Help System for CGMIMP, CGM import filter.

Table of Contents

Introduction

This HELP provides additional detailed information about the CGM import filter for MS Windows from Henderson Software Inc (HSI). This import filter is packaged with Office 97 (and with SGML Author, together with its companion CGM export filter).

Following some clarification, in this Introduction, of the history of CGM filters shipped with MS application products, there is description of tuning options available with the filter(s), for advanced users.

The new CGM filter released with Word/Office 97 was developed by HSI, to provide enhanced functionality over the CGM filter that was released with Office 95. These enhancements in particular have enabled this filter to be certified by NIST as complete and correct.

The new filter represents an Office-compatible version of earlier HSI CGM filters that were released with MS SGML Author 1.0 for Word 6 and MS SGML Author 1.1 for Word 95 (and they are released commercially directly by HSI as well).

Filter Options

The HSI *cgmimp* filter, for CGM import into Microsoft Windows applications, has a number of options available for the advanced user, to fine-tune the translation and import of CGM.

The filter (along with its companion CGM export filter) was first released with the product SGML Author 1.0 for Word 6.x (Windows 3.1) and then SGML Author 1.1 for Word 95. In the releases for Windows 3.1, the initial and save values for filter options are provided in a file called CGM.INI. In Windows 95 releases, the Registry is used instead.

In the remainder of this HELP, any reference to Registry is equivalent to CGM.INI for Windows 3.1 (16-bit) releases of the filters.

The ability of the user to access the visual dialog for the options is controlled by one of the options (OptionDialog) and the interaction of the filter with the calling application. It is possible to force the visual dialog to happen by editing this option in the Registry.

The *cgmimp* filter has the following options to control its translation of CGM to WMF.

The “[default]” designation in the following indicates the startup configuration of the filters as shipped. If the Registry does not have a setting for a given option, then the default is the value that the filter will use (unless it is changed in the visual dialog).

Note (see above) that unless the user takes actions to the contrary, each startup of the filters resumes with the option set in effect at the end of the last execution of the filters.

The number of options for *cgmimp* may seem daunting at first. Fortunately, for complete and correct CGMs you should not have to worry about the options very much at all. For incomplete CGMs, or those making certain common errors and omissions, the options may help you obtain useful results from the file.

Some of the options are provided to tune the installation, and can generally be ignored by most users who are happy with the defaults. Many of the options are intended to provide optimization and tuning of the functional mappings, or tradeoffs between picture goodness and speed, etc., and will only be of interest to advanced users.

The *cgmimp* options are arranged in a number of “pages”, with the page selector down the left of the dialog box. Where two option names exist, the first is the option name displayed in the dialog box and the second in {} is the name as it appears in the Registry.

Basic Page

SaveOptions

Save options, as defined at the end of the visual dialog, back into the Registry upon exiting the visual options dialog (via OK button).

True	Do it [default].
False	Don't.

BackgroundMode

How to handle background color information (or lack thereof) in the CGM, and what to put into the WMF:

Insert	Use the CGM background colour, insert WMF background rectangle.
Substitute	Substitute the value of the *BackgroundColor* option for the CGM background color. The user will be presented with a color-selection dialog to choose the new background color.
Suppress	Ignore CGM background color, suppress generation of WMF background rectangle [default].

Some CGM applications make a fundamental mistake in defining pictures, which can lead to white drawing on white background, or black on black. This option allows you to correct for such erroneous CGMs. If the import has apparently done some work, and you see a white or black screen, this may be the problem. It may also be useful in some cases (e.g. some kinds of clip art) to overlay the CGM picture without its opaque background.

Note — This Microsoft Office release of cgmimp *uses the value "Suppress" as default. While this may be unusual or undesirable for some CGM applications, it is considered to be the most useful setting for the common Office activity of inserting clip art into documents.*

Resources Page

Directory {ResourceDirectory}

Pathname to directory in which resource files for *cgmimp* will be found
[default: <install_directory>\hsiapps].

Font Configuration {FontMap}

Specify font mapping/substitution file [default: <ResourceDirectory>\hsi_font.cfg].

This is one of the most important options for getting good results with CGM to WMF translation. The *cgmimp* filter will try to translate each incoming font request into one that Windows can understand. It also allows *cgmimp* to compensate for the lack of a Font List element in a metafile, and allows some adjustment of the average font metrics in the mapping. The hsi_font.cfg file (the default mapping) is self documenting. If you have unusual fonts in your CGM, you will want to modify this file, or better yet, define a separate font mapping file for each class of metafile that you deal with.

See the end of this section for a complete example.

LogFile

Name of logfile (record of translation, errors, diagnostics) to use [default: <ResourceDirectory>\hsi_errs.log].

{WMFToDisk}

Controls whether *cgmimp* will write the WMF to disk, or just into memory for viewing.

True	Do it.
False	Don’t write disk file [default].

{WMFFileName}

Name of disk resident output WMF to create [default: *.wmf].

Picture Page

Picture

Selects which picture number to interpret, in multi-picture metafiles. The picture number must be valid for the metafile, or else an error is diagnosed [default: 1].

Metafiles are seen with multiple pictures, although they are rare. A tool such as MetaCheck™ can tell you if you have a multiple picture metafile. In the future, *cgmimp* may generate an informational message that more pictures remain (when picture #1 is selected). Some older metafiles have been seen with an empty first picture and a good second picture.

An error will be logged if the selected picture does not exist in the metafile (e.g., selecting picture #2 in a 1-picture metafile). The user must be aware of the SaveOptions feature — selecting picture #6 for a translation will result in this being the default picture number for the next translation.

Restrict Size

If the CGM Scaling Mode is 'abstract', a WMF picture size of 4x4 inches will be selected in any case — this option has no effect (and CGM pictures are mapped to 4x4 inch WMF pictures). If the CGM scaling mode is 'metric', then:

False do not constrain the WMF picture size if the CGM picture is 'metric' [default].

True if CGM picture is larger than 4x4 inches, then scale it down 4x4 inches in the WMF file.

CGMs that are 'abstract' have no inherent size — they are correctly displayed at any size. Those that are 'metric' have inherent physical size. This option allows the user to either preserve that scaling in mapping to WMF (which could lead to the attempt to put an E-size drawing on your screen), or to suppress the inherent size/scaling and translate the CGM to a 4x4 inch WMF.

Preserve Coord Range {PreserveVDC}

Whether or not to exactly preserve the range of the input CGM coordinates in the output WMF.

False scale up the coordinate range [default].

True preserve input coordinates on output.

Even CGMs which use low-precision integer coordinates exhibit a wide variation in the range of the coordinates in the input pictures. Coordinate ranges of 1000 or less are not unusual. This can lead to some less-than-optimal results when, for example, text or line simulations are selected. Therefore the default mode of operation of HSIview is to boost the input coordinate range to create a WMF with higher precision (around 16,000). This option allows that behavior to be surpressed, for those cases where it is desired to retain the exact input coordinate range for output.

Clip Horizontal Adjust Factor {HorizontalClipAdjust}

An epsilon to add/subtract to the width of Windows clip rectangles. Positive values increase to the size (subtract on left and add on right), and negative values decrease the size (add on left and subtract on right). This is a real number in the range -1.00 to +1.00, which is interpreted as a fraction of the VDC extent size. The “VDC extent size” is taken to be the length of the longest side of the VDC extent [default: 0.0].

The problem alluded to in the discussion of RectangleMode also pertains to WMF clipping (at least rectangular clipping). Perfectly valid CGMs may lose lines on the right and bottom of rectangular areas, and the exact result is dependent on the particular display device. This option allows the *cgmimp* user to compensate for this behavior of WMF/GDI.

Clip Vertical Adjust Factor {VerticalClipAdjust}

Epsilon to add/subtract to the height of Windows clip rectangles. Positive values increase to the size (subtract on bottom and add on top), and negative values decrease the size (add on bottom and subtract on top). This is a real number in the range -1.00 to +1.00, which is interpreted as a fraction of the VDC extent size. The “VDC extent size” is taken to be the length of the longest side of the VDC extent [default: 0.0].

See also Clip Horizontal Adjust Factor.

Text Page

SimulateText

Whether or not to simulate text, when the incoming font is a font for which this is possible. The default is to map to a TrueType font (with average metrics adjustment, see hsi_fonts.cfg). But *cgmimp* has a text simulator built in, for those cases where absolute accuracy is needed (at the cost of turning CGM Text elements into WMF lines).

False Use only TrueType substitute [default].

True Perform conversion to lines/polygons if possible.

CGM Character Set {DefaultCharSet}

Tells *cgmimp* to assume that the character set is:

ISOLatin1 [default].

ANSI Windows' "augmented" version of ISOLatin1.

OEM sometimes encountered in CGMs from other Windows applications.

If characters like "bullet" are expected in the display of the CGM, and are not appearing, then it is likely that the generator of the CGM passed OEM or ANSI characters directly into the CGM without proper declaration or mapping. The user can inform *cgmimp* that this is the case and sometimes recover the mismapped characters.

The application of this information is controlled by the OverRideCGMCharSet option.

Override CGM Character Set {OverRideCGMCharSet}

No Don't do it [default].

Yes Do it.

— If 'no', then any intelligible Character Set information in the CGM is used, regardless of the value of DefaultCharSet. If 'yes', then the value of the *DefaultCharSet* option overrides whatever information may be in the CGM file (i.e., the filter should assume that any information in the CGM is wrong).

WMF Character Set {MapCharSet}

Tells *cgmimp* what to do with the character set of the CGM (as established by the CGM contents itself, and the previous two options):

ANSI Map ISOLatin1 or ANSI in the input CGM, to ANSI in the output WMF [default].

OEM preserve ANSI exactly, OEM exactly.

This capability is still in development. The combination of OEM for DefaultCharSet and ANSI for MapCharSet will correct CGM which were generated with OEM codes. Other combinations may not be useful.

Restricted Text {RestrictedTextFit}

Tells *cgmimp* how to fit text to the restriction box for Restricted Text element:

BoxCap text should fit restriction box exactly in width, and the baseline to capline of the text exactly in height [default].

IsotropicCap text should fit restriction box in width or height in one dimension, with the other dimension being wholly contained, without distorting the aspect ratio of the text.

The precise meanings are as defined in CGM:1992.

Primitives Page

PatternMode

Pattern handling directive — how to map CGM Pattern Table definitions and Pattern Size to WMF:

Normal if CGM pattern is 8x8, then map straight to WMF pattern and ignore CGM pattern size; if CGM pattern is other than 8x8, then render the pattern exactly with a pattern size of max{ CGM pattern size, 1/75 VDC extent } [default].

Fast always map CGM patterns to 8x8 WMF patterns and ignore Pattern Size (risking possible partial loss of pattern integrity if the pattern is not 8x8).

Correct always render defined pattern exactly and preserve Pattern Size, unless Pattern Size appears too small or missing, in which case use 1/75 of VDC Extent.

Exact always do the defined pattern and preserve Pattern Size, regardless of whether it looks to be bogus.

One of the distinguishing capabilities of the *cgmimp* filter is its ability to accurately map CGM pattern-filled areas to WMF. WMF has very limited pattern interior capabilities compared to CGM. In the case that the CGMs came from a Windows graphics application, then it is likely that the straight mapping will produce accurate results. In other cases (e.g., some complex PIP applications use 32x32 patterns), the filter's pattern simulation will give accurate results. However, this simulation is very expensive in terms of both speed and memory. The two filters are able to "round-trip" in all pattern cases — CGM to WMF to CGM should produce the same CGM.

LineMode

Whether to render lines exactly, possibly requiring line type simulation, or to use WMF lines.

Never Always use the best approximation obtainable with WMF line attributes (which may be very far off in the case of some extended, CALS, linetypes).

Conditional Simulate if necessary, to accurately map the exact combination of CGM line type and line width to WMF, but otherwise use WMF line attributes if possible [default].

Always Always simulate to provide the most accurate rendering of the lines.

WMF only supports dashed lines at the nominal width, and only supports wide lines if they are solid. It does not support such line types as the CALS engineering line types. The *cgmimp* filter contains a linetype simulator that will produce exact results in all cases. But it is expensive in execution time, it creates larger WMF files, and it degenerates single CGM line segments into multiple WMF pieces.

The recommended setting is "conditional", which will give accurate results but use WMF lines when possible.

EdgeMode

Whether to render edges of filled areas (polygons, circles, etc) exactly, possibly requiring edge type simulation, or to use WMF edges.

Never	Always use the best approximation obtainable with WMF line attributes (which may be very far off in the case of some extended, CALS, linetypes).
Conditional	Simulate if necessary, to accurately map the exact combination of CGM line type and line width to WMF, but otherwise use WMF line attributes if possible [default].
Always .	Always simulate to provide the most accurate rendering of the lines

Use WMF Rectangles {Rectangles}

Whether to map CGM Rectangle to WMF Rectangle, or simulate CGM Rectangles with WMF Polygons.

On	Map to WMF rectangle.
Off	Map to WMF polygons [default].

WMF rectangles have some unusual rules regarding the extent of the interior with respect to the 4 edges. While it is desirable to preserve the identity of CGM rectangles, and more efficient, it also may lead to inaccurate results in some cases. Therefore WMF rectangles are avoided by default. But this option allows the user to force WMF rectangles to be generated.

Insert WMF Window {InsertWindow}

Controls whether the picture window/size information of the PictInfo structure is redundantly put into the WMF in the form of SetWindowOrg and SetWindowExt. Almost all standard WMF-reading applications must have it.

False	don’t do it.
True	do it [default].

Use WMF Palette Colors {UsePaletteColors}

Determines which of two color mapping directives will be embedded in the WMF file.

True	Use PALETTERGB macro to create colors.
False	Use RGB macro [default].

If set to ‘True’, the WMF contains instructions to its renderer to *not* use color dither. If set to ‘False’, the WMF displayer is expected to do the best it can to maintain color fidelity, including dither in the case of devices with limited hardware palette.

The results obtained by selecting ‘True’ are dependent upon the particular renderer of the generated WMF. Good results are obtained with Word.

Map To 5x9x5 Colors {MapColors}

Determines the treatment of color-rich Cell Array elements.

True Enable the capability.

False Disable it [default].

If Cell Array (raster) elements occur in the CGM which have a very large number of close but distinct colors, this option should be selected. The current design limits the number of distinct colors passed on to the WMF in a raster element to 256. If this option is selected, then raster colors will be mapped to the “closest” point in a 5 Red, 9 Green, 5 Blue, and 32 Gray-level grid of color space. The disadvantage of using this option will be slight inaccuracies in the color, but advantage is avoiding the risk of exhausting the 256-color limit in some narrow portion of the color spectrum (which can give *very* distorted results).

Make Bitmaps Printable {Printable}

Causes bitmaps to be created which are not compressed or flipped.

True Enable the capability [default].

False Disable it.

Some print drivers cannot process compressed or flipped raster elements. This option allows more simple raster elements to be generated, which print drivers may find easier to process. Note that this may result in much bigger imported files.

Note — This Microsoft Office release of cgmimp *uses the value “true” as default. While this may lead to larger WMF files for big CGM raster elements, it on the other hand gives reliable results across a wide range of printers.*

CGM Opts Page

MetafileSource

Specifies information to control deflavorizing:

suppress do not do deflavorizing, even if the metafile source is recognized as supported for deflavorizing.

default the metafile source is recognized as one in the set that *cgmimp* can deflavorize, then deflavorize the metafile during translation [default].

<prod_id> *future values, not yet implemented* — regardless of whether or not *cgmimp* recognizes the metafile source, use the prod_id identifier string to select the deflavorizing to be applied to the metafile.

In this release, metafiles from some selected older versions of the Harvard Graphics and Lotus Freelance products are supported in deflavorizing, plus metafiles from CorelDRAW! and HiJaak. Others will be added according to client demand. The "<prod_id>" values are not yet usefully implemented.

Specific deflavorizations performed when value is "default":

(1) — CorelDRAW! (1.0) Some CorelDRAW! CGM pictures (self-identifying as 1.0) are specified upside down. Deflavorization: invert such pictures.

(1) — CorelDRAW! (6.0) CorelDRAW! CGM pictures (self-identifying as 6.0) have too much white space within the picture frame. Deflavorization: strip the white space out.

(1) — HiJaak (2.0) As for CorelDRAW! (6.0).

Dialogs Page

Display

Selects whether the visual display generated by the filter consists only of the progress/cancel window, or whether the error window is included:

All Display both a progress and an error dialog [default].

Progress display only progress window.

It is not possible, in this version, to turn off the display of the progress window. It is needed by the filter for certain critical operations.

OptionDialog

Option dialog display.

Never Never show the visual options dialog.

Always Always show it on each *cgmimp* invocation.

Requested Only show it if the calling application (e.g. Word) requests it via one of the standard filter API functions (GetFilterPref).

ShowResources

A diagnostic/debug option, used to help debug system resource problems (in cooperation with HSI customer support).

True Show resources.

False Don’t show resources [default].

ShowErrors

Indicates if errors should be displayed in a dialog box and written to the log file, or only written to the log file.

True Show errors.

False Log errors only [default].

Deflavorizing

There is minimal support for deflavorizing (correcting the systematic mistakes made by some CGM generating applications, by automatically recognizing the source generator). More support will be added as demand dictates. HSI will do custom deflavorizing for important interchange sources of filter clients. See the MetafileSource option for more details.

Font Substitution File Format

The user is able to correct for unfamiliar fonts declared in the CGM Font List, or even lack of a CGM font list, using the font substitution mechanism. This involves an external file — the font configuration file — which is used to direct the mapping of unsupported or missing font declarations in the CGM to fonts that are locally available to the application.

The font configuration file consists of a series of statements, one per line. Each statement consists of a command, an argument list, and a font name string. The command, argument list, and font name string are separated by white space (blanks or tabs). If an argument list contains more than one argument, the arguments are separated by commas and optional white space. The font name string must be quoted. Lines beginning with a “#” in column 1 are ignored.

There are three commands: index, keyword, match, and default. The argument to the index keyword is a single integer. If the font index is equal to this integer, then the corresponding font name is used.

The argument to the keyword command is a list of strings to be searched for in the CGM font name. If all of the strings are found, then the corresponding font name is used.

The argument to the match command is a single quoted string. If the CGM font name is identical (ignoring case) to this string, then the corresponding font name is used.

The default keyword does not take an argument list. The font name string that is listed will be used if no other statements match the CGM font name.

In searching for a font name to use, the font configuration file is scanned from first statement to last. As soon as a statement matching the CGM font specification is found, the search terminates. The order of processing implies that the order of statements in the file must be from most specific to least specific.

Two additional fields may be added to the end of each command line. These consist of scale factors in the base and up vector directions respectively, to be applied to any text to which the substitution command applies. These fields are optional, but if one is present then both must be. If these fields are not present then no scaling is applied.

```
# HSI CGM Font Configuration File.
# --------------------------------
#
# Command syntax:
#     command arguments     “font_name_string”
#
#     commands are:
#         index n             “font_name_string”
#         match “string”      “font_name_string”
#           keyword word [, word, ...] “font_name_string”
#
# ----------------------------------------------------------------
#
#   The font name string provides an index into an internal table
#           1 - Courier New”, FW_NORMAL
#           2 Courier New”, FW_BOLD
#           3 Courier New”, FW_NORMAL,italic
#           4 Courier New”, FW_BOLD, italic
#           5 Arial”, FW_NORMAL
#           6 Arial”, FW_BOLD
#           7 Arial”, FW_NORMAL, italic
#           8 Arial”, FW_BOLD, italic
#           9 Times New Roman”   FW_NORAML
#           10 Times New Roman” FW_BOLD
#           11 Times New Roman”, FW_NORMAL, italic
#           12 Times New Roman”, FW_BOLD, italic
#           13 Symbol”, FW_NORMAL,
#           14 Wingdings”, FW_NORMAL
#           15 Script”, FW_NORMAL
#
#
keyword HERSHEY, CARTOGRAPHIC, ROMAN  “5”    1.19 1.0
keyword HERSHEY, CARTOGRAPHIC, GREEK  “13”   1.0  1.0
keyword HERSHEY, SIMPLEX, ROMAN       “5”    1.19 1.0
keyword HERSHEY, SIMPLEX, GREEK       “13”   1.0  1.0
keyword HERSHEY, SIMPLEX, SCRIPT      “11”   1.43 1.0
keyword HERSHEY, COMPLEX, ROMAN       “9”    1.1  1.0
keyword HERSHEY, COMPLEX, GREEK       “13”   1.0  1.0
keyword HERSHEY, COMPLEX, SCRIPT      “12”   1.39 1.0
keyword HERSHEY, COMPLEX, ITALIC      “11”   1.43 1.0
keyword HERSHEY, COMPLEX, CYRILLIC    “1”    0.91 1.0
keyword HERSHEY, DUPLEX, ROMAN        “6”    1.1  1.0
keyword HERSHEY, TRIPLEX, ROMAN       “10”   1.33 1.0
keyword HERSHEY, TRIPLEX, ITALIC      “12”   1.38 1.0
keyword HERSHEY, GOTHIC, GERMAN       “1”    0.91 1.0
keyword HERSHEY, GOTHIC, ENGLISH      “1”    0.91 1.0
keyword HERSHEY, GOTHIC, ITALIAN      “1”    0.91 1.0
keyword HERSHEY, SYMBOL, SET1         “13”   1.0  1.0
keyword HERSHEY, SYMBOL, SET2         “13”   1.0  1.0
keyword HERSHEY, SYMBOL, MATH         “13”   1.0  1.0
keyword HERSHEY, SYMBOL               “13”   1.0  1.0
#
# Handle a few generic keywords
#
keyword Sans, Serif, Bold, Italic            “8”
keyword Sans, Serif, Bold, Oblique              “8”
keyword Sans, Serif, Bol                        “6”
keyword Sans, Serif, Italic                     “7”
keyword Sans, Serif, Oblique            “7”
keyword Sans, Serif                     “5”
#
keyword Serif, Bold, Italic             “12”
keyword Serif, Bold, Oblique “12”
keyword Serif, Italic           “11”
keyword Serif, Oblique                  “11”
keyword Serif, Bold             “10”
keyword Serif                   “9”
#
keyword   Courier, Bold, Italic         “4”
keyword   Courier, Bold, Oblique        “4”
keyword   Courier, Bold                 “2”
keyword   Courier, Italic               “3”
keyword   Courier, Oblique    “3”
keyword   Courier                       “1”
#
keyword   Times, Bold, Italic “12”
keyword   Times, Bold, Oblique          “12”
keyword   Times, Bold                   “10”
keyword   Times, Italic                 “11”
keyword   Times, Oblique                “11”
keyword   Times                         “9”
#
keyword   Helvetica, Narrow, Bold, Italic       “8”
keyword   Helvetica, Narrow, Bold, Oblique “8”
keyword   Helvetica, Narrow, Bold               “6”
keyword   Helvetica, Narrow, Italic             “7”
keyword   Helvetica, Bold, Italic               “8”
keyword   Helvetica, Bold, Oblique              “8”
keyword   Helvetica, Oblique            “7”
keyword   Helvetica, Bold                       “6”
keyword   Helvetica, Italic                     “7”
keyword   Helvetica, Oblique            “7”
keyword   Helvetica, Narrow             “5”
keyword   Helvetica                     “5”
#
keyword AvantGarde, Bold, Italic                “8”
keyword AvantGarde, Bold, Oblique “8”
keyword AvantGarde, Bold                “6”
keyword AvantGarde, Italic              “7”
keyword AvantGarde, Oblique                     “7”
keyword AvantGarde                      “7”
```

```
#
keyword Bookman, Bold, Italic           “8”
keyword Bookman, Bold, Oblique          “8”
keyword Bookman, Bold                   “6”
keyword Bookman, Italic                 “7”
keyword Bookman, Oblique         “7”
keyword Bookman                  “5”
#
keyword LucidaSans, Bold, Italic        “12”
keyword LucidaSans, Bold, Oblique “12”
keyword LucidaSans, Bold         “10”
keyword LucidaSans, Italic       “11”
keyword LucidaSans, Oblique             “11”
keyword LucidaSans               “9”
#
keyword NewCenturySchlbk, Bold, Italic  “12”
keyword NewCenturySchlbk, Bold, Oblique “12”
keyword NewCenturySchlbk, Bold          “10”
keyword NewCenturySchlbk, Italic        “11”
keyword NewCenturySchlbk, Oblique       “11”
keyword NewCenturySchlbk         “9”
#
keyword Palatino, Bold, Italic          “8”
keyword Palatino, Bold, Oblique         “8”
keyword Palatino, Bold                  “6”
keyword Palatino, Italic                “7”
keyword Palatino, Oblique               “7”
keyword Palatino                        “5”
#
keyword Symbol                   “13”
#
keyword ZapfChancery             “14”
#
keyword ZapfDingbats             “14”
#
#      use Times for fonts
#      with no recognized type family name
#
keyword   Bold, Italic           “12”
keyword   Bold, Oblique          “12”
keyword   Bold                   “10”
keyword   Italic                 “11”
keyword   Oblique         “11”
#
#
```

keyword 10	“10”
keyword 11	“11”
keyword 12	“12”
keyword 13	“13”
keyword 14	“14”
keyword 15	“15”
keyword 1	“1”
keyword 2	“2”
keyword 3	“3”
keyword 4	“4”
keyword 5	“5”
keyword 6	“6”
keyword 7	“7”
keyword 8	“8”
keyword 9	“9”
#	
#	
#	
index 1	“5”
index 2	“7”
index 3	“6”
index 4	“8”
index 5	“5”
index 6	“7”
index 7	“6”
index 8	“8”

Selects basic options: saving options and controlling background color.

Selects resource options: location of support files.

Selects debug options: various test options.

Selects picture options: coordinate control; clipping rigidity; picture size limitation.

Selects text options: simulation; style of restricted text; character sets.

Selects primitive element options: line & edge controls; pattern simulation; color control.

Selects CGM options: deflavourizing.

Controls display of dialogs.

Check this box to save the options between sessions.

Controls background color options.

Selects background mode:
Insert - Use the background element if present.
Suppress - Do not interpret background color element.
Substitute - Use the color selected here as background.

Opens a dialog to allow selection of background color.

Specifies the full path of the resource directory.

Specifies the name of the font mapping file.

Selects name of file to which status messages are written.

Specifies WMF disk filename.

Specifies which picture in the CGM is to be imported.

Check this to limits the size of metric pictures to 4x4 inches.

Check this to preserve CGM coordinates in imported file.

Specifies percent of VDC to add to clipping area in horizontal direction.

Specifies percent of VDC to add to clipping area in vertical direction.

Check this to replace text with lines or polygons.

Selects restricted text type.

When checked, the character set element in the CGM is ignored.

Selects CGM character set.

Selects WMF character set.

Selects pattern mode:
Normal - if the pattern is 8x8, ignore CGM pattern size; otherwise render the pattern accurately with a pattern size of max { CGM pat size, 1/75 VDC extent}.
Fast - always map CGM pattern to 8x8, ignore pattern size.
Correct - always render the pattern accurately with a pattern size of max { CGM pat size, 1/75 VDC extent}.
Exact - always render the pattern accurately.

Selects edge rendering mode:
Never simulate - use native WMF edges.
Conditional - use native when it works accurately, otherwise break the edge into line segments to get accurate appearance.
Always simulate - always break the edge into line segments to get accurate appearance.

Selects line rendering mode:
Never simulate - use native WMF lines.
Conditional - use native when it works accurately, otherwise break the line into line segments to get accurate appearance.
Always simulate - always break the line into line segments to get accurate appearance.

When checked, inserts calls to SetWondowOrg, SetWindowExt.

When checked, renders rectangles as polygons.

When checked, uses color mapping of PALETTERGB instead of RGB.

When checked, maps colors to 5x9x5 color grid points.

Selects source of CGM file, as an aid to “deflavourizing”.

Selects what dialogs are displayed:

When checked, displays system resources in the progress dialog.

When not checked, disables display of interpreter errors.

Controls display of options dialog:
Always - always displays the options dialog when the import filter in invoked.
Never - never displays the options dialog.
Requested - displays the options dialog when asked to do so through the standard API.