Excel HTML format		XLS, HTM	adWORDPROCESSOR	
Portable_FloatMap_Fmt	1166	1097	Portable FloatMap (PFM) image	image/x-portable-floatmap	PFM	adRASTERIMAGE	
RGBE_Fmt	1167	1098	Radiance RGBE (HDR)	image/vnd.radiance	HDR, PIC, RGBE,	adRASTERIMAGE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
			image		XYZE		
APNG_Fmt	1168	1099	Animated Portable Network Graphics (Animated-PNG)	image/apng	APNG, PNG	adANIMATION	kppngrdr
Enhanced_Compressed_Wavelet_Fmt	1169	1100	Enhanced Compressed Wavelet image	image/ecw	ECW	adRASTERIMAGE	
Ensoniq_Waveset_Fmt	1170	1101	Ensoniq Waveset audio data file		ECW	adSOUND	
Corel_Photo_Paint_Fmt	1171	1102	Corel Photo Paint (version 7 and higher)	image/x-corelphotopaint	CPT	adRASTERIMAGE	
OpenRaster_Fmt	1172	1103	OpenRaster image	image/openraster	ORA	adRASTERIMAGE	
Krita_Fmt	1173	1104	Krita image	application/x-krita	KRA	adRASTERIMAGE	
Gerber_Fmt	1174	1105	Gerber image format	application/vnd.gerber	GBR	adVECTORGRAPHIC	
PGML_Fmt	1175	1106	Precision Graphics Markup Language		PGML	adVECTORGRAPHIC	
Away3D_Fmt	1176	1107	Away3D scene file		AWD	adCAD	
CAD_3MF_Fmt	1177	1108	3D Manufacturing Format document	application/vnd.ms-package.3dmanufacturing-3dmodel+xml	3MF	adCAD	
AMF_Fmt	1178	1109	Additive manufacturing file format (AMF) document	application/x-amf	AMF	adCAD	xmlsr
C3D_Fmt	1179	1110	Coordinate 3D (C3D) format		C3D	adCAD	
CAD_3DSystems_BFF_Fmt	1180	1111	3D Sprint (3D Systems) SLA Build file		BFF	adCAD	
NRRD_Fmt	1181	1112	NRRD (nearly raw raster data) image format		NRRD	adRASTERIMAGE	
Cinema_4D_Fmt	1182	1113	Cinema 4D model		C4D	adCAD	
FBX_ASCII_Fmt	1183	1114	Kaydara FBX project (ASCII)		FBX	adCAD	
FBX_Binary_Fmt	1184	1115	Kaydara FBX project (binary)		FBX	adCAD	
Wavefront_OBJ_Fmt	1185	1116	Wavefront OBJ geometry definition file		OBJ	adCAD	
Wavefront_MTL_Fmt	1186	1117	Wavefront Material Template Library (MTL)		MTL	adCAD	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
MS_Power_BI_Template_Fmt	1187	1118	Microsoft Power BI Desktop template format		PBIT	adANALYTICS	
Windows_Sticky_Notes_Fmt	1188	1119	Microsoft Windows Sticky Notes format		SNT	adWORDPROCESSOR	
BlakHole_Fmt	1189	1120	BlakHole compression format		BH	adENCAPSULATION	
PowerArchiver_Fmt	1190	1121	PowerArchiver PA compression format		PA	adENCAPSULATION	
PageMagic_Fmt	1191	1122	NEBS PageMagic format		DTP	adDESKTOPPUBLSH	
PIM_Archiver_Fmt	1192	1123	PIM Archiver format		PIM	adENCAPSULATION	
Softdisk_Text_Compressor_Fmt	1193	1124	Softdisk Text Compressor format		CTX	adENCAPSULATION	
Ability_PhotoPaint_Fmt	1194	1125	Ability Office PhotoPaint image		APX	adRASTERIMAGE	
Softlib_Fmt	1195	1126	Softdisk Softlib compression format		SLB	adENCAPSULATION	
Timeworks_Publisher_Fmt	1196	1127	Timeworks Publisher (Publish It) format		DTP	adDESKTOPPUBLSH	
Scribe_Fmt	1197	1128	Scribe markup language and word processing system		MSS	adWORDPROCESSOR	afsr
SQLite_Write_Ahead_Log_Fmt	1198	1129	SQLite Write-Ahead Log file		WAL	adDATABASE	
SQLite_WAL_Index_Fmt	1199	1130	SQLite WAL-index (shm) file		SHM	adDATABASE	
AutoForm_Design_Fmt	1200	1131	AutoForm Design file		AFD	adCAD	
TSV_Fmt	1201	1132	Tab-separated values (TSV) file	text/tab-separated-values	TSV, TAB	adWORDPROCESSOR	afsr , afsr
OpenStreetMap_XML_Fmt	1202	1133	OpenStreetMap XML data		OSM	adGIS	
OpenStreetMap_PBF_Fmt	1203	1134	OpenStreetMap Protocolbuffer Binary Format data file (.osm.pbf)		PBF	adGIS	
Nero_Audio_Compilation_Fmt	1204	1135	Nero Audio-CD compilation file		NRA	adMISC	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Nero_ISO_Compilation_Fmt	1205	1136	Nero ISO compilation file		NRI	adMISC	
WordStar_for_Windows_Fmt	1206	1137	WordStar for Windows file		WSD	adWORDPROCESSOR	stringsr
MS_Outlook_PAB_Fmt	1207	1138	Microsoft Outlook Personal Address Book (PAB)		PAB	adMISC	
HLSL_FXO_Fmt	1208	1139	DirectX High-Level Shader Language (HLSL) pre-compiled shader		FXO	adCAD	
HLSL_CSO_Fmt	1209	1140	DirectX High-Level Shader Language (HLSL) compiled shader object		CSO	adCAD	
Oberon_Document_Fmt	1210	1141	Component Pascal / Oberon Document file		ODC	adSOURCECODE	
Oberon_Symbol_Fmt	1211	1142	Component Pascal / Oberon Symbol file		OSF	adOBJECTMODULE	
Oberon_Code_Fmt	1212	1143	Component Pascal / Oberon Code (executable and loadable object) file		OCF	adEXECUTABLE	
Python_Bytecode_Fmt	1213	1144	Python compiled bytecode	application/x-bytecode.python	PYC	adEXECUTABLE	
PCPaint_Fmt	1214	1145	PCPaint / Pictor Paint image format		PIC	adRASTERIMAGE	
PCRaster_Map_Fmt	1215	1146	PCRaster Map / Cross System Format geographical data		MAP, CSF	adGIS	
COM_Type_Library_Fmt	1216	1147	Microsoft Component Object Model (COM) Type library		TLB	adLIBRARY	
MS_Visual_C_Export_Fmt	1217	1148	Microsoft Visual C++ Export file		EXP	adLIBRARY	
Lotus_Organizer_Report_Fmt	1218	1149	Lotus Organizer report document		REP	adSCHEDULE	
Audible_Audiobook_AA_Fmt	1219	1150	Audible Audiobook (AA) file	audio/audible	AA	adSOUND	
DOS_RED_Fmt	1220	1151	MS-DOS RED installer library format		RED	adLIBRARY	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
CA_ZIPXP_Fmt	1221	1152	CA Technologies ZIPXP compressed document		CAZ	adENCAPSULATION	
Kindle_Topaz_Fmt	1222	1153	Amazon Kindle Topaz eBook		AZW, AZW1, TPZ	adWORDPROCESSOR	
Windows_Shim_Database_Fmt	1223	1154	Microsoft Windows Shim Database file		SDB	adDATABASE	
MS_Incremental_Linker_Fmt	1224	1155	Microsoft Visual Studio incremental linker file		ILK	adMISC	
Lotus_Smart_Icon_Fmt	1225	1156	Lotus Smart Icon image file		SMI	adRASTERIMAGE	
Lotus_Organizer_Layout_Fmt	1226	1157	Lotus Organizer print/paper layout file		PLT	adSCHEDULE	
CMZ_Fmt	1227	1158	CMZ compression format		CMZ	adENCAPSULATION	
RFFlow_Fmt	1228	1159	RFFlow flowchart document		FLO	adPRESENTATION	
InstallShield_Script_Fmt	1229	1160	InstallShield script document		INS	adENCAPSULATION	
InstallShield_Rules_Fmt	1230	1161	InstallShield Compiled Rules file		INX	adENCAPSULATION	
Windows_FTS_Fmt	1231	1162	Microsoft Windows 95/NT help full-text-search file		FTS	adDATABASE	
DVD_Info_Fmt	1232	1163	DVD Information (IFO) file	content/dvd	IFO	adDATABASE	
Emacs_Lisp_Bytecode_Fmt	1233	1164	Byte-compiled Lisp (Emacs/XEmacs)	application/x-bytecode.elisp	ELC	adEXECUTABLE	
Windows_Resource_Fmt	1234	1165	Microsoft Windows binary resource file		RES	adMISC	
MS_Precompiled_Header_Fmt	1235	1166	Microsoft Visual C/C++ binary pre-compiled header		PCH	adMISC	
Borland_Turbo_Project_Fmt	1236	1167	Borland Turbo C project file		PRJ	adMISC	
PS_Font_Descriptor_Fmt	1237	1168	PostScript binary Font Descriptor file		NTF	adFONT	
MySQL_Index_Fmt	1238	1169	MySQL MyISAM Table index		MYI	adDATABASE	
MS_SQL_Fmt	1239	1170	Microsoft SQL Server primary database file		MDF	adDATABASE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
DNL_eBook_Fmt	1240	1171	DNAML DNL eBook		DNL	adWORDPROCESSOR	
GD_Image_Fmt	1241	1172	GD Library image		GD, GD2	adRASTERIMAGE	
iTunes_Library_Fmt	1242	1173	Apple iTunes music library		ITL	adDATABASE	
MS_SQM_Fmt	1243	1174	Microsoft Windows Live Messenger/Mail log file		SQM	adMISC	
VIFF_Fmt	1244	1175	Khoros Visualization Image File Format (VIFF)	image/x-viff	XV, VIF, VIFF	adRASTERIMAGE	
JBIG_Fmt	1245	1176	JBIG (JBIG1) image	image/jbig	JBG, JBIG, BIE	adRASTERIMAGE	
CodeWarrior_Project_Fmt	1246	1177	CodeWarrior C/C++ project		MCP	adMISC	
PaintShop_Pro_JBF_Fmt	1247	1178	PaintShop Pro JBF image cache file	image/jbf	JBF	adMISC	
Delphi_Diagram_Portfolio_Fmt	1248	1179	Delphi Diagram Portfolio file		DDP	adMISC	
Adobe_Swatch_Exchange_Fmt	1249	1180	Adobe Swatch Exchange Format		ASE, ASEF	adRASTERIMAGE	
ASCII_Scene_Exporter_Fmt	1250	1181	Autodesk 3ds Max ASCII Scene Exporter file		ASE	adCAD	
AVR_Fmt	1251	1182	AVR (Audio Visual Research) format		AVR	adSOUND	
Winamp_AVS_Fmt	1252	1183	Winamp AVS (Advanced Visualization Studio) plug-in file		AVS	adSOUND	
After_Effects_Project_Fmt	1253	1184	Adobe After Effects project		AEP	adMOVIE	
Anfy_Applet_Generator_Fmt	1254	1185	Anfy (Java) Applet Generator file		AJP	adMISC	
SmartCipher_Fmt	1255	1186	SmartCipher encrypted file			adENCAPSULATION	
General_Exchange_Fmt	1256	1187	General Exchange Format (GXF)	application/gxf	GXF	adMOVIE	
Maxis_XA_Fmt	1257	1188	Maxis XA audio file		XA	adSOUND	
NUT_Fmt	1258	1189	NUT Open Container Format		NUT	adMOVIE	
OpenMG_Audio_Fmt	1259	1190	Sony OpenMG Audio		OMA, OMG	adSOUND	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
			(OMA) container file				
TXD_Fmt	1260	1191	Renderware Texture Dictionary (TXD) file		TXD	adRASTERIMAGE	
DFA_Fmt	1261	1192	DreamForge DFA FMV format		DFA	adMOVIE	
FunCom_ISS_Fmt	1262	1193	FunCom ISS audio		ISS	adSOUND	
Sony_MSV_Fmt	1263	1194	Sony Compressed Audio (MSV/DVF)		DVF, ICS, MSV	adSOUND	
THP_Fmt	1264	1195	GameCube THP Video		THP	adMOVIE	
Smush_Animation_Fmt	1265	1196	Smush Animation Format (SAN)		SAN, NUT	adANIMATION	
SIFF_Audio_Fmt	1266	1197	Beam Software SIFF audio file		SON	adSOUND	
SNES_SPC_Fmt	1267	1198	SNES SPC700 audio file		SPC	adSOUND	
Sierra_VMD_Fmt	1268	1199	Sierra Video and Music Data format		VMD	adMOVIE	
VTech_MJP_Fmt	1269	1200	VTech MHP video format		MJP	adMOVIE	
Nullsoft_Video_Fmt	1270	1201	Nullsoft Video format (NSV)		NSV	adMOVIE	
Shorten_Fmt	1271	1202	Shorten audio file		SHN	adSOUND	
Leitch_Video_Fmt	1272	1203	Leitch Exchange Format video (LXF)		LXF	adMOVIE	
ETV_Fmt	1273	1204	ETV video file		ETV	adMOVIE	
TAK_Audio_Fmt	1274	1205	TAK audio file		TAK	adSOUND	
Maelstrom_ANM_Fmt	1275	1206	Maelstrom ANM animation		ANM	adANIMATION	
SW_ANM_Fmt	1276	1207	Savage Warriors ANM animation		ANM	adANIMATION	
DeluxePaint_Animation_Fmt	1277	1208	DeluxePaint animation		ANM	adANIMATION	
Crack_Art_Fmt	1278	1209	Crack Art image		CA1	adRASTERIMAGE	
Time_Shift_Video_Fmt	1279	1210	Time Shift Video (TSV) format		TSV	adMOVIE	
XBV_Fmt	1280	1211	XBV video		XBV	adMOVIE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
HNM4_Fmt	1281	1212	CRYO HNM4 video		HNM	adMOVIE	
HNM6_Fmt	1282	1213	CRYO HNM6 video		HNM, HNS	adMOVIE	
NXV_Fmt	1283	1214	NXV video		NXV	adMOVIE	
VP5_Fmt	1284	1215	On2 VP5 video		VP5	adMOVIE	
FutureVision_FST_Fmt	1285	1216	FutureVision FST video		FST	adMOVIE	
Electronic_Arts_Audio_Fmt	1286	1217	Electronic Arts audio file		STR	adSOUND	
YOP_Fmt	1287	1218	Psygnosis YOP video		YOP	adMOVIE	
Matrox_Setup_Program_Fmt	1288	1219	Matrox Setup Program Archive MVA file		MVA	adMISC	
Vivado_Design_Suite_Fmt	1289	1220	Xilinx Vivado Design Suite file		VDS	adMISC	
Meridian_Lossless_Packing_Fmt	1290	1221	Meridian Lossless Packing Audio file		MLP	adSOUND	
Electronic_Arts_SEAD_Fmt	1291	1222	Electronic Arts SEAD audio		TGV	adSOUND	
Electronic_Arts_MPC_Fmt	1292	1223	Electronic Arts MPC video		MPC	adMOVIE	
PMP_Fmt	1293	1224	PMP video		PMP	adMOVIE	
DEGAS_Fmt	1294	1225	DEGAS (Design & Entertainment Graphic Arts System) image		PI1, PI2, PI3	adRASTERIMAGE	
DEGAS_Compressed_Fmt	1295	1226	DEGAS (Design & Entertainment Graphic Arts System) compressed image		PC1, PC2, PC3	adRASTERIMAGE	
AutoCAD_Plotter_Fmt	1296	1227	AutoCAD Plot Style and Configuration files		CTB, STB, PC3, PMP	adCAD	
Tiny_Stuff_Fmt	1297	1228	Tiny Stuff image		TNY, TN1, TN2, TN3, TN4, TN5, TN6	adRASTERIMAGE	
JV_Video_Fmt	1298	1229	Bitmap Brothers JV video		JV	adMOVIE	
REDCode_Fmt	1299	1230	REDCode video format		R3D	adMOVIE	
SIFF_Video_Fmt	1300	1231	Beam Software SIFF video file		VB	adMOVIE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
VP6_Fmt	1301	1232	On2 VP6 video		VP6	adMOVIE	
MTV_Fmt	1302	1233	Chinese MP4/MTV video		MTV	adMOVIE	
RSO_Fmt	1303	1234	Mindstorm RSO audio		RSO	adSOUND	
Star3_Fmt	1304	1235	Creative Labs Star 3 audio		ST3	adSOUND	
DXA_Fmt	1305	1236	Runesoft DXA video		DXA	adMOVIE	
MTH_Fmt	1306	1237	Nintendo GameCube video file		MTH	adMOVIE	
MAD_Fmt	1307	1238	Electronic Arts MAD video file		MAD	adMOVIE	
Bink2_Fmt	1308	1239	Bink Video 2 audio-video container		BIK, BK2	adMOVIE	
PVA_Fmt	1309	1240	TechnoTrend PVA video		PVA	adMOVIE	
Interplay_ACMP_Fmt	1310	1241	Interplay ACMP audio			adSOUND	
Ipix_Fmt	1311	1242	Ipix spherical image		IPX	adRASTERIMAGE	
IVR_Fmt	1312	1243	RealNetworks Internet Video Recording (IVR) file		IVR	adMOVIE	
NuppelVideo_Fmt	1313	1244	NuppelVideo file		NUV	adMOVIE	
VFlash_PTX_Fmt	1314	1245	VTech V.Flash VTX image		PTX	adRASTERIMAGE	
PMD_Ringtone_Fmt	1315	1246	Polyphonic Ringtone PMD audio	application/x-pmd	PMD	adSOUND	
RoQ_Fmt	1316	1247	RoQ video		ROQ	adMOVIE	
CRYO_APC_Fmt	1317	1248	CRYO Interactive APC audio		APC, HNM, BF, ZIK	adSOUND	
VGZ_Fmt	1318	1249	VGZ video		VGZ	adMOVIE	
Novastorm_Video_Fmt	1319	1250	Novastorm Media video file		FA, FLM	adMOVIE	
UTalk_Fmt	1320	1251	MicroTalk/UTalk audio		UTK	adSOUND	
Xbox_XMV_Fmt	1321	1252	Microsoft Xbox XMV video		XMV	adMOVIE	
AbiWord_Fmt	1322	1253	AbiWord document	application/x-abiword	ABW	adWORDPROCESSOR	
AbiWord_Template_Fmt	1323	1254	AbiWord template		ABT	adWORDPROCESSOR	
Psion_Word_Fmt	1324	1255	Psion EPOC Word document		PSI, PSITEXT	adWORDPROCESSOR	stringssr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Psion_Sheet_Fmt	1325	1256	Psion EPOC Sheet spreadsheet		PSISHEET	adSPREADSHEET	
Psion_Sketch_Fmt	1326	1257	Psion EPOC Sketch image			adRASTERIMAGE	
Psion_Record_Fmt	1327	1258	Psion EPOC Record audio			adSOUND	
Psion_MBM_Fmt	1328	1259	Psion EPOC Multi-Bitmap (MBM) image		MBM	adRASTERIMAGE	
Psion_TextEd_Fmt	1329	1260	Psion EPOC TextEd file			adWORDPROCESSOR	stringssr
Psion_AIF_Fmt	1330	1261	Psion EPOC Application Information File (AIF)		AIF	adRASTERIMAGE	
Psion_PIC_Fmt	1331	1262	Psion 3 PIC bitmap		PIC	adRASTERIMAGE	
Psion_Object_Fmt	1332	1263	Psion 3 OPL Object File		OPA, OPO	adENCAPSULATION	
Psion_Executable_Fmt	1333	1264	Psion 3 IMG/APP executable		IMG, APP	adEXECUTABLE	
Psion_Sound_Fmt	1334	1265	Psion 3 Sound file		WVE	adSOUND	
Psion_Database_Fmt	1335	1266	Psion EPOC Database			adDATABASE	
Psion_Word_3_Fmt	1336	1267	Psion 3 Word document		WRD	adWORDPROCESSOR	stringssr
Psion_Sheet_3_Fmt	1337	1268	Psion 3 Sheet spreadsheet		SPR	adSPREADSHEET	
Zoner_Draw_Fmt	1338	1269	Zoner Draw / Zoner Callisto Metafile (ZMF)		ZMF	adVECTORGRAPHIC	
Zoner_BMI_Fmt	1339	1270	Zoner BMI image		BMI	adRASTERIMAGE	
TealDoc_Fmt	1340	1271	TealDoc PalmOS eBook		PDB	adWORDPROCESSOR	
TealPaint_Fmt	1341	1272	TealPaint PalmOS eBook		PDB	adWORDPROCESSOR	
PalmDOC_Fmt	1342	1273	PalmDOC / Aportis DOC eBook	application/x-aportisdoc	PRC, PDB	adWORDPROCESSOR	
QiOO_Fmt	1343	1274	QiOO mobile eBook		JAR	adWORDPROCESSOR	
Plucker_Fmt	1344	1275	Plucker eBook	application/prs.plucker	PDB	adWORDPROCESSOR	
eReader_Fmt	1345	1276	eReader (Palm Reader/ Peanut Reader) eBook		PDB	adWORDPROCESSOR	
Quickword_Fmt	1346	1277	PalmOS Quickword document		PRC	adWORDPROCESSOR	stringssr
Quicksheet_Fmt	1347	1278	PalmOS Quicksheet document		PRC	adSPREADSHEET	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Quickpoint_Fmt	1348	1279	PalmOS Quickpoint document		PRC	adPRESENTATION	
TealMeal_Fmt	1349	1280	TealMeal PalmOS database		PDB	adDATABASE	
zTXT_Fmt	1350	1281	zTXT eBook	application/x-pdb-ztxt-ebook	PDB	adWORDPROCESSOR	
TomeRaider_Fmt	1351	1282	TomeRaider eBook		TR	adWORDPROCESSOR	
TomeRaider_PDB_Fmt	1352	1283	TomeRaider PDB eBook		TR2, TR3	adWORDPROCESSOR	
WordSmith_Fmt	1353	1284	PalmOS Wordsmith document			adWORDPROCESSOR	
iSilo_Fmt	1354	1285	PalmOS iSilo document	application/x-pdb-isilo-ebook	PDB	adWORDPROCESSOR	
SuperMemo_Fmt	1355	1286	PalmOS SuperMemo document		KNO, PDB	adWORDPROCESSOR	
BDicty_Fmt	1356	1287	PalmOS BDicty document		PDB	adWORDPROCESSOR	
PalmOS_Executable_Fmt	1357	1288	PalmOS executable	application/vnd.palm	PRC	adEXECUTABLE	
PalmOS_Library_Fmt	1358	1289	PalmOS dynamic library		PRC	adLIBRARY	
Shanda_Bambook_Fmt	1359	1290	Shanda Bambook eBook	application/x-snb-ebook	SNB	adWORDPROCESSOR	
PMLZ_Fmt	1360	1291	Palm Markup Language (PMLZ) eBook		PMLZ	adWORDPROCESSOR	
Rocket_eBook_Fmt	1361	1292	Rocket eBook	application/x-rocketbook	RB	adWORDPROCESSOR	
iBooks_Author_Fmt	1362	1293	Apple iBooks Author eBook	application/vnd.apple.ibauthor	IBA	adWORDPROCESSOR	
Statistica_Spreadsheet_Fmt	1363	1294	Statsoft Statistica Spreadsheet		STA	adSPREADSHEET	
Statistica_Graph_Fmt	1364	1295	Statsoft Statistica Graph File		STG	adVECTORGRAPHIC	
Statistica_Scrollsheet_Fmt	1365	1296	Statsoft Statistica Scrollsheet		SCR	adSPREADSHEET	
Apple_Newton_Package_Fmt	1366	1297	Apple Newton executable/installer/file		PKG	adEXECUTABLE	
Adobe_Zip_Extension_Fmt	1367	1298	Adobe Zip Format Extension Package (ZXP)	application/vnd.adobe.air-ucf-package+zip	ZXP	adENCAPSULATION	
Uniform_Office_Fmt	1368	1299	Uniform Office Format document		UOF	adWORDPROCESSOR	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Uniform_Office_Text_Fmt	1369	1300	Uniform Office Format word processing document	application/vnd.uof.text	UOF, UOT	adWORDPROCESSOR	
Uniform_Office_Spreadsheet_Fmt	1370	1301	Uniform Office Format spreadsheet	application/vnd.uof.spreadsheet	UOF, UOS	adSPREADSHEET	
Uniform_Office_Presentation_Fmt	1371	1302	Uniform Office Format presentation	application/vnd.uof.presentation	UOF, UOP	adPRESENTATION	
Uniform_Office_Zip_Fmt	1372	1303	Uniform Office Format document, zip format		UOF	adWORDPROCESSOR	
Uniform_Office_Text_Zip_Fmt	1373	1304	Uniform Office Format word processing document, zip format	application/vnd.uof.text+zip	UOF, UOT	adWORDPROCESSOR	
Uniform_Office_Spreadsheet_Zip_Fmt	1374	1305	Uniform Office Format spreadsheet, zip format	application/vnd.uof.spreadsheet+zip	UOF, UOS	adSPREADSHEET	
Uniform_Office_Presentation_Zip_Fmt	1375	1306	Uniform Office Format presentation, zip format	application/vnd.uof.presentation+zip	UOF, UOP	adPRESENTATION	
MacDraft_Fmt	1376	1307	MacDraft drawing		DRW, MDD	adCAD	
RagTime_Fmt	1377	1308	RagTime document		RAG, RTD	adDESKTOPPUBLSH	
MacDraw_Fmt	1378	1309	MacDraw drawing			adVECTORGRAPHIC	
Wingz_Fmt	1379	1310	Wingz spreadsheet		WKZ	adSPREADSHEET	
Claris_Draw_Fmt	1380	1311	Claris Draw document			adVECTORGRAPHIC	
BeagleWorks_Word_Fmt	1381	1312	BeagleWorks (later WordPerfect Works) Word Processor document		BW, WPW	adWORDPROCESSOR	stringsr
BeagleWorks_Database_Fmt	1382	1313	BeagleWorks (later WordPerfect Works) Database document		BW, WPW	adDATABASE	
BeagleWorks_Spreadsheet_Fmt	1383	1314	BeagleWorks (later WordPerfect Works) Spreadsheet document		BW, WPW	adSPREADSHEET	
BeagleWorks_Paint_Fmt	1384	1315	BeagleWorks (later WordPerfect Works) Paint document		BW, WPW	adRASTERIMAGE	
BeagleWorks_Draw_Fmt	1385	1316	BeagleWorks (later WordPerfect Works) Draw document		BW, WPW	adVECTORGRAPHIC	
GreatWorks_Word_Fmt	1386	1317	Symantec GreatWorks			adWORDPROCESSOR	stringsr

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
			Word Processor document				
GreatWorks_Outline_Fmt	1387	1318	Symantec GreatWorks Outline document			adOUTLINE	
GreatWorks_Database_Fmt	1388	1319	Symantec GreatWorks Database document			adDATABASE	
GreatWorks_Spreadsheet_Fmt	1389	1320	Symantec GreatWorks Spreadsheet document			adSPREADSHEET	
GreatWorks_Draw_Fmt	1390	1321	Symantec GreatWorks Draw document			adVECTORGRAPHIC	
GreatWorks_Chart_Fmt	1391	1322	Symantec GreatWorks Chart document			adVECTORGRAPHIC	
MS_Works_3_Mac_WP_Fmt	1392	1323	Microsoft Works for Mac, version 3 and 4, Word Processor document	application/x-msworks	MSW, WPS	adWORDPROCESSOR	
MS_Works_3_Mac_DB_Fmt	1393	1324	Microsoft Works for Mac, version 3 and 4, Database	application/x-msworks	WDB	adDATABASE	
MS_Works_3_Mac_SS_Fmt	1394	1325	Microsoft Works for Mac, version 3 and 4, Spreadsheet	application/x-msworks	WKS	adSPREADSHEET	
MS_Works_3_Mac_Comm_Fmt	1395	1326	Microsoft Works for Mac, version 3 and 4, Communications document	application/x-msworks		adCOMMUNICATION	
MS_Works_3_Mac_Draw_Fmt	1396	1327	Microsoft Works for Mac, version 3 and 4, Draw document	application/x-msworks	MSW	adVECTORGRAPHIC	
SAP_VDS_Fmt	1397	1328	SAP 3d Visual Enterprise VDS document		VDS	adCAD	
ZIPVFS_Fmt	1398	1329	ZIPVFS SQLite compressed read/write database		SQLITE	adDATABASE	
Right_Hemisphere_Material_Fmt	1399	1330	Right Hemisphere Material file		RH, RHM	adCAD	
RH_Thumbnails_Fmt	1400	1331	Right Hemisphere thumbnail collection file		\$RH	adCAD	
Westwood_Studios_Audio_Fmt	1401	1332	Westwood Studios Audio file		AUD	adSOUND	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Shockwave_Stream_Fmt	1402	1333	Shockwave Stream audio-video file		STREAM	adMOVIE	
EGG_Video_Fmt	1403	1334	EGG video file		EGG	adMOVIE	
IRCAM_Fmt	1404	1335	IRCAM audio file		IRCAM	adSOUND	
Sierra_Audio_Fmt	1405	1336	Sierra Entertainment audio file		SOL	adSOUND	
TiVo_Video_Fmt	1406	1337	TiVo video		TY+	adMOVIE	
OptimFROG_Fmt	1407	1338	OptimFROG audio		OFR, OFS	adSOUND	
LPAC_Fmt	1408	1339	Lossless Predictive Audio Compression file		PAC	adSOUND	
RK_Audio_Fmt	1409	1340	RK Audio lossless compressed audio		RKA	adSOUND	
Asylum_Music_Fmt	1410	1341	Asylum Music Format		AMF	adSOUND	
Novastorm_Audio_Fmt	1411	1342	Novastorm Media audio file		SMP	adSOUND	
HHE_Fmt	1412	1343	HHE video		HHE	adMOVIE	
Portable_Voice_Fmt	1413	1344	Portable Voice Format audio		PVF	adSOUND	
CNM_Video_Fmt	1414	1345	Arxel CNM audio-video format		CNM	adMOVIE	
Phantom_Cine_Fmt	1415	1346	Phantom Cine video file		CINE	adMOVIE	
MPEG2_Transport_Stream_Fmt	1416	1347	MPEG-2 Transport Stream video		M2TS	adMOVIE	
Audacity_Project_Fmt	1417	1348	Audacity audio project file	application/x-audacity-project	AUP	adSOUND	
Voltage_VSF_Fmt	1418	1349	Micro Focus Voltage VSF encrypted file		VDF	adENCAPSULATION	
XLIFF_Fmt	1419	1350	XML Localization Interchange File Format (XLIFF)	application/xliff+xml	XLF	adWORDPROCESSOR	
XBRL_Fmt	1420	1351	Extensible Business Reporting Language (XBRL)		XBRL	adWORDPROCESSOR	
AuditXPressX_Fmt	1421	1352	AuditXPressX file		AXPX	adWORDPROCESSOR	
Box_Note_Fmt	1422	1353	Box Note document		BOXNOTE	adWORDPROCESSOR	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
Hikvision_DVR_Fmt	1423	1354	Hikvision DVR video			adMOVIE	
Electronic_Arts_TGV_Fmt	1424	1355	Electronic Arts TGV video		TGV	adMOVIE	
Electronic_Arts_TGQ_Fmt	1425	1356	Electronic Arts TGQ video		TGQ	adMOVIE	
Reaper_Video_Fmt	1426	1357	Reaper Video		FMV	adMOVIE	
Lightweight_Video_Fmt	1427	1358	Lightweight Video Format (LVF)		LVF	adMOVIE	
Liquid_Audio_Fmt	1428	1359	Liquid Audio		LQT	adSOUND	
Extended_Instrument_Fmt	1429	1360	eXtended Instrument generic audio tracker		XI	adSOUND	
MAML_Fmt	1430	1361	Microsoft Assistance Markup Language		AML	adWORDPROCESSOR	
MS_Chat_Character_Fmt	1431	1362	Microsoft Comic Chat Character		AVB	adRASTERIMAGE	
MS_Border_Fmt	1432	1363	Microsoft Office Border images		BDR	adRASTERIMAGE	
MS_Binary_Log_Fmt	1433	1364	Microsoft Binary Log file		BLG	adMISC	
MS_Reader_eBook_Fmt	1434	1365	Microsoft Reader eBook file		LIT	adWORDPROCESSOR	
MS_Reader_Annotations_Fmt	1435	1366	Microsoft Reader annotation file		EBO	adWORDPROCESSOR	
Amazon_KFX_Aux_Fmt	1436	1367	Amazon KFX eBook auxiliary format (2015)		KFX, AZW	adWORDPROCESSOR	
Amazon_KFX_Ion_Fmt	1437	1368	Amazon KFX eBook Ion format (2015)		KFX, AZW, ION	adWORDPROCESSOR	
MS_DPAPI_Fmt	1438	1369	Microsoft Data Protection API (DPAPI) data			adMISC	
MS_Streets_Fmt	1439	1370	Microsoft Streets & Trips map		EST	adGIS	
MS_Fast_Find_Index_Fmt	1440	1371	Microsoft Office Fast Find Index		FFX	adMISC	
MS_Fresh_Paint_Fmt	1441	1372	Microsoft Fresh Paint image		FPX	adRASTERIMAGE	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
MS_Mathematics_Fmt	1442	1373	Microsoft Mathematics worksheet		GCW	adSCIENTIFIC	
MS_Instrument_Definition_Fmt	1443	1374	Microsoft MIDI Instrument Definition File		IDF	adSOUND	
MS_Pocket_Streets_Fmt	1444	1375	Microsoft Pocket Streets map		MPS	adGIS	
Obfuscated_OpenType_Fmt	1445	1376	Obfuscated OpenType font (ODTTF)	application/vnd.ms-package.obfuscated-opentype	ODTTF	adFONT	
Pfaff_PCS_Fmt	1446	1377	Pfaff PCS embroidery image		PCS	adVECTORGRAPHIC	
Janome_JEF_Fmt	1447	1378	Janome JEF embroidery format		JEF	adVECTORGRAPHIC	
Husqvarna_HUS_Fmt	1448	1379	Husqvarna Viking HUS embroidery format		HUS	adVECTORGRAPHIC	
Husqvarna_VIP_Fmt	1449	1380	Husqvarna Viking-Pfaff VIP embroidery format		VIP	adVECTORGRAPHIC	
Brother_PEC_Fmt	1450	1381	Brother PEC embroidery format		PEC	adVECTORGRAPHIC	
Brother_PES_Fmt	1451	1382	Brother PES embroidery format		PES	adVECTORGRAPHIC	
Viking_SHV_Fmt	1452	1383	Viking SHV embroidery format		SHV	adVECTORGRAPHIC	
VP3_Fmt	1453	1384	VP3 embroidery format		VP3	adVECTORGRAPHIC	
SEW_Fmt	1454	1385	SEW embroidery format		SEW	adVECTORGRAPHIC	
Data_Stitch_Tajima_Fmt	1455	1386	Data Stitch Tajima (DST) embroidery image		DST	adVECTORGRAPHIC	
Singer_XXX_Fmt	1456	1387	Singer XXX embroidery image		XXX	adVECTORGRAPHIC	
Bernina_ART_Fmt	1457	1388	Bernina ART embroidery image		ART	adVECTORGRAPHIC	
MS_Prefetch_Fmt	1458	1389	Microsoft Windows Prefetch (uncompressed) file		PF	adMISC	
MS_Prefetch_Compressed_Fmt	1459	1390	Microsoft Windows Prefetch (compressed) file		PF	adMISC	

Format Name	Number	Category	Description	MIME Type	Extension	File Class	Readers
MS_MapPoint_Fmt	1460	1391	Microsoft MapPoint map		PTM	adGIS	
MS_Live_Meeting_Fmt	1461	1392	Microsoft Office Live Meeting Connection		RTC	adSCHEDULE	
MS_Speech_Definitions_Fmt	1462	1393	Microsoft text-to-speech Speech Definitions File		SDF	adMISC	
MS_Speech_Data_Fmt	1463	1394	Microsoft text-to-speech Speech Data File		SPD	adDATABASE	
MS_SQL_CE_Fmt	1464	1395	Microsoft SQL Server Compact (CE) edition database		SDF	adDATABASE	
MS_ICE_Project_Fmt	1465	1396	Microsoft Image Composite Editor (ICE) Project		SPJ	adMISC	
MS_DVR_Fmt	1466	1397	Microsoft Digital Video Recording (DVR-MS)	video/x-ms-dvr	DVR-MS	adMOVIE	
Symbol_Dynamics_EXP_Fmt	1467	1398	Symbol Dynamics EXP document		WXP	adWORDPROCESSOR	stringsr
XNA_Compiled_Fmt	1468	1399	Microsoft XNA Compiled Format		XNB	adENCAPSULATION	
Outlook_Shortcut_Fmt	1469	1400	Microsoft Outlook or Exchange folder shortcut		XNK	adMISC	

¹MHT, EML, and MBX files might return either format 2, 233, or 395, depending on the text in the file. In general, files that contain fields such as **To**, **From**, **Date**, or **Subject** are considered to be email messages; files that contain fields such as **content-type** and **mime-version** are considered to be MHT files; and files that do not contain any of those fields are considered to be text files.

²All CAT file extensions, for example CATDrawing, CATProduct, CATPart, and so on.

Appendix B: Document Readers

This section lists the KeyView document readers that are available to filter, export, and view supported file formats.

- [Key to Document Readers Table](#) 309
- [Document Readers](#) 311

Key to Document Readers Table

The document readers table includes the following information.

Column	Description
Reader	The name of the reader.
Description	A description of the reader.
Filter	Shows whether KeyView can filter text from the main content of the file.
Export	Shows whether KeyView supports export to HTML, XML, and PDF.
View	Shows whether KeyView provides viewing capability.
Extract	Shows whether KeyView can extract sub-files.
Metadata	Shows whether KeyView can extract metadata (properties such as title, author, and subject).
Charset	Shows whether KeyView can detect and extract the character set. Even though a file format might be able to provide character set information, some documents might not contain character set information. Therefore, the document reader would not be able to determine the character set of the document.
H/F	Shows whether KeyView can extract headers and footers.
Associated File Formats	The file formats that are supported by the reader.

Key to Symbols

Symbol	Description
Y	The feature is supported.
N	The feature is not supported.

Key to Symbols, continued

Symbol	Description
P	Partial metadata is extracted from this format. Some non-standard fields are not extracted.
T	Only text is extracted from this format. Formatting information is not extracted.
M	Only metadata (title, subject, author, and so on) is extracted from this format. Text and formatting information are not extracted.

Document Readers

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
ActiveX components	Microsoft Visio (2013)	N	N	Y ¹	N	Y	N	N	MS_Visio_2013_Fmt
ad1sr	AD1	N	N	Y	Y	N	n/a	N	AD1_Fmt
afsr	ASCII Text	Y	Y	Y	N	N	N	N	ASCII_Text_Fmt , GIS_World_File_Fmt , MSDOS_Batch_File_Fmt , SGML_Fmt , Scribe_Fmt , TSV_Fmt , TSV_Fmt , Wiki_Fmt ,
aiffr	Audio Interchange File Format	M	N	N	N	Y	N	N	AIFF_Fmt
asfsr	Advanced Systems Format (1.2)	N	N	N	N	Y	N	N	ASF_Fmt , WMA_Fmt , WMV_Fmt
assr	Applix Spreadsheets (4.2, 4.3, 4.4)	Y	Y	Y	N	N	Y	N	Applix_Spreadsheets_Fmt
awsr	Applix Words (3.11, 4, 4.1, 4.2,	Y	Y	Y	N	N	Y	Y	Applix_Words_Fmt

¹Visio 2013 is supported in Viewing only, with the support of ActiveX components from the Microsoft Visio 2013 Viewer. Image fidelity is supported but other features, such as highlighting, are not.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	4.3, 4.4)								
axsr	Applix Asterix	Y	T	T	N	N	N	N	Applix_Alis_Fmt
b1sr	B1	N	N	Y	Y	N	n/a	N	B1_Fmt
bkfsr	Microsoft Backup File	N	N	Y	Y	N	n/a	N	BKF_Fmt
bmpsr	Windows Bitmap	M	M	N	N	Y	N	N	BMP_Fmt
bzip2sr	Bzip2	N	N	Y	Y	N	n/a	N	BZIP2_Fmt
cabsr	Microsoft Cabinet format (1.3)	N	N	Y	Y	N	n/a	N	CAB_Fmt
cdsr	Convergent Technologies DEF	Y	T	T	N	N	N	N	CT_DEF_Fmt
cebsr ¹	Founder Chinese E-paper Basic (3.2.1)	Y	N	N	N	N	N	N	Founder_CEB_Fmt
chmsr	Microsoft Compiled HTML Help (3)	N	N	Y	Y	N	n/a	N	CHM_Fmt
csvsr	Comma Separated	Y	Y	Y	N	N	N	N	CSV_Fmt

¹This reader is only supported on Windows 32-bit platforms.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	Values								
dbfsr	dBase Database (III+, IV)	Y	Y	Y	N	N	N	N	dBase_Fmt
dbxsr	Microsoft Outlook DBX (5.0, 6.0)	N	N	Y	Y	Y	Y	N	MS_OEDBX_Fmt
dcasr	IBM DCA/RFT (Revisable Form Text) (SC23-0758-1)	Y	Y	Y	N	N	Y	N	DCA_RFT_Fmt
dcmsr	Digital Imaging & Communications in Medicine (DICOM)	M	N	N	N	Y	N	N	Dicom_Fmt
difsr	Data Interchange Format	Y	Y	Y	N	N	N	N	DIF_SpreadSheet_Fmt
dmgsr	Mac Disk Copy Disk Image	N	N	Y	Y	N	n/a	N	DMG_Fmt
dw4sr	DisplayWrite (4)	Y	Y	Y	N	N	Y	N	IBM_Display_Write_Fmt
dxlsr	Domino XML Language ¹	N	N	Y	Y	Y	N	N	Lotus_Domino_DXL_Fmt

¹Supports non-encrypted embedded files only.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
emlsr ¹	Text Mail (MIME) / Microsoft Outlook Express (Windows 6, MacIntosh 5)	Y	T	T	Y	Y	Y	N	SMTP_Fmt
emxsr	Legato EMailXtender Archive	N	N	Y	Y	N	n/a	N	EMX_Fmt
encase2sr	Expert Witness Compression Format (EnCase) (7)	N	N	Y	Y	N	n/a	N	EnCase_Fmt
encasesr	Expert Witness Compression Format (EnCase) (6)	N	N	Y	Y	N	n/a	N	EnCase_Fmt
entsr	Microsoft Entourage Database (2004)	N	N	Y	Y	Y	Y	N	ENT_Fmt
epubsr	Open Publication Structure eBook (2.0, 3.0)	Y	Y	Y	N	Y	Y	N	Epub_Fmt

¹This reader supports both clear signed and encrypted S/MIME. KeyView supports S/MIME for PST, EML, MBX, and MSG files.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
exesr	Executable	N	N	Y	N	N	n/a	N	MS_Executable_Fmt
foliosr	Folio Flat File (3.1)	Y	Y	Y	N	Y	Y	Y	Folio_Flat_Fmt
gdsiisr	GDSII	Y	T	T	N	N	N	N	GDSII_Fmt
gifsr	GIF (87, 89)	M	M	N	N	Y	N	N	GIF_87a_Fmt , GIF_89a_Fmt
gwfssr	GroupWise FileSurf	N	N	Y	Y	Y	N	N	GWFS_Email_Fmt
hl7sr	Health level7 (2.0)	Y	Y	Y	N	Y	Y	N	HI7_Fmt
htmsr	HTML/XHTML (3, 4)	Y	Y	Y	N	Y ¹	Y	N	HTML_Fmt , Netscape_Bookmark_File_Fmt
hwposr	Haansoft Hangul (2002, 2005, 2007, 2010)	Y	Y	Y	Y	Y	Y	N	HWP_Fmt
hwpsr	Haansoft Hangul (97)	Y	Y	Y	N	Y	Y	N	HWP_Fmt
ichatsr	Apple iChat Log (1, AV 2, AV 2.1, AV 3)	Y	Y	Y	N	N	N	N	Apple_iChat_Fmt
icssr	Microsoft Outlook	N	N	Y	Y	Y	Y	N	ICS_Fmt

¹HTML only supports partial metadata extraction

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	iCalendar (1.0, 2.0)								
isosr	ISO	N	N	Y	Y	N	n/a	N	ISO_Fmt
iwss13sr ¹	Apple iWork Numbers ('13, '16, '18, iCloud 2018)	Y	T	T	N	N	Y	N	IWSS13_Fmt
iwsssr	Apple iWork Numbers ('08, '09)	Y	Y	Y	N	Y	Y	N	IWSS_Fmt
iwwp13sr ²	Apple iWork Pages ('13, '16, '18, iCloud 2018)	Y	T	T	N	N	N	N	IWWP13_Fmt
iwwpsr	Apple iWork Pages ('08, '09)	Y	Y	Y	N	Y	Y	N	IWWP_Fmt
jp2000sr	JPEG (2000)	M	M	N	N	Y	N	N	ISO_JPEG2000_JP2_Fmt , ISO_JPEG2000_JPEG2000_JPM_Fmt , ISO_JPEG2000_JPX_Fmt , JPEG_2000_JP2_File_Fmt , JPEG_2000_PGX_Fmt , Motion_JPEG_2000_Fmt
jpgsr	JPEG (JFIF)	M	M	N	N	Y	N	N	JPEG_File_Interchange_Fmt

¹This reader is available only on Windows (32-bit and 64-bit), Linux (32-bit and 64-bit), and Solaris x86-64.

²This reader is available only on Windows (32-bit and 64-bit), Linux (32-bit and 64-bit), and Solaris x86-64.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
jtdsr	JustSystems Ichitaro (8 to 2013, 2018)	Y	Y	Y	N	P	N	Y	ICHITARO_Compr_Fmt , ICHITARO_Fmt
kpagrdr	Applix Presents (4.0, 4.2, 4.3, 4.4)	Y	Y	Y	N	N	N	N	Applix_Graphics_Fmt
kpanirdr	Windows Animated Cursor	N	Y	Y	N	N	N	N	Windows_Animated_Cursor_Fmt
kpbmprdr	Windows Bitmap	N	Y	Y	N	N	N	N	BMP_Fmt
kpCATrdr	CATIA formats (5)	Y	N	N	N	Y	N	N	CATIA_Fmt
kpcdrdr	CorelDRAW ¹ (through 9.0, 10, 11, 12, X3)	N	Y	Y	N	N	N	N	Corel_Draw_Fmt
kpcgmrdr ²	Computer Graphics Metafile	Y	Y	Y	N	N	N	N	CGM_Binary_Fmt , CGM_Character_Fmt , CGM_ClearText_Fmt
kpchtrdr	Microsoft Excel (2-7) and Lotus 1-2-3 Charts (2-5)	N	Y	Y	N	N	N	N	

¹CDR/CDR with TIFF header.

²Files with non-partitioned data are supported.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
kpdcxrdr	DCX Fax System	N	Y	Y	N	N	N	N	DCX_Fmt
kpDWGrdr ¹	AutoCAD Drawing (R13 onwards)	Y	Y	Y	N	Y	Y	N	AutoDesk_DWG_Fmt
kpDXFrdr ²	AutoCAD Drawing Exchange (R13 onwards)	Y	Y	Y	N	Y	Y	N	AutoCAD_DXF_Binary_Fmt , AutoCAD_DXF_Text_Fmt
kpemfrdr	Enhanced Metafile	Y	Y	Y	N	Y	N	N	Enhanced_Metafile_Fmt
kpepsrdr	Encapsulated PostScript (raster) (TIFF header)	N	Y	Y	N	N	N	N	EPSF_Fmt , Preview_EPSF_Fmt
kpGFLrdr	Omni Graffle	Y	N	N	N	Y	Y	N	Omni_Graffle_XML_Fmt
kpgifdr	GIF (87, 89)	N	Y	Y	N	N	N	N	GIF_87a_Fmt , GIF_89a_Fmt
kpiconrdr	Windows Icon Cursor	N	Y	Y	N	N	N	N	Windows_Icon_Fmt

¹The kpODArdr reader can filter, export, and view all versions but is supported only on Windows, Linux, and macOS. The kpDWGrdr reader is used on AIX, FreeBSD, Solaris, and SPARC platforms, but does not support graphics for versions after 2004 or text for versions after 2013.

²The kpODArdr reader can filter, export, and view all versions but is supported only on Windows, Linux, and macOS. The kpDXFrdr reader is used on AIX, FreeBSD, Solaris, and SPARC platforms, but does not support graphics for versions after 2004.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
kpIWPG13rdr ¹	Apple iWork Keynote ('13, '16, '18, iCloud 2018)	Y	T	N	N	N	N	N	IWPG13_Fmt
kpIWPGrdr	Apple iWork Keynote (2, 3, '08, '09)	Y	Y	Y	N	Y	Y	N	IWPG13_Fmt , IWPG_Fmt
kpJBIG2rdr	JBIG2	N	Y	Y	N	N	N	N	JBIG2_Fmt
kpjp2000rdr	JPEG (2000)	N	Y	Y	N	N	N	N	ISO_JPEG2000_JP2_Fmt , ISO_JPEG2000_JPM_Fmt , ISO_JPEG2000_JPX_Fmt , JPEG_2000_JP2_File_Fmt , JPEG_2000_PGX_Fmt , Motion_JPEG_2000_Fmt
kpjpgdr	JPEG (JFIF)	N	Y	Y	N	N	N	N	JPEG_File_Interchange_Fmt
kpmacrdr	MacPaint	N	Y	Y	N	N	N	N	MacPaint_Fmt
kpmsordr	Microsoft Office Drawing	N	Y	Y	N	N	N	N	MS_Office_Drawing_Fmt
kpODArdr	ODA	Y	Y	Y	N	Y	Y	N	AutoCAD_DXF_Binary_Fmt , AutoCAD_DXF_Text_Fmt , AutoDesk_DWG_Fmt
kpodfrdr	OASIS Open	Y	Y	Y	Y ³	Y	Y	N	ODF_Drawing_Fmt , ODF_Drawing_

¹This reader is available only on Windows (32-bit and 64-bit), Linux (32-bit and 64-bit), and Solaris x86-64.

³Supported using the olesr embedded objects reader.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	Document Format (1, 2 ¹)								Template_Fmt , ODF_Presentation_Fmt , ODF_Presentation_Template_Fmt , SO_Drawing_XML_Fmt , SO_Presentation_XML_Fmt
kpONErdr	Microsoft OneNote (2007, 2010, 2013, 2016)	Y	Y	Y	Y	N	Y	N	OneNote_Fmt
kpp40rdr	Microsoft PowerPoint (98)	Y	Y	Y	N	P ²	N	N	PowerPoint_Win_Fmt
kpp95rdr	Microsoft PowerPoint Windows (95)	Y	Y	Y	N	P	Y	N	PowerPoint_95_Fmt
kpp97rdr	Microsoft PowerPoint (97-2004)	Y	Y	Y	N	P	Y	Y ³	PowerPoint_2000_Fmt , PowerPoint_97_Fmt
kppctrdr	Macintosh Raster (2)	N	Y	Y	N	N	N	N	Mac_PICT_Fmt
kppcxrdr	PC PaintBrush (3)	N	Y	Y	N	N	N	N	PC_Paintbrush_Fmt

¹Generated by OpenOffice Impress 2.0, StarOffice 8 Impress, and IBM Lotus Symphony Presentation 3.0.

²Microsoft PowerPoint Windows only

³Microsoft PowerPoint Windows only

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
kppdf2rdr ¹	Adobe PDF (1.1 to 1.7, 2.0)	N	N	Y	N	N	N	N	PDF_Fmt
kppdfdr	Adobe PDF (1.1 to 1.7, 2.0)	N	Y	Y	N	N	N	N	PDF_Fmt
kppicrdr	Lotus Pic	Y	Y	Y	N	N	N	N	Lotus_PIC_Fmt
kppngrdr	Portable Network Graphics	N	Y	Y	N	N	N	N	APNG_Fmt , PNG_Fmt
kpppxrdr	Microsoft PowerPoint Windows XML (2007 onwards)	Y	Y	Y	Y	Y	Y	Y	MS_PPT_2007_Fmt , MS_PPT_Macro_2007_Fmt
kpprerdr	Lotus Freelance Graphics 2 (2)	Y	Y	Y	N	N	N	N	Freelance_OS2_Fmt , Freelance_Win_Fmt
kpprzrdr	Lotus Freelance Graphics (96, 97, 98, R9, 9.8)	Y	Y	Y	N	N	N	N	Freelance_96_Fmt , Freelance_97_Fmt , Freelance_DOS_Fmt
kpsddrdr	StarOffice Impress (3, 4, 5)	Y	T	N	N	N	N	N	SO_Presentation_Fmt
kpsdwrdr	Lotus AMIDraw Graphics	N	Y	Y	N	N	N	N	Ami_Pro_Draw_Fmt , SO_Text_Fmt

¹kppdf2rdr is an alternate graphic-based reader that produces high-fidelity output but does not support other features such as highlighting or text searching.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
kpsgirdr	SGI RGB Image	N	Y	Y	N	N	N	N	SGI_Image_Fmt
kpshwrdr	Corel Presentations (6, 7, 8, 9, 10, 11, 12, X3)	Y	Y	Y	N	N	N	N	Corel_Presentations_Fmt
kpsunrdr	Sun Raster Image	N	Y	Y	N	N	N	N	Sun_Raster_Fmt
kpTGArdr	Truevision Targa (2)	N	Y	Y	N	N	N	N	Targa_Fmt
kptifdr	Tagged Image File (through 6.0 ¹)	N	Y	Y	N	N	N	N	TIFF_Fmt
kpUGrdr	Unigraphics (UG) NX	Y	N	N	N	N	N	N	Unigraphics_NX_Fmt
kpVSD2rdr	Microsoft Visio (4, 5, 2000, 2002, 2003, 2007, 2010 ²)	Y	Y	Y	N	Y	Y	N	MS_Visio_Fmt
kpVSDXrdr	Microsoft Visio	Y	Y	Y	Y	Y	Y	N	MS_Visio_2013_Fmt , MS_Visio_2013_

¹The following compression types are supported: no compression, CCITT Group 3 1-Dimensional Modified Huffman, CCITT Group 3 T4 1-Dimensional, CCITT Group 4 T6, LZW, JPEG (only Gray, RGB and CMYK color space are supported), and PackBits.

²Viewing and Export use the graphic reader, kpVSD2rdr for Microsoft Visio 2003, 2007, and 2010, and vsdr for all earlier versions. Image fidelity in Viewing and Export is therefore only supported for versions 2003 and above. Filter uses the graphic reader kpVSD2rdr for Microsoft Visio 2003, 2007, and 2010, and vsdr for all earlier versions.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	(2013)								Macro_Fmt , MS_Visio_2013_Stencil_Fmt , MS_Visio_2013_Stencil_Macro_Fmt , MS_Visio_2013_Template_Fmt , MS_Visio_2013_Template_Macro_Fmt
kpwg2rdr	WordPerfect Graphics 2 (2, 7)	N	Y	Y	N	N	N	N	WordPerfect_Graphics_Fmt
kpwmfrdr	Windows Metafile (3)	Y ¹	Y	Y	N	N	N	N	Windows_Metafile_Fmt , Windows_Metafile_NoHdr_Fmt
kpwpgrdr	WordPerfect Graphics 1 (1)	N	Y	Y	N	N	N	N	WordPerfect_Graphics_Fmt
kpXFDLrdr	Extensible Forms Description Language	Y	Y	Y	N	Y	Y	N	XFDL_Fmt
kvgz	GZIP (2)	N	N	Y	N	N	n/a	N	GZ_Compress_Fmt
kvgzsr	GZIP (2)	N	N	N	Y	N	n/a	N	GZ_Compress_Fmt
kvhqsr	BinHex	N	N	Y	Y	N	n/a	N	BinHex_Fmt
kvzee	UNIX Compress	N	N	Y	N	N	n/a	N	Compress_Fmt
kvzeesr	UNIX Compress	N	N	N	Y	N	n/a	N	Compress_Fmt
l123sr	Lotus 1-2-3 (96,	Y	Y	Y	N	P	Y	N	Lotus_123_97_Fmt , Lotus_123_Format_

¹Windows Metafiles can contain both raster images (KeyView file class 4) and vector graphics (KeyView file class 5). Filtering is supported only for vector graphics (class 5).

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	97, R9, 9.8)								Fmt, Lotus_123_R9_Fmt
lasr	Lotus AMI Pro and Write Plus (2, 3)	Y	Y	Y	N	p ¹	Y ²	Y	Ami_Pro_Fmt, Ami_Pro_StyleSheet_Fmt
lwpsr	Lotus Word Pro and SmartMaster (96, 97, R9)	Y	Y	Y	N	p ³	N	Y ⁴	Lotus_Word_Pro_96_Fmt, Lotus_Word_Pro_97_Fmt
lzhsr	Microsoft Compressed Folder	N	N	N	Y	N	n/a	N	LZH_Fmt
macbinsr	MacBinary	N	N	Y	Y	N	n/a	N	MacBinary_Fmt
mbsr	Microsoft Word Macintosh (4, 5, 6, 98)	Y	Y	Y	N	Y	N	Y	MS_Word_Mac_4_Fmt, MS_Word_Mac_Fmt
mbxsr ⁵	Text Mail	Y ⁷	N	T	Y	Y	Y	N	MIME_Fmt

¹Lotus AMI Pro only

²Lotus AMI Pro only

³Lotus Word Pro only

⁴Lotus Word Pro only

⁵This reader supports both clear signed and encrypted S/MIME. KeyView supports S/MIME for PST, EML, MBX, and MSG files.

⁷Text Mail only

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	(MIME), Microsoft Outlook Express (Windows 6, MacIntosh 5), Mailbox ¹ (Thunderbird 1.0, Eudora 6.2)								
MCI	Microsoft Media Control Interface	N	N	Y	N	N	N	N	AIFF_Fmt , AU_Audio_Fmt , ISO_QuickTime_Fmt , MIDI_Audio_Fmt , MPEG_Audio_Fmt , MS_Video_Fmt , MS_WAVE_Audio_Fmt , Mobile_QuickTime_Fmt , QuickTime_Fmt
mdbsr	Microsoft Access (95 onwards)	Y	T	T	N	N	Y ²	N	MS_Access_2000_Fmt , MS_Access_2007_Fmt , MS_Access_95_Fmt , MS_Access_97_Fmt , MS_Access_Fmt
mhtsr	MIME HTML	Y	Y	Y	N	Y	Y	N	MHT_Fmt
mifsr	Adobe FrameMaker Interchange Format (5, 5.5, 6, 7)	Y	Y	Y	N	N	Y	N	Maker_Interchange_Fmt
misr	Microsoft Word	Y	Y	Y	N	N	N	Y	MS_Word_Win_Fmt

¹KeyView supports MBX files created by Eudora Email and Mozilla Thunderbird. MBX files created by other common mail applications are typically filtered, converted, and displayed.

²Charset is not supported for Microsoft Access 95 or 97.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	Windows (1.0, 2.0)								
mp3sr	MPEG-1 Audio layer3 (ID3 v1 and v2)	M	M	Y	N	Y	N	N	MPEG_Audio_Fmt
mpeg4sr	MPEG video	M	N	N	N	Y	N	N	Adobe_Flash_Audio_Book_Fmt , Adobe_Flash_Audio_Fmt , Adobe_Flash_Protected_Video_Fmt , Adobe_Flash_Video_Fmt , Audible_Audiobook_Fmt , ISO_3GPP2_Fmt , ISO_3GPP_Fmt , ISO_IEC_MPEG_4_Fmt , KDDI_Video_Fmt , MPEG4_AVC_Fmt , MPEG4_M4A_Fmt , MPEG4_M4B_Fmt , MPEG4_M4P_Fmt , MPEG4_M4V_Fmt , MPEG4_Sony_PSP_Fmt , MPEG_21_Fmt , NTT_MPEG4_Fmt , Nero_MPEG4_Audio_Fmt , QuickTime_Fmt , Sony_XAVC_Fmt
mppsr	Microsoft Project (2000 onwards)	Y	Y	Y	Y	Y	Y	N	MS_Project_2000_Fmt , MS_Project_2007_Fmt , MS_Project_41_Fmt , MS_Project_4_Fmt , MS_Project_98_Fmt
msgsr ¹	Microsoft	Y ²	T ³	Y ⁴	Y	Y	Y ⁵	N	EMCMF_Fmt , MS_Outlook_Fmt

¹This reader supports both clear signed and encrypted S/MIME. KeyView supports S/MIME for PST, EML, MBX, and MSG files.

²Except Documentum EMCMF

³Except Documentum EMCMF

⁴For Outlook this is Text only

⁵Returns "Unicode" character set for Outlook version 2003 and up, and "Unknown" character set for previous versions.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	Outlook (97 onwards), Documentum EMCMP								
mspubsr	Microsoft Publisher (98 to 2016)	Y	T	T	Y	Y	Y	N	MS_Publisher_98_Fmt , MS_Publisher_Fmt
msw6sr	Microsoft Works (6, 2000)	Y	Y	Y	N	N	N	Y	MS_Works_Win_WP_Fmt
mwsr	Microsoft Works (1, 2, 3, 4)	Y	Y	Y	N	N	N	Y	MS_Works_Win_WP_Fmt
multiarcsr ¹	Compressed formats	N	N	Y ²	Y	N	n/a	N	ARJ_Fmt , RAR5_Fmt , XZ_Fmt
mw6sr	Microsoft Word Windows (6, 7, 8, 95)	Y	Y	Y	N	Y	Y	Y	MS_Word_95_Fmt
mw8sr	Microsoft Word (97-2004)	Y	Y	Y	Y ³	Y	Y	Y ⁴	MS_Word_2000_Fmt , MS_Word_97_Fmt

¹Zip is supported with the multiarcsr reader on some platforms for Extract.

²7-zip and SUN PEX archives only

³Supported using the embedded objects reader olesr.

⁴Microsoft Word for Windows only

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
mwsr	Microsoft Word PC (4-6) and Windows Write (1-3)	Y	Y	Y	N	N	Y ¹	Y ²	MS_Windows_Write_Fmt , MS_Word_PC_Driver_Fmt , MS_Word_PC_Fmt , MS_Word_PC_Glossary_Fmt , MS_Word_PC_Misc_Fmt , MS_Word_PC_StyleSheet_Fmt
mwssr	Microsoft Works Spreadsheet (2, 3, 4)	Y	Y	Y	N	N	Y	N	MS_Works_DOS_SS_Fmt , MS_Works_Mac_SS_Fmt , MS_Works_Win_SS_Fmt
mwxsr	Microsoft Word XML (2007 onwards)	Y	Y	Y	Y	Y	Y	Y	MS_Word_2007_Flat_XML_Fmt , MS_Word_2007_Fmt , MS_Word_Macro_2007_Fmt
nnsr	NBI Net Archive	Y	T	T	N	N	N	N	NBI_Net_Archive_Fmt
nsfsr	Lotus Notes database (4, 5, 6.0, 6.5, 7.0, 8.0)	N	N	Y	Y	Y	N	N	Lotus_Notes_NSF_Fmt
oa2sr	Fujitsu Oasys (7)	Y	Y	Y	N	P	N	N	Oasys_Fmt
odfssr	OASIS Open Document Format (1, 2 ³)	Y	Y	Y	Y ⁴	Y	Y	N	ODF_Spreadsheet_Fmt , ODF_Spreadsheet_Template_Fmt

¹Microsoft Windows Write only

²Microsoft Word PC only

³Generated by OpenOffice Calc 2.0, StarOffice 8 Calc, and IBM Lotus Symphony Spreadsheet 3.0.

⁴Supported using the embedded objects reader olesr.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
odfwpsr	OASIS Open Document Format (1, 2 ¹)	Y	Y	Y	Y ²	Y	Y	Y	ODF_Text_Fmt , ODF_Text_Template_Fmt , SO_Text_XML_Fmt
olesr	Windows Scrap File	N	N	N	Y	N	n/a	N	OLE_Fmt , Scrap_Fmt , Windows_Installer_Fmt
olmsr	Microsoft Outlook for Macintosh (2011)	N	N	Y	Y	N	Y	N	MS_OutlookOLM_Fmt
onealtsr	Microsoft OneNote Alternate Format (2007 onwards)	Y	T	T	Y	N	N	N	OneNote_Alternate_Fmt
onmsr	Legato Extender	N	N	Y	Y	Y	N	N	Legato_Extender_ONM_Fmt
oo3sr	Omni Outliner (v3, OPML, OOutline)	Y	Y	Y	N	N	Y	N	OO3_Fmt , OOUTLINE_Fmt , OPML_Fmt
pbixsr	Microsoft Power BI (1.11)	Y	T	T	N	N	Y	N	MS_Power_BI_Fmt
pdf2sr	Adobe PDF (1.1 to 1.7, 2.0)	N	Y	N	N	N	N	N	PDF_Fmt

¹Generated by OpenOffice Writer 2.0, StarOffice 8 Writer, and IBM Lotus Symphony Documents 3.0.

²Supported using the embedded objects reader olesr.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
pdfsr	Adobe PDF (1.1 to 1.7, 2.0)	Y	Y	N	Y ¹	Y	Y	N	PDF_Fmt , Portfolio_PDF_Fmt
pdfsr ²	Microsoft Outlook Offline Storage File (97 onwards)	N	N	Y	Y	Y	Y	N	MS_OutlookOST_Fmt
pfilesr	pFiles	Y ³	T ⁴	T ⁵	N	Y	N	N	RMS_Protected_Fmt
pngsr	Portable Network Graphics	M	M	N	N	Y	N	N	PNG_Fmt
psdsr	Adobe Photoshop	N	N	N	N	Y ⁶	N	N	PSD_Fmt
pstnsr	Microsoft Outlook Personal Folder ⁷ (97 onwards)	N	N	Y	Y	Y	Y	N	MS_OutlookPST_Fmt

¹Includes support for extraction of subfiles from PDF Portfolio documents.

²The reader pdfsr is available only on Windows and Linux.

³KeyView filters only the internal redirection text. The underlying document text is not accessible without the decryption key.

⁴KeyView filters only the internal redirection text. The underlying document text is not accessible without the decryption key.

⁵KeyView filters only the internal redirection text. The underlying document text is not accessible without the decryption key.

⁶Only XMP metadata is extracted for this format.

⁷KeyView provides several readers capable of processing PST files. The pstsr reader uses the Microsoft Messaging Application Programming Interface (MAPI), works only on Windows, and requires that you have Microsoft Outlook installed. The pstxsr reader is available for Windows (32-bit and 64-bit) and Linux (64-bit only) and does not require Microsoft Outlook. The pstnsr reader is an alternative reader that does not require Microsoft Outlook, for all platforms not supported by pstxsr.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
pstsr ¹	Microsoft Outlook Personal Folder ² (97 onwards)	N	N	Y	Y	Y	N	N	MS_OutlookPST_Fmt
pstxsr	Microsoft Outlook Personal Folder ³ (97 onwards)	N	N	Y	Y	Y	Y	N	MS_OutlookPST_Fmt
pwsr	Primeword	Y	T	T	N	N	N	N	PRIMEWORD_Fmt
qpssr	Corel Quattro Pro (5, 6, 7, 8)	Y	Y	Y	N	P	Y	N	Quattro_Pro_Win_Fmt
qpwsr	Corel Quattro Pro (X4)	Y	N	Y	N	P	Y	N	QPW_Fmt
rarsr	RAR archive (2.0 through 3.5)	N	N	N	Y	N	n/a	N	RAR_Fmt
riffr	Microsoft Wave Sound	M	N	N	N	Y	N	N	MS_WAVE_Audio_Fmt

¹This reader supports both clear signed and encrypted S/MIME. KeyView supports S/MIME for PST, EML, MBX, and MSG files.

²KeyView provides several readers capable of processing PST files. The pstsr reader uses the Microsoft Messaging Application Programming Interface (MAPI), works only on Windows, and requires that you have Microsoft Outlook installed. The pstxsr reader is available for Windows (32-bit and 64-bit) and Linux (64-bit only) and does not require Microsoft Outlook. The pstnsr reader is an alternative reader that does not require Microsoft Outlook, for all platforms not supported by pstxsr.

³KeyView provides several readers capable of processing PST files. The pstsr reader uses the Microsoft Messaging Application Programming Interface (MAPI), works only on Windows, and requires that you have Microsoft Outlook installed. The pstxsr reader is available for Windows (32-bit and 64-bit) and Linux (64-bit only) and does not require Microsoft Outlook. The pstnsr reader is an alternative reader that does not require Microsoft Outlook, for all platforms not supported by pstxsr.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
rtfsr	Rich Text Format (1 through 1.7)	Y	Y	Y	N	P	Y	Y	MS_Pocket_Word_Fmt , MS_RTF_Fmt
skypesr	Skype Log (3)	Y	Y	Y	N	N	N	N	Skype_Fmt
sosr	OpenOffice, LibreOffice(1-5), StarOffice (6-9)	Y	T	T	N	Y	Y	N	SO_Spreadsheet_XML_Fmt
starcsr	StarOffice Calc (3, 4, 5)	Y	T	T	N	N	N	N	SO_Spreadsheet_Fmt
starwsr	StarOffice Writer (3, 4, 5)	Y	T	T	N	N	N	N	SO_Text_Fmt
stringsr	Generic 'strings' reader	Y	T	T	N	N	N	N	BeagleWorks_Word_Fmt , CEOwrite_Fmt , CPT_Comm_Fmt , CWK_Fmt , DG_CDS_Fmt , DSA101_Fmt , Data_Point_VistaWord_Fmt , Enable_WP_Fmt , GreatWorks_Word_Fmt , HP_Word_PC_Fmt , IBM_DCF_Script_Fmt , IBM_Writing_Assistant_Fmt , Lotus_Notes_CDF_Fmt , Lyrix_Fmt , MASS_11_Fmt , MS_Works_DOS_WP_Fmt , MS_Works_Mac_WP_Fmt , MacWrite_Fmt , MacWrite_II_Fmt , Multimate_Adv_Fmt , Multimate_Adv_Fnote_Fmt , Multimate_Adv_II_Fmt , Multimate_Adv_II_Fnote_Fmt , Multimate_Fmt , Multimate_Fnote_Fmt , Navy_DIF_Fmt , ODA_Q1_11_Fmt , ODA_Q1_12_Fmt , Office_Writer_Fmt , Psion_TextEd_Fmt , Psion_Word_3_Fmt , Psion_Word_

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
									Fmt, Q_A_DOS_Fmt, Q_A_Win_Fmt, Quadratron_Q_One_v1_Fmt, Quadratron_Q_One_v2_Fmt, Quickword_Fmt, SAMNA_Word_IV_Fmt, Symbol_Dynamics_EXP_Fmt, Targon_Word_Fmt, Uniplex_WP_Fmt, Volkswriter_Fmt, WANG_WITA_Fmt, WANG_WPS_Comm_Fmt, WPS_PLUS_Fmt, WordERA_Fmt, WordMARC_Fmt, WordPerfect_Fmt, WordStar_2000_Fmt, WordStar_Fmt, WordStar_for_Windows_Fmt, Word_Connection_Fmt, WriteNow_Fmt, Xerox_860_Comm_Fmt, Xerox_Writer_Fmt
swfsr	Macromedia Flash (through 8.0)	Y	Y	Y	N	N	Y ¹	N	Macromedia_Flash_Fmt
swsr	SmartWare II Word Processor	Y	T	T	N	N	N	N	SmartWare_II_WP_Fmt
tarsr	Tape Archive	N	N	Y	Y	N	n/a	N	TAR_Fmt
tifsr	Tagged Image File (through 6.0 ²)	M	M	N	N	Y	N	N	TIFF_Fmt
tnfsr	Transport	N	N	Y	Y	Y	Y	N	TNEF_Fmt

¹The character set cannot be determined for versions 5.x and lower.

²The following compression types are supported: no compression, CCITT Group 3 1-Dimensional Modified Huffman, CCITT Group 3 T4 1-Dimensional, CCITT Group 4 T6, LZW, JPEG (only Gray, RGB and CMYK color space are supported), and PackBits.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
	Neutral Encapsulation Format								
unihtmsr	Unicode HTML	Y	Y	Y	N	Y	Y	N	Unicode_HTML_Fmt
unistr	Unicode Text (3, 4)	Y	Y	Y	N	N	Y	N	Unicode_Fmt
unzip	PKZIP/Zip Compression	N	N	Y ¹	Y	N	n/a	N	Executable_JAR_Fmt , KMZ_Fmt , ODF_Formula_Fmt , ODF_Formula_Template_Fmt , PKZIP_Fmt , Tableau_Packaged_Data_Source_Fmt , Tableau_Packaged_Workbook_Fmt
uudsr	UUEncoding (all versions)	N	N	Y	Y	N	n/a	N	UUEncoded_Fmt
vcfsr	Microsoft Outlook vCard Contact (2.1, 3.0, 4.0)	Y	Y	T	N	Y	N	N	VCF_Fmt
vsdsr	Microsoft Visio (4, 5, 2000, 2002, 2003, 2007, 2010 ²)	Y	Y	Y	Y ³	Y	Y	N	MS_Visio_Fmt

¹PKZIP, WinZip, and Java Archive only

²Viewing and Export use the graphic reader, kpVSD2rdr for Microsoft Visio 2003, 2007, and 2010, and vsdsr for all earlier versions. Image fidelity in Viewing and Export is therefore only supported for versions 2003 and above. Filter uses the graphic reader kpVSD2rdr for Microsoft Visio 2003, 2007, and 2010, and vsdsr for all earlier versions.

³Extraction of embedded OLE objects is supported for Filter on Windows platforms only.

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
wkssr	Lotus 1-2-3 (2, 3, 4, 5)	Y	Y	Y	N	N	Y	N	Lotus_123_Worksheet_Fmt
wosr	Corel WordPerfect Windows (5, 5.1)	Y	Y	Y	N	P	Y	Y	WordPerfect_5_Fmt
wp6sr	Corel WordPerfect (6 onwards)	Y	Y	Y	N	P	Y	N	WordPerfect_6_Fmt
wpmsr	Corel WordPerfect Macintosh (1.02, 2, 2.1, 2.2, 3, 3.1)	Y	Y	Y	N	N	Y	N	WordPerfect_Mac_Fmt
xlsbsr	Microsoft Excel Binary Format (2007 onwards)	Y	Y	Y	N	Y	N	N	MS_Excel_Binary_2007_Fmt
xlssr	Microsoft Excel (2.2 to 2004)	Y	Y	Y	Y ¹	Y	Y	Y ²	Excel_2000_Fmt , Excel_95_Fmt , Excel_97_Fmt , Excel_Chart_Fmt , Excel_Fmt , Excel_Macro_Fmt
xlsxsr	Microsoft Excel Windows XML (2007 onwards)	Y	Y	Y	Y	Y	Y	Y	MS_Excel_2007_Fmt , MS_Excel_Macro_2007_Fmt

¹Supported using the embedded objects reader olesr.

²Microsoft Excel for Windows only

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
xmlsr	XML	Y	T	T	N	Y	Y	N	AMF_Fmt , Adobe_XML_Data_Package_Fmt , Atom_Syndication_Fmt , CDXML_Fmt , Chemical_Markup_Language_Fmt , Collada_DAE_Fmt , ESzigno_Fmt , FictionBook_Fmt , Grasshopper_GHX_Fmt , JNLP_Fmt , JavaView_JVX_Fmt , KML_Fmt , MARC_XML_Fmt , METS_Fmt , MODS_Fmt , MS_Excel_XML_Fmt , MS_Management_Pack_MPX_Fmt , MS_Visio_XML_Fmt , MS_Word_XML_Fmt , MXML_Fmt , Metalink_Fmt , Mozilla_XUL_Fmt , MusicXML_Fmt , Open_Diagnostic_Data_Exchange_Fmt , Open_eBook_Fmt , PDF_XML_Forms_Data_Fmt , PLS_Fmt , RDF_XML_Fmt , RSS_Fmt , Really_Simple_Discovery_Fmt , SBML_Fmt , SMIL_Fmt , SPARQL_Results_Fmt , SRGS_Fmt , SRU_Fmt , SSML_Fmt , SVG_Fmt , SyncML_Fmt , TEI_Fmt , Tableau_Data_Source_Fmt , Tableau_Map_Source_Fmt , Tableau_Preferences_Fmt , Tableau_Workbook_Fmt , VTK_XML_Fmt , VoiceXML_Fmt , WML_Fmt , Windows_Audio_Playlist_Fmt , XAML_Browser_Application_Fmt , XDF_Fmt , XML_Fmt , XML_Shareable_Playlist_Fmt , XSLT_Fmt , YIN_Fmt
xpssr	XML Paper Specification	Y	T	T	N	N	N	N	MS_XPS_Fmt
xywsr	XyWrite (4.12)	Y	Y	Y	N	N	N	N	XyWrite_Fmt

Reader	Description	Filter	Export	View	Extract	Metadata	Charset	H/F	Associated File Formats
yimsr ¹	Yahoo! Instant Messenger	Y	Y	Y	N	N	N	N	YIM_Fmt
z7zsr	7-Zip (4.57)	N	N	Y	Y	N	n/a	N	Z7Z_Fmt

¹To successfully use this reader, you must set the KV_YAHOO_ID environment variable to the Yahoo user ID. You can optionally set the KV_OTHER_YAHOO_ID environment variable to the other Yahoo user ID. If you do not set it, "Other" is used by default. If you enter incorrect values for the environment variables, erroneous data is generated.

Appendix C: Character Sets

This section provides information on the handling of character sets in the KeyView suite of products, which includes KeyView Filter SDK, KeyView Export SDK, and KeyView Viewing SDK.

- [Multibyte and Bidirectional Support](#) 338
- [Coded Character Sets](#) 346

Multibyte and Bidirectional Support

The KeyView SDKs can process files that contain multibyte characters. A multibyte character encoding represents a single character with consecutive bytes. KeyView can also process text from files that contain bidirectional text. Bidirectional text contains both Latin-based text which is read from left to right, and text that is read from right to left (Hebrew and Arabic).

The following table indicates which character encodings are supported by KeyView for each format.

Multibyte and bidirectional support

Format	Single-byte	Multibyte	Bidirectional
Archive			
7-Zip (7Z)	n/a	n/a	n/a
AD1 Evidence file	n/a	n/a	n/a
ADJ	n/a	n/a	n/a
B1	n/a	n/a	n/a
BinHex (Hqx)	n/a	n/a	n/a
Bzip2 (BZ2)	n/a	n/a	n/a
EnCase – Expert Witness Compression Format (E01)	n/a	n/a	n/a
GZIP (GZ)	n/a	n/a	n/a
ISO (ISO)	n/a	n/a	n/a
Java Archive (JAR)	n/a	n/a	n/a
Legato EMailXtender Archive (EMX)	n/a	n/a	n/a
MacBinary (BIN)	n/a	n/a	n/a
Mac Disk Copy Disk Image (DMG)	n/a	n/a	n/a
Microsoft Backup File (BKF)	n/a	n/a	n/a

Multibyte and bidirectional support, continued

Format	Single-byte	Multibyte	Bidirectional
Microsoft Cabinet format (CAB)	n/a	n/a	n/a
Microsoft Compiled HTML Help (CHM)	n/a	n/a	n/a
Microsoft Compressed Folder (LZH)	n/a	n/a	n/a
PKZip (ZIP)	n/a	n/a	n/a
Microsoft Outlook DBX (DBX)	Y	Y	Y
Microsoft Outlook Offline Storage File (OST)	Y	Y	Y
RAR Archive (RAR)	n/a	n/a	n/a
Tape Archive (TAR)	n/a	n/a	n/a
UNIX Compress (Z)	n/a	n/a	n/a
UUEncoding (UUE)	n/a	n/a	n/a
Windows Scrap File (SHS)	n/a	n/a	n/a
WinZip (ZIP)	n/a	n/a	n/a
Binary			
Executable (EXE)	n/a	n/a	n/a
Link Library (DLL)	n/a	n/a	n/a
Computer-aided Design			
AutoCAD Drawing (DWG)	Y	Y	Y
AutoCAD Drawing Exchange (DXF)	Y	Y	Y
CATIA formats (CAT)	Y	N	N
Microsoft Visio (VSD)	Y	Y	Y
Database			
dBase Database	Y	N	N
Microsoft Access (MDB)	Y	Y	N
Microsoft Project (MPP)	Y	Y	N
Desktop Publishing			
Microsoft Publisher	N	Y	N

Multibyte and bidirectional support, continued

Format	Single-byte	Multibyte	Bidirectional
Display			
Adobe Portable Document Format (PDF) (basic reader)	Y	Y ¹	Y
Adobe Portable Document Format (PDF) (graphic-based reader)	Y	Y ¹	Y
Graphics			
Computer Graphics Metafile (CGM)	Y	N	N
Corel DRAW (CDR)	n/a	n/a	n/a
DCX Fax System (DCX)	Y	N	N
DICOM – Digital Imaging and Communications in Medicine (DCM)	n/a	n/a	n/a
Encapsulated PostScript (EPS)	Y	N	N
Enhanced Metafile (EMF)	Y	Y	N
Graphic Interchange Format (GIF)	n/a	n/a	n/a
JBIG2	n/a	n/a	n/a
JPEG	n/a	n/a	n/a
JPEG 2000	n/a	n/a	n/a
Lotus AMIDraw Graphics (SDW)	n/a	n/a	n/a
Lotus Pic (PIC)	n/a	n/a	n/a
Macintosh Raster (PICT/PCT)	n/a	n/a	n/a
MacPaint (PNTG)	n/a	n/a	n/a
Microsoft Office Drawing (MSO)	n/a	n/a	n/a

¹Multibyte PDFs are supported, provided the PDF document is created by using either Character ID-keyed (CID) fonts, predefined CJK CMap files, or ToUnicode font encodings, and does not contain embedded fonts. See the Adobe website and the Adobe Acrobat documentation for more information. Any multibyte characters that are not supported are displayed using the replacement character. By default, the replacement character is a question mark (?).

To determine the type of font encodings that are used in a PDF, open the PDF in Adobe Acrobat, and select File > Document Info > Fonts. If the Encoding column lists Custom or Embedded encodings, you might encounter problems converting the PDF.

Multibyte and bidirectional support, continued

Format	Single-byte	Multibyte	Bidirectional
Omni Graffiti (GRAFFLE)	Y	N	N
PC PaintBrush (PCX)	n/a	n/a	n/a
Portable Network Graphics (PNG)	n/a	n/a	n/a
SGI RGB Image (RGB)	n/a	n/a	n/a
Sun Raster Image (RS)	n/a	n/a	n/a
Tagged Image File (TIFF)	Y	N	N
Truevision Targa (TGA)	n/a	n/a	n/a
Windows Animated Cursor (ANI)	n/a	n/a	n/a
Windows Bitmap (BMP)	n/a	n/a	n/a
Windows Icon Cursor (ICO)	n/a	n/a	n/a
Windows Metafile (WMF)	Y	Y	N
WordPerfect Graphics 1 (WPG)	Y	N	N
WordPerfect Graphics 2 (WPG)	Y	N	N
Mail			
Documentum EMC MF Format	Y	Y	Y
Domino XML Language (DXL)	Y	Y	N
GroupWise FileSurf	Y	N	N
Legato Extender (ONM)	Y	Y	N
Lotus Notes database (NSF)	Y	Y	Y
Mailbox (MBX)	Y	Y	Y
Microsoft Entourage Database	Y	Y	Y
Microsoft Outlook (MSG)	Y	Y	Y
Microsoft Outlook Express (EML)	Y	Y	Y
Microsoft Outlook iCalendar	Y	Y	Y
Microsoft Outlook for Macintosh	Y	Y	Y
Microsoft Outlook Offline Storage File	Y	Y	Y
Microsoft Outlook Personal File Folders (PST)	Y	Y	Y

Multibyte and bidirectional support, continued

Format	Single-byte	Multibyte	Bidirectional
Microsoft Outlook vCard Contact			
Text Mail (MIME)	Y	Y	Y
Transport Neutral Encapsulation Format	Y	Y	Y
Multimedia			
Advanced Systems Format (ASF)	n/a	n/a	n/a
Audio Interchange File Format (AIFF)	n/a	n/a	n/a
Microsoft Wave Sound (WAV)	n/a	n/a	n/a
MIDI (MID)	n/a	n/a	n/a
MPEG 1 Audio Layer 3 (MP3)	n/a	n/a	n/a
MPEG 1 Video (MPG)	n/a	n/a	n/a
MPEG 2 Audio (MPEGA)	n/a	n/a	n/a
MPEG 4 Audio (MP4)	n/a	n/a	n/a
NeXT/Sun Audio (AU)	n/a	n/a	n/a
QuickTime Movie (QT/MOV)	n/a	n/a	n/a
Windows Video (AVI)	n/a	n/a	n/a
Presentations			
Apple iWork Keynote (GZ)	Y	Y	N
Applix Presents (AG)	character set 1252 only	N	N
Corel Presentations (SHW)	character set 1252 only	N	N
Extensible Forms Description Language (XFD)	Y	Y	N
Lotus Freelance Graphics 2 (PRE)	character set 850 only	N	N
Lotus Freelance Graphics (PRZ)	Y	Japanese, Simple Chinese, Traditional Chinese, Thai only	N
Macromedia Flash (SWF)	Y	Y	N

Multibyte and bidirectional support, continued

Format	Single-byte	Multibyte	Bidirectional
Microsoft OneNote	Y	Y	N
Microsoft PowerPoint PC (PPT)	character set 1252 only	Traditional Chinese only	N
Microsoft PowerPoint Windows (PPT)	Y	Japanese, Simple Chinese, Traditional Chinese, Korean only	Hebrew only
Microsoft PowerPoint Macintosh (PPT)	Y	N	N
Microsoft PowerPoint Windows XML 2007 and 2010 (PPTX)	Y	Y	Y
OASIS Open Document (ODP)	Y	Y	N
OpenOffice Impress (ODP)	Y	Y	N
StarOffice Impress (ODP)	Y	Y	N
Spreadsheets			
Apple iWork Numbers (GZ)	Y	Y	N
Applix Spreadsheets (AS)	character set 1252 only	N	N
Comma Separated Values (CSV)	character set 1252 only	N	N
Corel Quattro Pro (QPW/WB3)	Y	N	N
Data Interchange Format (DIF)	Y	Y	Y ¹
Lotus 1-2-3 (123)	Y	Y	Y
Lotus 1-2-3 (WK4)	Y	Y	N
Lotus 123 Charts (123)	Y	Y	N
Microsoft Excel Charts (XLS)	Y	Y	N
Microsoft Excel Macintosh (XLS)	Y	N	N
Microsoft Excel Windows (XLS)	Y	Y	Y ²
Microsoft Excel Windows XML 2007 (XLSX)	Y	Y	N
Microsoft Office Excel Binary Format (XLSB)	Y	Y	N
Microsoft Works Spreadsheet	Y	N	N

Multibyte and bidirectional support, continued

Format	Single-byte	Multibyte	Bidirectional
(S30/S40)			
OASIS Open Document (ODS)	Y	Y	N
OpenOffice Calc (ODS)	Y	Y	N
StarOffice Calc (ODS)	Y	Y	N
Text and Markup			
ANSI (TXT)	Y	Y	Y ²
ASCII (TXT)	Y	Y	Y ²
HTML (HTM)	Y	Y	Y ² , ²
Microsoft Excel Windows XML 2003	Y	Y	Y
Microsoft Word for Windows XML 2003	Y	Y	Y
Microsoft Visio XML 2003	Y	Y	Y
Rich Text Format (RTF)	Y	Y	Y ³
Unicode HTML	Y	Y	Y ^{2,3}
Unicode Text (TXT)	Y	Y	Y ²
XHTML	Y	Y	Y ³
XML	Y	Y	Y
Word Processing			
Adobe Maker Interchange Format (MIF)	character set 1252 only	N	N
Apple iChat Log (ICHAT)	Y	Y	N
Apple iWork Pages (GZ)	Y	Y	N
Applix Words (AW)	character set 1252 only	N	N
DisplayWrite (IP)	character set 500, 1026 only	N	N
Folio Flat File (FFF)	character set 1252 only	N	N
Founder Chinese E-paper Basic (CEB)	Y	Y	N

Multibyte and bidirectional support, continued

Format	Single-byte	Multibyte	Bidirectional
Fujitsu Oasys (OA2)	Y	Y	N
Hangul (HWP)	Y	Y	N
Health level7 (HL7)	Y	Y	Y
IBM DCA/RTF (DC)	character sets 500, 1026 only	N	N
JustSystems Ichitaro (JTD)	Y	Y	N
Lotus AMI Pro (SAM)	Y	Simple Chinese, Traditional Chinese, Japanese, Thai only	Y
Lotus AMI Professional Write Plus (AMI)	Y	Simple Chinese, Traditional Chinese, Japanese, Thai only	N
Lotus Word Pro (LWP)	Y	Y	Y ³
Lotus SmartMaster (MWP)	Y	Y	N
Microsoft Word PC (DOC)	character set 1252 only	N	N
Microsoft Word Windows V1-2 (DOC)	Y	N	N
Microsoft Word Windows V6, 7, 8, 95 (DOC)	Y	Y	Hebrew only ³
Microsoft Word Windows V97 through 2003 (DOC)	Y	Y	Y ³
Microsoft Word Windows XML 2007 and 2010 (DOCX)	Y	Y	Y ³
Microsoft Word Macintosh (DOC)	Y	N	Y ³
Microsoft Works (WPS)	Y	Japanese only	N
Microsoft Write (WRI)	Y	Japanese only	N
OASIS Open Document (ODT)	Y	Y	N
Omni Outliner (OO3)	Y	Y	N
OpenOffice Writer (ODT)	Y	Y	N
Open Publication Structure eBook (EPUB)	Y	Y	Y
StarOffice Writer (ODT)	Y	Y	N

Multibyte and bidirectional support, continued

Format	Single-byte	Multibyte	Bidirectional
Skype Log (DBB)	Y	Y (null-terminated charsets)	N
WordPad (RTF)	Y	Y	Y
WordPerfect Linux (WPS)	Y	N	N
WordPerfect Macintosh (WPS)	Y	N	N
WordPerfect Windows (WO)	Y	N	N
XML Paper Specification (XPS)	Y	Y	N
XYWrite Windows (XY4)	character set 1252 only	N	N
Yahoo! Instant Messenger (DAT)	Y	Y (null-terminated charsets)	N

¹The text direction in the output file might not be correct.

²In Export SDK, a bidirectional right-to-left (RTL) tag is extracted from this format and included in the direction element (<dir=RTL>) of the output.

Coded Character Sets

This section lists which character set you can use to specify the target character set. The coded character sets are enumerated in `kvcharset.h` and defined in the class.

Code Character Sets

Coded Character Set	Description	Can be set as target charset?
KVCS_UNKNOWN	Unknown character set	N
KVCS_SJIS	Japanese (uses multibyte encoding), cp932	Y
KVCS_GB	Simplified Chinese (China, Singapore, Malaysia) cp936	Y
KVCS_BIG5	Traditional Chinese (Taiwan, Hong Kong, Macaw) cp950	Y
KVCS_KSC	Korean, cp949	Y
KVCS_1250	Windows Latin 2 (Central Europe)	Y
KVCS_1251	Windows Cyrillic (Slavic)	Y

Code Character Sets, continued

Coded Character Set	Description	Can be set as target charset?
KVCS_1252	Windows Latin 1 (ANSI)	Y
KVCS_1253	Windows Greek	Y
KVCS_1254	Windows Latin 5 (Turkish)	Y
KVCS_1255	Windows Hebrew	Y
KVCS_1256	Windows Arabic	Y
KVCS_1257	Windows Baltic Rim	Y
KVCS_1258	Windows Vietnamese	Y
KVCS_8859_1	ISO 8859-1 Latin 1 (Western Europe, Latin America)	Y
KVCS_8859_2	ISO 8859-2 Latin 2 (Central Eastern Europe)	Y
KVCS_8859_3	ISO 8859-3 Latin 3 (S.E. Europe)	Y
KVCS_8859_4	ISO 8859-4 Latin 4 (Scandinavia/Baltic)	Y
KVCS_8859_5	ISO 8859-5 Latin/Cyrillic	Y
KVCS_8859_6	ISO 8859-6 Latin/Arabic	Y
KVCS_8859_7	ISO 8859-7 Latin/Greek	Y
KVCS_8859_8	ISO 8859-8 Latin/Hebrew	Y
KVCS_8859_9	ISO 8859-9 Latin/Turkish	Y
KVCS_8859_14	ISO 8859-14	Y
KVCS_8859_15	ISO 8859-15	Y
KVCS_437	DOS Latin US	Y
KVCS_737	DOS Greek	Y
KVCS_775	DOS Baltic Rim	Y
KVCS_850	DOS Latin 1	Y
KVCS_851	DOS Greek	Y
KVCS_852	DOS Latin 2	Y
KVCS_855	DOS Cyrillic	Y

Code Character Sets, continued

Coded Character Set	Description	Can be set as target charset?
KVCS_857	DOS Turkish	Y
KVCS_860	DOS Portuguese	Y
KVCS_861	DOS Icelandic	Y
KVCS_862	DOS Hebrew	Y
KVCS_863	DOS Canadian French	Y
KVCS_864	DOS Arabic	Y
KVCS_865	DOS Nordic	Y
KVCS_866	DOS Cyrillic Russian	Y
KVCS_869	DOS Greek 2	Y
KVCS_874	Thai	Y
KVCS_PDFMACDOC	PDF MAC DOC	N
KVCS_PDFWINDOC	PDF WIN DOC	N
KVCS_STDENC	Adobe Standard Encoding	N
KVCS_PDFDOC	Adobe standard PDF character set	N
KVCS_037	EBCDIC code page 037	Y
KVCS_1026	EBCDIC code page 1026	Y
KVCS_500	EBCDIC code page 500	Y
KVCS_875	EBCDIC code page 875	Y
KVCS_LMBCS	Lotus multibyte character set Group 1 and Group 2	N
KVCS_UNICODE	Unicode, UCS-2	Y
KVCS_UTF16	16-bit Unicode transformation format	Y
KVCS_UTF8	8-bit Unicode transformation format	Y
KVCS_UTF7	7-bit Unicode transformation format	Y
KVCS_2022_JP	ISO 2022-JP, Japanese mail and news safe encoding (JIS-7)	N

Code Character Sets, continued

Coded Character Set	Description	Can be set as target charset?
KVCS_2022_CN	ISO 2022-CN, Chinese mail and news safe encoding	N
KVCS_2022_KR	ISO 2022-KR, Korean mail and news safe encoding	N
KVCS_WP6X	Word Perfect 6.x and higher character mapping	N
KVCS_10000	Western European (Macintosh)	Y
KVCS_KSC5601	Unified Hangul	Y
KVCS_GB2312	Simplified Chinese (China, Singapore, Hong Kong)	Y
KVCS_GB12345	Traditional Chinese (China) - analogue of GB2312	Y
KVCS_CNS11643	Traditional Chinese - Taiwan. Supplement to Big5	Y
KVCS_JIS0201	Japanese - contains ASCII character set (JIS-Roman)	N
KVCS_JIS0212	Japanese. Supplement to JIS0208.	Y
KVCS_EUC_JP	Japanese Extended UNIX Code	Y
KVCS_EUC_GB	Simplified Chinese Extended UNIX Code	Y
KVCS_EUC_BIG5	Traditional Chinese Extended UNIX Code	N
KVCS_EUC_KSC	Korean Extended UNIX Code	N
KVCS_424	EBCDIC Hebrew	N
KVCS_856	PC Hebrew (old)	N
KVCS_1006	IBM AIX Pakistan (Urdu)	N
KVCS_KOI8R	Cyrillic (Russian)	Y
KVCS_PDF_JAPAN1	Adobe-Japan1-2 character collection	N
KVCS_PDF_KOREA1	Adobe-Korea1-0 character collection	N
KVCS_PDF_GB1	Adobe-GB1-3 character collection	N
KVCS_PDF_	Adobe-CNS1-2 character collection	N

Code Character Sets, continued

Coded Character Set	Description	Can be set as target charset?
CNS1		
KVCS_2022_JP_8	ISO 2022-JP, Japanese mail and news safe encoding (JIS8)	N
KVCS_720	Arabic DOS-720	Y
KVCS_VISCII	Vietnamese VISCII	Y
KVCS_8859_10	ISO 8859-10 (Latin 6 Nordic)	Y ¹
KVCS_8859_13	ISO 8859-13 (Latin 7 Baltic)	Y ¹
KVCS_57002	ISCII Devanagari (x-iscii-de)	Y ¹
KVCS_57003	ISCII Bengali (x-iscii-be)	Y ¹
KVCS_57004	ISCII Tamil (x-iscii-ta)	Y ¹
KVCS_57005	ISCII Telugu (x-iscii-te)	Y ¹
KVCS_57006	ISCII Assamese (x-iscii-as)	Y ¹
KVCS_57007	ISCII Oriya (x-iscii-or)	Y ¹
KVCS_57008	ISCII Kannada (x-iscii-ka)	Y ¹
KVCS_57009	ISCII Malayalam (x-iscii-ma)	Y ¹
KVCS_57010	ISCII Gujarathi (x-iscii-gu)	Y ¹
KVCS_57011	ISCII Panjabi (x-iscii-pa)	Y ¹
KVCS_GB18030b2	Reserved for internal use	n/a
KVCS_GB18030	GB18030 (Chinese 4-byte character set)	Y
KVCS_8859_11	ISO 8859-11 (Thai)	Y
KVCS_8859_16	ISO 8859-16 (Latin-10 South-Eastern Europe)	Y
KVCS_ARABICMAC	Arabic Mac (x-mac-arabic)	Y
KVCS_KOI8U	Cyrillic (KOI8U Ukrainian)	Y
KVCS_HZGB2312	The 7-bit representation of GB 2312 / RFC 1842	n/a
KVCS_UTF32	32-bit Unicode transformation format	

¹The character set cannot be forced as output in Export SDK and Viewing SDK because the character set is not supported by the major browsers.

Appendix D: File Format Detection

This section describes how file formats are detected in KeyView Viewing SDK.

- [Introduction](#) 352
- [Extract Format Information](#) 352
- [Determine Format Support](#) 352
- [Translate Format Information](#) 353
- [Determine a Document Reader](#) 354
- [Category Values in the Initialization File and Registry](#) 354

Introduction

The KeyView format detection module (kwad) detects a file's format, and reports the information to the API, which in turn reports the information to the developer's application. If the detected format is supported by the KeyView SDK, the detection module also loads the appropriate structured access layer and format reader for further processing.

For a list of supported formats, see [Document Readers, on page 309](#).

Extract Format Information

To extract format information from a document, use the [VAPIMWP_INIT_GETDOCFORMAT](#) parameter of the [VAPIM_INIT](#) message. This parameter gets format information (such as the file class, format, and version), and populates the [ADDOCINFO](#) structure. This structure, which is defined in the header file `adinfo.h`, specifies the formats that KeyView can detect. If required, this format information can then be reported to the developer's application.

For information on how to translate the extracted format information, see [Translate Format Information, on the next page](#).

Determine Format Support

After the file format is extracted, the detection module then uses an initialization file (`kvsdk.ini`) or the Windows registry to determine whether the format is supported by KeyView, and the appropriate structured access layer and reader to load. See [View Initialization Information, on page 22](#) for more information.

The initialization file and Windows registry contain the following information:

- Coded format information. To translate this information, see [Translate Format Information, on the next page](#).

- The reader associated with each format. See [Determine a Document Reader, on the next page](#).
- Initialization information. See [View Initialization Information, on page 22](#).

Below are some entries from the initialization file:

```
153=afsr.dll
207=afsr.dll
210=htmsr.dll
251=htmsr.dll
296=htmsr.dll
282=foliosr.dll
```

NOTE: The information in the initialization file and Windows registry applies to all formats except graphics. Detection of graphics formats is handled by an internal module named KeyView Picture Interchange Format (KPIF).

Translate Format Information

Format information can include file attributes in the following categories:

- Major format
- File class
- Minor format
- Major version
- Minor version

Not all categories are required. Many formats include only major format and file class, or major format only.

The format information has the following structure:

MajorFormat.FileClass.MinorFormat.MajorVersion.MinorVersion

For example:

81.2.0.9.0

Each number in the format information represents a file attribute. The entry 81.2.0.9.0 represents a Lotus 1-2-3 Spreadsheet file version 9.0, where

81 = Lotus 1-2-3 Spreadsheet (major format)

2 = Spreadsheet (file class)

0 = not defined (minor format)

9 = 9 (major version)

0 = 0 (minor version)

The example above applies to the initialization file and the Windows registry. When extracting format information by using the `VAPIMWP_INIT_GETDOCFORMAT` parameter, the same format information is represented as `294.2.0.9`.

NOTE: The format values returned by `VAPIMWP_INIT_GETDOCFORMAT` differ from those in the initialization file and Windows registry because the former defines a unique ID for each major format, whereas the latter uses a major version, minor version, and minor format to distinguish between formats.

Distinguish Between Formats

The `ADDDOCINFO` structure provides a unique ID for each major format. For example, `VAPIMWP_INIT_GETDOCFORMAT` returns `351.1.0` for a Microsoft Word 2003 XML format. The major format `351` is unique to this format.

Unlike `ADDDOCINFO`, the initialization file and the Windows registry use the major version number to distinguish between formats. For example, in the initialization file, a Microsoft Word 2003 XML format is defined as `285.1.0.100.0`. The major format `285` and file class `1` are the same values for generic XML. The major version `100` distinguishes the format as Microsoft Word 2003 XML.

The major version is used in the `kvsdk.ini` file or the Windows registry to specify the following formats:

- The Microsoft Office 2003 XML format has the same major format and file class as generic XML (`285.1`). It is distinguished from generic XML by using the following major versions:
 - Word: `100`
 - Excel: `101`
 - Visio: `110`
- The XHTML format has the same major format and file class as HTML (`210.1`). It is distinguished from HTML by using the major version `100`.

Determine a Document Reader

The entries in the initialization file or the Windows registry list each format's coded value, and the reader used to parse that format. For example, the entry below specifies that a Lotus 1-2-3 Spreadsheet file version 9.0 is parsed by the Lotus 1-2-3 reader, `1123sr.dll`:

```
81.2.0.9.0=1123sr.dll
```

[List of Files Required for Redistribution, on page 371](#) lists the document readers provided with KeyView.

Category Values in the Initialization File and Registry

This section lists the possible category values for format information in the initialization file and the Windows registry. The corresponding values for the format information extracted by using the `VAPIMWP_INIT_GETDOCFORMAT` parameter are listed in `adinfo.h`.

- [File Classes](#)
- [Minor Formats](#)

File Classes

Attribute Number	Description	File class
0	No file class	AutoDetNoFormat
01	Word processor	adWORDPROCESSOR
02	Spreadsheet	adSPREADSHEET
03	Database	adDATABASE
04	Raster image	adRASTERIMAGE
05	Vector graphic	adVECTORGRAPHIC
06	Presentation	adPRESENTATION
07	Executable	adEXECUTABLE
08	Encapsulation	adENCAPSULATION
09	Sound	adSOUND
10	Desktop publishing	adDESKTOPPUBLSH
11	Outline/planning	adOUTLINE
12	Miscellaneous	adMISC
13	Mixed format	adMIXED
14	Font	adFONT
15	Time scheduling	adSCHEDULE
16	Communications	adCOMMUNICATION
17	Object module	adOBJECTMODULE
18	Library module	adLIBRARY
19	Fax	adFAXFORMAT
20	Movie	adMOVIE
21	Animation	adANIMATION
22	Source Code	adSOURCECODE

File Classes, continued

Attribute Number	Description	File class
23	Computer-Aided Design	adCAD
24	BI and analysis tools	adANALYTICS
25	Scientific data	adSCIENTIFIC
26	Geographic Info System	adGIS

Minor Formats

Attribute Number	Minor Format
00	Minor format not defined
01	Standard
02	Book
03	Chart
04	Macro
05	Text
06	Binary
07	PC
08	Windows
09	DOS
10	Macintosh
11	RGB
12	TIFF
13	IFF
14	Experimental
15	Format Information
16	RLE
17	Symbol
18	Old

Minor Formats, continued

Attribute Number	Minor Format
19	Footnote
20	Style
21	Palette
22	Configuration
23	Activity
24	Resource
25	Calculation
26	Glossary
27	Spelling
28	Thesaurus
29	Hyphenation
30	Miscellaneous
31	UNIX
32	VAX
33	Driver
34	Archive

Appendix E: Extract and Format Lotus Notes Subfiles

This section describes how to create XML templates to alter the appearance of extracted Lotus mail note subfiles so that they maintain the look and feel of the original notes.

- [Overview](#) 358
- [Customize XML Templates](#) 358
- [Template Elements and Attributes](#) 360
- [Date and Time Formats](#) 365

Overview

KeyView uses the NSF reader, nsfsr, to extract Lotus database files, and places Lotus mail notes in subfiles. The NSF reader uses a set of default XML templates to extract the notes and apply formatting, thereby approximating the look and feel of the original notes.

In some cases, you might need to customize the XML templates, for instance if your notes contain custom data. In such cases, you can modify the existing XML templates or create your own.

During extraction, the NSF reader loads all XML files in the NSFtemplates directory and its subdirectories (except for the NSFtemplates\images directory, which is reserved for images). During initialization, the KeyView XML parser verifies the XML templates. If the templates contain any invalid XML, elements, or attributes, initialization fails and errors are recorded in the nsfsr.log file.

Customize XML Templates

XML templates are enabled by default. In most cases, the default templates should be sufficient; however, you can customize them or create your own as required.

To customize XML templates for Lotus note extraction

1. Modify the template files in the following directory.

```
install\OS\bin\NSFtemplates
```

The main.xml file must exist in the NSFtemplates directory. It is the top-level template file that extracts all subfiles, usually by calling other templates.

2. Make sure that any modifications or additional XML files conform to the supported elements and attributes described in [Template Elements and Attributes, on page 360](#).
3. Extract the Lotus database file.

Use Demo Templates

For testing purposes, you can extract notes by using a set of demo templates, which are provided to demonstrate the proper usage of all the XML elements and attributes, because the default templates do not use all the XML elements.

The demo templates are available at:

install\OS\bin\NSFtemplates

To use the demo XML templates

1. In the `formats.ini` file, set the following parameter.

```
[nsfsr]
UseDemoTemplate=1
```

2. In the `main.xml` file, uncomment the following section.

```
<ifini name="UseDemoTemplate" text="1">
  <call file="demo.xml"/>
  <quit/>
</ifini>
```

Use Old Templates

For testing purposes, you can extract notes by using legacy templates, which produce MHTML output. You can generate similar output by disabling the XML templates, but using the old templates enables you to see the XML code and compare it to the standard and demo templates.

To use the old XML templates

1. In the `formats.ini` file, set the following parameter.

```
[nsfsr]
UseOldTemplate=1
```

2. In the `main.xml` file, uncomment the following section.

```
<ifini name="UseOldTemplate" text="1">
  <call file="default_old.xml">
  <quit>
</ifini>
```

Disable XML Templates

For testing purposes, you can disable XML templates; KeyView extracts the notes in MHTML format. You can compare the MHTML output directly by the NSF reader with the MHTML output indirectly by the NSF reader through the XML templates.

To disable XML templates

1. In the `formats.ini` file, set the following parameter.

```
[nsfsr]
ExtractByTemplate=0
```

Template Elements and Attributes

This section lists the valid XML elements and attributes that you can use when creating or modifying templates. See the demo templates for examples.

Conditional Elements

The following table lists the valid conditional elements.

Conditional elements

Element	Description
<keyview>	The KeyView XML template container ("root") element
<if*>	<p>If the condition from the comparison is true, process the XML. Conditions can be nested up to 25 levels deep.</p> <p>Attributes</p> <ul style="list-style-type: none"> • <code>name</code>. (Required) The name of the main item to compare to <code>item</code> or <code>text</code>. • <code>item</code>. (Required if no <code>text</code>) The name of the item to compare to the item specified by <code>name</code>. • <code>text</code>. (Required if no <code>item</code>) The text to compare to the item specified by <code>name</code>.
<ifex>, <ifnx>	<p>If <code>name</code> item exists and has a <code>text</code> value or not.</p> <p>The Notes item might have a value that cannot be converted to text, such as an image.</p>
<ifeq>, <ifne>, <iflt>, <ifle>, <ifgt>, <ifge>	<p>Respectively, if <code>text</code> ==, !=, <, >, <=, >, >=.</p> <p>Text comparison uses a case-insensitive string compare.</p>
<iftdeq>, <iftdne>, <iftdlt>, <iftdle>, <iftdgt>, <iftdge>	<p>Respectively, if time/date ==, !=, <, >, <=, >, >=.</p> <p>Time/date comparison converts dates to text in local time using the Notes default, <code>TZFMT_NEVER</code>, because Notes also sometimes converts fields to text internally. For example:</p> <pre>text="06/30/2005 02:52:04 PM"</pre>

Conditional elements, continued

Element	Description
<iftzeq>, <iftzne>	Respectively, if the time zone equals or does not equal the comparison text, for example CDT, EST, and so on.
<ifini>	If the value of the INI option specified in name equals the text value.
<else>	If the condition from the last <if> or <switch> was false, process XML.
<switch>	If a name value exists, process XML. Attributes <ul style="list-style-type: none"> name. (Required) The name of the main item to compare in <case> subelements.
<case>	If the comparison condition is true, process XML, then stop processing the rest of <switch>. Attributes <ul style="list-style-type: none"> text. (Required) The text to compare to the name item of <switch>.
<default>	If all <case> conditions were false, process XML. This element must be the last element in <switch>, after all the <case> elements. Any <case> elements after the <default> element are ignored.
<for>	If a name value exists, process XML. Process for each part of the name item. Attributes <ul style="list-style-type: none"> name. (Required) The name of the main item. max. (Optional) The maximum index to process. By default, all are processed.
<index>	Output <for> loop index (1-based). <index> is only valid within a <for> element.

Control Elements

The following table lists the valid control elements.

Control Elements

Element	Description
<call>	Call another XML template. You can nest templates up to 10 levels deep. Attributes

Control Elements, continued

Element	Description
	<ul style="list-style-type: none"> file. (Required) The template file name. This name must be unique.
<log>	<p>Log message to the NSF log file.</p> <p>Attributes</p> <ul style="list-style-type: none"> text. (Required) The text to log. type. (Optional) The type of log message. The following values are valid: <ul style="list-style-type: none"> ERROR WARN INFO DIAG (the default option) DEBUG DUMP
<quit>	<p>Stop processing the template. Exits without error.</p> <p>Attributes</p> <ul style="list-style-type: none"> text. (Optional) The text to log. type. (Optional) The type of log message. See <log>, above.
<stop>	<p>Stop processing the template. Exits with an ERROR log message.</p> <p>Attributes</p> <ul style="list-style-type: none"> text. (Required) The text to log.

Data Elements

The following table lists the valid data elements.

Data elements

Element	Description
<text>	<p>Output text.</p> <p>Attributes</p> <ul style="list-style-type: none"> name. (Required if there is no parent) The name of the item to output.
<rich>	<p>Output rich text (MHTML). Images are output in the next part or parts of the MHTML, after the first <HTML> part.</p>

Data elements, continued

Element	Description
	<p>Attributes</p> <ul style="list-style-type: none"> name. (Required if there is no parent) The name of the item to output.
<body>	Output the message body in rich text (MHTML). As with <rich>, on the previous page, images are output in the next part or parts of the MHTML.
<form>	Output the message form (usually \$Body field) in rich text (MHTML). <p>Attributes</p> <ul style="list-style-type: none"> name. (Required if there is no parent) The name of the item to output.
<addr>	Output an address. <p>Attributes</p> <ul style="list-style-type: none"> name. (Required if there is no parent) The name of the item to output. type. (Optional) The type of address to output. Set this attribute to CN (Common Name), which is the only supported type.
<name>	Output the name of the last name item, or in other words the current main item. The item must exist.
<format>	Set the default format for <date> and <date_kv>. This element does not set the <text> format. See Date and Time Formats, on page 365 for a list of all Notes and KeyView date and time formats and integer values. <p>Attributes</p> <ul style="list-style-type: none"> format. (Optional. Omit to reset to defaults) The Notes and KeyView date and time format. You can set the following formats: <ul style="list-style-type: none"> TD=int. The Time Date format (TDFMT_*) TS=int. The Time Show format (TSFMT_*) TT=int. The Time Time format (TTFMT_*) TZ=int. The Time Zone format (TZFMT_*) KV=int. The KeyView date and time format <p>where int is an integer value that corresponds to the desired format.</p> <p>Separate multiple formats with commas. For example:</p> <pre>format="TD=0,TS=2,TT=1,TZ=1,KV=55"</pre>
<date>	Output a Notes date. <p>Attributes</p> <ul style="list-style-type: none"> name. (Required if there is no parent) The name of the item to output.

Data elements, continued

Element	Description
	<ul style="list-style-type: none"> • format. (Optional) See <format>, on the previous page. You can set the following values: <ul style="list-style-type: none"> ◦ TD ◦ TS ◦ TT ◦ TZ
<p><date_kv></p>	<p>Output a KeyView date.</p> <p>Attributes</p> <ul style="list-style-type: none"> • name. (Required if there is no parent) The name of the item to output. • format. (Optional) See <format>, on the previous page. You can set the following values: <ul style="list-style-type: none"> ◦ TZ ◦ KV
<p><time></p>	<p>Output a time range, for example 1 hour, 30 minutes.</p> <p>Attributes</p> <ul style="list-style-type: none"> • name. (Required if there is no parent) The item name of the start date or time. • item. (Required) The item name of the end date or time.
<p><zone></p>	<p>Output a Notes time zone mnemonic, for example MST.</p> <p>Attributes</p> <ul style="list-style-type: none"> • name. (Required if there is no parent) The name of date item to output.
<p><zone_utc></p>	<p>Output a time zone as UTC, for example (UTC-06:00).</p>
<p><logo></p>	<p>Output the mail header logo.</p> <p>The image link is included in the output; the actual image is output to a different part of the MHTML subfile.</p>
<p><image></p>	<p>Output an image.</p> <p>The image link is included in the output; the actual image is output to the MHTML next part, as with <rich>, on page 362 and <body>, on the previous page.</p>
<p><image_uri></p>	<p>Output an image URI, in quotation marks. The actual image is output to a different part of the MHTML subfile.</p> <p>Attributes</p>

Data elements, continued

Element	Description
	<ul style="list-style-type: none"> • <code>link</code>. (Required if there is no <code>file</code>) The image link, such as a form or title name. For example: <ul style="list-style-type: none"> • <code>link="StdNotesLtr0"</code> • <code>file</code>. (Required if there is no <code>link</code>) The name of the image file. The file must exist in the <code>../../templates/images</code> directory. For example: <ul style="list-style-type: none"> • <code>file="boxcheck.gif"</code>

Date and Time Formats

This section lists the supported Notes and KeyView date and time formats for use with `<format>`, `<date>`, and `<date_kv>`.

Lotus Notes Date and Time Formats

This section lists supported Lotus Notes date and time formats, and the integer values that specify each one.

Lotus Notes date and time formats

Format	Integer Value	Description
TDFMT_FULL	0	(The Notes default) Year, month, and day
TDFMT_CPARTIAL	1	Month and day, year if not this year
TDFMT_PARTIAL	2	Month and day
TDFMT_DPARTIAL	3	Year and month
TDFMT_FULL4	4	Four-digit year, month, and day
TDFMT_CPARTIAL4	5	Month and day, four-digit year if not this year
TDFMT_DPARTIAL4	6	Four-digit year and month
TTFMT_FULL	0	(Notes default) Hour, minute, and second
TTFMT_PARTIAL	1	Hour and minute
TTFMT_HOUR	2	Hour

Lotus Notes date and time formats, continued

Format	Integer Value	Description
TZFMT_NEVER	0	(Notes default) All time zones are converted to the current time zone
TZFMT_SOMETIMES	1	Show only when outside the current time zone
TZFMT_ALWAYS	2	Show for all time zones
TSFMT_DATE	0	Date
TSFMT_TIME	1	Time
TSFMT_DATETIME	2	(The Notes default) Date and time
TSFMT_CDATETIME	4	Date and time, or time today or time yesterday

KeyView Date and Time Formats

This section lists KeyView date and time formats. The KeyView formats use the following syntax:

- Month Month = full month name
 Mon = abbreviated month name
 m = month (number)
 mm = two-digit month (leading 0)
- Weekday Weekday = full weekday name
 Wday = abbreviated weekday name
- Year yy = two-digit year
 yyyy = four-digit year
- >Day d = day (number)
 dd = two-digit day (leading 0)
- Time h = 12-hour
 H = 24-hour
 m = minutes
 s = seconds
 P = AM/PM
 p = am/pm

Separators _ = space
 c = comma
 s = slash
 a = dash
 o = dot

KeyView date and time formats

Format	Output	Integer Value
12-Hour and 24-Hour Time Formats		
KVDTF_P	P	1
KVDTF_P_hmm	P h:mm	2
KVDTF_hmm_P	h:mm P	3
KVDTF_P_hhmm	P hh:mm	4
KVDTF_hhmm_P	hh:mm P	5
KVDTF_P_hmmss	P h:mm:ss	6
KVDTF_hmmss_P	h:mm:ss P	7
KVDTF_P_hhmmss	P hh:mm:ss	8
KVDTF_hhmmss_P	hh:mm:ss P	9
KVDTF_Hmm	H:mm	10
KVDTF_HHmm	HH:mm	11
KVDTF_mmss	mm:ss	12
KVDTF_Hmmss	H:mm:ss	13
KVDTF_HHmss	HH:mm:ss	14
Numerical Date Formats with Slashes		
KVDTF_mmsdd	mm/dd	15
KVDTF_msdsyy	m/d/yy	16
KVDTF_mmsddsyy	mm/dd/yy	17
KVDTF_mmsddsyyyy	mm/dd/yyyy	18
KVDTF_ddsmm	dd/mm	19

KeyView date and time formats, continued

Format	Output	Integer Value
KVDTF_ddsmsyy	dd/mm/yy	20
KVDTF_ddsmsyy_Hmm	dd/mm/yy H:mm	21
KVDTF_ddsmm_P_hmm	dd/mm P h:mm	22
KVDTF_ddsmm_hmm_P	dd/mm h:mm P	23
KVDTF_ddsmm_P_hhmm	dd/mm P hh:mm	24
KVDTF_ddsmm_hhmm_P	dd/mm hh:mm P	25
KVDTF_ddsmsyy_P_hmm	dd/mm/yy P h:mm	26
KVDTF_ddsmsyy_hmm_P	dd/mm/yy h:mm P	27
KVDTF_ddsmsyy_P_hmmss	dd/mm/yy P h:mm:ss	28
KVDTF_ddsmsyy_hmmss_P	dd/mm/yy h:mm:ss P	29
KVDTF_ddsmsyy_P_hhmmss	dd/mm/yy P hh:mm:ss	30
KVDTF_ddsmsyy_hhmmss_P	dd/mm/yy hh:mm:ss P	31
KVDTF_yysmmsdd_P_hhmmss	yy/mm/dd P hh:mm:ss	32
KVDTF_yysmmsdd_hhmmss_P	yy/mm/dd hh:mm:ss P	33
KVDTF_msdsyy_Hmm	m/d/yy H:mm	34
KVDTF_mmsddsyy_Hmm	mm/dd/yy H:mm	35
KVDTF_msdsyy_P_hmm	m/d/yy P h:mm	36
KVDTF_msdsyy_hmm_P	m/d/yy h:mm P	37
KVDTF_mmsddsyy_hmm_P	mm/dd/yy h:mm P	38
KVDTF_mmsdd_P_hhmm	mm/dd P hh:mm	39
KVDTF_mmsdd_hhmm_P	mm/dd hh:mm P	40
KVDTF_mmsddsyy_P_hhmmss	mm/dd/yy P hh:mm:ss	41
KVDTF_mmsddsyy_hhmmss_P	mm/dd/yy hh:mm:ss P	42
KVDTF_msd	m/d	43
KVDTF_yysm	yy/m	44
KVDTF_yysmm	yy/mm	45

KeyView date and time formats, continued

Format	Output	Integer Value
KVDTF_ysmsd	yy/m/d	46
KVDTF_ysmmsdd	yy/mm/dd	47
KVDTF_yyyysmmsdd	yyyy/mm/dd	48
Numerical Date Formats with Dashes		
KVDTF_ddammayy	dd-mm-yy	49
KVDTF_mmadd	mm-dd	50
KVDTF_mmayy	mm-yy	51
KVDTF_yyamadd	yy-mm-dd	52
KVDTF_yyyyamadd	yyyy-mm-dd	53
KVDTF_yyyyamaddaHHmss	yyyy-mm-dd-HH:mm:ss	54
Numerical Date Formats with Dots		
KVDTF_yyomod	yy.m.d	55
KVDTF_yyommodd	yy.mm.dd	56
KVDTF_mod	m.d	57
KVDTF_mmodd	mm.dd	58
Numerical and String Date Formats with Dashes, Commas, and Spaces		
KVDTF_ddaMon	dd-Mon	59
KVDTF_daMonayy	d-Mon-yy	60
KVDTF_ddaMonayy	dd-Mon-yy	61
KVDTF_ddaMonayyyy	dd-Mon-yyyy	62
KVDTF_Mon	Mon	63
KVDTF_Monayy	Mon-yy	64
KVDTF_Monayyyy	Mon-yyyy	65
KVDTF_Monaddayy	Mon-dd-yy	66
KVDTF_yyamadd_P_hhmmss	yy-mm-dd P hh:mm:ss	67
KVDTF_mmadd_P_hhmm	mm-dd P hh:mm	68

KeyView date and time formats, continued

Format	Output	Integer Value
KVDTF_Mon_yy	Mon yy	69
KVDTF_Monc_yy	Mon, yy	70
KVDTF_Month	Month	71
KVDTF_Monthyy	Month-yy	72
KVDTF_Month_yy	Month yy	73
KVDTF_Monthc_yy	Month, yy	74
KVDTF_Monthayyyy	Month-yyyy	75
KVDTF_Month_yyyy	Month yyyy	76
KVDTF_Monthc_yyyy	Month, yyyy	77
KVDTF_Mon_dc_yyyy	Mon d, yyyy	78
KVDTF_d_Monc_yyyy	d Mon, yyyy	79
KVDTF_yyyy_Mon_d	yyyy Mon d	80
KVDTF_Month_dc_yyyy	Month d, yyyy	81
KVDTF_d_Monthc_yyyy	d Month, yyyy	82
KVDTF_yyyy_Month_d	yyyy Month d	83
Weekday Date Formats		
KVDTF_wday	wday	84
KVDTF_Weekday	Weekday	85
KVDTF_wdayc_Mon_dc_yyyy	wday, Mon d, yyyy	86
KVDTF_Weekdayc_Month_dc_yyyy	Weekday, Month d, yyyy	87
KVDTF_Weekdayc_d_Monthc_yyyy	Weekday, d Month, yyyy	88

Appendix F: List of Files Required for Redistribution

This section lists the files required for redistributing applications based on Viewing SDK.

- [Core Files](#) 371
- [Support Files](#) 372
- [Document Readers and Writers](#) 373
- [Miscellaneous Functionality](#) 381
- [Viewing ActiveX Control](#) 382

The following files should be installed to the `\bin` directory of your application's installation directory.

Core Files

The following core files can be redistributed with your application.

File	Description
chartb1s.ux	Character mapping tables.
htmconv.dll	HTML converter for the document token stream.
kvarcve.dll	Archive format viewing engine.
kvdecrypt.dll	Decryption utility functions.
kvdocve.dll	Word processing format viewing engine.
kvmailve.dll	Mail format viewing engine.
kvmve.dll	Multimedia format viewing engine.
kvpicve.dll	Picture format viewing engine.
kvolefio.dll	Embedded OLE object reader.
kvsdk.ini	Initialization file.
kvutil.dll	Utility.
kvvapi.dll	Viewing API.
kvwkbve.dll	Spreadsheet format viewing engine.
kvxssa.dll	Interface between spreadsheet readers and Viewing API.
kvxtract.dll	File Extraction interface for container file support.
kvxwpsa.dll	Interface between word processing document readers and Viewing API.

File	Description
kvzip.dll	Zip writer.
kwad.dll	Format detection module.
kwcm.dll	Conversion Manager.
kwlm.dll	URL Launch Manager.
kwres.dll	Resources.
vcredist*	Microsoft Visual C++ Redistributable Packages. NOTE: This folder can be found in the Viewing SDK installation directory.

Support Files

The following support files can be redistributed with your application.

File	Description
datafiles*	(Folder) Required by kvlangdetect.
NSFtemplates*	(Folder) Templates used by nsfsrc to format Lotus mail notes.
7z.dll	Required by z7zsrc and multiarcsrc.
bentofio.dll	Required by 1123src.dll and kpprzrdr.dll.
cbmap.map	Character mappings for Adobe Portable Document Format (PDF).
chmdll.dll	Required by chmsrc.dll.
cpstsdk*	Required by pstxsrc.
kp3dwrld.dll	Required for 3D charts.
kpifcnvt.dll	Picture conversion routines.
kpifutil.dll	Picture utility routines.
kpjpeg.dll	JPEG file interchange format shared routines.
kppng.dll	Portable Network Graphics (PNG) utilities.
kv.lic	Contains license information for KeyView products. This file is opened and validated when a KeyView API is used.
kvaxcc.dll	Required for viewing HTML using Internet Explorer within View API window
kvgraph.dll	Required for all spreadsheets (chart support).

File	Description
kvlangdetect.dll	Utility functions for language and character set detection.
kvpageve.dll	An alternate viewing engine for Word processing formats.
kvpie.dll	Required for all spreadsheets (chart support).
kvplug.dll	Required for PDF support through the Acrobat plug-in if you use a version of Acrobat Reader earlier than 4.0. You might have to manually install the Acrobat plug-in nppdf32.dll. This is determined by the browser you use. If you use Netscape, nppdf32.dll installs automatically with Acrobat Reader. However, if you use other browsers, such as Internet Explorer, you must manually install nppdf32.dll into a subdirectory of the Viewing Home directory called plugins. You must then set up the registry or initialization file according to the description for PDF in kvsdk.ini or install.reg.
kvradar.dll	Required for all spreadsheet formats (chart support).
kvreg.dll	Sheet Registry processing.
kvssvwr.dll	Required for all spreadsheet formats.
kvthread.*	Required for multithreaded out-of-process filtering.
kvxmlve.dll	XML format viewing engine.
langdetecttext.dll	Required by kvlangdetect.dll.
libcrypto*	SSL utility functions used by KeyView mail format readers.
libpff.dll	Required by pffsr.
unzipjpg.dll	Required for JPEG decompression.
wpmap.dll	Extended character mapping for WordPerfect and Corel Presentation.
xmlsh.dll	Contains a library of content handlers for each XML file type. Required by the Expat XML parser.

Document Readers and Writers

The following readers and writers can be redistributed with your application.

Archive Formats

File	Description
ad1sr.dll	AD1 Evidence file reader

File	Description
b1sr.dll	B1 archive reader
bkfsr.dll	Microsoft Backup File reader
bzip2sr.dll	Bzip2 reader
cabsr.dll	Microsoft Cabinet format reader
chmsr.dll	Microsoft Compiled HTML Help reader
dmgsr.dll	Mac Disk Copy Disk Image File reader
emxsr.dll	Legato EMailXtender archive (EMX) reader
encasesr.dll	Expert Witness Compression Format (EnCase) v6 reader
encase2sr.dll	Expert Witness Compression Format (EnCase) v7 reader
isosr.dll	ISO-9660 CD Disc Image Format reader
kvgz.dll	GZIP reader
kvgzsr.dll	GZIP reader
kvhqx.dll	BinHex reader
kvzee.dll	Unix Compress reader
kw2hqx.dll	BinHex writer
kw2tar.dll	Tape Archive writer
kw2uue.dll	UUEncoding writer
kw2zee.dll	Unix Compressed writer
kw2zip.dll	ZIP writer
lzhsr.dll	Microsoft Compression Folder reader.
macbinsr.dll	MacBinary reader
multiarcsr.dll	ARJ reader
rarsr.dll	RAR Archive reader
tarsr.dll	Tape Archive (TAR) reader
unzip.dll	ZIP reader
uudsr.dll	UUEncoding reader
z7zsr.dll	7-Zip reader

Binary Formats

File	Description
exesr.dll	DOS/Windows Executables/DLLs

Computer-Aided Design Formats

File	Description
gdsiisr.dll	Graphic Database System (GDSII) reader
kpodandr.dll	AutoCAD reader (Windows only)
kpvsd2rdr.dll	Microsoft Visio reader
kpvdxrdr.dll	Microsoft Visio 2013 reader
vsdsr.dll	Microsoft Visio reader

Database Formats

File	Description
dbfsr.dll	dBase Database reader
mdbsr.dll	Microsoft Access reader
mppsр.dll	Microsoft Project reader

Desktop Publishing Formats

File	Description
mspubsr.dll	Microsoft Publisher reader

Display Formats

File	Description
kppdfrdr.dll	Adobe Portable Document File (PDF) graphic-based reader
kppdf2rdr.dll	High-fidelity Adobe Portable Document File (PDF) graphic-based reader

Graphic Formats

File	Description
jp2000sr.dll	JPEG 2000 metadata reader
kpanirdr.dll	Windows Animated cursor reader
kpbmprdr.dll	Windows Bitmap reader
kpbmpwrt.dll	Windows Bitmap writer
kpcdrdr.dll	Corel Draw reader
kpcgmrdr.dll	Computer Graphics Metafile (CGM)
kpdcxrdr.dll	DCX (fax) reader
kpem2rdr.dll	Enhanced Windows Metafile (EMF) reader
kpemfrdr.dll	Enhanced Windows Metafile (EMF) reader
kpemfwrt.dll	Enhanced Metafile writer
kpepsrdr.dll	Encapsulated PostScript (EPS) reader
kpgflrdr.dll	OmniGraffle Picture reader
kpgifrdr.dll	Graphic Interchange Format (GIF) reader
kpicordr.dll	Windows Icon reader
kpjbig2rdr.dll	JBIG2 reader
kpjp2000rdr.dll	JPEG 2000 reader
kpjpgdrdr.dll	JPEG file interchange format reader
kpjpgwrt.dll	JPEG file interchange format writer
kpmacrdr.dll	MacPaint reader
kpmsordr.dll	Microsoft Office Drawing Objects reader
kpnbmrdr.dll	Lotus Notes Bitmap reader (for embedded images in DXL files)
kpoxdrdr.dll	Open Office XML Diagram Graphics reader
kppctrdr.dll	Macintosh Quick Draw Picture (PICT) reader
kppcxrdr.dll	PC Paintbrush (PCX) reader
kppicdrdr.dll	Pictor PC Paint format (PIC) reader

File	Description
kppngrdr.dll	Portable Network Graphics (PNG) reader
kppngwrt.dll	Portable Network Graphics (PNG) writer
kprawdr.dll	ODA Internal Raster (RAW) Picture reader
kpsdwrdr.dll	Lotus Ami Pro Graphics reader
kpsgirdr.dll	SGI RGB reader
kpsprdr.dll	Shape Stream reader
kpsunrdr.dll	Sun Raster reader
kptgardr.dll	Truevision Targa reader
kptifrdr.dll	Tagged Image File Format reader
kptifwrt.dll	Tagged Image File Format writer
kpwg2rdr.dll	WordPerfect Graphics 2.0 reader
kpwm2rdr.dll	Windows Metafile (WMF) reader
kpwmfrdr.dll	Windows Metafile (WMF) reader
kpwmfwrt.dll	Windows Metafile writer
kpwpgrdr.dll	WordPerfect Graphics 1.0 reader

Mail Formats

File	Description
dbxsr.dll	Microsoft Outlook Express DBX reader
dxl1sr.dll	Domino XML Language reader
eml1sr.dll	Microsoft Outlook Express (EML) reader
entsr.dll	Microsoft Entourage Database Format reader
gwfssr.dll	GroupWise FileSurf reader
icssr.dll	Microsoft Outlook iCalendar reader
msgsr.dll	Microsoft Outlook (MSG) reader
mbxsr.dll	Mailbox (MBX) and Microsoft Outlook Express (EML) reader. This reader is considered an advanced feature and is sold and licensed separately from the Viewing SDK. See License Information, on page 17 .

File	Description
nsfsrc.dll	Lotus Notes Database reader. This reader is considered an advanced feature and is sold and licensed separately from the Viewing SDK. See License Information, on page 17 .
olmsrc.dll	Microsoft Outlook for Macintosh reader
onmsrc.dll	Legato EMailXtender Native Message reader
pffsrc.dll	Microsoft Outlook Offline Storage File reader
pstsrc.dll	Microsoft Outlook Personal Folders file MAPI-based reader (supported on Windows platform only). This reader is considered an advanced feature and is sold and licensed separately from the Viewing SDK. See License Information, on page 17 .
pstnsr.dll	Microsoft Outlook Personal Folders file native reader. This reader is considered an advanced feature and is sold and licensed separately from the Viewing SDK. See License Information, on page 17 .
pstxsr.dll	Microsoft Outlook Personal Folders file native reader. This reader is considered an advanced feature and is sold and licensed separately from the Viewing SDK. See License Information, on page 17 .
tnefsrc.dll	Transfer Neutral Encapsulation Format reader
vcfsrc.dll	Microsoft Outlook vCard Contact reader

Presentation Formats

File	Description
kpagrdr.dll	Applix Presentations reader
kpiwpgdr.dll	Apple iWork Keynote reader
kpodfrdr.dll	Oasis Open Document Format presentation (ODP) reader
kpONErdr.dll	Microsoft OneNote reader
kpp40rdr.dll	Microsoft PowerPoint 4.0 reader
kpp95rdr.dll	Microsoft PowerPoint 95 reader
kpp97rdr.dll	Microsoft PowerPoint 97, 2000, and 2002 reader
kpppxrdr.dll	Microsoft PowerPoint XML reader 2007
kpprerdr.dll	Lotus Freelance 96/97 reader
kpprzrdr.dll	Lotus Freelance 2.x reader
kpshwrdr.dll	Corel Presentation Graphics reader

File	Description
kpxfd1rdr.dll	Extensible Forms Description Language reader
swfsr.dll	Macromedia Flash reader
vdsr.dll	Microsoft Visio reader

Spreadsheet Formats

File	Description
assr.dll	Applix spreadsheet reader
csvsr.dll	Comma Separated Values reader
difsr.dll	Data Interchange Format reader
htmss.dll	Required to save spreadsheets as HTML.
iwssr.dll	Apple iWork Numbers reader
kpchtrdr.dll	Required for all spreadsheets (chart support)
l123sr.dll	Lotus 123 V96/97 reader
mwssr.dll	Microsoft Works Spreadsheet reader
odfssr.dll	Oasis Open Document Format spreadsheets (ODS) reader
pbixsr.dll	Microsoft Power BI file (PBIX) reader
qpssr.dll	Quattro Pro Spreadsheet reader
qpwsr.dll	Corel Quattro Pro version X4 spreadsheet reader
starcsr.dll	StarOffice Calc reader
wkssr.dll	Lotus 123 V2 to 5 reader
xlsbsr.dll	Microsoft Office 2007 Excel Binary Format reader
xlssr.dll	Microsoft Excel reader
xlsxsr.dll	Microsoft Excel 2007 XML reader

Word Processor Formats

File	Description
afsr.dll	ASCII reader

File	Description
awsr.dll	Applix Words V4.x reader
dcasr.dll	Document Content Architecture/Revisable Form Text (DCA/RFT) reader
dw4sr.dll	DisplayWrite 4 reader
epubsr.dll	Open Publication Structure eBook reader
foliosr.dll	Folio Flat File 3.1 reader
hexsr.dll	Hexadecimal reader
h17sr.dll	Health level7 reader
htmsr.dll	Hypertext Markup Language (HTML) reader
hwposr.dll	Hangul reader
hwpsr.dll	Hangul 97 reader
ichatsr.dll	Apple iChat Log reader
iwwp13sr.dll	iWork 13 Pages reader
iwwpsr.dll	Apple iWork Pages reader
jtdsr.dll	JustSystems Ichitaro reader
lasr.dll	Lotus AMI Pro reader
1tbenn30.dll	Lotus Word Pro support
1tscsn10.dll	Lotus Word Pro support
1wpapin.dll	Lotus Word Pro support
1wppann.dll	Lotus Word Pro support
1wpsr.dll	Lotus Word Pro reader.
mbsr.dll	Microsoft Word Mac reader
mhtsr.dll	MIME HTML reader
mifsr.dll	Adobe Maker Interchange Format (.mif) reader
misr.dll	Microsoft Word 2 reader
msw6sr.dll	Microsoft Works 6, 2000 reader
mswsr.dll	Microsoft Works 1, 2, 3, 4 reader
mw6sr.dll	Microsoft Word 95 reader

File	Description
mw8sr.dll	Microsoft Word 97, 2000, XP reader
mwsr.dll	Microsoft Word for DOS and Microsoft Write reader
mwxsr.dll	Microsoft Word 2007 XML reader
oa2sr.dll	Fujitsu Oasys reader
odfwpsr.dll	Oasis Open Document Format word processing (ODS) reader
onealtsr.dll	Microsoft OneNote Alternate Format reader
onesr.dll	Microsoft OneNote Format reader
oo3sr.dll	Omni Outliner reader
pfilesr.dll	Microsoft Rights Management System encryption file reader
rtfsr.dll	Microsoft Rich Text Format reader
skypesr.dllk	Skype log file reader
sosr.dll	StarOffice/OpenOffice reader
starwsr.dll	StarOffice Writer reader
unihtmsr.dll	Unicode HTML reader
unisr.dll	Unicode reader
wosr.dll	WordPerfect 5.x reader
wp6sr.dll	WordPerfect 6.0 through 10.0 reader
wpmsr.dll	WordPerfect for Macintosh reader
xmlsr.dll	XML reader
xpssr.dll	XML Paper Specification reader
xywsr.dll	XyWrite reader
yimsr.dll	Yahoo! Instant Messenger reader

Miscellaneous Functionality

File	Description
kvcnv.dll	SaveAs
kvtlbar.dll	Toolbar with MFC library dynamically loaded (need to redistribute mfc42.dll).

File	Description
kvtlbst.dll	Toolbar with MFC library statically linked
rtfcnv.dll	SaveAs RTF (through SaveAs dialog box or VAPIM_CONVERT), Copy to Clipboard
rtfss.dll	SaveAs RTF, Copy to Clipboard
txtcnv.dll	SaveAs Text (through SaveAs dialog box) Copy to Clipboard

Viewing ActiveX Control

File	Description
kvocx.ocx	Viewing ActiveX control

Appendix G: Configuration Options in kvsdk.ini

This appendix lists and explains configuration parameters available in the kvsdk.ini file.

- [kvsdk.ini Options, below](#)

kvsdk.ini Options

The following table lists configuration parameters available in the kvsdk.ini file.

kvsdk.ini configuration options

Configuration option	Description
[DiskCache] section	
DiskCacheSize type = integer default = 64	Specify the amount of memory in KB that KeyView will use for caching. Generally, when you increase the memory, performance improves. To determine a reasonable value, divide the maximum amount of memory you want KeyView to use by the number of threads. For example, if you want KeyView to use 50MB of memory and have 10 threads, set the value to 5MB, or 5120. The minimum amount of memory that you can use for file caching is 64 KB.
[nsfsr] section	
ExtractAllNotes type = Boolean default = 0	Set to 1 to extract all classes of notes and all subfiles regardless of whether they contain mail headers.
ExtractAllFields type = Boolean default = 0	Set to 1 to extract all fields to a subfile. Applies to non-mail subfiles only.
TempDir type = file path default = current temp directory	(Windows only) Specify a new temp directory.
ExportDXL type = Boolean default = 0	Export as DXL instead of MHT.

kvsdk.ini configuration options, continued

Configuration option	Description
[pdf_flags] section	
remove_invisible_text type = Boolean default = 0	Set to 1 if you do not want to include invisible text from PDF documents in your output.
[ss_flags] section	
process_images_with_min_width type = integer	The minimum width (in pixels) that an image in a spreadsheet must have for it to be exported from or viewed in a spreadsheet. This option can improve performance for documents that have lots of very small images.
process_images_with_min_height type = integer	The minimum height (in pixels) that an image in a spreadsheet must have for it to be exported from or viewed in a spreadsheet. This option can improve performance for documents that have lots of very small images.

Appendix H: Password Protected Files

This section lists supported password-protected container and non-container files and describes how to open them.

- [Supported Password Protected File Types, below](#)
- [View Password Protected Files, on the next page](#)

Supported Password Protected File Types

The following table lists the password-protected file types that KeyView supports.

Key to support table

Symbol	Description
Y	Format is supported.
N	Format is not supported.
S	Support for viewing subfiles.
V	Support for viewing content.
P	Password required.
C	Password and certificate or User ID file required.

Supported password-protected file types

File Type	Version	Filter	Export	Extract	View	Credentials
PST (Windows)	n/a	N	N	Y	S	P
PST (non-Windows) ¹	n/a	N	N	Y	S	N
ZIP	n/a	N	N	Y	S	P
7-Zip	n/a	N	N	Y	S	P
RAR	n/a	N	N	Y	S	P
SMIME in MSG, EML, MBX	n/a	N	N	Y	N	C

¹The native PST readers, pstxsr and pstnsr, do not require credentials to open password-protected PST files that use compressible encryption.

Supported password-protected file types, continued

File Type	Version	Filter	Export	Extract	View	Credentials
Lotus Notes NSF	n/a	N	N	Y	N	C
Adobe PDF	n/a	Y	Y	Y	V	P
Microsoft Office	97-2003 2007 2010	Y	Y	Y	V	P

View Password Protected Files

This section describes how to view password-protected files by using the Viewing API.

To view password-protected files

- Set the password with the [VAPIMWP_INIT_SETPASSWORD](#) message parameter.
 - For password-protected PST files, this message must be called before the `VAPIMWP_INIT_OPEN_DOCUMENT` message.
 - For password-protected Microsoft Office 2007 and 2010 files, this message must be called before the `VAPIMWP_INIT_OPEN_DOCUMENT` message.
 - For password-protected ZIP files, this message can be called after the `VAPIMWP_INIT_OPEN_DOCUMENT`, but must be called before the protected subfile is extracted or viewed.

Send documentation feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Viewing SDK Programming Guide (Micro Focus KeyView 12.6)

Add your feedback to the email and click **Send**.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to swpdl.idoldocsfeedback@microfocus.com.

We appreciate your feedback!