

libemf

1.0.9

Generated on Tue Apr 14 2026 14:55:53 for libemf by Doxygen 1.15.0

Tue Apr 14 2026 14:55:53

<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Data Structure Index</b>	<b>3</b>
2.1 Data Structures	3
<b>3 File Index</b>	<b>7</b>
3.1 File List	7
<b>4 Data Structure Documentation</b>	<b>7</b>
4.1 EMF::BRUSH Class Reference	7
4.1.1 Detailed Description	8
4.1.2 Constructor & Destructor Documentation	8
4.1.3 Member Function Documentation	9
4.2 EMF::BYTEARRAY Struct Reference	9
4.2.1 Detailed Description	10
4.2.2 Constructor & Destructor Documentation	10
4.3 EMF::CHARSTR Struct Reference	10
4.3.1 Detailed Description	11
4.3.2 Constructor & Destructor Documentation	11
4.4 EMF::DATASTREAM Class Reference	11
4.4.1 Detailed Description	13
4.4.2 Constructor & Destructor Documentation	13
4.4.3 Member Function Documentation	13
4.5 EMF::DWORDARRAY Struct Reference	34
4.5.1 Detailed Description	34
4.5.2 Constructor & Destructor Documentation	35
4.6 EMF::EMRARC Class Reference	35
4.6.1 Detailed Description	36
4.6.2 Constructor & Destructor Documentation	36
4.6.3 Member Function Documentation	37
4.7 EMF::EMRARCTO Class Reference	37
4.7.1 Detailed Description	38
4.7.2 Constructor & Destructor Documentation	38
4.7.3 Member Function Documentation	39
4.8 EMF::EMRBEGINPATH Class Reference	40
4.8.1 Detailed Description	40
4.8.2 Constructor & Destructor Documentation	40
4.8.3 Member Function Documentation	41
4.9 EMF::EMRCLOSEFIGURE Class Reference	42
4.9.1 Detailed Description	42
4.9.2 Constructor & Destructor Documentation	42
4.9.3 Member Function Documentation	43

4.10 EMF::EMRCREATEBRUSHINDIRECT Class Reference	44
4.10.1 Detailed Description	44
4.10.2 Constructor & Destructor Documentation	44
4.10.3 Member Function Documentation	45
4.11 EMF::EMRCREATEPALETTE Class Reference	46
4.11.1 Detailed Description	46
4.11.2 Constructor & Destructor Documentation	46
4.11.3 Member Function Documentation	47
4.12 EMF::EMRCREATEPEN Class Reference	48
4.12.1 Detailed Description	48
4.12.2 Constructor & Destructor Documentation	48
4.12.3 Member Function Documentation	49
4.13 EMF::EMRDELETEOBJECT Class Reference	50
4.13.1 Detailed Description	50
4.13.2 Constructor & Destructor Documentation	50
4.13.3 Member Function Documentation	51
4.14 EMF::EMRELIPSE Class Reference	52
4.14.1 Detailed Description	52
4.14.2 Constructor & Destructor Documentation	52
4.14.3 Member Function Documentation	53
4.15 EMF::EMRENDPATH Class Reference	54
4.15.1 Detailed Description	54
4.15.2 Constructor & Destructor Documentation	55
4.15.3 Member Function Documentation	55
4.16 EMF::EMREOF Class Reference	56
4.16.1 Detailed Description	57
4.16.2 Constructor & Destructor Documentation	57
4.16.3 Member Function Documentation	57
4.17 EMF::EMREXTCREATEFONTINDIRECTW Class Reference	58
4.17.1 Detailed Description	59
4.17.2 Constructor & Destructor Documentation	59
4.17.3 Member Function Documentation	59
4.18 EMF::EMREXTCREATEPEN Class Reference	60
4.18.1 Detailed Description	61
4.18.2 Constructor & Destructor Documentation	61
4.18.3 Member Function Documentation	62
4.19 EMF::EMREXTTEXTOUTA Class Reference	63
4.19.1 Detailed Description	63
4.19.2 Constructor & Destructor Documentation	63
4.19.3 Member Function Documentation	64
4.20 EMF::EMREXTTEXTOUTW Class Reference	65
4.20.1 Detailed Description	66

4.20.2 Constructor & Destructor Documentation	66
4.20.3 Member Function Documentation	67
4.21 EMF::EMRFILLPATH Class Reference	68
4.21.1 Detailed Description	68
4.21.2 Constructor & Destructor Documentation	68
4.21.3 Member Function Documentation	69
4.22 EMF::EMRLINETO Class Reference	70
4.22.1 Detailed Description	70
4.22.2 Constructor & Destructor Documentation	70
4.22.3 Member Function Documentation	71
4.23 EMF::EMRMODIFYWORLDTRANSFORM Class Reference	72
4.23.1 Detailed Description	72
4.23.2 Constructor & Destructor Documentation	72
4.23.3 Member Function Documentation	73
4.24 EMF::EMRMOVETOEX Class Reference	74
4.24.1 Detailed Description	74
4.24.2 Constructor & Destructor Documentation	74
4.24.3 Member Function Documentation	75
4.25 EMF::EMRPOLYBEZIER Class Reference	76
4.25.1 Detailed Description	76
4.25.2 Constructor & Destructor Documentation	76
4.25.3 Member Function Documentation	77
4.26 EMF::EMRPOLYBEZIER16 Class Reference	78
4.26.1 Detailed Description	79
4.26.2 Constructor & Destructor Documentation	79
4.26.3 Member Function Documentation	80
4.27 EMF::EMRPOLYBEZIERTO Class Reference	81
4.27.1 Detailed Description	81
4.27.2 Constructor & Destructor Documentation	82
4.27.3 Member Function Documentation	83
4.28 EMF::EMRPOLYBEZIERTO16 Class Reference	83
4.28.1 Detailed Description	84
4.28.2 Constructor & Destructor Documentation	84
4.28.3 Member Function Documentation	85
4.29 EMF::EMRPOLYGON Class Reference	86
4.29.1 Detailed Description	87
4.29.2 Constructor & Destructor Documentation	87
4.29.3 Member Function Documentation	88
4.30 EMF::EMRPOLYGON16 Class Reference	89
4.30.1 Detailed Description	89
4.30.2 Constructor & Destructor Documentation	90
4.30.3 Member Function Documentation	91

4.31 EMF::EMRPOLYLINE Class Reference . . . . .	92
4.31.1 Detailed Description . . . . .	92
4.31.2 Constructor & Destructor Documentation . . . . .	92
4.31.3 Member Function Documentation . . . . .	93
4.32 EMF::EMRPOLYLINE16 Class Reference . . . . .	94
4.32.1 Detailed Description . . . . .	94
4.32.2 Constructor & Destructor Documentation . . . . .	95
4.32.3 Member Function Documentation . . . . .	96
4.33 EMF::EMRPOLYLINETO Class Reference . . . . .	97
4.33.1 Detailed Description . . . . .	97
4.33.2 Constructor & Destructor Documentation . . . . .	97
4.33.3 Member Function Documentation . . . . .	98
4.34 EMF::EMRPOLYLINETO16 Class Reference . . . . .	99
4.34.1 Detailed Description . . . . .	99
4.34.2 Constructor & Destructor Documentation . . . . .	100
4.34.3 Member Function Documentation . . . . .	101
4.35 EMF::EMRPOLYPOLYGON Class Reference . . . . .	102
4.35.1 Detailed Description . . . . .	102
4.35.2 Constructor & Destructor Documentation . . . . .	102
4.35.3 Member Function Documentation . . . . .	103
4.36 EMF::EMRPOLYPOLYGON16 Class Reference . . . . .	104
4.36.1 Detailed Description . . . . .	105
4.36.2 Constructor & Destructor Documentation . . . . .	105
4.36.3 Member Function Documentation . . . . .	106
4.37 EMF::EMRRECTANGLE Class Reference . . . . .	107
4.37.1 Detailed Description . . . . .	108
4.37.2 Constructor & Destructor Documentation . . . . .	108
4.37.3 Member Function Documentation . . . . .	109
4.38 EMF::EMRRESTOREDC Class Reference . . . . .	109
4.38.1 Detailed Description . . . . .	110
4.38.2 Constructor & Destructor Documentation . . . . .	110
4.38.3 Member Function Documentation . . . . .	111
4.39 EMF::EMRSAVEDC Class Reference . . . . .	111
4.39.1 Detailed Description . . . . .	112
4.39.2 Constructor & Destructor Documentation . . . . .	112
4.39.3 Member Function Documentation . . . . .	113
4.40 EMF::EMRSCALEVIEWPORTEXTEX Class Reference . . . . .	113
4.40.1 Detailed Description . . . . .	114
4.40.2 Constructor & Destructor Documentation . . . . .	114
4.40.3 Member Function Documentation . . . . .	115
4.41 EMF::EMRSCALEWINDOWEXTEx Class Reference . . . . .	116
4.41.1 Detailed Description . . . . .	116

4.41.2 Constructor & Destructor Documentation	116
4.41.3 Member Function Documentation	117
4.42 EMF::EMRSELECTOBJECT Class Reference	118
4.42.1 Detailed Description	118
4.42.2 Constructor & Destructor Documentation	119
4.42.3 Member Function Documentation	119
4.43 EMF::EMRSETBKCOLOR Class Reference	120
4.43.1 Detailed Description	121
4.43.2 Constructor & Destructor Documentation	121
4.43.3 Member Function Documentation	121
4.44 EMF::EMRSETBKMODE Class Reference	122
4.44.1 Detailed Description	123
4.44.2 Constructor & Destructor Documentation	123
4.44.3 Member Function Documentation	124
4.45 EMF::EMRSETMAPMODE Class Reference	124
4.45.1 Detailed Description	125
4.45.2 Constructor & Destructor Documentation	125
4.45.3 Member Function Documentation	126
4.46 EMF::EMRSETMETARGN Class Reference	126
4.46.1 Detailed Description	127
4.46.2 Constructor & Destructor Documentation	127
4.46.3 Member Function Documentation	128
4.47 EMF::EMRSETMITERLIMIT Class Reference	128
4.47.1 Detailed Description	129
4.47.2 Constructor & Destructor Documentation	129
4.47.3 Member Function Documentation	130
4.48 EMF::EMRSETPIXELV Class Reference	130
4.48.1 Detailed Description	131
4.48.2 Constructor & Destructor Documentation	131
4.48.3 Member Function Documentation	132
4.49 EMF::EMRSETPOLYFILLMODE Class Reference	133
4.49.1 Detailed Description	133
4.49.2 Constructor & Destructor Documentation	133
4.49.3 Member Function Documentation	134
4.50 EMF::EMRSETTEXTALIGN Class Reference	135
4.50.1 Detailed Description	135
4.50.2 Constructor & Destructor Documentation	135
4.50.3 Member Function Documentation	136
4.51 EMF::EMRSETTEXTCOLOR Class Reference	137
4.51.1 Detailed Description	137
4.51.2 Constructor & Destructor Documentation	137
4.51.3 Member Function Documentation	138

4.52 EMF::EMRSETVIEWPORTEXTEX Class Reference	139
4.52.1 Detailed Description	139
4.52.2 Constructor & Destructor Documentation	139
4.52.3 Member Function Documentation	140
4.53 EMF::EMRSETVIEWPORTORGEX Class Reference	141
4.53.1 Detailed Description	141
4.53.2 Constructor & Destructor Documentation	141
4.53.3 Member Function Documentation	142
4.54 EMF::EMRSETWINDOWTEX Class Reference	143
4.54.1 Detailed Description	143
4.54.2 Constructor & Destructor Documentation	143
4.54.3 Member Function Documentation	144
4.55 EMF::EMRSETWINDOWORGEX Class Reference	145
4.55.1 Detailed Description	145
4.55.2 Constructor & Destructor Documentation	145
4.55.3 Member Function Documentation	146
4.56 EMF::EMRSETWORLDTRANSFORM Class Reference	147
4.56.1 Detailed Description	147
4.56.2 Constructor & Destructor Documentation	147
4.56.3 Member Function Documentation	148
4.57 EMF::EMRSTROKEANDFILLPATH Class Reference	149
4.57.1 Detailed Description	149
4.57.2 Constructor & Destructor Documentation	149
4.57.3 Member Function Documentation	150
4.58 EMF::EMRSTROKEPATH Class Reference	151
4.58.1 Detailed Description	151
4.58.2 Constructor & Destructor Documentation	151
4.58.3 Member Function Documentation	152
4.59 EMF::ENHMETAHEADER Class Reference	153
4.59.1 Detailed Description	153
4.59.2 Constructor & Destructor Documentation	153
4.59.3 Member Function Documentation	154
4.60 EMF::EXTPEN Class Reference	155
4.60.1 Detailed Description	156
4.60.2 Constructor & Destructor Documentation	156
4.60.3 Member Function Documentation	156
4.61 EMF::FONT Class Reference	157
4.61.1 Detailed Description	158
4.61.2 Constructor & Destructor Documentation	158
4.61.3 Member Function Documentation	158
4.62 EMF::GLOBALOBJECTS Class Reference	159
4.62.1 Detailed Description	161

4.62.2 Member Function Documentation . . . . .	162
4.63 EMF::GRAPHICSOBJECT Class Reference . . . . .	163
4.63.1 Detailed Description . . . . .	164
4.63.2 Member Function Documentation . . . . .	164
4.63.3 Field Documentation . . . . .	165
4.64 EMF::INTARRAY Struct Reference . . . . .	165
4.64.1 Detailed Description . . . . .	165
4.64.2 Constructor & Destructor Documentation . . . . .	166
4.65 EMF::METAFILEDEVICECONTEXT Class Reference . . . . .	166
4.65.1 Detailed Description . . . . .	168
4.65.2 Constructor & Destructor Documentation . . . . .	168
4.65.3 Member Function Documentation . . . . .	169
4.65.4 Field Documentation . . . . .	170
4.66 EMF::METARECORD Class Reference . . . . .	172
4.66.1 Detailed Description . . . . .	172
4.66.2 Constructor & Destructor Documentation . . . . .	173
4.66.3 Member Function Documentation . . . . .	173
4.67 EMF::OBJECT Class Reference . . . . .	174
4.67.1 Detailed Description . . . . .	175
4.67.2 Constructor & Destructor Documentation . . . . .	175
4.67.3 Member Function Documentation . . . . .	175
4.67.4 Field Documentation . . . . .	175
4.68 EMF::PADDING Struct Reference . . . . .	176
4.68.1 Detailed Description . . . . .	176
4.68.2 Constructor & Destructor Documentation . . . . .	176
4.69 EMF::PALETTE Class Reference . . . . .	177
4.69.1 Detailed Description . . . . .	177
4.69.2 Constructor & Destructor Documentation . . . . .	178
4.69.3 Member Function Documentation . . . . .	178
4.70 EMF::PEN Class Reference . . . . .	179
4.70.1 Detailed Description . . . . .	179
4.70.2 Constructor & Destructor Documentation . . . . .	180
4.70.3 Member Function Documentation . . . . .	180
4.71 EMF::POINT16ARRAY Struct Reference . . . . .	180
4.71.1 Detailed Description . . . . .	181
4.71.2 Constructor & Destructor Documentation . . . . .	181
4.72 EMF::POINTLARRAY Struct Reference . . . . .	181
4.72.1 Detailed Description . . . . .	182
4.72.2 Constructor & Destructor Documentation . . . . .	182
4.73 EMF::WCHARSTR Struct Reference . . . . .	182
4.73.1 Detailed Description . . . . .	183
4.73.2 Constructor & Destructor Documentation . . . . .	183

<b>5 File Documentation</b>	<b>183</b>
5.1 emf.h . . . . .	183
5.2 basetsd.h . . . . .	184
5.3 guiddef.h . . . . .	186
5.4 poppack.h . . . . .	187
5.5 pshpack2.h . . . . .	187
5.6 pshpack4.h . . . . .	188
5.7 w16.h . . . . .	188
5.8 winbase.h . . . . .	189
5.9 windef.h . . . . .	210
5.10 winerror.h . . . . .	213
5.11 wingdi.h . . . . .	235
5.12 winnt.h . . . . .	273
5.13 winuser.h . . . . .	329
5.14 libemf.h . . . . .	377

## 1 Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

<b>EMF::BYTEARRAY</b>	<b>9</b>
<b>EMF::CHARSTR</b>	<b>10</b>
<b>EMF::DATASTREAM</b>	<b>11</b>
<b>EMF::DWORDARRAY</b>	<b>34</b>
<b>EMF::GLOBALOBJECTS</b>	<b>159</b>
<b>EMF::INTARRAY</b>	<b>165</b>
<b>EMF::METARECORD</b>	<b>172</b>
<b>EMF::EMRARC</b>	<b>35</b>
<b>EMF::EMRARCTO</b>	<b>37</b>
<b>EMF::EMRBEGINPATH</b>	<b>40</b>
<b>EMF::EMRCLOSEFIGURE</b>	<b>42</b>
<b>EMF::EMRCREATEBRUSHINDIRECT</b>	<b>44</b>
<b>EMF::EMRCREATEPALETTE</b>	<b>46</b>
<b>EMF::EMRCREATEPEN</b>	<b>48</b>
<b>EMF::EMRDELETEOBJECT</b>	<b>50</b>

EMF::EMRELLIPSE	52
EMF::EMRENDPATH	54
EMF::EMREOF	56
EMF::EMREXTCREATEFONTINDIRECTW	58
EMF::EMREXTCREATEPEN	60
EMF::EMREXTTEXTOUTA	63
EMF::EMREXTTEXTOUTW	65
EMF::EMRFILLPATH	68
EMF::EMRLINETO	70
EMF::EMRMODIFYWORLDTRANSFORM	72
EMF::EMRMOVETOEX	74
EMF::EMRPOLYBEZIER	76
EMF::EMRPOLYBEZIERTO	81
EMF::EMRPOLYBEZIER16	78
EMF::EMRPOLYBEZIERTO16	83
EMF::EMRPOLYBEZIERTO	81
EMF::EMRPOLYBEZIERTO16	83
EMF::EMRPOLYGON	86
EMF::EMRPOLYGON16	89
EMF::EMRPOLYLINE	92
EMF::EMRPOLYLINE16	94
EMF::EMRPOLYLINETO	97
EMF::EMRPOLYLINETO16	99
EMF::EMRPOLYPOLYGON	102
EMF::EMRPOLYPOLYGON16	104
EMF::EMRRECTANGLE	107
EMF::EMRRESTOREDC	109
EMF::EMRSAVEDC	111
EMF::EMRSCALEVIEWPORTEXTEX	113
EMF::EMRSCALEWINDOWEXTEX	116
EMF::EMRSELECTOBJECT	118
EMF::EMRSETBKCOLOR	120

EMF::EMRSETBKMODE	122
EMF::EMRSETMAPMODE	124
EMF::EMRSETMETARGN	126
EMF::EMRSETMITERLIMIT	128
EMF::EMRSETPIXELV	130
EMF::EMRSETPOLYFILLMODE	133
EMF::EMRSETTEXTALIGN	135
EMF::EMRSETTEXTCOLOR	137
EMF::EMRSETVIEWPORTEXTEX	139
EMF::EMRSETVIEWPORTORGEX	141
EMF::EMRSETWINDOWEXTEX	143
EMF::EMRSETWINDOWORGEX	145
EMF::EMRSETWORLDTRANSFORM	147
EMF::EMRSTROKEANDFILLPATH	149
EMF::EMRSTROKEPATH	151
EMF::ENHMETAHEADER	153
EMF::OBJECT	174
EMF::GRAPHICSOBJECT	163
EMF::BRUSH	7
EMF::EXTPEN	155
EMF::FONT	157
EMF::PALETTE	177
EMF::PEN	179
EMF::METAFILEDEVICECONTEXT	166
EMF::PADDING	176
EMF::POINT16ARRAY	180
EMF::POINTLARRAY	181
EMF::WCHARSTR	182

## 2 Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<b>EMF::BRUSH</b>	
Graphics Brush	7
<b>EMF::BYTEARRAY</b>	
Represent a byte array in a simple way	9
<b>EMF::CHARSTR</b>	
Represent an ASCII character string in a simple way	10
<b>EMF::DATASTREAM</b>	
Support different endian modes when reading and writing the metafile	11
<b>EMF::DWORDARRAY</b>	
Represent an array of double word integers in a simple way	34
<b>EMF::EMRARC</b>	
EMF Arc	35
<b>EMF::EMRARCTO</b>	
EMF Arc To	37
<b>EMF::EMRBEGINPATH</b>	
EMF Begin Path	40
<b>EMF::EMRCLOSEFIGURE</b>	
EMF Close Figure	42
<b>EMF::EMRCREATEBRUSHINDIRECT</b>	
EMF Brush	44
<b>EMF::EMRCREATEPALETTE</b>	
EMF Palette	46
<b>EMF::EMRCREATEPEN</b>	
EMF Pen	48
<b>EMF::EMRDELETEOBJECT</b>	
EMF Delete Object	50
<b>EMF::EMRELLIPSE</b>	
EMF Ellipse	52
<b>EMF::EMRENDPATH</b>	
EMF End Path	54
<b>EMF::EMREOF</b>	
EMF End of File Record	56
<b>EMF::EMREXTCREATEFONTINDIRECTW</b>	
EMF Font	58
<b>EMF::EMREXTCREATEPEN</b>	
EMF Extended Pen	60
<b>EMF::EMREXTTEXTOUTA</b>	
EMF Extended Text Output ASCII	63
<b>EMF::EMREXTTEXTOUTW</b>	
EMF Extended Text Output Wide character	65
<b>EMF::EMRFILLPATH</b>	
EMF Fill path	68

<b>EMF::EMRLINETO</b>	
EMF Line To	70
<b>EMF::EMRMODIFYWORLDTRANSFORM</b>	
EMF Modify World Transform	72
<b>EMF::EMRMOVETOEX</b>	
EMF MoveTo (ex)	74
<b>EMF::EMRPOLYBEZIER</b>	
EMF Polybezier	76
<b>EMF::EMRPOLYBEZIER16</b>	
EMF Polybezier16	78
<b>EMF::EMRPOLYBEZIERTO</b>	
EMF PolyBezierTo	81
<b>EMF::EMRPOLYBEZIERTO16</b>	
EMF PolyBezierTo16	83
<b>EMF::EMRPOLYGON</b>	
EMF Filled Polygon	86
<b>EMF::EMRPOLYGON16</b>	
EMF Filled Polygon16	89
<b>EMF::EMRPOLYLINE</b>	
EMF Polyline	92
<b>EMF::EMRPOLYLINE16</b>	
EMF Polyline16	94
<b>EMF::EMRPOLYLINETO</b>	
EMF PolylineTo	97
<b>EMF::EMRPOLYLINETO16</b>	
EMF PolylineTo16	99
<b>EMF::EMRPOLYPOLYGON</b>	
EMF Poly Polygon	102
<b>EMF::EMRPOLYPOLYGON16</b>	
EMF Poly Polygon16	104
<b>EMF::EMRRECTANGLE</b>	
EMF Rectangle	107
<b>EMF::EMRRESTOREDC</b>	
EMF Restore DC	109
<b>EMF::EMRSAVEDC</b>	
EMF Save DC	111
<b>EMF::EMRSCALEVIEWPORTEXTTEX</b>	
EMF Scale Viewport Extents (ex)	113
<b>EMF::EMRSCALEWINDOWEXTTEX</b>	
EMF Scale Window Extents (ex)	116
<b>EMF::EMRSELECTOBJECT</b>	
EMF Select Object	118

<b>EMF::EMRSETBKCOLOR</b>	
EMF Set Background Color	120
<b>EMF::EMRSETBKMODE</b>	
EMF Set Background Mode	122
<b>EMF::EMRSETMAPMODE</b>	
EMF Set Mapping Mode	124
<b>EMF::EMRSETMETARGN</b>	
EMF Set Meta Region	126
<b>EMF::EMRSETMITERLIMIT</b>	
EMF SetMiterLimit	128
<b>EMF::EMRSETPIXELV</b>	
EMF Set Pixel	130
<b>EMF::EMRSETPOLYFILLMODE</b>	
EMF Set the Polygon Fill Mode	133
<b>EMF::EMRSETTEXTALIGN</b>	
EMF Set Text Alignment	135
<b>EMF::EMRSETTEXTCOLOR</b>	
EMF Set Text Color	137
<b>EMF::EMRSETVIEWPORTEXTEX</b>	
EMF Set Viewport Extents (ex)	139
<b>EMF::EMRSETVIEWPORTORGEX</b>	
EMF Set Viewport Origin (ex)	141
<b>EMF::EMRSETWINDOWEXTEX</b>	
EMF Set Window Extent (ex)	143
<b>EMF::EMRSETWINDOWORGEX</b>	
EMF Set Window Origin (ex)	145
<b>EMF::EMRSETWORLDTRANSFORM</b>	
EMF Set World Transform	147
<b>EMF::EMRSTROKEANDFILLPATH</b>	
EMF Stroke and Fill path	149
<b>EMF::EMRSTROKEPATH</b>	
EMF Stroke path	151
<b>EMF::ENHMETAHEADER</b>	
Enhanced Metafile Header Record	153
<b>EMF::EXTPEN</b>	
Extended Graphics Pen	155
<b>EMF::FONT</b>	
Graphics Font	157
<b>EMF::GLOBALOBJECTS</b>	
	159
<b>EMF::GRAPHICSOBJECT</b>	
A global graphics object	163

<b>EMF::INTARRAY</b>	
Represent an array of integers in a simple way	165
<b>EMF::METAFILEDEVICECONTEXT</b>	
Graphics Device Context	166
<b>EMF::METARECORD</b>	
The base class of all metafile records	172
<b>EMF::OBJECT</b>	
Global GDI object	174
<b>EMF::PADDING</b>	
All metafile records must be padded out to a multiple of 4 bytes	176
<b>EMF::PALETTE</b>	
Graphics Palette	177
<b>EMF::PEN</b>	
Graphics Pen	179
<b>EMF::POINT16ARRAY</b>	
Represent an array of 16-bit point in a simple way	180
<b>EMF::POINTLARRAY</b>	
Represent an array of points in a simple way	181
<b>EMF::WCHARSTR</b>	
Represent a wide (UNICODE) character string in a simple way	182

## 3 File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<b>emf.h</b>	183
<b>basetsd.h</b>	184
<b>guiddef.h</b>	186
<b>poppack.h</b>	187
<b>pshpack2.h</b>	187
<b>pshpack4.h</b>	188
<b>w16.h</b>	188
<b>winbase.h</b>	189
<b>windef.h</b>	210
<b>winerror.h</b>	213
<b>wingdi.h</b>	235

<a href="#">winnt.h</a>	273
<a href="#">winuser.h</a>	329
<a href="#">libemf.h</a>	377

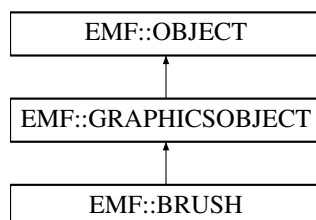
## 4 Data Structure Documentation

### 4.1 EMF::BRUSH Class Reference

Graphics Brush.

```
#include <libemf.h>
```

Inheritance diagram for EMF::BRUSH:



#### Public Member Functions

- [BRUSH](#) (const LOGBRUSH \*lbrush)
- OBJECTTYPE [getType](#) (void) const
- METARECORD \* [newEMR](#) (HDC dc, HGDIOBJ emf\_handle)

#### Public Member Functions inherited from [EMF::GRAPHICSOBJECT](#)

- virtual [~GRAPHICSOBJECT](#) ()  
*GRAPHICSOBJECTs has a virtual destructor.*

#### Public Member Functions inherited from [EMF::OBJECT](#)

- virtual [~OBJECT](#) ()  
*OBJECTs have a virtual destructor.*
- [OBJECT](#) (void)

#### Additional Inherited Members

#### Data Fields inherited from [EMF::GRAPHICSOBJECT](#)

- std::map< HDC, HGDIOBJ > [contexts](#)

**Data Fields inherited from [EMF::OBJECT](#)**

- [HGDIOBJ handle](#)

**4.1.1 Detailed Description**

Graphics Brush.

Brushes are used for filling shapes.

**4.1.2 Constructor & Destructor Documentation****BRUSH()**

```
EMF::BRUSH::BRUSH (
    const LOGBRUSH * lbrush) [inline]
```

**Parameters**

<i>lbrush</i>	the (logical?) brush definition.
---------------	----------------------------------

**4.1.3 Member Function Documentation****getType()**

```
OBJECTTYPE EMF::BRUSH::getType (
    void ) const [inline], [virtual]
```

Return the type of this object (could probably do better with RTTI()).

Implements [EMF::OBJECT](#).

**newEMR()**

```
METARECORD * EMF::BRUSH::newEMR (
    HDC dc,
    HGDIOBJ emf_handle) [inline], [virtual]
```

Return a new metarecord for this object. And record its selection into the given device context.

**Parameters**

<i>dc</i>	handle of device context into which this object is being selected.
<i>emf_handle</i>	the EMF handle associated with the <a href="#">BRUSH</a> .

Implements [EMF::GRAPHICSOBJECT](#).

References [EMF::GRAPHICSOBJECT::contexts](#).

The documentation for this class was generated from the following file:

- [libemf.h](#)

## 4.2 EMF::BYTEARRAY Struct Reference

Represent a byte array in a simple way.

```
#include <libemf.h>
```

### Public Member Functions

- [BYTEARRAY](#) (BYTE \*const array, const int n)

### Data Fields

- BYTE \*const **array\_**  
*Array of unsigned bytes.*
- const int **n\_**  
*Number of bytes in array.*

#### 4.2.1 Detailed Description

Represent a byte array in a simple way.

Evidently, an unsigned array of bytes with no particular encoding implied.

#### 4.2.2 Constructor & Destructor Documentation

##### BYTEARRAY()

```
EMF::BYTEARRAY::BYTEARRAY (  
    BYTE *const array,  
    const int n) [inline]
```

Simple constructor.

##### Parameters

<i>array</i>	pointer to array of bytes
<i>n</i>	number of bytes in array

References [array\\_](#), and [n\\_](#).

The documentation for this struct was generated from the following file:

- libemf.h

## 4.3 EMF::CHARSTR Struct Reference

Represent an ASCII character string in a simple way.

```
#include <libemf.h>
```

### Public Member Functions

- [CHARSTR](#) (CHAR \*const string, const int length)

### Data Fields

- CHAR \*const **string\_**  
*Array of single byte characters.*
- const int **length\_**  
*Number of single byte characters in array.*

#### 4.3.1 Detailed Description

Represent an ASCII character string in a simple way.

ASCII strings don't have to be byte swapped, but this structure allows us to provide a uniform stream-like interface for writing out all the components of metafiles.

#### 4.3.2 Constructor & Destructor Documentation

##### CHARSTR()

```
EMF::CHARSTR::CHARSTR (
    CHAR *const string,
    const int length) [inline]
```

Simple constructor.

##### Parameters

<i>string</i>	pointer to array of single byte characters.
<i>length</i>	number of bytes in array.

References [length\\_](#), and [string\\_](#).

The documentation for this struct was generated from the following file:

- libemf.h

## 4.4 EMF::DATASTREAM Class Reference

Support different endian modes when reading and writing the metafile.

```
#include <libemf.h>
```

### Public Member Functions

- [DATASTREAM](#) (::FILE \*fp=0)
- void [setStream](#) (::FILE \*fp)
- [DATASTREAM](#) & [operator<<](#) (const BYTE &byte)
- [DATASTREAM](#) & [operator>>](#) (BYTE &byte)
- [DATASTREAM](#) & [operator<<](#) (const WORD &word)
- [DATASTREAM](#) & [operator>>](#) (WORD &word)
- [DATASTREAM](#) & [operator<<](#) (const INT16 &word)
- [DATASTREAM](#) & [operator>>](#) (INT16 &word)
- [DATASTREAM](#) & [operator<<](#) (const DWORD &dword)
- [DATASTREAM](#) & [operator>>](#) (DWORD &dword)
- [DATASTREAM](#) & [operator<<](#) (const LONG &long\_)
- [DATASTREAM](#) & [operator>>](#) (LONG &long\_)
- [DATASTREAM](#) & [operator<<](#) (const INT &int\_)
- [DATASTREAM](#) & [operator>>](#) (INT &int\_)
- [DATASTREAM](#) & [operator<<](#) (const UINT &uint)
- [DATASTREAM](#) & [operator>>](#) (UINT &uint)
- [DATASTREAM](#) & [operator<<](#) (const FLOAT &float\_)
- [DATASTREAM](#) & [operator>>](#) (FLOAT &float\_)
- [DATASTREAM](#) & [operator<<](#) (const [PADDING](#) &padding)
- [DATASTREAM](#) & [operator<<](#) (const RECTL &rectl)
- [DATASTREAM](#) & [operator>>](#) (RECTL &rectl)
- [DATASTREAM](#) & [operator<<](#) (const SIZEL &szel)
- [DATASTREAM](#) & [operator>>](#) (SIZEL &szel)
- [DATASTREAM](#) & [operator<<](#) (const [WCHARSTR](#) &wcharstr)
- [DATASTREAM](#) & [operator>>](#) ([WCHARSTR](#) &wcharstr)
- [DATASTREAM](#) & [operator<<](#) (const [CHARSTR](#) &charstr)
- [DATASTREAM](#) & [operator>>](#) ([CHARSTR](#) &charstr)
- [DATASTREAM](#) & [operator<<](#) (const ::EMR &emr)
- [DATASTREAM](#) & [operator>>](#) (::EMR &emr)
- [DATASTREAM](#) & [operator<<](#) (const POINT &point)
- [DATASTREAM](#) & [operator>>](#) (POINT &point)
- [DATASTREAM](#) & [operator<<](#) (const POINTL &pointl)
- [DATASTREAM](#) & [operator>>](#) (POINTL &pointl)
- [DATASTREAM](#) & [operator<<](#) (const POINT16 &point)
- [DATASTREAM](#) & [operator>>](#) (POINT16 &point)
- [DATASTREAM](#) & [operator<<](#) (const XFORM &xform)
- [DATASTREAM](#) & [operator>>](#) (XFORM &xform)
- [DATASTREAM](#) & [operator<<](#) (const [BYTEARRAY](#) &array)
- [DATASTREAM](#) & [operator>>](#) ([BYTEARRAY](#) &array)
- [DATASTREAM](#) & [operator<<](#) (const [POINTLARRAY](#) &array)
- [DATASTREAM](#) & [operator>>](#) ([POINTLARRAY](#) &array)
- [DATASTREAM](#) & [operator<<](#) (const [POINT16ARRAY](#) &array)
- [DATASTREAM](#) & [operator>>](#) ([POINT16ARRAY](#) &array)
- [DATASTREAM](#) & [operator<<](#) (const [INTARRAY](#) &array)
- [DATASTREAM](#) & [operator>>](#) ([INTARRAY](#) &array)

- **DATASTREAM** & **operator**<< (const **DWORDARRAY** &array)
- **DATASTREAM** & **operator**>> (**DWORDARRAY** &array)
- **DATASTREAM** & **operator**<< (const ::EMRTEXT &text)
- **DATASTREAM** & **operator**>> (::EMRTEXT &text)
- **DATASTREAM** & **operator**<< (const LOGPEN &pen)
- **DATASTREAM** & **operator**>> (LOGPEN &pen)
- **DATASTREAM** & **operator**<< (const EXTLOGPEN &pen)
- **DATASTREAM** & **operator**>> (EXTLOGPEN &pen)
- **DATASTREAM** & **operator**<< (const LOGBRUSH &brush)
- **DATASTREAM** & **operator**>> (LOGBRUSH &brush)
- **DATASTREAM** & **operator**<< (const LOGFONTW &font)
- **DATASTREAM** & **operator**>> (LOGFONTW &font)
- **DATASTREAM** & **operator**<< (const PANOSE &panose)
- **DATASTREAM** & **operator**>> (PANOSE &panose)
- **DATASTREAM** & **operator**<< (const EXTLOGFONTW &font)
- **DATASTREAM** & **operator**>> (EXTLOGFONTW &font)
- **DATASTREAM** & **operator**<< (const LOGPALETTE &palette)
- **DATASTREAM** & **operator**>> (LOGPALETTE &palette)

#### 4.4.1 Detailed Description

Support different endian modes when reading and writing the metafile.

To support different endian modes, rather than just writing the structures directly to a file via `fwrite( &mr, ...)`, we have to write each element of the structure separately, swapping bytes as necessary. `datastream` supports this. Remarkably similar to the `QDataStream` class from Qt. So, too, for reading.

#### 4.4.2 Constructor & Destructor Documentation

## DATASTREAM()

```
EMF::DATASTREAM::DATASTREAM (
    ::FILE * fp = 0) [inline]
```

Constructor for **DATASTREAM**.

## Parameters

<i>fp</i>	optional file pointer (but must be assigned before any output occurs.)
-----------	--

Referenced by operator<<(), operator<<(), operator<<(), operator<<(), operator<<(), operator<<(),  
operator<<(), operator<<(), operator<<(), operator<<(), operator<<(), operator<<(),  
operator<<(), operator<<(), operator<<(), operator<<(), operator<<(), operator<<(), operator<<(),  
operator<<(), operator<<(), operator<<(), operator<<(), operator>>(), operator>>(), operator>>(),  
operator>>(), operator>>(), operator>>(), operator>>(), operator>>(), operator>>(), operator>>(),  
operator>>(), operator>>(), operator>>(), operator>>(), operator>>(), operator>>(), operator>>(),  
operator>>(), operator>>(), operator>>(), operator>>(), operator>>(), and operator>>().

### 4.4.3 Member Function Documentation

#### **operator<<()** [1/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const ::EMR & emr) [inline]
```

Output an Enhanced Metafile Record header.

#### **Parameters**

<i>emr</i>	Enhanced Metafile Record header to output.
------------	--

References [DATASTREAM\(\)](#).

#### **operator<<()** [2/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const ::EMRTEXT & text) [inline]
```

Output an Enhanced Metafile Text Record.

#### **Parameters**

<i>text</i>	Enhanced Metafile Text Record to output.
-------------	--

References [DATASTREAM\(\)](#).

#### **operator<<()** [3/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const BYTE & byte) [inline]
```

Output a byte to the stream (not swabbed or anything).

#### **Parameters**

<i>byte</i>	byte to output.
-------------	-----------------

References [DATASTREAM\(\)](#).

**operator<<()** [4/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (
    const BYTEARRAY & array) [inline]
```

Output an array of BYTES.

**Parameters**

<i>array</i>	array of BYTES to output.
--------------	---------------------------

References [EMF::BYTEARRAY::array\\_](#), [DATASTREAM\(\)](#), and [EMF::BYTEARRAY::n\\_](#).

**operator<<()** [5/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (
    const CHARSTR & charstr) [inline]
```

Output a single byte character string.

**Parameters**

<i>charstr</i>	structure to output.
----------------	----------------------

References [DATASTREAM\(\)](#), [EMF::CHARSTR::length\\_](#), and [EMF::CHARSTR::string\\_](#).

**operator<<()** [6/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (
    const DWORD & dword) [inline]
```

Output a double word (long) to the stream (swabbed).

**Parameters**

<i>dword</i>	double word (long) to output.
--------------	-------------------------------

References [DATASTREAM\(\)](#).

**operator<<()** [7/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const DWORDARRAY & array) [inline]
```

Output an array of double words (longs).

**Parameters**

<i>array</i>	array of double words (longs) to output.
--------------	--

References [DATASTREAM\(\)](#), [EMF::DWORDARRAY::dwords\\_](#), and [EMF::DWORDARRAY::n\\_](#).

**operator<<()** [8/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const EXTLOGFONTW & font) [inline]
```

Output an Extended Logical Font definition (using WCHAR strings).

**Parameters**

<i>font</i>	Extended Logical Font definition to output.
-------------	---

References [DATASTREAM\(\)](#).

**operator<<()** [9/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const EXTLOGPEN & pen) [inline]
```

Output an Extended Logical Pen definition.

**Parameters**

<i>pen</i>	Extended Logical Pen definition to output.
------------	--

References [DATASTREAM\(\)](#).

**operator<<()** [10/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (
    const FLOAT & float_) [inline]
```

Output a single precision float to the stream (swabbed).

**Parameters**

<i>float</i> ↔	single precision float to output.
—	

References [DATASTREAM\(\)](#).

**operator<<()** [11/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (
    const INT & int_) [inline]
```

Output a (long) int to the stream (swabbed).

**Parameters**

<i>int</i> ↔	(long) int to output.
—	

References [DATASTREAM\(\)](#).

**operator<<()** [12/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (
    const INT16 & word) [inline]
```

Output a (short, 16-bit) word to the stream (swabbed).

**Parameters**

<i>word</i>	(short, 16-bit) word to output.
-------------	---------------------------------

References [DATASTREAM\(\)](#).

**operator<<()** [13/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const INTARRAY & array) [inline]
```

Output an array of (long) ints.

**Parameters**

<i>array</i>	array of (long) ints to output.
--------------	---------------------------------

References [DATASTREAM\(\)](#), [EMF::INTARRAY::ints\\_](#), and [EMF::INTARRAY::n\\_](#).

**operator<<()** [14/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const LOGBRUSH & brush) [inline]
```

Output a Logical Brush definition.

**Parameters**

<i>brush</i>	Logical Brush definition to output.
--------------	-------------------------------------

References [DATASTREAM\(\)](#).

**operator<<()** [15/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const LOGFONTW & font) [inline]
```

Output a Logical Font definition (using WCHAR strings).

**Parameters**

<i>font</i>	Logical Font definition to output.
-------------	------------------------------------

References [DATASTREAM\(\)](#).

**operator<<()** [16/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const LOGPALETTE & palette) [inline]
```

Output a Logical Palette.

**Parameters**

<i>palette</i>	Logical Palette to output.
----------------	----------------------------

References [DATASTREAM\(\)](#).

**operator<<()** [17/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const LOGPEN & pen) [inline]
```

Output a Logical Pen definition.

**Parameters**

<i>pen</i>	Logical Pen definition to output.
------------	-----------------------------------

References [DATASTREAM\(\)](#).

**operator<<()** [18/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const LONG & long_) [inline]
```

Output a long int to the stream (swabbed).

**Parameters**

<i>long</i> $\leftrightarrow$	long int to output.
—	

References [DATASTREAM\(\)](#).

**operator<<()** [19/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const PADDING & padding) [inline]
```

Output a series of '\0's to pad out a record.

**Parameters**

<i>padding</i>	simple padding structure (length and number of nulls).
----------------	--

References [DATASTREAM\(\)](#), [EMF::PADDING::padding\\_](#), and [EMF::PADDING::size\\_](#).

**operator<<()** [20/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const PANOSE & panose) [inline]
```

Output a Panose structure.

**Parameters**

<i>panose</i>	Panose structure to output.
---------------	-----------------------------

References [DATASTREAM\(\)](#).

**operator<<()** [21/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const POINT & point) [inline]
```

Output a POINT structure.

**Parameters**

<i>point</i>	POINT to output.
--------------	------------------

References [DATASTREAM\(\)](#).

**operator<<()** [22/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const POINT16 & point) [inline]
```

Output a POINT16 structure.

**Parameters**

<i>point</i>	POINT16 to output.
--------------	--------------------

References [DATASTREAM\(\)](#).

**operator<<()** [23/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const POINT16ARRAY & array) [inline]
```

Output an array of POINT16s.

**Parameters**

<i>array</i>	array of POINT16s to output.
--------------	------------------------------

References [DATASTREAM\(\)](#), [EMF::POINT16ARRAY::n\\_](#), and [EMF::POINT16ARRAY::points\\_](#).

**operator<<()** [24/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const POINTL & pointl) [inline]
```

Output a POINTL structure.

**Parameters**

<i>pointl</i>	POINTL to output.
---------------	-------------------

References [DATASTREAM\(\)](#).

**operator<<()** [25/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const POINTLARRAY & array) [inline]
```

Output an array of POINTLs.

**Parameters**

<i>array</i>	array of POINTLs to output.
--------------	-----------------------------

References [DATASTREAM\(\)](#), [EMF::POINTLARRAY::n\\_](#), and [EMF::POINTLARRAY::points\\_](#).

**operator<<()** [26/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const RECTL & rectl) [inline]
```

Output a RECTL structure.

**Parameters**

<i>rectl</i>	structure to output.
--------------	----------------------

References [DATASTREAM\(\)](#).

**operator<<()** [27/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const SIZEL & sizel) [inline]
```

Output a SIZEL structure.

**Parameters**

<i>sizel</i>	structure to output.
--------------	----------------------

References [DATASTREAM\(\)](#).

**operator<<()** [28/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const UINT & uint) [inline]
```

Output a (long) unsigned int to the stream (swabbed).

**Parameters**

<i>uint</i>	(long) unsigned int to output.
-------------	--------------------------------

References [DATASTREAM\(\)](#).

**operator<<()** [29/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const WCHARSTR & wcharstr) [inline]
```

Output a WCHAR string (note: the individual characters are swabbed).

**Parameters**

<i>wcharstr</i>	structure to output.
-----------------	----------------------

References [DATASTREAM\(\)](#), [EMF::WCHARSTR::length\\_](#), and [EMF::WCHARSTR::string\\_](#).

**operator<<()** [30/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const WORD & word) [inline]
```

Output a (short) word to the stream (swabbed).

**Parameters**

<i>word</i>	(short) word to output.
-------------	-------------------------

References [DATASTREAM\(\)](#).

**operator<<()** [31/31]

```
DATASTREAM & EMF::DATASTREAM::operator<< (  
    const XFORM & xform) [inline]
```

Output an XFORM structure.

**Parameters**

<i>xform</i>	XFORM to output.
--------------	------------------

References [DATASTREAM\(\)](#).

**operator>>()** [1/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    ::EMR & emr) [inline]
```

Input an Enhanced Metafile Record header.

**Parameters**

<i>emr</i>	destination of Enhanced Metafile Record header.
------------	---

References [DATASTREAM\(\)](#).

**operator>>()** [2/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    ::EMRTEXT & text) [inline]
```

Input an Enhanced Metafile Text Record.

**Parameters**

<i>text</i>	destination of Enhanced Metafile Text Record.
-------------	---

References [DATASTREAM\(\)](#).

**operator>>()** [3/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (
    BYTE & byte) [inline]
```

Input a byte from the stream (not swabbed or anything).

**Parameters**

<i>byte</i>	destination for input byte.
-------------	-----------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [4/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (
    BYTEARRAY & array) [inline]
```

Input an array of BYTES.

**Parameters**

<i>array</i>	destination of array of input BYTES.
--------------	--------------------------------------

References [EMF::BYTEARRAY::array\\_](#), [DATASTREAM\(\)](#), and [EMF::BYTEARRAY::n\\_](#).

**operator>>()** [5/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (
    CHARSTR & charstr) [inline]
```

Input a single byte character string.

**Parameters**

<i>charstr</i>	destination of input CHAR string.
----------------	-----------------------------------

References [DATASTREAM\(\)](#), [EMF::CHARSTR::length\\_](#), and [EMF::CHARSTR::string\\_](#).

**operator>>()** [6/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    DWORD & dword) [inline]
```

Input a double word (long) from the stream (swabbed).

**Parameters**

<i>dword</i>	destination for double word (long).
--------------	-------------------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [7/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    DWORDARRAY & array) [inline]
```

Input an array of double words (longs).

**Parameters**

<i>array</i>	destination of array of input double words (longs).
--------------	---

References [DATASTREAM\(\)](#), [EMF::DWORDARRAY::dwords\\_](#), and [EMF::DWORDARRAY::n\\_](#).

**operator>>()** [8/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    EXTLOGFONTW & font) [inline]
```

Input an Extended Logical Font definition (using WCHAR strings).

**Parameters**

<i>font</i>	destination of Extended Logical Font definition.
-------------	--

References [DATASTREAM\(\)](#).

**operator>>()** [9/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    EXTLOGPEN & pen) [inline]
```

Input an Extended Logical Pen definition.

**Parameters**

<i>pen</i>	destination of Extended Logical Pen definition.
------------	---

References [DATASTREAM\(\)](#).

**operator>>()** [10/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    FLOAT & float_) [inline]
```

Input a single precision float from the stream (swabbed).

**Parameters**

<i>float</i> ↔ —	destination for single precision float.
---------------------	---

References [DATASTREAM\(\)](#).

**operator>>()** [11/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    INT & int_) [inline]
```

Input a (long) int from the stream (swabbed).

**Parameters**

<i>int</i> ↔ —	destination for (long) int.
-------------------	-----------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [12/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    INT16 & word) [inline]
```

Input a (short, 16-bit) word from the stream (swabbed).

**Parameters**

<i>word</i>	destination for (short, 16-bit) word.
-------------	---------------------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [13/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    INTARRAY & array) [inline]
```

Input an array of (long) ints.

**Parameters**

<i>array</i>	destination of array of input (long) ints.
--------------	--

References [DATASTREAM\(\)](#), [EMF::INTARRAY::ints\\_](#), and [EMF::INTARRAY::n\\_](#).

**operator>>()** [14/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    LOGBRUSH & brush) [inline]
```

Input a Logical Brush definition.

**Parameters**

<i>brush</i>	destination of Logical Brush definition.
--------------	--

References [DATASTREAM\(\)](#).

**operator>>()** [15/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    LOGFONTW & font) [inline]
```

Input a Logical Font definition (using WCHAR strings).

**Parameters**

<i>font</i>	destination of Logical Font definition.
-------------	---

References [DATASTREAM\(\)](#).

**operator>>()** [16/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    LOGPALETTE & palette) [inline]
```

Input a Logical Palette.

**Parameters**

<i>palette</i>	destination of input Logical Palette.
----------------	---------------------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [17/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    LOGPEN & pen) [inline]
```

Input a Logical Pen definition.

**Parameters**

<i>pen</i>	destination of Logical Pen definition.
------------	--

References [DATASTREAM\(\)](#).

**operator>>()** [18/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    LONG & long_) [inline]
```

Input a long int from the stream (swabbed).

**Parameters**

<i>long</i> ↔ —	destination for long int.
--------------------	---------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [19/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    PANOSE & panose) [inline]
```

Input a Panose structure.

**Parameters**

<i>panose</i>	destinatino of input Panose structure.
---------------	--

References [DATASTREAM\(\)](#).

**operator>>()** [20/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    POINT & point) [inline]
```

Input a POINT structure.

**Parameters**

<i>point</i>	destination of input POINT.
--------------	-----------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [21/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    POINT16 & point)    [inline]
```

Input a POINT16 structure.

**Parameters**

<i>point</i>	destination of input POINT16.
--------------	-------------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [22/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    POINT16ARRAY & array)    [inline]
```

Input an array of POINT16s.

**Parameters**

<i>array</i>	destination of array of input POINT16s.
--------------	---

References [DATASTREAM\(\)](#), [EMF::POINT16ARRAY::n\\_](#), and [EMF::POINT16ARRAY::points\\_](#).

**operator>>()** [23/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    POINTL & pointl)    [inline]
```

Input a POINTL structure.

**Parameters**

<i>pointl</i>	destination of input POINTL.
---------------	------------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [24/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    POINTLARRAY & array) [inline]
```

Input an array of POINTLs.

**Parameters**

<i>array</i>	destination of array of input POINTLs.
--------------	--

References [DATASTREAM\(\)](#), [EMF::POINTLARRAY::n\\_](#), and [EMF::POINTLARRAY::points\\_](#).

**operator>>()** [25/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    RECTL & rectl) [inline]
```

Input a RECTL structure.

**Parameters**

<i>rectl</i>	destination of input RECTL.
--------------	-----------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [26/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    SIZEL & sizel) [inline]
```

Input a SIZEL structure.

**Parameters**

<i>sizel</i>	destination of input SIZEL.
--------------	-----------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [27/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    UINT & uint) [inline]
```

Input a (long) unsigned int from the stream (swabbed).

**Parameters**

<i>uint</i>	destination for (long) unsigned int.
-------------	--------------------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [28/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    WCHARSTR & wcharstr) [inline]
```

Input a WCHAR string (note: the individual characters are swabbed.)

**Parameters**

<i>wcharstr</i>	destination of input WCHAR string.
-----------------	------------------------------------

References [DATASTREAM\(\)](#), [EMF::WCHARSTR::length\\_](#), and [EMF::WCHARSTR::string\\_](#).

**operator>>()** [29/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (  
    WORD & word) [inline]
```

Input a (short) word from the stream (swabbed).

**Parameters**

<i>word</i>	destination for (short) word.
-------------	-------------------------------

References [DATASTREAM\(\)](#).

**operator>>()** [30/30]

```
DATASTREAM & EMF::DATASTREAM::operator>> (
    XFORM & xform) [inline]
```

Input an XFORM structure.

#### Parameters

<i>xform</i>	destination of input XFORM.
--------------	-----------------------------

References [DATASTREAM\(\)](#).

#### setStream()

```
void EMF::DATASTREAM::setStream (
    ::FILE * fp) [inline]
```

Use the given FILE stream as the input/output destination.

#### Parameters

<i>fp</i>	file point for i/o.
-----------	---------------------

The documentation for this class was generated from the following file:

- libemf.h

## 4.5 EMF::DWORDARRAY Struct Reference

Represent an array of double word integers in a simple way.

```
#include <libemf.h>
```

#### Public Member Functions

- [DWORDARRAY](#) (DWORD \*const dwords, const DWORD n)

#### Data Fields

- DWORD \*const **dwords\_**  
*Array of double words.*
- const DWORD **n\_**  
*Number of double words in array.*

### 4.5.1 Detailed Description

Represent an array of double word integers in a simple way.

Allow an array of DWORD's to be written out at once.

### 4.5.2 Constructor & Destructor Documentation

#### DWORDARRAY()

```
EMF::DWORDARRAY::DWORDARRAY (
    DWORD *const dwords,
    const DWORD n) [inline]
```

simple constructor.

#### Parameters

<i>dwords</i>	pointer to double words.
<i>n</i>	number double words in array.

References [dwords\\_](#), and [n\\_](#).

The documentation for this struct was generated from the following file:

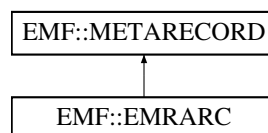
- libemf.h

## 4.6 EMF::EMRARC Class Reference

EMF Arc.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRARC:



#### Public Member Functions

- [EMRARC](#) (INT left, INT top, INT right, INT bottom, INT xstart, INT ystart, INT xend, INT yend)
- [EMRARC](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.6.1 Detailed Description

EMF Arc.

Draw an arc. Not sure what the specification here means, though.

### 4.6.2 Constructor & Destructor Documentation

#### [EMRARC\(\)](#) [1/2]

```
EMF::EMRARC::EMRARC (  
    INT left,  
    INT top,  
    INT right,  
    INT bottom,  
    INT xstart,  
    INT ystart,  
    INT xend,  
    INT yend) [inline]
```

Take these descriptions with a grain of salt...

#### Parameters

<i>left</i>	x position of left edge of arc box.
<i>top</i>	y position of top edge of arc box.
<i>right</i>	x position of right edge of arc box.
<i>bottom</i>	y position bottom edge of arc box.
<i>xstart</i>	x position of arc start.
<i>ystart</i>	y position of arc start.
<i>xend</i>	x position of arc end.
<i>yend</i>	y position of arc end.

References [EMRARC\(\)](#).

Referenced by [EMRARC\(\)](#).

**EMRARC()** [2/2]

```
EMF::EMRARC::EMRARC (
    DATASTREAM & ds) [inline]
```

Construct an Arc record from the input datastream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.6.3 Member Function Documentation****execute()**

```
void EMF::EMRARC::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRARC::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRARC::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

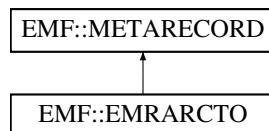
- libemf.h

## 4.7 EMF::EMRARCTO Class Reference

EMF Arc To.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRARCTO:



### Public Member Functions

- [EMRARCTO](#) (INT left, INT top, INT right, INT bottom, INT xstart, INT ystart, INT xend, INT yend)
- [EMRARCTO](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.7.1 Detailed Description

EMF Arc To.

Draw another arc. Not sure what the specification here means, though.

#### 4.7.2 Constructor & Destructor Documentation

##### EMRARCTO() [1/2]

```

EMF::EMRARCTO::EMRARCTO (
    INT left,
    INT top,
    INT right,
    INT bottom,
    INT xstart,
    INT ystart,
    INT xend,
    INT yend) [inline]
  
```

Take these descriptions with a grain of salt...

**Parameters**

<i>left</i>	x position of left edge of arc box.
<i>top</i>	y position of top edge of arc box.
<i>right</i>	x position of right edge of arc box.
<i>bottom</i>	y position bottom edge of arc box.
<i>xstart</i>	x position of arc start.
<i>ystart</i>	y position of arc start.
<i>xend</i>	x position of arc end.
<i>yend</i>	y position of arc end.

References [EMRARCTO\(\)](#).

Referenced by [EMRARCTO\(\)](#).

**EMRARCTO() [2/2]**

```
EMF::EMRARCTO::EMRARCTO (
    DATASTREAM & ds) [inline]
```

Construct an ArcTo record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.7.3 Member Function Documentation****execute()**

```
void EMF::EMRARCTO::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRARCTO::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRARCTO::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

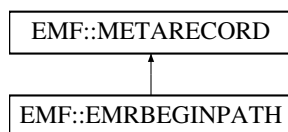
- libemf.h

**4.8 EMF::EMRBEGINPATH Class Reference**

EMF Begin Path.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRBEGINPATH:

**Public Member Functions**

- [EMRBEGINPATH](#) (void)
- [EMRBEGINPATH](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

### 4.8.1 Detailed Description

EMF Begin Path.

Begin the current path definition.

### 4.8.2 Constructor & Destructor Documentation

#### EMRBEGINPATH() [1/2]

```
EMF::EMRBEGINPATH::EMRBEGINPATH (  
    void )    [inline]
```

Create a Begin Path record.

References [EMRBEGINPATH\(\)](#).

Referenced by [EMRBEGINPATH\(\)](#).

#### EMRBEGINPATH() [2/2]

```
EMF::EMRBEGINPATH::EMRBEGINPATH (  
    DATASTREAM & ds)    [inline]
```

Construct a BeginPath record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.8.3 Member Function Documentation

#### execute()

```
void EMF::EMRBEGINPATH::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const    [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRBEGINPATH::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRBEGINPATH::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

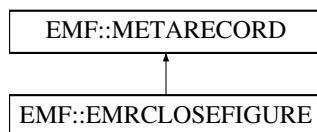
- libemf.h

**4.9 EMF::EMRCLOSEFIGURE Class Reference**

EMF Close Figure.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRCLOSEFIGURE:

**Public Member Functions**

- [EMRCLOSEFIGURE](#) (void)
- [EMRCLOSEFIGURE](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

### 4.9.1 Detailed Description

EMF Close Figure.

Close the current figure.

### 4.9.2 Constructor & Destructor Documentation

#### EMRCLOSEFIGURE() [1/2]

```
EMF::EMRCLOSEFIGURE::EMRCLOSEFIGURE (  
    void )    [inline]
```

Create a Close Figure record.

References [EMRCLOSEFIGURE\(\)](#).

Referenced by [EMRCLOSEFIGURE\(\)](#).

#### EMRCLOSEFIGURE() [2/2]

```
EMF::EMRCLOSEFIGURE::EMRCLOSEFIGURE (  
    DATASTREAM & ds)    [inline]
```

Construct a CloseFigure record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.9.3 Member Function Documentation

#### execute()

```
void EMF::EMRCLOSEFIGURE::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const    [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRCLOSEFIGURE::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRCLOSEFIGURE::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

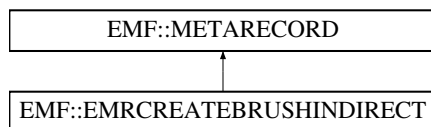
- libemf.h

**4.10 EMF::EMRCREATEBRUSHINDIRECT Class Reference**

EMF Brush.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRCREATEBRUSHINDIRECT:

**Public Member Functions**

- [EMRCREATEBRUSHINDIRECT](#) ([BRUSH](#) \*brush, [HGDIOBJ](#) handle)
- [EMRCREATEBRUSHINDIRECT](#) ([DATASTREAM](#) &ds)
- bool [serialize](#) ([DATASTREAM](#) ds)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \*source, [HDC](#) dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

### 4.10.1 Detailed Description

EMF Brush.

Create a new brush (used for filling shapes).

### 4.10.2 Constructor & Destructor Documentation

#### EMRCREATEBRUSHINDIRECT() [1/2]

```
EMRCREATEBRUSHINDIRECT::EMRCREATEBRUSHINDIRECT (
    BRUSH * brush,
    HGDIOBJ handle)
```

##### Parameters

<i>brush</i>	an instance of a <a href="#">BRUSH</a> object.
<i>handle</i>	the <a href="#">BRUSH</a> object's handle.

References [EMRCREATEBRUSHINDIRECT\(\)](#).

Referenced by [EMRCREATEBRUSHINDIRECT\(\)](#).

#### EMRCREATEBRUSHINDIRECT() [2/2]

```
EMRCREATEBRUSHINDIRECT::EMRCREATEBRUSHINDIRECT (
    DATASTREAM & ds)
```

Create a CreateBrushIndirect record from the input stream.

##### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.10.3 Member Function Documentation

#### execute()

```
void EMRCREATEBRUSHINDIRECT::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

References [EMF::METAFILEDEVICECONTEXT::emf\\_handles](#).

**serialize()**

```
bool EMF::EMRCREATEBRUSHINDIRECT::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRCREATEBRUSHINDIRECT::size (  
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following files:

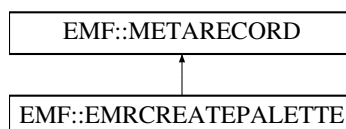
- libemf.h
- libemf.cpp

## 4.11 EMF::EMRCREATEPALETTE Class Reference

EMF Palette.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRCREATEPALETTE:



## Public Member Functions

- [EMRCREATEPALETTE](#) ([PALETTE](#) \*palette, HGDIOBJ handle)
- [EMRCREATEPALETTE](#) ([DATASTREAM](#) &ds)
- bool [serialize](#) ([DATASTREAM](#) ds)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.11.1 Detailed Description

EMF Palette.

Create a new palette.

### 4.11.2 Constructor & Destructor Documentation

#### [EMRCREATEPALETTE\(\)](#) [1/2]

```
EMRCREATEPALETTE::EMRCREATEPALETTE (
    PALETTE * palette,
    HGDIOBJ handle)
```

#### Parameters

<i>palette</i>	an instance of a <a href="#">PALETTE</a> object.
<i>handle</i>	the <a href="#">PALETTE</a> object's handle.

References [EMRCREATEPALETTE\(\)](#).

Referenced by [EMRCREATEPALETTE\(\)](#).

#### [EMRCREATEPALETTE\(\)](#) [2/2]

```
EMF::EMRCREATEPALETTE::EMRCREATEPALETTE (
    DATASTREAM & ds)
```

Construct a CreatePalette record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.11.3 Member Function Documentation

#### execute()

```
void EMRCREATEPALETTE::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRCREATEPALETTE::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRCREATEPALETTE::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following files:

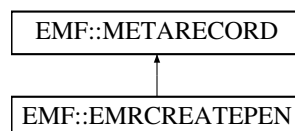
- libemf.h
- libemf.cpp

## 4.12 EMF::EMRCREATEPEN Class Reference

EMF Pen.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRCREATEPEN:



## Public Member Functions

- [EMRCREATEPEN](#) ([PEN](#) \*pen, [HGDIOBJ](#) handle)
- [EMRCREATEPEN](#) ([DATASTREAM](#) &ds)
- bool [serialize](#) ([DATASTREAM](#) ds)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \*source, [HDC](#) dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.12.1 Detailed Description

EMF Pen.

Create a new pen (used for drawing lines, arcs, rectangles, etc.).

### 4.12.2 Constructor & Destructor Documentation

#### [EMRCREATEPEN\(\)](#) [1/2]

```
EMRCREATEPEN::EMRCREATEPEN (
    PEN * pen,
    HGDIOBJ handle)
```

#### Parameters

<i>pen</i>	an instance of a <a href="#">PEN</a> object.
<i>handle</i>	the <a href="#">PEN</a> object's handle.

References [EMRCREATEPEN\(\)](#).

Referenced by [EMRCREATEPEN\(\)](#).

#### [EMRCREATEPEN\(\)](#) [2/2]

```
EMRCREATEPEN::EMRCREATEPEN (
    DATASTREAM & ds)
```

Construct a CreatePen record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.12.3 Member Function Documentation

#### **execute()**

```
void EMRCREATEPEN::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [virtual]
```

Execute this record in the context of the given device context.

#### **Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

References [EMF::METAFILEDEVICECONTEXT::emf\\_handles](#).

#### **serialize()**

```
bool EMF::EMRCREATEPEN::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### **Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### **size()**

```
int EMF::EMRCREATEPEN::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following files:

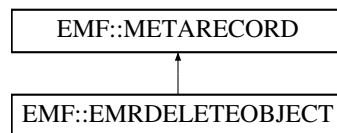
- libemf.h
- libemf.cpp

## 4.13 EMF::EMRDELETEOBJECT Class Reference

EMF Delete Object.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRDELETEOBJECT:



### Public Member Functions

- [EMRDELETEOBJECT](#) (HGDIOBJ object)
- [EMRDELETEOBJECT](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.13.1 Detailed Description

EMF Delete Object.

Delete the given object, such as a pen, brush or font.

#### 4.13.2 Constructor & Destructor Documentation

##### EMRDELETEOBJECT() [1/2]

```
EMF::EMRDELETEOBJECT::EMRDELETEOBJECT (
    HGDIOBJ object) [inline]
```

### Parameters

<i>object</i>	the object to delete.
---------------	-----------------------

References [EMRDELETEOBJECT\(\)](#).

Referenced by [EMRDELETEOBJECT\(\)](#).

**EMRDELETEOBJECT() [2/2]**

```
EMF::EMRDELETEOBJECT::EMRDELETEOBJECT (
    DATASTREAM & ds) [inline]
```

Construct a DeleteObject record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.13.3 Member Function Documentation****execute()**

```
void EMRDELETEOBJECT::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

References [EMF::METAFILEDEVICECONTEXT::emf\\_handles](#).

**serialize()**

```
bool EMF::EMRDELETEOBJECT::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRDELETEOBJECT::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following files:

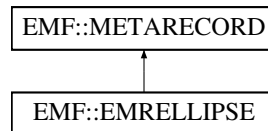
- libemf.h
- libemf.cpp

## 4.14 EMF::EMRELLIPSE Class Reference

EMF Ellipse.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRELLIPSE:



### Public Member Functions

- [EMRELLIPSE](#) (INT left, INT top, INT right, INT bottom)
- [EMRELLIPSE](#) ([DATASTREAM](#) &ds)
- bool [serialize](#) ([DATASTREAM](#) ds)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.14.1 Detailed Description

EMF Ellipse.

Draw an ellipse. (I have no idea how the ellipse is defined!)

#### 4.14.2 Constructor & Destructor Documentation

##### EMRELLIPSE() [1/2]

```

EMF::EMRELLIPSE::EMRELLIPSE (
    INT left,
    INT top,
    INT right,
    INT bottom) [inline]
  
```

Take these descriptions with a grain of salt...

### Parameters

<i>left</i>	x position of left extrema of ellipse.
-------------	--

<i>top</i>	y position of top extrema of ellipse.
<i>right</i>	x position of right extrema of ellipse.
<i>bottom</i>	y position of bottom extrema of ellipse.

References [EMRELLIPSE\(\)](#).

Referenced by [EMRELLIPSE\(\)](#).

## EMRELLIPSE() [2/2]

```
EMF::EMRELLIPSE::EMRELLIPSE (  
    DATASTREAM & ds) [inline]
```

Construct an Ellipse record from the input stream.

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

## 4.14.3 Member Function Documentation

### execute()

```
void EMF::EMRELLIPSE::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

### serialize()

```
bool EMF::EMRELLIPSE::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRELLIPSE::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

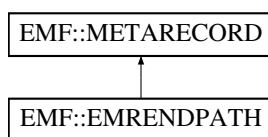
- libemf.h

**4.15 EMF::EMRENDPATH Class Reference**

EMF End Path.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRENDPATH:

**Public Member Functions**

- [EMRENDPATH](#) (void)
- [EMRENDPATH](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

**4.15.1 Detailed Description**

EMF End Path.

End the current path definition.

### 4.15.2 Constructor & Destructor Documentation

#### EMRENDPATH() [1/2]

```
EMF::EMRENDPATH::EMRENDPATH (  
    void )    [inline]
```

Create an End Path record.

References [EMRENDPATH\(\)](#).

Referenced by [EMRENDPATH\(\)](#).

#### EMRENDPATH() [2/2]

```
EMF::EMRENDPATH::EMRENDPATH (  
    DATASTREAM & ds)    [inline]
```

Construct an EndPath record from the input stream.

##### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.15.3 Member Function Documentation

#### execute()

```
void EMF::EMRENDPATH::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const    [inline], [virtual]
```

Execute this record in the context of the given device context.

##### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRENDPATH::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRENDPATH::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

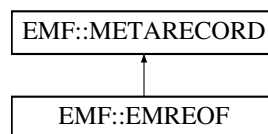
- libemf.h

**4.16 EMF::EMREOF Class Reference**

EMF End of File Record.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMREOF:

**Public Member Functions**

- [EMREOF](#) (void)
- [EMREOF](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

### 4.16.1 Detailed Description

EMF End of File Record.

Every metafile must have an End of File record. A palette may also be recorded in the EOF record, but it is currently unused by this library (all colors are specified in full RGB coordinates).

### 4.16.2 Constructor & Destructor Documentation

#### EMREOF() [1/2]

```
EMF::EMREOF::EMREOF (  
    void ) [inline]
```

Constructor contains no user serviceable parts.

References [EMREOF\(\)](#).

Referenced by [EMREOF\(\)](#).

#### EMREOF() [2/2]

```
EMF::EMREOF::EMREOF (  
    DATASTREAM & ds) [inline]
```

Construct an EOF record from the input stream

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.16.3 Member Function Documentation

#### execute()

```
void EMF::EMREOF::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMREOF::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMREOF::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

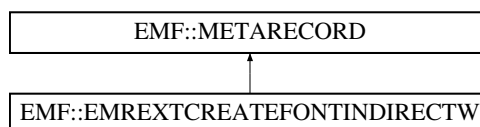
- libemf.h

**4.17 EMF::EMREXTCREATEFONTINDIRECTW Class Reference**

EMF Font.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMREXTCREATEFONTINDIRECTW:

**Public Member Functions**

- [EMREXTCREATEFONTINDIRECTW](#) ([FONT](#) \*font, HGDIOBJ handle)
- [EMREXTCREATEFONTINDIRECTW](#) ([DATASTREAM](#) &ds)
- bool [serialize](#) ([DATASTREAM](#) ds)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

#### 4.17.1 Detailed Description

EMF Font.

Create a new font.

#### 4.17.2 Constructor & Destructor Documentation

##### EMREXTCREATEFONTINDIRECTW() [1/2]

```
EMREXTCREATEFONTINDIRECTW::EMREXTCREATEFONTINDIRECTW (
    FONT * font,
    HGDIOBJ handle)
```

##### Parameters

<i>font</i>	an instance of a <a href="#">FONT</a> object.
<i>handle</i>	the <a href="#">FONT</a> object's handle.

References [EMREXTCREATEFONTINDIRECTW\(\)](#).

Referenced by [EMREXTCREATEFONTINDIRECTW\(\)](#).

##### EMREXTCREATEFONTINDIRECTW() [2/2]

```
EMREXTCREATEFONTINDIRECTW::EMREXTCREATEFONTINDIRECTW (
    DATASTREAM & ds)
```

Construct a CreateFontIndirectW record from the input stream.

##### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

#### 4.17.3 Member Function Documentation

##### execute()

```
void EMREXTCREATEFONTINDIRECTW::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

References [EMF::METAFILEDEVICECONTEXT::emf\\_handles](#).

**serialize()**

```
bool EMF::EMREXTCREATEFONTINDIRECTW::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMREXTCREATEFONTINDIRECTW::size (
    void ) const [inline], [virtual]
```

**Returns**

the size of the record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following files:

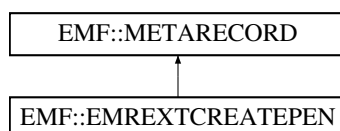
- libemf.h
- libemf.cpp

**4.18 EMF::EMREXTCREATEPEN Class Reference**

EMF Extended Pen.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMREXTCREATEPEN:



## Public Member Functions

- [EMREXTCREATEPEN](#) ([EXTPEN](#) \*pen, HGDIOBJ handle)
- [EMREXTCREATEPEN](#) ([DATASTREAM](#) &ds)
- bool [serialize](#) ([DATASTREAM](#) ds)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.18.1 Detailed Description

EMF Extended Pen.

Create a new pen (used for drawing lines, arcs, rectangles, etc.). Apparently uses extended attributes such as a bitmap mask.

### 4.18.2 Constructor & Destructor Documentation

#### EMREXTCREATEPEN() [1/2]

```
EMREXTCREATEPEN::EMREXTCREATEPEN (
    EXTPEN * pen,
    HGDIOBJ handle)
```

#### Parameters

<i>pen</i>	an instance of a <a href="#">PEN</a> object.
<i>handle</i>	the <a href="#">PEN</a> object's handle.

References [EMREXTCREATEPEN\(\)](#).

Referenced by [EMREXTCREATEPEN\(\)](#).

#### EMREXTCREATEPEN() [2/2]

```
EMREXTCREATEPEN::EMREXTCREATEPEN (
    DATASTREAM & ds)
```

Construct a ExtCreatePen record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.18.3 Member Function Documentation

#### execute()

```
void EMREXTCREATEPEN::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

References [EMF::METAFILEDEVICECONTEXT::emf\\_handles](#).

#### serialize()

```
bool EMF::EMREXTCREATEPEN::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMREXTCREATEPEN::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following files:

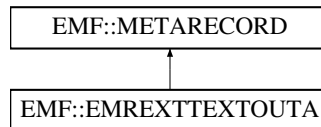
- libemf.h
- libemf.cpp

## 4.19 EMF::EMREXTTEXTOUTA Class Reference

EMF Extended Text Output ASCII.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMREXTTEXTOUTA:



### Public Member Functions

- [EMREXTTEXTOUTA](#) (const RECTL \*bounds, DWORD graphicsMode, FLOAT xScale, FLOAT yScale, const PEMRTEXT text, LPCSTR string, const INT \*dx)
- [EMREXTTEXTOUTA](#) (DATASTREAM &ds)
- [~EMREXTTEXTOUTA](#) ()
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.19.1 Detailed Description

EMF Extended Text Output ASCII.

Draw this text string with the current font, in the color of the current pen and with the given text background color. Individual character positioning can be given in the dx array.

#### 4.19.2 Constructor & Destructor Documentation

##### EMREXTTEXTOUTA() [1/2]

```

EMF::EMREXTTEXTOUTA::EMREXTTEXTOUTA (
    const RECTL * bounds,
    DWORD graphicsMode,
    FLOAT xScale,
    FLOAT yScale,
    const PEMRTEXT text,
    LPCSTR string,
    const INT * dx) [inline]
  
```

**Parameters**

<i>bounds</i>	bounding box of text string.
<i>graphicsMode</i>	(not entirely sure?)
<i>xScale</i>	width scale factor (of what?)
<i>yScale</i>	height scale factor (of what?)
<i>text</i>	a text metarecord containing the rendering style.
<i>string</i>	the text to render
<i>dx</i>	an array of positions for each character in string.

References [EMREXTTEXTOUTA\(\)](#).

Referenced by [EMREXTTEXTOUTA\(\)](#).

**EMREXTTEXTOUTA()** [2/2]

```
EMF::EMREXTTEXTOUTA::EMREXTTEXTOUTA (
    DATASTREAM & ds) [inline]
```

Construct a ExtTextOutA record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**~EMREXTTEXTOUTA()**

```
EMF::EMREXTTEXTOUTA::~~EMREXTTEXTOUTA () [inline]
```

Destructor frees its copy of the string and its character offset array

**4.19.3 Member Function Documentation****execute()**

```
void EMF::EMREXTTEXTOUTA::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
---------------	---

<i>dc</i>	device context for execute.
-----------	-----------------------------

Implements [EMF::METARECORD](#).

### serialize()

```
bool EMF::EMREXTTEXTOUTA::serialize (
    DATASTREAM ds) [inline], [virtual]
```

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

### size()

```
int EMF::EMREXTTEXTOUTA::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

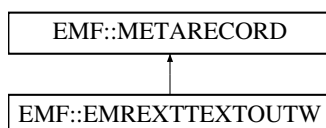
- libemf.h

## 4.20 EMF::EMREXTTEXTOUTW Class Reference

EMF Extended Text Output Wide character.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMREXTTEXTOUTW:



### Public Member Functions

- [EMREXTTEXTOUTW](#) (const RECTL \*bounds, DWORD graphicsMode, FLOAT xScale, FLOAT yScale, const PEMRTEXT text, LPCWSTR string, const INT \*dx)
- [EMREXTTEXTOUTW](#) (DATASTREAM &ds)
- [~EMREXTTEXTOUTW](#) ()
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

**4.20.1 Detailed Description**

EMF Extended Text Output Wide character.

Draw this text string with the current font, in the color of the current pen and with the given text background color. Individual character positioning can be given in the dx array.

**4.20.2 Constructor & Destructor Documentation****EMREXTTEXTOUTW() [1/2]**

```
EMF::EMREXTTEXTOUTW::EMREXTTEXTOUTW (
    const RECTL * bounds,
    DWORD graphicsMode,
    FLOAT xScale,
    FLOAT yScale,
    const PEMRTEXT text,
    LPCWSTR string,
    const INT * dx) [inline]
```

**Parameters**

<i>bounds</i>	bounding box of text string.
<i>graphicsMode</i>	(not entirely sure?)
<i>xScale</i>	width scale factor (of what?)
<i>yScale</i>	height scale factor (of what?)
<i>text</i>	a text metarecord containing the rendering style.
<i>string</i>	the text to render
<i>dx</i>	an array of positions for each character in string.

References [EMREXTTEXTOUTW\(\)](#).

Referenced by [EMREXTTEXTOUTW\(\)](#).

**EMREXTTEXTOUTW() [2/2]**

```
EMF::EMREXTTEXTOUTW::EMREXTTEXTOUTW (
    DATASTREAM & ds) [inline]
```

Construct a ExtTextOutA record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

### ~EMREXTTEXTOUTW()

```
EMF::EMREXTTEXTOUTW::~~EMREXTTEXTOUTW () [inline]
```

Destructor frees its copy of the string and its character offset array

## 4.20.3 Member Function Documentation

### execute()

```
void EMF::EMREXTTEXTOUTW::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

### serialize()

```
bool EMF::EMREXTTEXTOUTW::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

### size()

```
int EMF::EMREXTTEXTOUTW::size (  
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

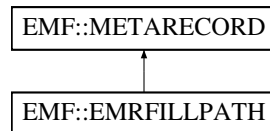
- libemf.h

## 4.21 EMF::EMRFILLPATH Class Reference

EMF Fill path.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRFILLPATH:



### Public Member Functions

- [EMRFILLPATH](#) (const RECTL \*bounds)
- [EMRFILLPATH](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.21.1 Detailed Description

EMF Fill path.

Fill the path.

#### 4.21.2 Constructor & Destructor Documentation

##### EMRFILLPATH() [1/2]

```
EMF::EMRFILLPATH::EMRFILLPATH (
    const RECTL * bounds) [inline]
```

### Parameters

<i>bounds</i>	overall bounding box of polygon.
---------------	----------------------------------

References [EMRFILLPATH\(\)](#).

Referenced by [EMRFILLPATH\(\)](#).

**EMRFILLPATH()** [2/2]

```
EMF::EMRFILLPATH::EMRFILLPATH (
    DATASTREAM & ds) [inline]
```

Create a FillPath record from input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.21.3 Member Function Documentation****execute()**

```
void EMF::EMRFILLPATH::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRFILLPATH::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRFILLPATH::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

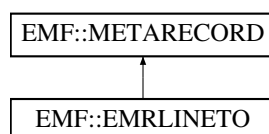
- libemf.h

## 4.22 EMF::EMRLINETO Class Reference

EMF Line To.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRLINETO:



### Public Member Functions

- [EMRLINETO](#) (INT x, INT y)
- [EMRLINETO](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.22.1 Detailed Description

EMF Line To.

Draw a line using the current pen to the given position.

#### 4.22.2 Constructor & Destructor Documentation

##### EMRLINETO() [1/2]

```
EMF::EMRLINETO::EMRLINETO (
    INT x,
    INT y)    [inline]
```

### Parameters

<i>x</i>	x position to draw line to in logical coordinates.
<i>y</i>	y position to draw line to in logical coordinates.

References [EMRLINETO\(\)](#).

Referenced by [EMRLINETO\(\)](#).

**EMRLINETO()** [2/2]

```
EMF::EMRLINETO::EMRLINETO (
    DATASTREAM & ds) [inline]
```

Construct a LineTo record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream
-----------	---------------------

**4.22.3 Member Function Documentation****execute()**

```
void EMF::EMRLINETO::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRLINETO::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream
-----------	---------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRLINETO::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

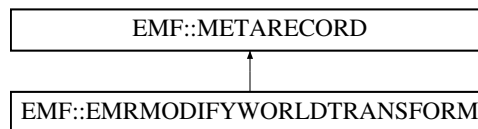
- libemf.h

## 4.23 EMF::EMRMODIFYWORLDTRANSFORM Class Reference

EMF Modify World Transform.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRMODIFYWORLDTRANSFORM:



### Public Member Functions

- [EMRMODIFYWORLDTRANSFORM](#) (const XFORM \*transform, DWORD mode)
- [EMRMODIFYWORLDTRANSFORM](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.23.1 Detailed Description

EMF Modify World Transform.

Enhanced metafiles have a Coordinate Transformation which allows the contents to be rotated and transformed. Does not appear to work properly in StarOffice (but it's also possible I don't understand how it's supposed to work either).

#### 4.23.2 Constructor & Destructor Documentation

##### EMRMODIFYWORLDTRANSFORM() [1/2]

```
EMF::EMRMODIFYWORLDTRANSFORM::EMRMODIFYWORLDTRANSFORM (
    const XFORM * transform,
    DWORD mode) [inline]
```

### Parameters

<i>transform</i>	the transformation to apply
<i>mode</i>	the mode of the transformation application (namely, pre- or post-multiply)

References [EMRMODIFYWORLDTRANSFORM\(\)](#).

Referenced by [EMRMODIFYWORLDTRANSFORM\(\)](#).

## EMRMODIFYWORLDTRANSFORM() [2/2]

```
EMF::EMRMODIFYWORLDTRANSFORM::EMRMODIFYWORLDTRANSFORM (  
    DATASTREAM & ds) [inline]
```

Construct a ModifyWorldTransform from the input datastream.

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

## 4.23.3 Member Function Documentation

### execute()

```
void EMF::EMRMODIFYWORLDTRANSFORM::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

### serialize()

```
bool EMF::EMRMODIFYWORLDTRANSFORM::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

### size()

```
int EMF::EMRMODIFYWORLDTRANSFORM::size (  
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

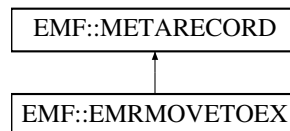
- libemf.h

## 4.24 EMF::EMRMOVETOEX Class Reference

EMF MoveTo (ex).

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRMOVETOEX:



### Public Member Functions

- [EMRMOVETOEX](#) (INT x, INT y)
- [EMRMOVETOEX](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.24.1 Detailed Description

EMF MoveTo (ex).

Move the drawing point to the given position.

#### 4.24.2 Constructor & Destructor Documentation

##### EMRMOVETOEX() [1/2]

```
EMF::EMRMOVETOEX::EMRMOVETOEX (
    INT x,
    INT y) [inline]
```

### Parameters

<i>x</i>	new x position in logical coordinates.
<i>y</i>	new y position in logical coordinates.

References [EMRMOVETOEX\(\)](#).

Referenced by [EMRMOVETOEX\(\)](#).

**EMRMOVETOEX()** [2/2]

```
EMF::EMRMOVETOEX::EMRMOVETOEX (  
    DATASTREAM & ds) [inline]
```

Construct a MoveToEx record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.24.3 Member Function Documentation****execute()**

```
void EMF::EMRMOVETOEX::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRMOVETOEX::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRMOVETOEX::size (  
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

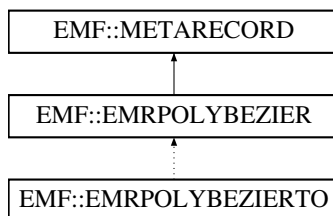
- libemf.h

## 4.25 EMF::EMRPOLYBEZIER Class Reference

EMF Polybezier.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYBEZIER:



### Public Member Functions

- [EMRPOLYBEZIER](#) (const RECTL \*bounds, const POINT \*points, INT n)
- [EMRPOLYBEZIER](#) (DATASTREAM &ds)
- [~EMRPOLYBEZIER](#) ()
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.25.1 Detailed Description

EMF Polybezier.

Draw a polygonal Bezier curve to (what?)

#### 4.25.2 Constructor & Destructor Documentation

##### EMRPOLYBEZIER() [1/2]

```
EMF::EMRPOLYBEZIER::EMRPOLYBEZIER (
    const RECTL * bounds,
    const POINT * points,
    INT n) [inline]
```

### Parameters

<i>bounds</i>	overall bounding box of polybezier curve.
---------------	---

<i>points</i>	array of polybezier vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYBEZIER\(\)](#).

Referenced by [EMRPOLYBEZIER\(\)](#), and [EMRPOLYBEZIER\(\)](#).

## EMRPOLYBEZIER() [2/2]

```
EMF::EMRPOLYBEZIER::EMRPOLYBEZIER (
    DATASTREAM & ds) [inline]
```

Construct a PolyBezier record from the input stream.

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYBEZIER\(\)](#).

## ~EMRPOLYBEZIER()

```
EMF::EMRPOLYBEZIER::~~EMRPOLYBEZIER () [inline]
```

Destructor frees a copy of the points it buffered.

## 4.25.3 Member Function Documentation

### execute()

```
void EMF::EMRPOLYBEZIER::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

Reimplemented in [EMF::EMRPOLYBEZIERTO](#).

**serialize()**

```
bool EMF::EMRPOLYBEZIER::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

Reimplemented in [EMF::EMRPOLYBEZIERTO](#).

**size()**

```
int EMF::EMRPOLYBEZIER::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

Reimplemented in [EMF::EMRPOLYBEZIERTO](#).

The documentation for this class was generated from the following file:

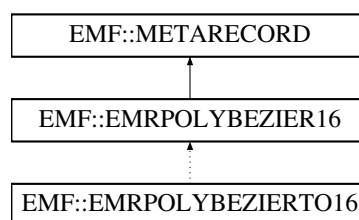
- libemf.h

**4.26 EMF::EMRPOLYBEZIER16 Class Reference**

EMF Polybezier16.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYBEZIER16:

**Public Member Functions**

- [EMRPOLYBEZIER16](#) (const RECTL \*bounds, const POINT16 \*points, INT n)
- [EMRPOLYBEZIER16](#) (const RECTL \*bounds, const POINT \*points, INT n)
- [EMRPOLYBEZIER16](#) (DATASTREAM &ds)
- [~EMRPOLYBEZIER16](#) ()
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.26.1 Detailed Description

EMF Polybezier16.

Draw a polygonal Bezier curve to (what?) using 16-bit points.

### 4.26.2 Constructor & Destructor Documentation

#### **EMRPOLYBEZIER16()** [1/3]

```
EMF::EMRPOLYBEZIER16::EMRPOLYBEZIER16 (  
    const RECTL * bounds,  
    const POINT16 * points,  
    INT n)    [inline]
```

#### Parameters

<i>bounds</i>	overall bounding box of polybezier curve.
<i>points</i>	array of polybezier vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYBEZIER16\(\)](#).

Referenced by [EMRPOLYBEZIER16\(\)](#), [EMRPOLYBEZIER16\(\)](#), and [EMRPOLYBEZIER16\(\)](#).

#### **EMRPOLYBEZIER16()** [2/3]

```
EMF::EMRPOLYBEZIER16::EMRPOLYBEZIER16 (  
    const RECTL * bounds,  
    const POINT * points,  
    INT n)    [inline]
```

Convenience constructor with POINTs.

#### Parameters

<i>bounds</i>	overall bounding box of polybezier curve.
<i>points</i>	array of polybezier vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYBEZIER16\(\)](#).

**EMRPOLYBEZIER16()** [3/3]

```
EMF::EMRPOLYBEZIER16::EMRPOLYBEZIER16 (
    DATASTREAM & ds) [inline]
```

Construct a PolyBezier record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYBEZIER16\(\)](#).

**~EMRPOLYBEZIER16()**

```
EMF::EMRPOLYBEZIER16::~~EMRPOLYBEZIER16 () [inline]
```

Destructor frees a copy of the points it buffered.

**4.26.3 Member Function Documentation****execute()**

```
void EMF::EMRPOLYBEZIER16::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

Reimplemented in [EMF::EMRPOLYBEZIERTO16](#).

**serialize()**

```
bool EMF::EMRPOLYBEZIER16::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

Reimplemented in [EMF::EMRPOLYBEZIERTO16](#).

**size()**

```
int EMF::EMRPOLYBEZIER16::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

Reimplemented in [EMF::EMRPOLYBEZIERTO16](#).

The documentation for this class was generated from the following file:

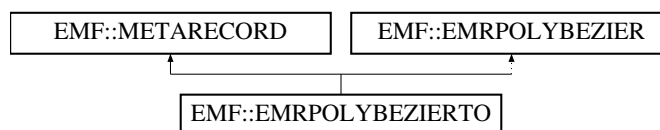
- libemf.h

**4.27 EMF::EMRPOLYBEZIERTO Class Reference**

EMF PolyBezierTo.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYBEZIERTO:

**Public Member Functions**

- [EMRPOLYBEZIERTO](#) (const RECTL \*bounds, const POINT \*points, INT n)
- [EMRPOLYBEZIERTO](#) (DATASTREAM &ds)
- [~EMRPOLYBEZIERTO](#) ()
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

**4.27.1 Detailed Description**

EMF PolyBezierTo.

Draw a polygonal Bezier curve to (what?)

### 4.27.2 Constructor & Destructor Documentation

#### EMRPOLYBEZIERTO() [1/2]

```
EMF::EMRPOLYBEZIERTO::EMRPOLYBEZIERTO (
    const RECTL * bounds,
    const POINT * points,
    INT n) [inline]
```

##### Parameters

<i>bounds</i>	overall bounding box of polybezier curve.
<i>points</i>	array of polybezier vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYBEZIERTO\(\)](#).

Referenced by [EMRPOLYBEZIERTO\(\)](#), and [EMRPOLYBEZIERTO\(\)](#).

#### EMRPOLYBEZIERTO() [2/2]

```
EMF::EMRPOLYBEZIERTO::EMRPOLYBEZIERTO (
    DATASTREAM & ds) [inline]
```

Construct a PolyBezier record from the input stream.

##### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYBEZIERTO\(\)](#).

#### ~EMRPOLYBEZIERTO()

```
EMF::EMRPOLYBEZIERTO::~~EMRPOLYBEZIERTO () [inline]
```

Destructor frees a copy of the points it buffered.

### 4.27.3 Member Function Documentation

#### execute()

```
void EMF::EMRPOLYBEZIERTO::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRPOLYBEZIERTO::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRPOLYBEZIERTO::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

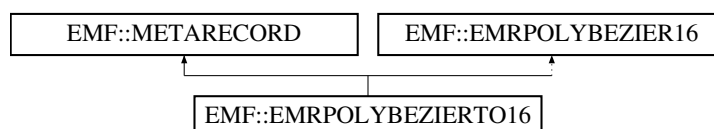
- libemf.h

## 4.28 EMF::EMRPOLYBEZIERTO16 Class Reference

EMF PolyBezierTo16.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYBEZIERTO16:



### Public Member Functions

- [EMRPOLYBEZIERTO16](#) (const RECTL \*bounds, const POINT16 \*points, INT n)
- [EMRPOLYBEZIERTO16](#) (const RECTL \*bounds, const POINT \*points, INT n)
- [EMRPOLYBEZIERTO16](#) (DATASTREAM &ds)
- [~EMRPOLYBEZIERTO16](#) ()
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.28.1 Detailed Description

EMF PolyBezierTo16.

Draw a polygonal Bezier curve to (what?) using 16-bit points

#### 4.28.2 Constructor & Destructor Documentation

##### EMRPOLYBEZIERTO16() [1/3]

```
EMF::EMRPOLYBEZIERTO16::EMRPOLYBEZIERTO16 (
    const RECTL * bounds,
    const POINT16 * points,
    INT n) [inline]
```

##### Parameters

<i>bounds</i>	overall bounding box of polybezier curve.
<i>points</i>	array of polybezier vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYBEZIERTO16\(\)](#).

Referenced by [EMRPOLYBEZIERTO16\(\)](#), [EMRPOLYBEZIERTO16\(\)](#), and [EMRPOLYBEZIERTO16\(\)](#).

**EMRPOLYBEZIERTO16()** [2/3]

```
EMF::EMRPOLYBEZIERTO16::EMRPOLYBEZIERTO16 (
    const RECTL * bounds,
    const POINT * points,
    INT n) [inline]
```

Convenience constructor with POINTs.

**Parameters**

<i>bounds</i>	overall bounding box of polybezier curve.
<i>points</i>	array of polybezier vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYBEZIERTO16\(\)](#).

**EMRPOLYBEZIERTO16()** [3/3]

```
EMF::EMRPOLYBEZIERTO16::EMRPOLYBEZIERTO16 (
    DATASTREAM & ds) [inline]
```

Construct a PolyBezier record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYBEZIERTO16\(\)](#).

**~EMRPOLYBEZIERTO16()**

```
EMF::EMRPOLYBEZIERTO16::~~EMRPOLYBEZIERTO16 () [inline]
```

Destructor frees a copy of the points it buffered.

**4.28.3 Member Function Documentation****execute()**

```
void EMF::EMRPOLYBEZIERTO16::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRPOLYBEZIERTO16::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRPOLYBEZIERTO16::size (  
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

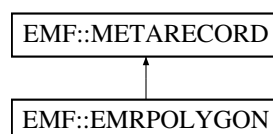
- libemf.h

**4.29 EMF::EMRPOLYGON Class Reference**

EMF Filled Polygon.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYGON:



## Public Member Functions

- [EMRPOLYGON](#) (const RECTL \*bounds, const POINT \*points, INT n)
- [EMRPOLYGON](#) (DATASTREAM &ds)
- [~EMRPOLYGON](#) ()
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.29.1 Detailed Description

EMF Filled Polygon.

Draw a filled polygon.

### 4.29.2 Constructor & Destructor Documentation

#### [EMRPOLYGON\(\)](#) [1/2]

```
EMF::EMRPOLYGON::EMRPOLYGON (
    const RECTL * bounds,
    const POINT * points,
    INT n) [inline]
```

#### Parameters

<i>bounds</i>	overall bounding box of polygon.
<i>points</i>	array of polygon vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYGON\(\)](#).

Referenced by [EMRPOLYGON\(\)](#), and [EMRPOLYGON\(\)](#).

**EMRPOLYGON()** [2/2]

```
EMF::EMRPOLYGON::EMRPOLYGON (
    DATASTREAM & ds) [inline]
```

Construct a Polygon record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYGON\(\)](#).

**~EMRPOLYGON()**

```
EMF::EMRPOLYGON::~~EMRPOLYGON () [inline]
```

Destructor frees a copy of the points it buffered.

**4.29.3 Member Function Documentation****execute()**

```
void EMF::EMRPOLYGON::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRPOLYGON::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRPOLYGON::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

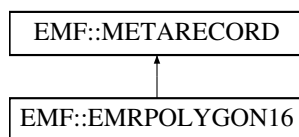
- libemf.h

**4.30 EMF::EMRPOLYGON16 Class Reference**

EMF Filled Polygon16.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYGON16:

**Public Member Functions**

- [EMRPOLYGON16](#) (const RECTL \*bounds, const POINT \*points, INT16 n)
- [EMRPOLYGON16](#) (const RECTL \*bounds, const POINT16 \*points, INT16 n)
- [EMRPOLYGON16](#) (DATASTREAM &ds)
- [~EMRPOLYGON16](#) ()
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

**4.30.1 Detailed Description**

EMF Filled Polygon16.

Draw a filled polygon (with 16-bit points).

### 4.30.2 Constructor & Destructor Documentation

#### EMRPOLYGON16() [1/3]

```
EMF::EMRPOLYGON16::EMRPOLYGON16 (
    const RECTL * bounds,
    const POINT * points,
    INT16 n) [inline]
```

##### Parameters

<i>bounds</i>	overall bounding box of polygon.
<i>points</i>	array of polygon vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYGON16\(\)](#).

Referenced by [EMRPOLYGON16\(\)](#), [EMRPOLYGON16\(\)](#), and [EMRPOLYGON16\(\)](#).

#### EMRPOLYGON16() [2/3]

```
EMF::EMRPOLYGON16::EMRPOLYGON16 (
    const RECTL * bounds,
    const POINT16 * points,
    INT16 n) [inline]
```

Additional constructor which takes a POINT16 array.

##### Parameters

<i>bounds</i>	overall bounding box of polygon.
<i>points</i>	array of polygon vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYGON16\(\)](#).

#### EMRPOLYGON16() [3/3]

```
EMF::EMRPOLYGON16::EMRPOLYGON16 (
    DATASTREAM & ds) [inline]
```

Construct a Polygon record from the input stream.

##### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYGON16\(\)](#).

## ~EMRPOLYGON16()

```
EMF::EMRPOLYGON16::~~EMRPOLYGON16 () [inline]
```

Destructor frees a copy of the points it buffered.

### 4.30.3 Member Function Documentation

#### execute()

```
void EMF::EMRPOLYGON16::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRPOLYGON16::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRPOLYGON16::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

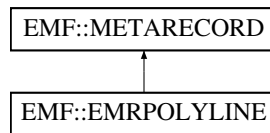
- libemf.h

## 4.31 EMF::EMRPOLYLINE Class Reference

EMF Polyline.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYLINE:



### Public Member Functions

- [EMRPOLYLINE](#) (const RECTL \*bounds, const POINT \*points, INT n)
- [~EMRPOLYLINE](#) ()
- [EMRPOLYLINE](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.31.1 Detailed Description

EMF Polyline.

Draw a series of connected lines.

#### 4.31.2 Constructor & Destructor Documentation

##### EMRPOLYLINE() [1/2]

```

EMF::EMRPOLYLINE::EMRPOLYLINE (
    const RECTL * bounds,
    const POINT * points,
    INT n) [inline]
  
```

##### Parameters

<i>bounds</i>	overall bounding box of polyline.
<i>points</i>	array of polyline vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYLINE\(\)](#).

Referenced by [EMRPOLYLINE\(\)](#), and [EMRPOLYLINE\(\)](#).

## ~EMRPOLYLINE()

```
EMF::EMRPOLYLINE::~~EMRPOLYLINE () [inline]
```

Destructor frees a copy of the points it buffered.

## EMRPOLYLINE() [2/2]

```
EMF::EMRPOLYLINE::EMRPOLYLINE (  
    DATASTREAM & ds) [inline]
```

Construct a Polyline record from the input stream.

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYLINE\(\)](#).

## 4.31.3 Member Function Documentation

### execute()

```
void EMF::EMRPOLYLINE::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

### serialize()

```
bool EMF::EMRPOLYLINE::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRPOLYLINE16::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

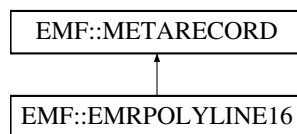
- libemf.h

**4.32 EMF::EMRPOLYLINE16 Class Reference**

EMF Polyline16.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYLINE16:

**Public Member Functions**

- [EMRPOLYLINE16](#) (const RECTL \*bounds, const POINT16 \*points, INT n)
- [EMRPOLYLINE16](#) (const RECTL \*bounds, const POINT \*points, INT n)
- [~EMRPOLYLINE16](#) ()
- [EMRPOLYLINE16](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

**4.32.1 Detailed Description**

EMF Polyline16.

Draw a series of connected lines using 16-bit points.

## 4.32.2 Constructor & Destructor Documentation

### EMRPOLYLINE16() [1/3]

```
EMF::EMRPOLYLINE16::EMRPOLYLINE16 (
    const RECTL * bounds,
    const POINT16 * points,
    INT n) [inline]
```

#### Parameters

<i>bounds</i>	overall bounding box of polyline.
<i>points</i>	array of polyline vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYLINE16\(\)](#).

Referenced by [EMRPOLYLINE16\(\)](#), [EMRPOLYLINE16\(\)](#), and [EMRPOLYLINE16\(\)](#).

### EMRPOLYLINE16() [2/3]

```
EMF::EMRPOLYLINE16::EMRPOLYLINE16 (
    const RECTL * bounds,
    const POINT * points,
    INT n) [inline]
```

Constructor with POINTs.

#### Parameters

<i>bounds</i>	overall bounding box of polyline.
<i>points</i>	array of polyline vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYLINE16\(\)](#).

### ~EMRPOLYLINE16()

```
EMF::EMRPOLYLINE16::~~EMRPOLYLINE16 () [inline]
```

Destructor frees a copy of the points it buffered.

**EMRPOLYLINE16()** [3/3]

```
EMF::EMRPOLYLINE16::EMRPOLYLINE16 (
    DATASTREAM & ds) [inline]
```

Construct a Polyline record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYLINE16\(\)](#).

**4.32.3 Member Function Documentation****execute()**

```
void EMF::EMRPOLYLINE16::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRPOLYLINE16::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRPOLYLINE16::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

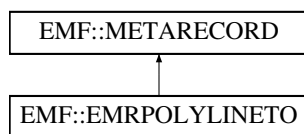
- libemf.h

### 4.33 EMF::EMRPOLYLINETO Class Reference

EMF PolylineTo.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYLINETO:



#### Public Member Functions

- [EMRPOLYLINETO](#) (const RECTL \*bounds, const POINT \*points, INT n)
- [EMRPOLYLINETO](#) (DATASTREAM &ds)
- [~EMRPOLYLINETO](#) ()
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

#### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.33.1 Detailed Description

EMF PolylineTo.

Draw a polygonal line curve to (what?)

#### 4.33.2 Constructor & Destructor Documentation

##### EMRPOLYLINETO() [1/2]

```

EMF::EMRPOLYLINETO::EMRPOLYLINETO (
    const RECTL * bounds,
    const POINT * points,
    INT n) [inline]
  
```

#### Parameters

<i>bounds</i>	overall bounding box of polybezier curve.
<i>points</i>	array of polybezier vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYLINETO\(\)](#).

Referenced by [EMRPOLYLINETO\(\)](#), and [EMRPOLYLINETO\(\)](#).

**EMRPOLYLINETO()** [2/2]

```
EMF::EMRPOLYLINETO::EMRPOLYLINETO (
    DATASTREAM & ds) [inline]
```

Construct a PolylineTo record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYLINETO\(\)](#).

**~EMRPOLYLINETO()**

```
EMF::EMRPOLYLINETO::~~EMRPOLYLINETO () [inline]
```

Destructor frees a copy of the points it buffered.

**4.33.3 Member Function Documentation****execute()**

```
void EMF::EMRPOLYLINETO::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRPOLYLINETO::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRPOLYLINETO::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

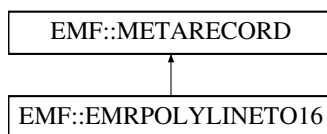
- libemf.h

## 4.34 EMF::EMRPOLYLINETO16 Class Reference

EMF PolylineTo16.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYLINETO16:



### Public Member Functions

- [EMRPOLYLINETO16](#) (const RECTL \*bounds, const POINT16 \*points, INT n)
- [EMRPOLYLINETO16](#) (const RECTL \*bounds, const POINT \*points, INT n)
- [EMRPOLYLINETO16](#) (DATASTREAM &ds)
- [~EMRPOLYLINETO16](#) ()
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.34.1 Detailed Description

EMF PolylineTo16.

Draw a polygonal line curve to (what?)

### 4.34.2 Constructor & Destructor Documentation

#### EMRPOLYLINETO16() [1/3]

```
EMF::EMRPOLYLINETO16::EMRPOLYLINETO16 (
    const RECTL * bounds,
    const POINT16 * points,
    INT n) [inline]
```

##### Parameters

<i>bounds</i>	overall bounding box of polybezier curve.
<i>points</i>	array of polybezier vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYLINETO16\(\)](#).

Referenced by [EMRPOLYLINETO16\(\)](#), [EMRPOLYLINETO16\(\)](#), and [EMRPOLYLINETO16\(\)](#).

#### EMRPOLYLINETO16() [2/3]

```
EMF::EMRPOLYLINETO16::EMRPOLYLINETO16 (
    const RECTL * bounds,
    const POINT * points,
    INT n) [inline]
```

Convenience constructor with POINTs.

##### Parameters

<i>bounds</i>	overall bounding box of polybezier curve.
<i>points</i>	array of polybezier vertices.
<i>n</i>	number of vertices in points.

References [EMRPOLYLINETO16\(\)](#).

#### EMRPOLYLINETO16() [3/3]

```
EMF::EMRPOLYLINETO16::EMRPOLYLINETO16 (
    DATASTREAM & ds) [inline]
```

Construct a PolylineTo record from the input stream.

##### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYLINETO16\(\)](#).

## ~EMRPOLYLINETO16()

```
EMF::EMRPOLYLINETO16::~~EMRPOLYLINETO16 () [inline]
```

Destructor frees a copy of the points it buffered.

### 4.34.3 Member Function Documentation

#### execute()

```
void EMF::EMRPOLYLINETO16::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRPOLYLINETO16::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRPOLYLINETO16::size (  
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

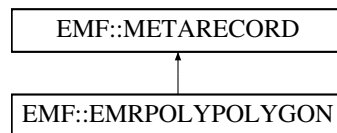
- libemf.h

## 4.35 EMF::EMRPOLYPOLYGON Class Reference

EMF Poly Polygon.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYPOLYGON:



### Public Member Functions

- [EMRPOLYPOLYGON](#) (const RECTL \*bounds, const POINT \*points, const INT \*counts, UINT polygons)
- [~EMRPOLYPOLYGON](#) ()
- [EMRPOLYPOLYGON](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.35.1 Detailed Description

EMF Poly Polygon.

Draw several filled polygons.

#### 4.35.2 Constructor & Destructor Documentation

##### EMRPOLYPOLYGON() [1/2]

```

EMF::EMRPOLYPOLYGON::EMRPOLYPOLYGON (
    const RECTL * bounds,
    const POINT * points,
    const INT * counts,
    UINT polygons) [inline]
  
```

### Parameters

<i>bounds</i>	overall bounding box of polygon.
<i>points</i>	array of polygon vertices.

<i>counts</i>	array of number of vertices in each polygon.
<i>polygons</i>	number of polygons.

References [EMRPOLYPOLYGON\(\)](#).

Referenced by [EMRPOLYPOLYGON\(\)](#), and [EMRPOLYPOLYGON\(\)](#).

### ~EMRPOLYPOLYGON()

```
EMF::EMRPOLYPOLYGON::~~EMRPOLYPOLYGON () [inline]
```

Destructor frees a copy of the counts and points it buffered.

### EMRPOLYPOLYGON() [2/2]

```
EMF::EMRPOLYPOLYGON::EMRPOLYPOLYGON (  
    DATASTREAM & ds) [inline]
```

Construct a Polygon record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYPOLYGON\(\)](#).

## 4.35.3 Member Function Documentation

### execute()

```
void EMF::EMRPOLYPOLYGON::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRPOLYPOLYGON::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRPOLYPOLYGON::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

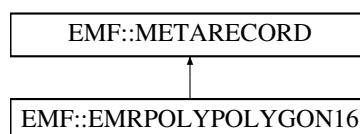
- libemf.h

**4.36 EMF::EMRPOLYPOLYGON16 Class Reference**

EMF Poly Polygon16.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRPOLYPOLYGON16:

**Public Member Functions**

- [EMRPOLYPOLYGON16](#) (const RECTL \*bounds, const POINT \*points, const INT \*counts, UINT polygons)
- [EMRPOLYPOLYGON16](#) (const RECTL \*bounds, const POINT16 \*points, const INT \*counts, UINT16 polygons)
- [~EMRPOLYPOLYGON16](#) ()
- [EMRPOLYPOLYGON16](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.36.1 Detailed Description

EMF Poly Polygon16.

Draw several filled polygons (with 16-bit points).

### 4.36.2 Constructor & Destructor Documentation

#### EMRPOLYPOLYGON16() [1/3]

```
EMF::EMRPOLYPOLYGON16::EMRPOLYPOLYGON16 (
    const RECTL * bounds,
    const POINT * points,
    const INT * counts,
    UINT polygons) [inline]
```

#### Parameters

<i>bounds</i>	overall bounding box of polygon.
<i>points</i>	array of polygon vertices.
<i>counts</i>	array of number of vertices in each polygon.
<i>polygons</i>	number of polygons.

References [EMRPOLYPOLYGON16\(\)](#).

Referenced by [EMRPOLYPOLYGON16\(\)](#), [EMRPOLYPOLYGON16\(\)](#), and [EMRPOLYPOLYGON16\(\)](#).

#### EMRPOLYPOLYGON16() [2/3]

```
EMF::EMRPOLYPOLYGON16::EMRPOLYPOLYGON16 (
    const RECTL * bounds,
    const POINT16 * points,
    const INT * counts,
    UINT16 polygons) [inline]
```

Additional constructor which takes a POINT16 structure.

#### Parameters

<i>bounds</i>	overall bounding box of polygon.
<i>points</i>	array of polygon vertices.
<i>counts</i>	array of number of vertices in each polygon.

<i>polygons</i>	number of polygons.
-----------------	---------------------

References [EMRPOLYPOLYGON16\(\)](#).

### ~EMRPOLYPOLYGON16()

```
EMF::EMRPOLYPOLYGON16::~~EMRPOLYPOLYGON16 () [inline]
```

Destructor frees a copy of the counts and points it buffered.

### EMRPOLYPOLYGON16() [3/3]

```
EMF::EMRPOLYPOLYGON16::EMRPOLYPOLYGON16 (
    DATASTREAM & ds) [inline]
```

Construct a Polygon record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

References [EMRPOLYPOLYGON16\(\)](#).

## 4.36.3 Member Function Documentation

### execute()

```
void EMF::EMRPOLYPOLYGON16::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRPOLYPOLYGON16::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRPOLYPOLYGON16::size (  
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

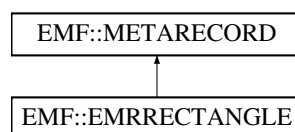
- libemf.h

## 4.37 EMF::EMRRECTANGLE Class Reference

EMF Rectangle.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRRECTANGLE:

**Public Member Functions**

- [EMRRECTANGLE](#) (INT left, INT top, INT right, INT bottom)
- [EMRRECTANGLE](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

### 4.37.1 Detailed Description

EMF Rectangle.

Draw a rectangle.

### 4.37.2 Constructor & Destructor Documentation

#### EMRRECTANGLE() [1/2]

```
EMF::EMRRECTANGLE::EMRRECTANGLE (
    INT left,
    INT top,
    INT right,
    INT bottom) [inline]
```

#### Parameters

<i>left</i>	x position of left side of rectangle.
<i>top</i>	y position of top side of rectangle.
<i>right</i>	x position of right edge of rectangle.
<i>bottom</i>	y position of bottom edge of rectangle.

References [EMRRECTANGLE\(\)](#).

Referenced by [EMRRECTANGLE\(\)](#).

#### EMRRECTANGLE() [2/2]

```
EMF::EMRRECTANGLE::EMRRECTANGLE (
    DATASTREAM & ds) [inline]
```

Construct a Rectangle record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.37.3 Member Function Documentation

#### execute()

```
void EMF::EMRRECTANGLE::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRRECTANGLE::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRRECTANGLE::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

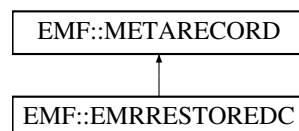
- libemf.h

## 4.38 EMF::EMRRESTOREDC Class Reference

EMF Restore DC.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRRESTOREDC:



## Public Member Functions

- [EMRSTOREDC](#) (INT n)
- [EMRSTOREDC](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.38.1 Detailed Description

EMF Restore DC.

Use the stored device context in this context(?)

### 4.38.2 Constructor & Destructor Documentation

#### EMRSTOREDC() [1/2]

```
EMF::EMRSTOREDC::EMRSTOREDC (
    INT n) [inline]
```

Create a Restore DC record.

References [EMRSTOREDC\(\)](#).

Referenced by [EMRSTOREDC\(\)](#).

#### EMRSTOREDC() [2/2]

```
EMF::EMRSTOREDC::EMRSTOREDC (
    DATASTREAM & ds) [inline]
```

Construct an Restoredc record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.38.3 Member Function Documentation

#### execute()

```
void EMF::EMRRESTOREDC::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRRESTOREDC::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRRESTOREDC::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

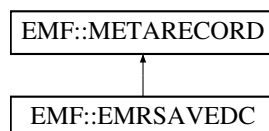
- libemf.h

## 4.39 EMF::EMRSAVEDC Class Reference

EMF Save DC.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSAVEDC:



## Public Member Functions

- [EMRSAVEDC](#) (void)
- [EMRSAVEDC](#) ([DATASTREAM](#) &ds)
- bool [serialize](#) ([DATASTREAM](#) ds)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.39.1 Detailed Description

EMF Save DC.

Save the device context (i.e., push contents on a stack of some variety?)

### 4.39.2 Constructor & Destructor Documentation

#### [EMRSAVEDC\(\)](#) [1/2]

```
EMF::EMRSAVEDC::EMRSAVEDC (
    void ) [inline]
```

Create a Save DC record.

References [EMRSAVEDC\(\)](#).

Referenced by [EMRSAVEDC\(\)](#).

#### [EMRSAVEDC\(\)](#) [2/2]

```
EMF::EMRSAVEDC::EMRSAVEDC (
    DATASTREAM & ds) [inline]
```

Construct an Savedc record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.39.3 Member Function Documentation

#### execute()

```
void EMF::EMRSAVEDC::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRSAVEDC::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRSAVEDC::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

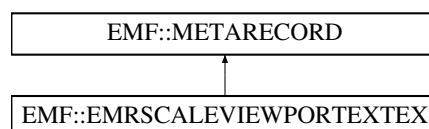
- libemf.h

## 4.40 EMF::EMRSCALEVIEWPORTEXTEX Class Reference

EMF Scale Viewport Extents (ex).

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSCALEVIEWPORTEXTEX:



## Public Member Functions

- [EMRSCALEVIEWPORTEXTEX](#) (LONG *x\_num*, LONG *x\_den*, LONG *y\_num*, LONG *y\_den*)
- [EMRSCALEVIEWPORTEXTEX](#) ([DATASTREAM](#) &ds)
- bool [serialize](#) ([DATASTREAM](#) ds)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.40.1 Detailed Description

EMF Scale Viewport Extents (ex).

The viewport extent is the device coordinate (i.e. pixels) size of the viewport. Scale the viewport extents by the ratios of the given values. (OpenOffice accepts this, but not SETVIEWPORTEXT(?))

### 4.40.2 Constructor & Destructor Documentation

#### EMRSCALEVIEWPORTEXTEX() [1/2]

```
EMF::EMRSCALEVIEWPORTEXTEX::EMRSCALEVIEWPORTEXTEX (
    LONG x_num,
    LONG x_den,
    LONG y_num,
    LONG y_den) [inline]
```

#### Parameters

<i>x_num</i>	numerator of x scale
<i>x_den</i>	denominator of x scale
<i>y_num</i>	numerator of y scale
<i>y_den</i>	denominator of y scale

References [EMRSCALEVIEWPORTEXTEX\(\)](#).

Referenced by [EMRSCALEVIEWPORTEXTEX\(\)](#).

**EMRSCALEVIEWPORTEXTEx() [2/2]**

```
EMF::EMRSCALEVIEWPORTEXTEx::EMRSCALEVIEWPORTEXTEx (
    DATASTREAM & ds) [inline]
```

Construct a ScaleViewportExtEx record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.40.3 Member Function Documentation****execute()**

```
void EMF::EMRSCALEVIEWPORTEXTEx::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSCALEVIEWPORTEXTEx::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSCALEVIEWPORTEXTEx::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

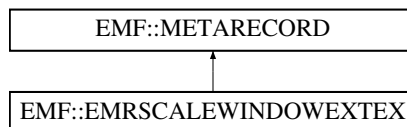
- libemf.h

## 4.41 EMF::EMRSCALEWINDOWEXTEx Class Reference

EMF Scale Window Extents (ex).

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSCALEWINDOWEXTEx:



### Public Member Functions

- [EMRSCALEWINDOWEXTEx](#) (LONG x\_num, LONG x\_den, LONG y\_num, LONG y\_den)
- [EMRSCALEWINDOWEXTEx](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.41.1 Detailed Description

EMF Scale Window Extents (ex).

The window extent is the logical coordinate size of the window. Scale the window extents by the ratios of the given values.

#### 4.41.2 Constructor & Destructor Documentation

##### EMRSCALEWINDOWEXTEx() [1/2]

```
EMF::EMRSCALEWINDOWEXTEx::EMRSCALEWINDOWEXTEx (
    LONG x_num,
    LONG x_den,
    LONG y_num,
    LONG y_den) [inline]
```

### Parameters

<i>x_num</i>	numerator of x scale
<i>x_den</i>	denominator of x scale

<i>y_num</i>	numerator of y scale
<i>y_den</i>	denominator of y scale

References [EMRSCALEWINDOWEXTEx\(\)](#).

Referenced by [EMRSCALEWINDOWEXTEx\(\)](#).

## EMRSCALEWINDOWEXTEx() [2/2]

```
EMF::EMRSCALEWINDOWEXTEx::EMRSCALEWINDOWEXTEx (  
    DATASTREAM & ds) [inline]
```

Construct a ScaleWindowExtEx record from the input stream.

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

## 4.41.3 Member Function Documentation

### execute()

```
void EMF::EMRSCALEWINDOWEXTEx::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

### serialize()

```
bool EMF::EMRSCALEWINDOWEXTEx::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSCALEWINDOWEXTEX::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

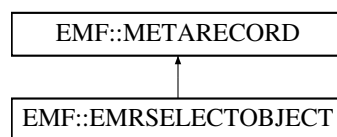
- libemf.h

**4.42 EMF::EMRSELECTOBJECT Class Reference**

EMF Select Object.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSELECTOBJECT:

**Public Member Functions**

- [EMRSELECTOBJECT](#) (HGDIOBJ object)
- [EMRSELECTOBJECT](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

**4.42.1 Detailed Description**

EMF Select Object.

Activate (make current) the given object, such as a pen, brush or font.

#### 4.42.2 Constructor & Destructor Documentation

##### EMRSELECTOBJECT() [1/2]

```
EMF::EMRSELECTOBJECT::EMRSELECTOBJECT (
    HGDIOBJ object) [inline]
```

###### Parameters

<i>object</i>	the object to make active.
---------------	----------------------------

References [EMRSELECTOBJECT\(\)](#).

Referenced by [EMRSELECTOBJECT\(\)](#).

##### EMRSELECTOBJECT() [2/2]

```
EMF::EMRSELECTOBJECT::EMRSELECTOBJECT (
    DATASTREAM & ds) [inline]
```

Construct a SelectObject record from the input stream.

###### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

#### 4.42.3 Member Function Documentation

##### execute()

```
void EMF::EMRSELECTOBJECT::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [virtual]
```

Execute this record in the context of the given device context.

###### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

References [EMF::METAFILEDEVICECONTEXT::emf\\_handles](#).

**serialize()**

```
bool EMF::EMRSELECTOBJECT::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSELECTOBJECT::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following files:

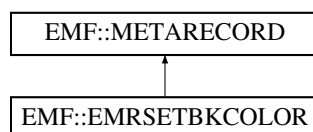
- libemf.h
- libemf.cpp

**4.43 EMF::EMRSETBKCOLOR Class Reference**

EMF Set Background Color.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETBKCOLOR:

**Public Member Functions**

- [EMRSETBKCOLOR](#) (COLORREF color)
- [EMRSETBKCOLOR](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

#### 4.43.1 Detailed Description

EMF Set Background Color.

Sets the background color.

#### 4.43.2 Constructor & Destructor Documentation

##### EMRSETBKCOLOR() [1/2]

```
EMF::EMRSETBKCOLOR::EMRSETBKCOLOR (
    COLORREF color) [inline]
```

##### Parameters

<i>color</i>	background color
--------------	------------------

References [EMRSETBKCOLOR\(\)](#).

Referenced by [EMRSETBKCOLOR\(\)](#).

##### EMRSETBKCOLOR() [2/2]

```
EMF::EMRSETBKCOLOR::EMRSETBKCOLOR (
    DATASTREAM & ds) [inline]
```

Construct a SetBkColor record from the input datastream.

##### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

#### 4.43.3 Member Function Documentation

##### execute()

```
void EMF::EMRSETBKCOLOR::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

##### Parameters

<i>source</i>	the device context from which this record is taken.
---------------	---

<i>dc</i>	device context for execute.
-----------	-----------------------------

Implements [EMF::METARECORD](#).

### serialize()

```
bool EMF::EMRSETBKCOLOR::serialize (
    DATASTREAM ds) [inline], [virtual]
```

### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

### size()

```
int EMF::EMRSETBKCOLOR::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

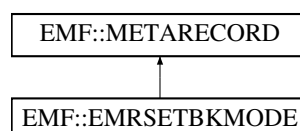
- libemf.h

## 4.44 EMF::EMRSETBKMODE Class Reference

EMF Set Background Mode.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETBKMODE:



### Public Member Functions

- [EMRSETBKMODE](#) (DWORD mode)
- [EMRSETBKMODE](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

**Public Member Functions inherited from [EMF::METARECORD](#)**

- virtual [~METARECORD](#) ()

**4.44.1 Detailed Description**

EMF Set Background Mode.

Set the background mode: transparent or opaque. Seems to be ignored by StarOffice. (Appears to work for text, though.)

**4.44.2 Constructor & Destructor Documentation****EMRSETBKMODE() [1/2]**

```
EMF::EMRSETBKMODE::EMRSETBKMODE (  
    DWORD mode) [inline]
```

**Parameters**

<i>mode</i>	background mode.
-------------	------------------

References [EMRSETBKMODE\(\)](#).

Referenced by [EMRSETBKMODE\(\)](#).

**EMRSETBKMODE() [2/2]**

```
EMF::EMRSETBKMODE::EMRSETBKMODE (  
    DATASTREAM & ds) [inline]
```

Construct a SetBkMode record from the input datastream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.44.3 Member Function Documentation

#### execute()

```
void EMF::EMRSETBKMODE::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRSETBKMODE::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRSETBKMODE::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

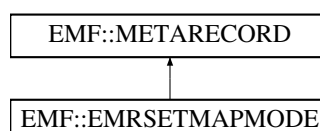
- libemf.h

## 4.45 EMF::EMRSETMAPMODE Class Reference

EMF Set Mapping Mode.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETMAPMODE:



## Public Member Functions

- [EMRSETMAPMODE](#) (DWORD mode)
- [EMRSETMAPMODE](#) ([DATASTREAM](#) &ds)
- bool [serialize](#) ([DATASTREAM](#) ds)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.45.1 Detailed Description

EMF Set Mapping Mode.

Set the mapping mode: HI (X style), LO (OpenGL style). Totally ignored by StarOffice as near as I can tell.

### 4.45.2 Constructor & Destructor Documentation

#### [EMRSETMAPMODE\(\)](#) [1/2]

```
EMF::EMRSETMAPMODE::EMRSETMAPMODE (
    DWORD mode)    [inline]
```

#### Parameters

<i>mode</i>	window mapping mode
-------------	---------------------

References [EMRSETMAPMODE\(\)](#).

Referenced by [EMRSETMAPMODE\(\)](#).

#### [EMRSETMAPMODE\(\)](#) [2/2]

```
EMF::EMRSETMAPMODE::EMRSETMAPMODE (
    DATASTREAM & ds)    [inline]
```

Construct a SetMapMode record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.45.3 Member Function Documentation

#### execute()

```
void EMF::EMRSETMAPMODE::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRSETMAPMODE::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRSETMAPMODE::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

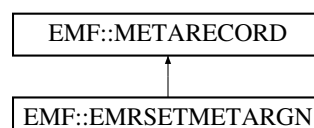
- libemf.h

## 4.46 EMF::EMRSETMETARGN Class Reference

EMF Set Meta Region.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETMETARGN:



## Public Member Functions

- [EMRSETMETARGN](#) (void)
- [EMRSETMETARGN](#) ([DATASTREAM](#) &ds)
- bool [serialize](#) ([DATASTREAM](#) ds)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.46.1 Detailed Description

EMF Set Meta Region.

I really have no idea.

### 4.46.2 Constructor & Destructor Documentation

#### [EMRSETMETARGN\(\)](#) [1/2]

```
EMF::EMRSETMETARGN::EMRSETMETARGN (
    void )    [inline]
```

Create a Set Meta Rgn record.

References [EMRSETMETARGN\(\)](#).

Referenced by [EMRSETMETARGN\(\)](#).

#### [EMRSETMETARGN\(\)](#) [2/2]

```
EMF::EMRSETMETARGN::EMRSETMETARGN (
    DATASTREAM & ds)    [inline]
```

Construct an Setmetargn record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.46.3 Member Function Documentation

#### execute()

```
void EMF::EMRSETMETARGN::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRSETMETARGN::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRSETMETARGN::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

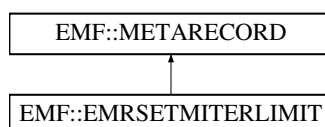
- libemf.h

## 4.47 EMF::EMRSETMITERLIMIT Class Reference

EMF SetMiterLimit.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETMITERLIMIT:



## Public Member Functions

- [EMRSETMITERLIMIT](#) (FLOAT limit)
- [EMRSETMITERLIMIT](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.47.1 Detailed Description

EMF SetMiterLimit.

Sets the length limit for miter joins.

### 4.47.2 Constructor & Destructor Documentation

#### EMRSETMITERLIMIT() [1/2]

```
EMF::EMRSETMITERLIMIT::EMRSETMITERLIMIT (
    FLOAT limit) [inline]
```

#### Parameters

<i>limit</i>	miter length limit.
--------------	---------------------

References [EMRSETMITERLIMIT\(\)](#).

Referenced by [EMRSETMITERLIMIT\(\)](#).

#### EMRSETMITERLIMIT() [2/2]

```
EMF::EMRSETMITERLIMIT::EMRSETMITERLIMIT (
    DATASTREAM & ds) [inline]
```

Construct a SetMiterLimit record from the input stream.

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.47.3 Member Function Documentation

#### execute()

```
void EMF::EMRSETMITERLIMIT::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRSETMITERLIMIT::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRSETMITERLIMIT::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

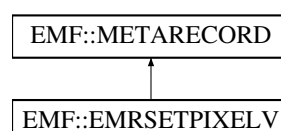
- libemf.h

## 4.48 EMF::EMRSETPIXELV Class Reference

EMF Set Pixel.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETPIXELV:



## Public Member Functions

- [EMRSETPIXELV](#) (INT *x*, INT *y*, COLORREF *color*)
- [EMRSETPIXELV](#) ([DATASTREAM](#) &*ds*)
- bool [serialize](#) ([DATASTREAM](#) *ds*)
- int [size](#) (void) const
- void [execute](#) ([METAFILEDEVICECONTEXT](#) \**source*, HDC *dc*) const

## Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

### 4.48.1 Detailed Description

EMF Set Pixel.

Set the given pixel to the given color.

### 4.48.2 Constructor & Destructor Documentation

#### [EMRSETPIXELV\(\)](#) [1/2]

```
EMF::EMRSETPIXELV::EMRSETPIXELV (
    INT x,
    INT y,
    COLORREF color) [inline]
```

#### Parameters

<i>x</i>	x position at which to draw pixel.
<i>y</i>	y position at which to draw pixel.
<i>color</i>	color of pixel.

References [EMRSETPIXELV\(\)](#).

Referenced by [EMRSETPIXELV\(\)](#).

#### [EMRSETPIXELV\(\)](#) [2/2]

```
EMF::EMRSETPIXELV::EMRSETPIXELV (
    DATASTREAM & ds) [inline]
```

Construct a SetPixelV record from the input stream.

#### Parameters

---

<i>ds</i>	Metafile datastream.
-----------	----------------------

### 4.48.3 Member Function Documentation

#### execute()

```
void EMF::EMRSETPIXELV::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

#### serialize()

```
bool EMF::EMRSETPIXELV::serialize (
    DATASTREAM ds) [inline], [virtual]
```

#### Parameters

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

#### size()

```
int EMF::EMRSETPIXELV::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

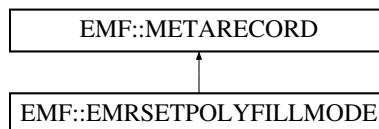
- libemf.h

## 4.49 EMF::EMRSETPOLYFILLMODE Class Reference

EMF Set the Polygon Fill Mode.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETPOLYFILLMODE:



### Public Member Functions

- [EMRSETPOLYFILLMODE](#) (DWORD mode)
- [EMRSETPOLYFILLMODE](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.49.1 Detailed Description

EMF Set the Polygon Fill Mode.

Set the polygon fill mode: ALTERNATE or WINDING

#### 4.49.2 Constructor & Destructor Documentation

##### EMRSETPOLYFILLMODE() [1/2]

```
EMF::EMRSETPOLYFILLMODE::EMRSETPOLYFILLMODE (
    DWORD mode) [inline]
```

### Parameters

<i>mode</i>	background mode.
-------------	------------------

References [EMRSETPOLYFILLMODE\(\)](#).

Referenced by [EMRSETPOLYFILLMODE\(\)](#).

**EMRSETPOLYFILLMODE()** [2/2]

```
EMF::EMRSETPOLYFILLMODE::EMRSETPOLYFILLMODE (
    DATASTREAM & ds) [inline]
```

Construct a SetPolyFillMode record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.49.3 Member Function Documentation****execute()**

```
void EMF::EMRSETPOLYFILLMODE::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSETPOLYFILLMODE::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSETPOLYFILLMODE::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

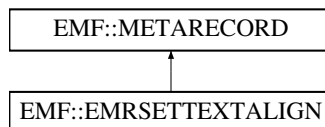
- libemf.h

## 4.50 EMF::EMRSETTEXTALIGN Class Reference

EMF Set Text Alignment.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETTEXTALIGN:



### Public Member Functions

- [EMRSETTEXTALIGN](#) (UINT mode)
- [EMRSETTEXTALIGN](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.50.1 Detailed Description

EMF Set Text Alignment.

Determines the justification of the text with respect to its position.

#### 4.50.2 Constructor & Destructor Documentation

##### EMRSETTEXTALIGN() [1/2]

```
EMF::EMRSETTEXTALIGN::EMRSETTEXTALIGN (
    UINT mode) [inline]
```

##### Parameters

<i>mode</i>	text alignment mode.
-------------	----------------------

References [EMRSETTEXTALIGN\(\)](#).

Referenced by [EMRSETTEXTALIGN\(\)](#).

**EMRSETTEXTALIGN()** [2/2]

```
EMF::EMRSETTEXTALIGN::EMRSETTEXTALIGN (
    DATASTREAM & ds) [inline]
```

Construct a SetTextAlign record from the input datastream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.50.3 Member Function Documentation****execute()**

```
void EMF::EMRSETTEXTALIGN::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSETTEXTALIGN::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSETTEXTALIGN::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

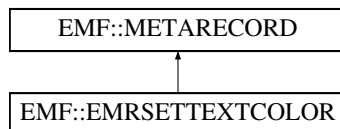
- libemf.h

## 4.51 EMF::EMRSETTEXTCOLOR Class Reference

EMF Set Text Color.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETTEXTCOLOR:



### Public Member Functions

- [EMRSETTEXTCOLOR](#) (COLORREF color)
- [EMRSETTEXTCOLOR](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.51.1 Detailed Description

EMF Set Text Color.

Sets the foreground color of text.

#### 4.51.2 Constructor & Destructor Documentation

##### EMRSETTEXTCOLOR() [1/2]

```
EMF::EMRSETTEXTCOLOR::EMRSETTEXTCOLOR (
    COLORREF color) [inline]
```

### Parameters

<i>color</i>	text foreground color
--------------	-----------------------

References [EMRSETTEXTCOLOR\(\)](#).

Referenced by [EMRSETTEXTCOLOR\(\)](#).

**EMRSETTEXTCOLOR()** [2/2]

```
EMF::EMRSETTEXTCOLOR::EMRSETTEXTCOLOR (
    DATASTREAM & ds) [inline]
```

Construct a SetTextColor record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.51.3 Member Function Documentation****execute()**

```
void EMF::EMRSETTEXTCOLOR::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSETTEXTCOLOR::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSETTEXTCOLOR::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

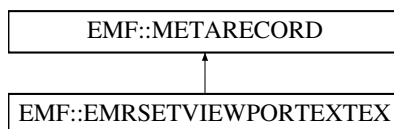
- libemf.h

## 4.52 EMF::EMRSETVIEWPORTETEX Class Reference

EMF Set Viewport Extents (ex).

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETVIEWPORTETEX:



### Public Member Functions

- [EMRSETVIEWPORTETEX](#) (INT cx, INT cy)
- [EMRSETVIEWPORTETEX](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.52.1 Detailed Description

EMF Set Viewport Extents (ex).

The viewport extent is the device coordinate (i.e. pixels) size of the viewport. Since W32 doesn't do any clipping, the purpose of this is not clear.

#### 4.52.2 Constructor & Destructor Documentation

##### EMRSETVIEWPORTETEX() [1/2]

```

EMF::EMRSETVIEWPORTETEX::EMRSETVIEWPORTETEX (
    INT cx,
    INT cy) [inline]
  
```

### Parameters

<code>cx</code>	width of viewport in device coordinates
<code>cy</code>	height of viewport in device coordinates

References [EMRSETVIEWPORTETEX\(\)](#).

Referenced by [EMRSETVIEWPORTETEX\(\)](#).

**EMRSETVIEWPORTEXTEX()** [2/2]

```
EMF::EMRSETVIEWPORTEXTEX::EMRSETVIEWPORTEXTEX (
    DATASTREAM & ds) [inline]
```

Construct a SetViewportExtEx record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.52.3 Member Function Documentation****execute()**

```
void EMF::EMRSETVIEWPORTEXTEX::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSETVIEWPORTEXTEX::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSETVIEWPORTEXTEX::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

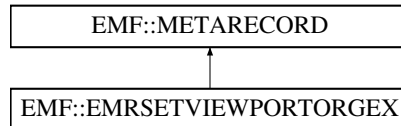
- libemf.h

## 4.53 EMF::EMRSETVIEWPORTORGEX Class Reference

EMF Set Viewport Origin (ex).

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETVIEWPORTORGEX:



### Public Member Functions

- [EMRSETVIEWPORTORGEX](#) (INT x, INT y)
- [EMRSETVIEWPORTORGEX](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.53.1 Detailed Description

EMF Set Viewport Origin (ex).

The viewport origin is a point in device coordinates (i.e., pixels) where the viewport starts. (For example, if you want to put several different views on the same page, you might use different viewports.)

#### 4.53.2 Constructor & Destructor Documentation

##### EMRSETVIEWPORTORGEX() [1/2]

```
EMF::EMRSETVIEWPORTORGEX::EMRSETVIEWPORTORGEX (
    INT x,
    INT y) [inline]
```

### Parameters

<i>x</i>	x position of the viewport in device coordinates
<i>y</i>	y position of the viewport in device coordinates

References [EMRSETVIEWPORTORGEX\(\)](#).

Referenced by [EMRSETVIEWPORTORGEX\(\)](#).

**EMRSETVIEWPORTORGEX()** [2/2]

```
EMF::EMRSETVIEWPORTORGEX::EMRSETVIEWPORTORGEX (
    DATASTREAM & ds) [inline]
```

Construct a SetVieportOrgEx record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.53.3 Member Function Documentation****execute()**

```
void EMF::EMRSETVIEWPORTORGEX::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSETVIEWPORTORGEX::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSETVIEWPORTORGEX::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

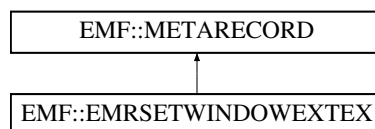
- libemf.h

## 4.54 EMF::EMRSETWINDOWEXTEx Class Reference

EMF Set Window Extent (ex).

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETWINDOWEXTEx:



### Public Member Functions

- [EMRSETWINDOWEXTEx](#) (INT cx, INT cy)
- [EMRSETWINDOWEXTEx](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.54.1 Detailed Description

EMF Set Window Extent (ex).

The window extents define the scale of the logical coordinates. For example, if your XY plot is from [-10,-10] to [10,10], then the window extents are [20,20].

#### 4.54.2 Constructor & Destructor Documentation

##### EMRSETWINDOWEXTEx() [1/2]

```
EMF::EMRSETWINDOWEXTEx::EMRSETWINDOWEXTEx (
    INT cx,
    INT cy) [inline]
```

### Parameters

<i>cx</i>	width of window in logical coordinates.
<i>cy</i>	height of window in logical coordinates.

References [EMRSETWINDOWEXTEx](#)().

Referenced by [EMRSETWINDOWEXTEx](#)().

**EMRSETWINDOWEXTEx()** [2/2]

```
EMF::EMRSETWINDOWEXTEx::EMRSETWINDOWEXTEx (  
    DATASTREAM & ds) [inline]
```

Construct a SetWindowExtEx record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.54.3 Member Function Documentation****execute()**

```
void EMF::EMRSETWINDOWEXTEx::execute (  
    METAFILEDEVICECONTEXT * source,  
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSETWINDOWEXTEx::serialize (  
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSETWINDOWEXTEx::size (  
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

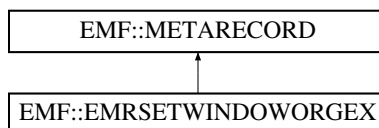
- libemf.h

## 4.55 EMF::EMRSETWINDOWORGEX Class Reference

EMF Set Window Origin (ex).

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETWINDOWORGEX:



### Public Member Functions

- [EMRSETWINDOWORGEX](#) (INT x, INT y)
- [EMRSETWINDOWORGEX](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.55.1 Detailed Description

EMF Set Window Origin (ex).

The window origin specifies the logical (i.e., real) coordinates of the upper, left corner of the viewport. (For example, if you want your XY plot's axis origin to be in the middle of the viewport, you'd set the window origin to something like [-1,-1].)

#### 4.55.2 Constructor & Destructor Documentation

##### EMRSETWINDOWORGEX() [1/2]

```

EMF::EMRSETWINDOWORGEX::EMRSETWINDOWORGEX (
    INT x,
    INT y) [inline]
  
```

### Parameters

<i>x</i>	x coordinate of window origin in logical coordinates
<i>y</i>	y coordinate of window origin in logical coordinates

References [EMRSETWINDOWORGEX\(\)](#).

Referenced by [EMRSETWINDOWORGEX\(\)](#).

**EMRSETWINDOWORGEX()** [2/2]

```
EMF::EMRSETWINDOWORGEX::EMRSETWINDOWORGEX (
    DATASTREAM & ds) [inline]
```

Construct a SetWindowOrgEx record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.55.3 Member Function Documentation****execute()**

```
void EMF::EMRSETWINDOWORGEX::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSETWINDOWORGEX::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSETWINDOWORGEX::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

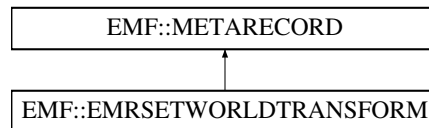
- libemf.h

## 4.56 EMF::EMRSETWORLDTRANSFORM Class Reference

EMF Set World Transform.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSETWORLDTRANSFORM:



### Public Member Functions

- [EMRSETWORLDTRANSFORM](#) (const XFORM \*transform)
- [EMRSETWORLDTRANSFORM](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.56.1 Detailed Description

EMF Set World Transform.

Enhanced metafiles have a Coordinate Transformation which allows the contents to be rotated and transformed. Does not appear to work properly in StarOffice (but it's also possible I don't understand how it's supposed to work either).

#### 4.56.2 Constructor & Destructor Documentation

##### EMRSETWORLDTRANSFORM() [1/2]

```
EMF::EMRSETWORLDTRANSFORM::EMRSETWORLDTRANSFORM (
    const XFORM * transform) [inline]
```

##### Parameters

<i>transform</i>	the new transformation
------------------	------------------------

References [EMRSETWORLDTRANSFORM\(\)](#).

Referenced by [EMRSETWORLDTRANSFORM\(\)](#).

**EMRSETWORLDTRANSFORM()** [2/2]

```
EMF::EMRSETWORLDTRANSFORM::EMRSETWORLDTRANSFORM (
    DATASTREAM & ds) [inline]
```

Construct a SetWorldTransform record from the input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.56.3 Member Function Documentation****execute()**

```
void EMF::EMRSETWORLDTRANSFORM::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSETWORLDTRANSFORM::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSETWORLDTRANSFORM::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

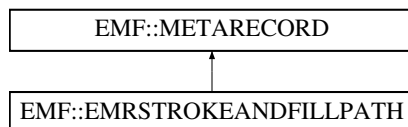
- libemf.h

## 4.57 EMF::EMRSTROKEANDFILLPATH Class Reference

EMF Stroke and Fill path.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSTROKEANDFILLPATH:



### Public Member Functions

- [EMRSTROKEANDFILLPATH](#) (const RECTL \*bounds)
- [EMRSTROKEANDFILLPATH](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.57.1 Detailed Description

EMF Stroke and Fill path.

Stroke and Fill the path.

#### 4.57.2 Constructor & Destructor Documentation

##### EMRSTROKEANDFILLPATH() [1/2]

```
EMF::EMRSTROKEANDFILLPATH::EMRSTROKEANDFILLPATH (
    const RECTL * bounds) [inline]
```

### Parameters

<i>bounds</i>	overall bounding box of polygon.
---------------	----------------------------------

References [EMRSTROKEANDFILLPATH\(\)](#).

Referenced by [EMRSTROKEANDFILLPATH\(\)](#).

**EMRSTROKEANDFILLPATH()** [2/2]

```
EMF::EMRSTROKEANDFILLPATH::EMRSTROKEANDFILLPATH (
    DATASTREAM & ds) [inline]
```

Create a StrokeandfillPath record from input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.57.3 Member Function Documentation****execute()**

```
void EMF::EMRSTROKEANDFILLPATH::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSTROKEANDFILLPATH::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSTROKEANDFILLPATH::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

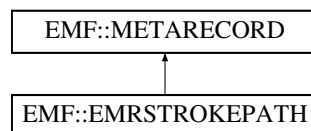
- libemf.h

## 4.58 EMF::EMRSTROKEPATH Class Reference

EMF Stroke path.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EMRSTROKEPATH:



### Public Member Functions

- [EMRSTROKEPATH](#) (const RECTL \*bounds)
- [EMRSTROKEPATH](#) (DATASTREAM &ds)
- bool [serialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.58.1 Detailed Description

EMF Stroke path.

Stroke the path.

#### 4.58.2 Constructor & Destructor Documentation

##### EMRSTROKEPATH() [1/2]

```
EMF::EMRSTROKEPATH::EMRSTROKEPATH (
    const RECTL * bounds) [inline]
```

##### Parameters

<i>bounds</i>	overall bounding box of polygon.
---------------	----------------------------------

References [EMRSTROKEPATH\(\)](#).

Referenced by [EMRSTROKEPATH\(\)](#).

**EMRSTROKEPATH()** [2/2]

```
EMF::EMRSTROKEPATH::EMRSTROKEPATH (
    DATASTREAM & ds) [inline]
```

Create a StrokePath record from input stream.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

**4.58.3 Member Function Documentation****execute()**

```
void EMF::EMRSTROKEPATH::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::EMRSTROKEPATH::serialize (
    DATASTREAM ds) [inline], [virtual]
```

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::EMRSTROKEPATH::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

The documentation for this class was generated from the following file:

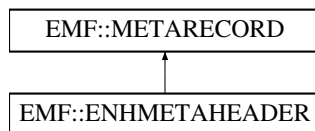
- libemf.h

## 4.59 EMF::ENHMETAHEADER Class Reference

Enhanced Metafile Header Record.

```
#include <libemf.h>
```

Inheritance diagram for EMF::ENHMETAHEADER:



### Public Member Functions

- [ENHMETAHEADER](#) (LPCWSTR description=0)
- [~ENHMETAHEADER](#) ()
- bool [serialize](#) (DATASTREAM ds)
- bool [unserialize](#) (DATASTREAM ds)
- int [size](#) (void) const
- void [execute](#) (METAFILEDEVICECONTEXT \*source, HDC dc) const

### Public Member Functions inherited from [EMF::METARECORD](#)

- virtual [~METARECORD](#) ()

#### 4.59.1 Detailed Description

Enhanced Metafile Header Record.

The [ENHMETAHEADER](#) serves two purposes in this library: it keeps track of the size of the metafile (in physical dimensions) and number of records and handles that are ultimately to be written to the disk file. It is also a real record that must be written out.

#### 4.59.2 Constructor & Destructor Documentation

##### ENHMETAHEADER()

```
EMF::ENHMETAHEADER::ENHMETAHEADER (
    LPCWSTR description = 0) [inline]
```

##### Parameters

<i>description</i>	an optional description argument is a UNICODE-like string with the following format: "some text\0some more text\0\0". The W32 interface defines UNICODE characters to be two-byte (unsigned short strings). The constructor makes a copy of the argument.
--------------------	---

References [ENHMETAHEADER\(\)](#).

Referenced by [ENHMETAHEADER\(\)](#), and [unserialize\(\)](#).

**~ENHMETAHEADER()**

```
EMF::ENHMETAHEADER::~~ENHMETAHEADER () [inline]
```

Destructor deletes memory allocated for description.

**4.59.3 Member Function Documentation****execute()**

```
void EMF::ENHMETAHEADER::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [inline], [virtual]
```

Execute this record in the context of the given device context.

**Parameters**

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	device context for execute.

Implements [EMF::METARECORD](#).

**serialize()**

```
bool EMF::ENHMETAHEADER::serialize (
    DATASTREAM ds) [inline], [virtual]
```

Serializing the header is an example of an extended record.

**Parameters**

<i>ds</i>	Metafile datastream.
-----------	----------------------

Implements [EMF::METARECORD](#).

**size()**

```
int EMF::ENHMETAHEADER::size (
    void ) const [inline], [virtual]
```

Internally computed size of this record.

Implements [EMF::METARECORD](#).

**unserialize()**

```
bool EMF::ENHMETAHEADER::unserialize (
    DATASTREAM ds) [inline]
```

Read a header record from the datastream.

References [ENHMETAHEADER\(\)](#).

The documentation for this class was generated from the following file:

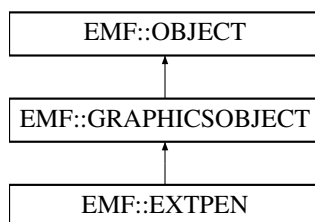
- libemf.h

**4.60 EMF::EXTPEN Class Reference**

Extended Graphics Pen.

```
#include <libemf.h>
```

Inheritance diagram for EMF::EXTPEN:

**Public Member Functions**

- [EXTPEN](#) (const EXTLOGPEN \*lpen)
- OBJECTTYPE [getType](#) (void) const
- METARECORD \* [newEMR](#) (HDC dc, HGDIOBJ emf\_handle)

**Public Member Functions inherited from [EMF::GRAPHICSOBJECT](#)**

- virtual [~GRAPHICSOBJECT](#) ()  
*GRAPHICSOBJECTs has a virtual destructor.*

**Public Member Functions inherited from [EMF::OBJECT](#)**

- virtual [~OBJECT](#) ()  
*OBJECTs have a virtual destructor.*
- [OBJECT](#) (void)

## Additional Inherited Members

### Data Fields inherited from [EMF::GRAPHICSOBJECT](#)

- `std::map< HDC, HGDIOBJ >` [contexts](#)

### Data Fields inherited from [EMF::OBJECT](#)

- HGDIOBJ [handle](#)

## 4.60.1 Detailed Description

Extended Graphics Pen.

Pens are used for drawing lines, arc, rectangles, etc.

## 4.60.2 Constructor & Destructor Documentation

### EXTPEN()

```
EMF::EXTPEN::EXTPEN (
    const EXTLOGPEN * lpen) [inline]
```

#### Parameters

<i>lpen</i>	the (logical?) pen definition.
-------------	--------------------------------

## 4.60.3 Member Function Documentation

### getType()

```
OBJECTTYPE EMF::EXTPEN::getType (
    void ) const [inline], [virtual]
```

Return the type of this object (could probably do better with RTTI()).

Implements [EMF::OBJECT](#).

**newEMR()**

```
METARECORD * EMF::EXTPEN::newEMR (
    HDC dc,
    HGDIOBJ emf_handle) [inline], [virtual]
```

Return a new metarecord for this object. And record its selection into the given device context.

**Parameters**

<i>dc</i>	handle of device context into which this object is being selected.
<i>emf_handle</i>	the EMF handle associated with the <a href="#">PEN</a> .

Implements [EMF::GRAPHICSOBJECT](#).

References [EMF::GRAPHICSOBJECT::contexts](#).

The documentation for this class was generated from the following file:

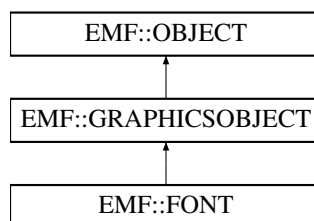
- libemf.h

**4.61 EMF::FONT Class Reference**

Graphics Font.

```
#include <libemf.h>
```

Inheritance diagram for EMF::FONT:

**Public Member Functions**

- [FONT](#) (const LOGFONTW \*lfont)
- OBJECTTYPE [getType](#) (void) const
- METARECORD \* [newEMR](#) (HDC dc, HGDIOBJ emf\_handle)

**Public Member Functions inherited from [EMF::GRAPHICSOBJECT](#)**

- virtual [~GRAPHICSOBJECT](#) ()  
*GRAPHICSOBJECTs has a virtual destructor.*

**Public Member Functions inherited from [EMF::OBJECT](#)**

- virtual `~OBJECT()`  
*OBJECTs have a virtual destructor.*
- [OBJECT](#) (void)

**Additional Inherited Members****Data Fields inherited from [EMF::GRAPHICSOBJECT](#)**

- `std::map< HDC, HGDIOBJ >` [contexts](#)

**Data Fields inherited from [EMF::OBJECT](#)**

- HGDIOBJ [handle](#)

**4.61.1 Detailed Description**

Graphics Font.

Fonts are used for drawing text (obviously).

**4.61.2 Constructor & Destructor Documentation****FONT()**

```
EMF::FONT::FONT (
    const LOGFONTW * lfont) [inline]
```

**Parameters**

<i>lfont</i>	the (logical?) font definition.
--------------	---------------------------------

**4.61.3 Member Function Documentation****getType()**

```
OBJECTTYPE EMF::FONT::getType (
    void ) const [inline], [virtual]
```

Return the type of this object (could probably do better with RTTI()).

Implements [EMF::OBJECT](#).

**newEMR()**

```
METARECORD * EMF::FONT::newEMR (
    HDC dc,
    HGDIOBJ emf_handle) [inline], [virtual]
```

Return a new metarecord for this object. And record its selection into the given device context.

**Parameters**

<i>dc</i>	handle of device context into which this object is being selected.
<i>emf_handle</i>	the EMF handle associated with the <a href="#">FONT</a> .

Implements [EMF::GRAPHICSOBJECT](#).

References [EMF::GRAPHICSOBJECT::contexts](#).

The documentation for this class was generated from the following file:

- libemf.h

**4.62 EMF::GLOBALOBJECTS Class Reference**

```
#include <libemf.h>
```

**Public Member Functions**

- HGDIOBJ [add](#) ([OBJECT](#) \*object)
- [OBJECT](#) \* [find](#) (const HGDIOBJ handle)
- void [remove](#) (const [OBJECT](#) \*object)
- auto [begin](#) (void) const
- auto [end](#) (void) const
- METARECORDCTOR [newRecord](#) (DWORD iType) const

**Static Public Member Functions**

- static [EMF::METARECORD](#) \* [new\\_eof](#) ([DATASTREAM](#) &ds)  
*Create a new [EMREOF](#) record.*
- static [EMF::METARECORD](#) \* [new\\_setviewportorgex](#) ([DATASTREAM](#) &ds)  
*Create a new [EMRSETVIEWPORTORGEX](#) record.*
- static [EMF::METARECORD](#) \* [new\\_setwindoworgex](#) ([DATASTREAM](#) &ds)  
*Create a new [EMRSETWINDOWORGEX](#) record.*
- static [EMF::METARECORD](#) \* [new\\_setviewporttextex](#) ([DATASTREAM](#) &ds)  
*Create a new [EMRSETVIEWPORTEXTEX](#) record.*
- static [EMF::METARECORD](#) \* [new\\_setwindowextex](#) ([DATASTREAM](#) &ds)  
*Create a new [EMRSETWINDOWEXTEX](#) record.*
- static [EMF::METARECORD](#) \* [new\\_scaleviewporttextex](#) ([DATASTREAM](#) &ds)  
*Create a new [SCALEVIEWPORTEXTEX](#) record.*
- static [EMF::METARECORD](#) \* [new\\_scalewindowextex](#) ([DATASTREAM](#) &ds)

- Create a new SCALEWINDOWEXTEN record.*

  - static [EMF::METARECORD](#) \* [new\\_modifyworldtransform](#) ([DATASTREAM](#) &ds)

*Create a new MODIFYWORLDTRANSFORM record.*
- static [EMF::METARECORD](#) \* [new\\_setworldtransform](#) ([DATASTREAM](#) &ds)

*Create a new SETWORLDTRANSFORM record.*
- static [EMF::METARECORD](#) \* [new\\_settextalign](#) ([DATASTREAM](#) &ds)

*Create a new SETTEXTALIGN record.*
- static [EMF::METARECORD](#) \* [new\\_settextcolor](#) ([DATASTREAM](#) &ds)

*Create a new SETTEXTCOLOR record.*
- static [EMF::METARECORD](#) \* [new\\_setbkcolor](#) ([DATASTREAM](#) &ds)

*Create a new SETBKCOLOR record.*
- static [EMF::METARECORD](#) \* [new\\_setbkmode](#) ([DATASTREAM](#) &ds)

*Create a new SETBKMODE record.*
- static [EMF::METARECORD](#) \* [new\\_setpolyfillmode](#) ([DATASTREAM](#) &ds)

*Create a new SETPOLYFILLMODE record.*
- static [EMF::METARECORD](#) \* [new\\_setmapmode](#) ([DATASTREAM](#) &ds)

*Create a new SETMAPMODE record.*
- static [EMF::METARECORD](#) \* [new\\_selectobject](#) ([DATASTREAM](#) &ds)

*Create a new SELECTOBJECT record.*
- static [EMF::METARECORD](#) \* [new\\_deleteobject](#) ([DATASTREAM](#) &ds)

*Create a new DELETEOBJECT record.*
- static [EMF::METARECORD](#) \* [new\\_movetoex](#) ([DATASTREAM](#) &ds)

*Create a new MOVETOEX record.*
- static [EMF::METARECORD](#) \* [new\\_lineto](#) ([DATASTREAM](#) &ds)

*Create a new LINETO record.*
- static [EMF::METARECORD](#) \* [new\\_arc](#) ([DATASTREAM](#) &ds)

*Create a new ARC record.*
- static [EMF::METARECORD](#) \* [new\\_arcto](#) ([DATASTREAM](#) &ds)

*Create a new ARCTO record.*
- static [EMF::METARECORD](#) \* [new\\_rectangle](#) ([DATASTREAM](#) &ds)

*Create a new RECTANGLE record.*
- static [EMF::METARECORD](#) \* [new\\_ellipse](#) ([DATASTREAM](#) &ds)

*Create a new ELLIPSE record.*
- static [EMF::METARECORD](#) \* [new\\_polyline](#) ([DATASTREAM](#) &ds)

*Create a new POLYLINE record.*
- static [EMF::METARECORD](#) \* [new\\_polyline16](#) ([DATASTREAM](#) &ds)

*Create a new POLYLINE16 record.*
- static [EMF::METARECORD](#) \* [new\\_polygon](#) ([DATASTREAM](#) &ds)

*Create a new POLYGON record.*
- static [EMF::METARECORD](#) \* [new\\_polygon16](#) ([DATASTREAM](#) &ds)

*Create a new POLYGON16 record.*
- static [EMF::METARECORD](#) \* [new\\_polypolygon](#) ([DATASTREAM](#) &ds)

*Create a new POLYPOLYGON record.*
- static [EMF::METARECORD](#) \* [new\\_polypolygon16](#) ([DATASTREAM](#) &ds)

*Create a new POLYPOLYGON16 record.*
- static [EMF::METARECORD](#) \* [new\\_polybezier](#) ([DATASTREAM](#) &ds)

*Create a new POLYBEZIER record.*
- static [EMF::METARECORD](#) \* [new\\_polybezier16](#) ([DATASTREAM](#) &ds)

*Create a new POLYBEZIER16 record.*
- static [EMF::METARECORD](#) \* [new\\_polybezierto](#) ([DATASTREAM](#) &ds)

*Create a new POLYBEZIERTO record.*

- static **EMF::METARECORD** \* **new\_polybezierto16** (**DATASTREAM** &ds)  
*Create a new POLYBEZIERTO16 record.*
- static **EMF::METARECORD** \* **new\_polylineto** (**DATASTREAM** &ds)  
*Create a new POLYLINETO record.*
- static **EMF::METARECORD** \* **new\_polylineto16** (**DATASTREAM** &ds)  
*Create a new POLYLINETO16 record.*
- static **EMF::METARECORD** \* **new\_exttextouta** (**DATASTREAM** &ds)  
*Create a new EXTTEXTOUTA record.*
- static **EMF::METARECORD** \* **new\_exttextoutw** (**DATASTREAM** &ds)  
*Create a new EXTTEXTOUTW record.*
- static **EMF::METARECORD** \* **new\_setpixelv** (**DATASTREAM** &ds)  
*Create a new SETPIXELV record.*
- static **EMF::METARECORD** \* **new\_createpen** (**DATASTREAM** &ds)  
*Create a new CREATEPEN record.*
- static **EMF::METARECORD** \* **new\_extcreatepen** (**DATASTREAM** &ds)  
*Create a new EXTCREATEPEN record.*
- static **EMF::METARECORD** \* **new\_createbrushindirect** (**DATASTREAM** &ds)  
*Create a new CREATEBRUSHINDIRECT record.*
- static **EMF::METARECORD** \* **new\_extcreatefontindirectw** (**DATASTREAM** &ds)  
*Create a new EXTCREATEFONTINDIRECTW record.*
- static **EMF::METARECORD** \* **new\_fillpath** (**DATASTREAM** &ds)  
*Create a new FILLPATH record.*
- static **EMF::METARECORD** \* **new\_strokepath** (**DATASTREAM** &ds)  
*Create a new STROKEPATH record.*
- static **EMF::METARECORD** \* **new\_strokeandfillpath** (**DATASTREAM** &ds)  
*Create a new STROKEANDFILLPATH record.*
- static **EMF::METARECORD** \* **new\_beginpath** (**DATASTREAM** &ds)  
*Create a new BEGINPATH record.*
- static **EMF::METARECORD** \* **new\_endpath** (**DATASTREAM** &ds)  
*Create a new ENDPATH record.*
- static **EMF::METARECORD** \* **new\_closefigure** (**DATASTREAM** &ds)  
*Create a new CLOSEFIGURE record.*
- static **EMF::METARECORD** \* **new\_savedc** (**DATASTREAM** &ds)  
*Create a new SAVEDC record.*
- static **EMF::METARECORD** \* **new\_restoredc** (**DATASTREAM** &ds)  
*Create a new RESTOREDC record.*
- static **EMF::METARECORD** \* **new\_setmetargn** (**DATASTREAM** &ds)  
*Create a new SETMETARGN record.*
- static **EMF::METARECORD** \* **new\_setmiterlimit** (**DATASTREAM** &ds)  
*Create a new SETMITERLIMIT record.*

#### 4.62.1 Detailed Description

Stores all the objects in a single database within a process.

### 4.62.2 Member Function Documentation

#### add()

```
HGDIOBJ EMF::GLOBALOBJECTS::add (  
    OBJECT * object)
```

Add an object to the global vector. The object's handle is simply its index in the global object vector, which is computed by the very interesting "difference between two iterators" method.

#### Parameters

<i>object</i>	pointer to a real instance of an object, not its handle.
---------------	--

#### begin()

```
auto EMF::GLOBALOBJECTS::begin (  
    void ) const [inline]
```

#### Returns

an iterator pointing to the first global object.

#### end()

```
auto EMF::GLOBALOBJECTS::end (  
    void ) const [inline]
```

#### Returns

an iterator pointing to (one past) the final global object.

#### find()

```
OBJECT * EMF::GLOBALOBJECTS::find (  
    const HGDIOBJ handle)
```

Look up a object by handle in the global object vector. Note: Stock objects (like a gray brush or the black pen) have their high order bit set, so this has to be masked out when using their handles.

#### Parameters

<i>handle</i>	the object's handle.
---------------	----------------------

#### Returns

pointer to object.

**newRecord()**

```
METARECORDCTOR EMF::GLOBALOBJECTS::newRecord (
    DWORD iType) const
```

See if we have a constructor for a record of the given type.

**Parameters**

<i>iType</i>	metarecord type.
--------------	------------------

**Returns**

pointer to "virtual" constructor.

**remove()**

```
void EMF::GLOBALOBJECTS::remove (
    const OBJECT * object)
```

A call to the metafile function DeleteObject() allows a particular object's handle to be reused, so some care has to be taken to erase it.

**Parameters**

<i>object</i>	pointer to object to delete.
---------------	------------------------------

The documentation for this class was generated from the following files:

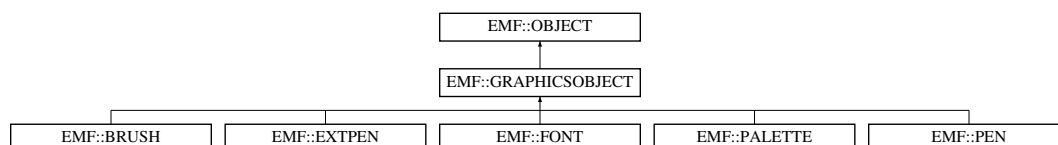
- libemf.h
- libemf.cpp

**4.63 EMF::GRAPHICSOBJECT Class Reference**

A global graphics object.

```
#include <libemf.h>
```

Inheritance diagram for EMF::GRAPHICSOBJECT:



## Public Member Functions

- virtual `~GRAPHICSOBJECT ()`  
*GRAPHICSOBJECTs has a virtual destructor.*
- virtual `METARECORD * newEMR (HDC dc, HGDIOBJ handle)=0`

## Public Member Functions inherited from EMF::OBJECT

- virtual `~OBJECT ()`  
*OBJECTs have a virtual destructor.*
- `OBJECT (void)`
- virtual `OBJECTTYPE getType (void) const =0`

## Data Fields

- `std::map< HDC, HGDIOBJ > contexts`

## Data Fields inherited from EMF::OBJECT

- `HGDIOBJ handle`

### 4.63.1 Detailed Description

A global graphics object.

Graphics objects have some additional properties: When an object is Select'ed into a device context, the handle for that context is added to the list of context's in which this object is used.

### 4.63.2 Member Function Documentation

#### newEMR()

```
virtual METARECORD * EMF::GRAPHICSOBJECT::newEMR (
    HDC dc,
    HGDIOBJ handle) [pure virtual]
```

Create a new metarecord which describes this object.

#### Parameters

<i>dc</i>	the handle to the device context.
<i>handle</i>	(appears not to used. Note the handle is really assigned at serialization time.)

Implemented in [EMF::BRUSH](#), [EMF::EXTPEN](#), [EMF::FONT](#), [EMF::PALETTE](#), and [EMF::PEN](#).

References [EMF::OBJECT::handle](#).

### 4.63.3 Field Documentation

#### contexts

```
std::map< HDC, HGDIOBJ > EMF::GRAPHICSOBJECT::contexts
```

A set of all the contexts into which this object has been selected and the associated metafile handle for the object.

Referenced by [EMF::BRUSH::newEMR\(\)](#), [EMF::EXTPEN::newEMR\(\)](#), [EMF::FONT::newEMR\(\)](#), [EMF::PALETTE::newEMR\(\)](#), and [EMF::PEN::newEMR\(\)](#).

The documentation for this class was generated from the following file:

- libemf.h

## 4.64 EMF::INTARRAY Struct Reference

Represent an array of integers in a simple way.

```
#include <libemf.h>
```

### Public Member Functions

- [INTARRAY](#) (INT \*const ints, const DWORD n)

### Data Fields

- INT \*const **ints\_**  
*Array of ints.*
- const DWORD **n\_**  
*Number of ints in array.*

#### 4.64.1 Detailed Description

Represent an array of integers in a simple way.

Allow an array of INT's to be written out at once.

### 4.64.2 Constructor & Destructor Documentation

#### INTARRAY()

```
EMF::INTARRAY::INTARRAY (
    INT *const ints,
    const DWORD n) [inline]
```

simple constructor.

#### Parameters

<i>ints</i>	pointer to ints.
<i>n</i>	number ints in array.

References [ints\\_](#), and [n\\_](#).

The documentation for this struct was generated from the following file:

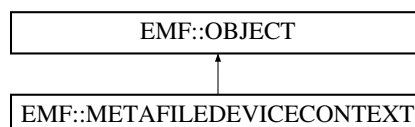
- libemf.h

## 4.65 EMF::METAFILEDEVICECONTEXT Class Reference

Graphics Device Context.

```
#include <libemf.h>
```

Inheritance diagram for EMF::METAFILEDEVICECONTEXT:



#### Public Member Functions

- [METAFILEDEVICECONTEXT](#) (FILE \*fp\_, const RECT \*size, LPCWSTR description\_w)
- virtual [~METAFILEDEVICECONTEXT](#) ()
- OBJECTTYPE [getType](#) (void) const
- DWORD [nextHandle](#) (void)
- void [clearHandle](#) (DWORD [handle](#))
- void [appendRecord](#) (METARECORD \*record)
- void [appendHandle](#) (METARECORD \*record)
- void [deleteMetafile](#) (void)
- void [mergePoint](#) (const LONG &x, const LONG &y)
- void [mergePoint](#) (const POINT &p)

## Public Member Functions inherited from [EMF::OBJECT](#)

- virtual `~OBJECT()`  
*OBJECTs have a virtual destructor.*
- [OBJECT](#)(void)

## Data Fields

- `::FILE * fp`
- [DATASTREAM ds](#)
- `ENHMETAHEADER * header`
- `std::vector< EMF::METARECORD * > records`
- `SIZEL resolution`  
*The resolution in DPI of the reference DC.*
- `SIZEL viewport_ext`  
*The extent of the viewport.*
- `POINT viewport_org`  
*The origin of the viewport.*
- `SIZEL window_ext`  
*The extent of the window.*
- `POINT window_org`  
*The origin of the window.*
- `bool update_frame`  
*Update the frame automatically?*
- `POINT min_device_point`  
*The lft/top-most painted point in device units.*
- `POINT max_device_point`  
*The rgt/btm-most painted point in device units.*
- `POINT point`  
*The current point.*
- [PEN](#) \* `pen`  
*The current pen.*
- [BRUSH](#) \* `brush`  
*The current brush.*
- [FONT](#) \* `font`  
*The current font.*
- [PALETTE](#) \* `palette`  
*The current palette.*
- `UINT text_alignment`  
*The current text alignment.*
- `COLORREF text_color`  
*The current text foreground color.*
- `COLORREF bk_color`  
*The current background color.*
- `INT bk_mode`  
*The current background mode.*
- `INT polyfill_mode`  
*The current polygon fill mode.*
- `INT map_mode`  
*The current mapping mode.*
- `FLOAT miter_limit`  
*The current miter length limit.*
- `std::vector< bool > handles`
- `std::map< HGDIOBJ, HGDIOBJ > emf_handles`

## Data Fields inherited from EMF::OBJECT

- HGDIOBJ [handle](#)

### 4.65.1 Detailed Description

Graphics Device Context.

Almost all GDI graphics calls require a device context (except those which create graphics objects such as pens and fonts). This is a specific context which renders to a metafile. There is a one-to-one correspondence between the device context and the metafile.

### 4.65.2 Constructor & Destructor Documentation

#### METAFILEDEVICECONTEXT()

```
EMF::METAFILEDEVICECONTEXT::METAFILEDEVICECONTEXT (
    FILE * fp_,
    const RECT * size,
    LPCWSTR description_w) [inline]
```

Most graphics programs seem to want to handle the opening and closing of files themselves, so this is an extension to the w32 interface.

#### Parameters

<i>fp_</i>	stdio pointer to an open file. May be null.
<i>size</i>	the rectangle describing the position and size of the metafile on the "page". May be null.
<i>description_w</i>	a UNICODE string describing the metafile. The format must be "some text\0some more text\0\0". May be null.

References [ds](#), and [fp](#).

#### ~METAFILEDEVICECONTEXT()

```
virtual EMF::METAFILEDEVICECONTEXT::~~METAFILEDEVICECONTEXT () [inline], [virtual]
```

Destructor frees all the graphics objects which may have been allocated. Now, it also frees any metarecords which it might hold, too.

References [deleteMetafile\(\)](#), and [records](#).

### 4.65.3 Member Function Documentation

#### appendHandle()

```
void EMF::METAFILEDEVICECONTEXT::appendHandle (
    METARECORD * record) [inline]
```

Add this record to the metafile.

##### Parameters

<i>record</i>	this record is an object so it increments the handle count as well.
---------------	---

References [header](#), [records](#), and [EMF::METARECORD::size\(\)](#).

#### appendRecord()

```
void EMF::METAFILEDEVICECONTEXT::appendRecord (
    METARECORD * record) [inline]
```

Add this record to the metafile.

##### Parameters

<i>record</i>	standard graphics record
---------------	--------------------------

References [header](#), [records](#), and [EMF::METARECORD::size\(\)](#).

#### clearHandle()

```
void EMF::METAFILEDEVICECONTEXT::clearHandle (
    DWORD handle) [inline]
```

Clear the usage of this handle

References [EMF::OBJECT::handle](#), and [handles](#).

#### deleteMetafile()

```
void EMF::METAFILEDEVICECONTEXT::deleteMetafile (
    void ) [inline]
```

Delete all the records from the metafile. This would seem to include deleting the header record as well.

References [records](#).

Referenced by [~METAFILEDEVICECONTEXT\(\)](#).

**getType()**

```
OBJECTTYPE EMF::METAFILEDEVICECONTEXT::getType (
    void ) const [inline], [virtual]
```

Return the type of this object (could probably do better with RTTI()).

Implements [EMF::OBJECT](#).

**mergePoint()** [1/2]

```
void EMF::METAFILEDEVICECONTEXT::mergePoint (
    const LONG & x,
    const LONG & y) [inline]
```

Somewhat superfluous, except checker doesn't understand the initialization of automatic structures in the declaration.

References [mergePoint\(\)](#).

Referenced by [mergePoint\(\)](#).

**mergePoint()** [2/2]

```
void EMF::METAFILEDEVICECONTEXT::mergePoint (
    const POINT & p) [inline]
```

Take the given point and determine if it enlarges the "painted" area of the device.

References [header](#), [max\\_device\\_point](#), [min\\_device\\_point](#), [update\\_frame](#), [viewport\\_ext](#), [viewport\\_org](#), [window\\_ext](#), and [window\\_org](#).

**nextHandle()**

```
DWORD EMF::METAFILEDEVICECONTEXT::nextHandle (
    void ) [inline]
```

Scan the bit vector of used handles and return the index of the first free bit as this objects metafile handle.

References [handles](#), and [header](#).

**4.65.4 Field Documentation****ds**

```
DATASTREAM EMF::METAFILEDEVICECONTEXT::ds
```

All i/o to the metafile is wrapped by this class so that byte swapping on big-endian machines is transparent.

Referenced by [METAFILEDEVICECONTEXT\(\)](#).

## emf\_handles

```
std::map< HGDIOBJ, HGDIOBJ > EMF::METAFILEDEVICECONTEXT::emf_handles
```

This map holds the *current* mapping between EMF handles and global object handles as a metafile is played back (with `PlayEnhMetaFile`).

Referenced by [EMF::EMRCREATEBRUSHINDIRECT::execute\(\)](#), [EMF::EMRCREATEPEN::execute\(\)](#), [EMF::EMRDELETEOBJECT::execute\(\)](#), [EMF::EMREXTCREATEFONTINDIRECTW::execute\(\)](#), [EMF::EMREXTCREATEPEN::execute\(\)](#), and [EMF::EMRSELECTOBJECT::execute\(\)](#).

## fp

```
::FILE* EMF::METAFILEDEVICECONTEXT::fp
```

If it is a file-based metafile, then this pointer is not null.

Referenced by [METAFILEDEVICECONTEXT\(\)](#).

## handles

```
std::vector< bool > EMF::METAFILEDEVICECONTEXT::handles
```

For compatibility, it appears that metafile handles are reused as objects are deleted. Attempt to emulate that behavior with a bit vector of used metafile handles.

Referenced by [clearHandle\(\)](#), and [nextHandle\(\)](#).

## header

```
ENHMETAHEADER* EMF::METAFILEDEVICECONTEXT::header
```

Serves double duty as the physical device description.

Referenced by [appendHandle\(\)](#), [appendRecord\(\)](#), [mergePoint\(\)](#), and [nextHandle\(\)](#).

## records

```
std::vector< EMF::METARECORD* > EMF::METAFILEDEVICECONTEXT::records
```

All of the metafile records are stored in memory.

Referenced by [appendHandle\(\)](#), [appendRecord\(\)](#), [deleteMetafile\(\)](#), and [~METAFILEDEVICECONTEXT\(\)](#).

The documentation for this class was generated from the following file:

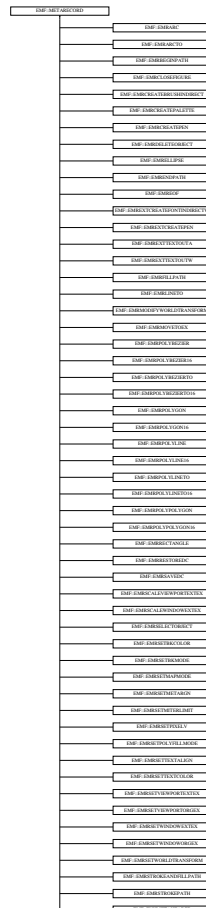
- libemf.h

#### 4.66 EMF::METARECORD Class Reference

The base class of all metafile records.

```
#include <libemf.h>
```

Inheritance diagram for EMF::METARECORD:



## Public Member Functions

- virtual void **execute** (**METAFILEDVICECONTEXT** \*source, HDC dc) const =0
- virtual bool **serialize** (**DATASTREAM** ds)=0
- virtual int **size** (void) const =0
- virtual **~METARECORD** ()

#### 4.66.1 Detailed Description

The base class of all metafile records.

A metafile consists off a sequence of graphics records "executed" in order. This is a common base class that allows each, different, record to be stored in a common list. An interface is specified for each record to write itself to a file.

## 4.66.2 Constructor & Destructor Documentation

### ~METARECORD()

```
virtual EMF::METARECORD::~~METARECORD () [inline], [virtual]
```

The virtual destructor allows records which allocated additional memory to release it when they are deleted. Simple records just use the default destructor defined here.

## 4.66.3 Member Function Documentation

### execute()

```
virtual void EMF::METARECORD::execute (
    METAFILEDEVICECONTEXT * source,
    HDC dc) const [pure virtual]
```

Execute the graphics command in the given context. Used by PlayEnhMetaFile to "copy" one metafile into another.

#### Parameters

<i>source</i>	the device context from which this record is taken.
<i>dc</i>	the destination context.

Implemented in [EMF::EMRARC](#), [EMF::EMRARCTO](#), [EMF::EMRBEGINPATH](#), [EMF::EMRCLOSEFIGURE](#), [EMF::EMRCREATEBRUSHINDIRECT](#), [EMF::EMRCREATEPALETTE](#), [EMF::EMRCREATEPEN](#), [EMF::EMRDELETEOBJECT](#), [EMF::EMRELLIPSE](#), [EMF::EMRENDPATH](#), [EMF::EMREOF](#), [EMF::EMREXTCREATEFONTINDIRECTW](#), [EMF::EMREXTCREATEPEN](#), [EMF::EMREXTTEXTOUTA](#), [EMF::EMREXTTEXTOUTW](#), [EMF::EMRFILLPATH](#), [EMF::EMRLINETO](#), [EMF::EMRMODIFYWORLDTRANSFORM](#), [EMF::EMRMOVETOEX](#), [EMF::EMRPOLYBEZIER16](#), [EMF::EMRPOLYBEZIER](#), [EMF::EMRPOLYBEZIERTO16](#), [EMF::EMRPOLYBEZIERTO](#), [EMF::EMRPOLYGON16](#), [EMF::EMRPOLYGON](#), [EMF::EMRPOLYLINE16](#), [EMF::EMRPOLYLINE](#), [EMF::EMRPOLYLINETO16](#), [EMF::EMRPOLYLINETO](#), [EMF::EMRPOLYPOLYGON16](#), [EMF::EMRPOLYPOLYGON](#), [EMF::EMRRECTANGLE](#), [EMF::EMRRESTORED](#), [EMF::EMRSAVE](#), [EMF::EMRSCALEVIEWPORTEX](#), [EMF::EMRSCALEWINDOWEX](#), [EMF::EMRSELECTOBJECT](#), [EMF::EMRSETBKCOLOR](#), [EMF::EMRSETBKMODE](#), [EMF::EMRSETMAPMODE](#), [EMF::EMRSETMETARGN](#), [EMF::EMRSETMITERLIMIT](#), [EMF::EMRSETPIXELV](#), [EMF::EMRSETPOLYFILLMODE](#), [EMF::EMRSETTEXTALIGN](#), [EMF::EMRSETTEXTCOLOR](#), [EMF::EMRSETVIEWPORTEX](#), [EMF::EMRSETVIEWPORTORGEX](#), [EMF::EMRSETWINDOWEX](#), [EMF::EMRSETWINDOWORGEX](#), [EMF::EMRSETWORLDTRANSFORM](#), [EMF::EMRSTROKEANDFILLPATH](#), [EMF::EMRSTROKEPATH](#), and [EMF::ENHMETAHEADER](#).

### serialize()

```
virtual bool EMF::METARECORD::serialize (
    DATASTREAM ds) [pure virtual]
```

Write yourself to the given file. This is virtual since some records are of arbitrary length and need to write additional information after their EMR structure.

#### Parameters

<i>ds</i>	the datastream to write oneself to.
-----------	-------------------------------------

Implemented in [EMF::EMRARC](#), [EMF::EMRARCTO](#), [EMF::EMRBEGINPATH](#), [EMF::EMRCLOSEFIGURE](#), [EMF::EMRCREATEBRUSHINDIRECT](#), [EMF::EMRCREATEPALETTE](#), [EMF::EMRCREATEPEN](#), [EMF::EMRDELETEOBJECT](#), [EMF::EMRELLIPSE](#), [EMF::EMRENDPATH](#), [EMF::EMREOF](#), [EMF::EMREXTCREATEFONTINDIRECTW](#), [EMF::EMREXTCREATEPEN](#), [EMF::EMREXTTEXTOUTA](#), [EMF::EMREXTTEXTOUTW](#), [EMF::EMRFILLPATH](#), [EMF::EMRLINETO](#), [EMF::EMRMODIFYWORLDTRANSFORM](#), [EMF::EMRMOVETOEX](#), [EMF::EMRPOLYBEZIER16](#), [EMF::EMRPOLYBEZIER](#), [EMF::EMRPOLYBEZIERTO16](#), [EMF::EMRPOLYBEZIERTO](#), [EMF::EMRPOLYGON16](#), [EMF::EMRPOLYGON](#), [EMF::EMRPOLYLINE16](#), [EMF::EMRPOLYLINE](#), [EMF::EMRPOLYLINETO16](#), [EMF::EMRPOLYLINETO](#), [EMF::EMRPOLYPOLYGON16](#), [EMF::EMRPOLYPOLYGON](#), [EMF::EMRRECTANGLE](#), [EMF::EMRRESTOREDC](#), [EMF::EMRSAVEDC](#), [EMF::EMRSCALEVIEWPORTEXTEX](#), [EMF::EMRSCALEWINDOWEXTEX](#), [EMF::EMRSELECTOBJECT](#), [EMF::EMRSETBKCOLOR](#), [EMF::EMRSETBKMODE](#), [EMF::EMRSETMAPMODE](#), [EMF::EMRSETMETARGN](#), [EMF::EMRSETMITERLIMIT](#), [EMF::EMRSETPIXELV](#), [EMF::EMRSETPOLYFILLMODE](#), [EMF::EMRSETTEXTALIGN](#), [EMF::EMRSETTEXTCOLOR](#), [EMF::EMRSETVIEWPORTEXTEX](#), [EMF::EMRSETVIEWPORTORGEX](#), [EMF::EMRSETWINDOWEXTEX](#), [EMF::EMRSETWINDOWORGEX](#), [EMF::EMRSETWORLDTRANSFORM](#), [EMF::EMRSTROKEANDFILLPATH](#), [EMF::EMRSTROKEPATH](#), and [EMF::ENHMETAHEADER](#).

### size()

```
virtual int EMF::METARECORD::size (
    void ) const [pure virtual]
```

The header record of a metafile records the total size of the metafile in bytes, so as each record is added to the list, it updates the total size.

Implemented in [EMF::EMRARC](#), [EMF::EMRARCTO](#), [EMF::EMRBEGINPATH](#), [EMF::EMRCLOSEFIGURE](#), [EMF::EMRCREATEBRUSHINDIRECT](#), [EMF::EMRCREATEPALETTE](#), [EMF::EMRCREATEPEN](#), [EMF::EMRDELETEOBJECT](#), [EMF::EMRELLIPSE](#), [EMF::EMRENDPATH](#), [EMF::EMREOF](#), [EMF::EMREXTCREATEFONTINDIRECTW](#), [EMF::EMREXTCREATEPEN](#), [EMF::EMREXTTEXTOUTA](#), [EMF::EMREXTTEXTOUTW](#), [EMF::EMRFILLPATH](#), [EMF::EMRLINETO](#), [EMF::EMRMODIFYWORLDTRANSFORM](#), [EMF::EMRMOVETOEX](#), [EMF::EMRPOLYBEZIER16](#), [EMF::EMRPOLYBEZIER](#), [EMF::EMRPOLYBEZIERTO16](#), [EMF::EMRPOLYBEZIERTO](#), [EMF::EMRPOLYGON16](#), [EMF::EMRPOLYGON](#), [EMF::EMRPOLYLINE16](#), [EMF::EMRPOLYLINE](#), [EMF::EMRPOLYLINETO16](#), [EMF::EMRPOLYLINETO](#), [EMF::EMRPOLYPOLYGON16](#), [EMF::EMRPOLYPOLYGON](#), [EMF::EMRRECTANGLE](#), [EMF::EMRRESTOREDC](#), [EMF::EMRSAVEDC](#), [EMF::EMRSCALEVIEWPORTEXTEX](#), [EMF::EMRSCALEWINDOWEXTEX](#), [EMF::EMRSELECTOBJECT](#), [EMF::EMRSETBKCOLOR](#), [EMF::EMRSETBKMODE](#), [EMF::EMRSETMAPMODE](#), [EMF::EMRSETMETARGN](#), [EMF::EMRSETMITERLIMIT](#), [EMF::EMRSETPIXELV](#), [EMF::EMRSETPOLYFILLMODE](#), [EMF::EMRSETTEXTALIGN](#), [EMF::EMRSETTEXTCOLOR](#), [EMF::EMRSETVIEWPORTEXTEX](#), [EMF::EMRSETVIEWPORTORGEX](#), [EMF::EMRSETWINDOWEXTEX](#), [EMF::EMRSETWINDOWORGEX](#), [EMF::EMRSETWORLDTRANSFORM](#), [EMF::EMRSTROKEANDFILLPATH](#), [EMF::EMRSTROKEPATH](#), and [EMF::ENHMETAHEADER](#).

Referenced by [EMF::METAFILEDEVICECONTEXT::appendHandle\(\)](#), and [EMF::METAFILEDEVICECONTEXT::appendRecord\(\)](#).

The documentation for this class was generated from the following file:

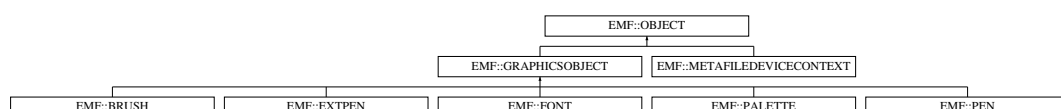
- libemf.h

## 4.67 EMF::OBJECT Class Reference

Global GDI object.

```
#include <libemf.h>
```

Inheritance diagram for EMF::OBJECT:



## Public Member Functions

- virtual `~OBJECT()`  
*OBJECTs have a virtual destructor.*
- `OBJECT()` (void)
- virtual `OBJECTTYPE getType()` (void) const =0

## Data Fields

- `HGDIOBJ handle`

### 4.67.1 Detailed Description

Global GDI object.

The GDI interface defines objects in terms of handles (rather than pointers). In order to emulate this, each object is placed into a global list and the index in that list becomes their handle.

### 4.67.2 Constructor & Destructor Documentation

#### OBJECT()

```
EMF::OBJECT::OBJECT (  
    void ) [inline]
```

Create a new object. It's up to a subclass to create a real handle for this object.

References [handle](#).

### 4.67.3 Member Function Documentation

#### getType()

```
virtual OBJECTTYPE EMF::OBJECT::getType (  
    void ) const [pure virtual]
```

Return the type of the object.

Implemented in [EMF::BRUSH](#), [EMF::EXTPEN](#), [EMF::FONT](#), [EMF::METAFILEDEVICECONTEXT](#), [EMF::PALETTE](#), and [EMF::PEN](#).

### 4.67.4 Field Documentation

#### handle

```
HGDIOBJ EMF::OBJECT::handle
```

The handle of a GDI object.

Referenced by [EMF::METAFILEDEVICECONTEXT::clearHandle\(\)](#), [EMF::GRAPHICSOBJECT::newEMR\(\)](#), and [OBJECT\(\)](#).

The documentation for this class was generated from the following file:

- `libemf.h`

## 4.68 EMF::PADDING Struct Reference

All metafile records must be padded out to a multiple of 4 bytes.

```
#include <libemf.h>
```

### Public Member Functions

- [PADDING](#) (const int size)

### Data Fields

- const int **size\_**  
*Number of bytes of padding.*

### Static Public Attributes

- static const char **padding\_** [4] = { 0, 0, 0, 0 }  
*Pad with '0's.*

#### 4.68.1 Detailed Description

All metafile records must be padded out to a multiple of 4 bytes.

Write out a few bytes of padding if necessary.

#### 4.68.2 Constructor & Destructor Documentation

##### PADDING()

```
EMF::PADDING::PADDING (  
    const int size) [inline]
```

simple constructor.

##### Parameters

<i>size</i>	number of bytes of padding to use.
-------------	------------------------------------

References [size\\_](#).

The documentation for this struct was generated from the following files:

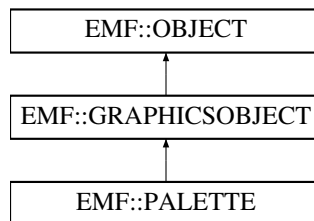
- libemf.h
- libemf.cpp

## 4.69 EMF::PALETTE Class Reference

Graphics Palette.

```
#include <libemf.h>
```

Inheritance diagram for EMF::PALETTE:



### Public Member Functions

- [PALETTE](#) (const LOGPALETTE \*lpalette)
- OBJECTTYPE [getType](#) (void) const
- METARECORD \* [newEMR](#) (HDC dc, HGDIOBJ emf\_handle)

### Public Member Functions inherited from [EMF::GRAPHICSOBJECT](#)

- virtual [~GRAPHICSOBJECT](#) ()  
*GRAPHICSOBJECTs has a virtual destructor.*

### Public Member Functions inherited from [EMF::OBJECT](#)

- virtual [~OBJECT](#) ()  
*OBJECTs have a virtual destructor.*
- [OBJECT](#) (void)

### Additional Inherited Members

#### Data Fields inherited from [EMF::GRAPHICSOBJECT](#)

- std::map< HDC, HGDIOBJ > [contexts](#)

#### Data Fields inherited from [EMF::OBJECT](#)

- HGDIOBJ [handle](#)

### 4.69.1 Detailed Description

Graphics Palette.

Not entirely sure how palettes are used in general.

## 4.69.2 Constructor & Destructor Documentation

### PALETTE()

```
EMF::PALETTE::PALETTE (  
    const LOGPALETTE * lpalette) [inline]
```

#### Parameters

<i>lpalette</i>	the (logical?) palette definition.
-----------------	------------------------------------

## 4.69.3 Member Function Documentation

### getType()

```
OBJECTTYPE EMF::PALETTE::getType (  
    void ) const [inline], [virtual]
```

Return the type of this object (could probably do better with RTTI()).

Implements [EMF::OBJECT](#).

### newEMR()

```
METARECORD * EMF::PALETTE::newEMR (  
    HDC dc,  
    HGDIOBJ emf_handle) [inline], [virtual]
```

Return a new metarecord for this object. And record its selection into the given device context.

#### Parameters

<i>dc</i>	handle of device context into which this object is being selected.
<i>emf_handle</i>	the EMF handle associated with the <a href="#">FONT</a> .

Implements [EMF::GRAPHICSOBJECT](#).

References [EMF::GRAPHICSOBJECT::contexts](#).

The documentation for this class was generated from the following file:

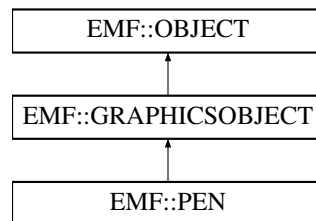
- libemf.h

## 4.70 EMF::PEN Class Reference

Graphics Pen.

```
#include <libemf.h>
```

Inheritance diagram for EMF::PEN:



### Public Member Functions

- [PEN](#) (const LOGPEN \*|pen)
- OBJECTTYPE [getType](#) (void) const
- METARECORD \* [newEMR](#) (HDC dc, HGDIOBJ emf\_handle)

### Public Member Functions inherited from [EMF::GRAPHICSOBJECT](#)

- virtual [~GRAPHICSOBJECT](#) ()  
*GRAPHICSOBJECTs has a virtual destructor.*

### Public Member Functions inherited from [EMF::OBJECT](#)

- virtual [~OBJECT](#) ()  
*OBJECTs have a virtual destructor.*
- [OBJECT](#) (void)

### Additional Inherited Members

#### Data Fields inherited from [EMF::GRAPHICSOBJECT](#)

- std::map< HDC, HGDIOBJ > [contexts](#)

#### Data Fields inherited from [EMF::OBJECT](#)

- HGDIOBJ [handle](#)

### 4.70.1 Detailed Description

Graphics Pen.

Pens are used for drawing lines, arc, rectangles, etc.

### 4.70.2 Constructor & Destructor Documentation

#### PEN()

```
EMF::PEN::PEN (
    const LOGPEN * lpen)    [inline]
```

##### Parameters

<i>lpen</i>	the (logical?) pen definition.
-------------	--------------------------------

### 4.70.3 Member Function Documentation

#### getType()

```
OBJECTTYPE EMF::PEN::getType (
    void ) const    [inline], [virtual]
```

Return the type of this object (could probably do better with RTTI()).

Implements [EMF::OBJECT](#).

#### newEMR()

```
METARECORD * EMF::PEN::newEMR (
    HDC dc,
    HGDIOBJ emf_handle)    [inline], [virtual]
```

Return a new metarecord for this object. And record its selection into the given device context.

##### Parameters

<i>dc</i>	handle of device context into which this object is being selected.
<i>emf_handle</i>	the EMF handle associated with the <a href="#">PEN</a> .

Implements [EMF::GRAPHICSOBJECT](#).

References [EMF::GRAPHICSOBJECT::contexts](#).

The documentation for this class was generated from the following file:

- libemf.h

## 4.71 EMF::POINT16ARRAY Struct Reference

Represent an array of 16-bit point in a simple way.

```
#include <libemf.h>
```

### Public Member Functions

- [POINT16ARRAY](#) (POINT16 \*const points, const DWORD n)

### Data Fields

- POINT16 \*const **points\_**  
*Array of POINT16s.*
- const DWORD **n\_**  
*Number of POINT16s in array.*

#### 4.71.1 Detailed Description

Represent an array of 16-bit point in a simple way.

Allow an array of POINT16's to be written out at once.

#### 4.71.2 Constructor & Destructor Documentation

##### POINT16ARRAY()

```
EMF::POINT16ARRAY::POINT16ARRAY (
    POINT16 *const points,
    const DWORD n) [inline]
```

Simple constructor.

##### Parameters

<i>points</i>	pointer to array of POINT16s.
<i>n</i>	number POINT16s in array.

References [n\\_](#), and [points\\_](#).

The documentation for this struct was generated from the following file:

- libemf.h

#### 4.72 EMF::POINTLARRAY Struct Reference

Represent an array of points in a simple way.

```
#include <libemf.h>
```

### Public Member Functions

- [POINTLARRAY](#) (POINTL \*const points, const DWORD n)

## Data Fields

- POINTL \*const **points\_**  
*Array of POINTLs.*
- const DWORD **n\_**  
*Number of POINTLs in array.*

### 4.72.1 Detailed Description

Represent an array of points in a simple way.

Allow an array of POINTL's to be written out at once.

### 4.72.2 Constructor & Destructor Documentation

#### POINTLARRAY()

```
EMF::POINTLARRAY::POINTLARRAY (  
    POINTL *const points,  
    const DWORD n) [inline]
```

Simple constructor.

#### Parameters

<i>points</i>	pointer to array of POINTLs.
<i>n</i>	number POINTLs in array.

References [n\\_](#), and [points\\_](#).

The documentation for this struct was generated from the following file:

- libemf.h

## 4.73 EMF::WCHARSTR Struct Reference

Represent a wide (UNICODE) character string in a simple way.

```
#include <libemf.h>
```

## Public Member Functions

- [WCHARSTR](#) (WCHAR \*const string, const int length)

## Data Fields

- WCHAR \*const **string\_**  
*String of WCHARs.*
- const int **length\_**  
*Number of WCHARs in string.*

### 4.73.1 Detailed Description

Represent a wide (UNICODE) character string in a simple way.

Even (wchar) strings have to be byte swapped. This structure allows us to provide a uniform stream-like interface for writing out all the components of metafiles.

### 4.73.2 Constructor & Destructor Documentation

#### WCHARSTR()

```
EMF::WCHARSTR::WCHARSTR (
    WCHAR *const string,
    const int length) [inline]
```

Simple constructor.

#### Parameters

<i>string</i>	pointer to string of WCHARs.
<i>length</i>	number of WCHARs in string.

References [length\\_](#), and [string\\_](#).

The documentation for this struct was generated from the following file:

- libemf.h

## 5 File Documentation

### 5.1 emf.h

```
00001 /*
00002  * EMF: A library for generating ECMA-234 Enhanced Metafiles
00003  * Copyright (C) 2002 lignum Computing, Inc. <dallenbarnett@users.sourceforge.net>
00004  * $Id: emf.h 93 2020-04-18 13:30:11Z dallenbarnett $
00005  *
00006  * This library is free software; you can redistribute it and/or
00007  * modify it under the terms of the GNU Lesser General Public
00008  * License as published by the Free Software Foundation; either
00009  * version 2.1 of the License, or (at your option) any later version.
00010  *
00011  * This library is distributed in the hope that it will be useful,
00012  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00013  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
00014  * Lesser General Public License for more details.
```

```

00015  *
00016  * You should have received a copy of the GNU Lesser General Public
00017  * License along with this library; if not, write to the Free Software
00018  * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
00019  *
00020  */
00021 #ifndef _EMF_H
00022 #define _EMF_H
00023
00024 #include <stdio.h>
00025 #include <string.h>
00026
00027 #include <libEMF/wine/windef.h>
00028 #include <libEMF/wine/winbase.h>
00029 #include <libEMF/wine/wingdi.h>
00030 #include <libEMF/wine/winuser.h>
00031 #include <libEMF/wine/winerror.h>
00032
00033 #ifdef __cplusplus
00034 extern "C" {
00035 #endif
00036 /*
00037  * Here are additional, non-"standard" routines which the author deems useful.
00038  */
00039 HDC CreateEnhMetaFileWithFILEA( HDC context, FILE* fp, const RECT* size,
00040                                LPCSTR description );
00041 HDC CreateEnhMetaFileWithFILEW( HDC context, FILE* fp, const RECT* size,
00042                                LPCWSTR description );
00043 HENHMETAFILE CloseEnhMetaFileWithFILE( HDC context );
00044 /*
00045  * This function will only produce output if the library has been compiled with
00046  * editing enabled (e.g., ./configure --enable-editing).
00047  */
00048 void WINAPI EditEnhMetaFile ( HENHMETAFILE metafile );
00049 #ifdef __cplusplus
00050 }
00051 #endif
00052
00053 #endif /* _EMF_H */

```

## 5.2 basetsd.h

```

00001 /*
00002  * Compilers that uses ILP32, LP64 or P64 type models
00003  * for both Win32 and Win64 are supported by this file.
00004  */
00005
00006 #ifndef __WINE_BASSETSD_H
00007 #define __WINE_BASSETSD_H
00008
00009 #ifdef __cplusplus
00010 extern "C" {
00011 #endif /* defined(__cplusplus) */
00012
00013 /*
00014  * Win32 was easy to implement under Unix since most (all?) 32-bit
00015  * Unices uses the same type model (ILP32) as Win32, where int, long
00016  * and pointer are 32-bit.
00017  *
00018  * Win64, however, will cause some problems when implemented under Unix.
00019  * Linux/{Alpha, Sparc64} and most (all?) other 64-bit Unices uses
00020  * the LP64 type model where int is 32-bit and long and pointer are
00021  * 64-bit. Win64 on the other hand uses the P64 (sometimes called LLP64)
00022  * type model where int and long are 32 bit and pointer is 64-bit.
00023  */
00024
00025 /* Type model indepent typedefs */
00026
00027 typedef char      __int8;
00028 typedef unsigned char __uint8;
00029
00030 typedef short     __int16;
00031 typedef unsigned short __uint16;
00032
00033 typedef int       __int32;
00034 typedef unsigned int __uint32;
00035
00036 // wogl
00037 //typedef long long    __int64;
00038 //typedef unsigned long long __uint64;
00039 typedef long       __int64;
00040 typedef unsigned long __uint64;
00041
00042 #if defined(_WIN64)

```

```
00043
00044 typedef __uint32 __ptr32;
00045 typedef void *__ptr64;
00046
00047 #else /* FIXME: defined(_WIN32) */
00048
00049 typedef void *__ptr32;
00050 typedef __uint64 __ptr64;
00051
00052 #endif
00053
00054 /* Always signed and 32 bit wide */
00055
00056 typedef __int32 LONG32;
00057 typedef __int32 INT32;
00058
00059 typedef LONG32 *PLONG32;
00060 typedef INT32 *PINT32;
00061
00062 /* Always unsigned and 32 bit wide */
00063
00064 typedef __uint32 ULONG32;
00065 typedef __uint32 DWORD32;
00066 typedef __uint32 UINT32;
00067
00068 typedef ULONG32 *PULONG32;
00069 typedef DWORD32 *PDWORD32;
00070 typedef UINT32 *PUINT32;
00071
00072 /* Always signed and 64 bit wide */
00073
00074 typedef __int64 LONG64;
00075 typedef __int64 INT64;
00076
00077 typedef LONG64 *PLONG64;
00078 typedef INT64 *PINT64;
00079
00080 /* Always unsigned and 64 bit wide */
00081
00082 typedef __uint64 ULONG64;
00083 typedef __uint64 DWORD64;
00084 typedef __uint64 UINT64;
00085
00086 typedef ULONG64 *PULONG64;
00087 typedef DWORD64 *PDWORD64;
00088 typedef UINT64 *PUINT64;
00089
00090 /* Win32 or Win64 dependent typedef/defines. */
00091
00092 #ifdef _WIN64
00093
00094 typedef __int64 INT_PTR, *PINT_PTR;
00095 typedef __uint64 UINT_PTR, *PUINT_PTR;
00096
00097 #define MAXINT_PTR 0x7fffffffffffffff
00098 #define MININT_PTR 0x8000000000000000
00099 #define MAXUINT_PTR 0xffffffffffffffff
00100
00101 typedef __int32 HALF_PTR, *PHALF_PTR;
00102 typedef __int32 UHALF_PTR, *PUHALF_PTR;
00103
00104 #define MAXHALF_PTR 0x7fffffff
00105 #define MINHALF_PTR 0x80000000
00106 #define MAXUHALF_PTR 0xffffffff
00107
00108 typedef __int64 LONG_PTR, *PLONG_PTR;
00109 typedef __uint64 ULONG_PTR, *PULONG_PTR;
00110 typedef __uint64 DWORD_PTR, *PDWORD_PTR;
00111
00112 #else /* FIXME: defined(_WIN32) */
00113
00114 typedef __int32 INT_PTR, *PINT_PTR;
00115 typedef __uint32 UINT_PTR, *PUINT_PTR;
00116
00117 #define MAXINT_PTR 0x7fffffff
00118 #define MININT_PTR 0x80000000
00119 #define MAXUINT_PTR 0xffffffff
00120
00121 typedef __int16 HALF_PTR, *PHALF_PTR;
00122 typedef __uint16 UHALF_PTR, *PUHALF_PTR;
00123
00124 #define MAXUHALF_PTR 0xffff
00125 #define MAXHALF_PTR 0x7fff
00126 #define MINHALF_PTR 0x8000
00127
00128 typedef __int32 LONG_PTR, *PLONG_PTR;
00129 typedef __uint32 ULONG_PTR, *PULONG_PTR;
```

```

00130 typedef __uint32 DWORD_PTR, *PDWORD_PTR;
00131
00132 #endif /* defined(_WIN64) || defined(_WIN32) */
00133
00134 typedef INT_PTR SSIZE_T, *PSSIZE_T;
00135 typedef UINT_PTR SIZE_T, *PSIZE_T;
00136
00137 #ifdef __cplusplus
00138 } /* extern "C" */
00139 #endif /* defined(__cplusplus) */
00140
00141 #endif /* !defined(__WINE_BASSETSD_H) */
00142
00143
00144

```

## 5.3 guiddef.h

```

00001 #ifndef GUID_DEFINED
00002 #define GUID_DEFINED
00003 typedef struct _GUID
00004 {
00005     unsigned long Data1;
00006     unsigned short Data2;
00007     unsigned short Data3;
00008     unsigned char Data4[ 8 ];
00009 } GUID;
00010 #endif
00011
00012 #undef DEFINE_GUID
00013
00014 #ifdef INITGUID
00015 #define DEFINE_GUID(name, l, w1, w2, b1, b2, b3, b4, b5, b6, b7, b8) \
00016     const GUID name = \
00017     { l, w1, w2, { b1, b2, b3, b4, b5, b6, b7, b8 } }
00018 #else
00019 #define DEFINE_GUID(name, l, w1, w2, b1, b2, b3, b4, b5, b6, b7, b8) \
00020     extern const GUID name
00021 #endif
00022
00023 #define DEFINE_OLEGUID(name, l, w1, w2) \
00024     DEFINE_GUID(name, l, w1, w2, 0xC0,0,0,0,0,0,0,0x46)
00025
00026 #ifndef _GUIDDEF_H_
00027 #define _GUIDDEF_H_
00028
00029 typedef GUID *LPGUID;
00030 typedef GUID CLSID,*LPCLSID;
00031 typedef GUID IID,*LPIID;
00032 typedef GUID FMTID,*LPFMTID;
00033
00034 #if defined(__cplusplus) && !defined(CINTERFACE)
00035 #define REFGUID const GUID &
00036 #define REFCLSID const CLSID &
00037 #define REFIID const IID &
00038 #define REFFMTID const FMTID &
00039 #else /* !defined(__cplusplus) && !defined(CINTERFACE) */
00040 #define REFGUID const GUID* const
00041 #define REFCLSID const CLSID* const
00042 #define REFIID const IID* const
00043 #define REFFMTID const FMTID* const
00044 #endif /* !defined(__cplusplus) && !defined(CINTERFACE) */
00045
00046 #if defined(__cplusplus) && !defined(CINTERFACE)
00047 #define IsEqualGUID(rguid1, rguid2) (!memcmp(&(rguid1), &(rguid2), sizeof(GUID)))
00048 #else /* defined(__cplusplus) && !defined(CINTERFACE) */
00049 #define IsEqualGUID(rguid1, rguid2) (!memcmp(rguid1, rguid2, sizeof(GUID)))
00050 #endif /* defined(__cplusplus) && !defined(CINTERFACE) */
00051 #define IsEqualIID(riid1, riid2) IsEqualGUID(riid1, riid2)
00052 #define IsEqualCLSID(rclsid1, rclsid2) IsEqualGUID(rclsid1, rclsid2)
00053
00054 #if defined(__cplusplus) && !defined(CINTERFACE)
00055 #include <string.h>
00056 inline bool operator==(const GUID& guidOne, const GUID& guidOther)
00057 {
00058     return !memcmp(&guidOne,&guidOther,sizeof(GUID));
00059 }
00060 inline bool operator!=(const GUID& guidOne, const GUID& guidOther)
00061 {
00062     return !(guidOne == guidOther);
00063 }
00064 #endif
00065
00066 extern const IID GUID_NULL;

```

```
00067 #define IID_NULL          GUID_NULL
00068 #define CLSID_NULL GUID_NULL
00069 #define FMTID_NULL          GUID_NULL
00070
00071 #endif /* _GUIDDEF_H_ */
```

## 5.4 poppack.h

```
00001 #if defined(__WINE_PSHPACK_H3)
00002 #   ifndef __WINE_INTERNAL_POPPACK
00003 #       undef __WINE_PSHPACK_H3
00004 #   endif
00005 /* Depth == 3 */
00006
00007 #   if defined(__GNUC__) || defined(__SUNPRO_C) || defined(__SUNPRO_CC)
00008 #       if __WINE_PSHPACK_H2 == 1
00009 #           pragma pack(1)
00010 #       elif __WINE_PSHPACK_H2 == 2
00011 #           pragma pack(2)
00012 #       elif __WINE_PSHPACK_H2 == 8
00013 #           pragma pack(8)
00014 #       else
00015 #           pragma pack(4)
00016 #       endif
00017 #   elif !defined(RC_INVOKED)
00018 #       error "Adjusting the alignment is not supported with this compiler"
00019 #   endif
00020
00021 #elif defined(__WINE_PSHPACK_H2)
00022 #   ifndef __WINE_INTERNAL_POPPACK
00023 #       undef __WINE_PSHPACK_H2
00024 #   endif
00025 /* Depth == 2 */
00026
00027 #   if defined(__GNUC__) || defined(__SUNPRO_C) || defined(__SUNPRO_CC)
00028 #       if __WINE_PSHPACK_H == 1
00029 #           pragma pack(1)
00030 #       elif __WINE_PSHPACK_H == 2
00031 #           pragma pack(2)
00032 #       elif __WINE_PSHPACK_H == 8
00033 #           pragma pack(8)
00034 #       else
00035 #           pragma pack(4)
00036 #       endif
00037 #   elif !defined(RC_INVOKED)
00038 #       error "Adjusting the alignment is not supported with this compiler"
00039 #   endif
00040
00041 #elif defined(__WINE_PSHPACK_H)
00042 #   ifndef __WINE_INTERNAL_POPPACK
00043 #       undef __WINE_PSHPACK_H
00044 #   endif
00045 /* Depth == 1 */
00046
00047 #   if defined(__GNUC__) || defined(__SUNPRO_C)
00048 #       pragma pack()
00049 #   elif defined(__SUNPRO_CC)
00050 /*#       warning "Assuming a default alignment of 4"*/
00051 #       pragma pack(4)
00052 #   elif !defined(RC_INVOKED)
00053 #       error "Adjusting the alignment is not supported with this compiler"
00054 #   endif
00055 #else
00056 #   if __WINE_PSHPACK_H == 0
00057 /* Depth == 0 ! */
00058
00059 #error "Popping alignment isn't possible since no alignment has been pushed"
00060
00061 #endif
00062
00063 #undef __WINE_INTERNAL_POPPACK
```

## 5.5 pshpack2.h

```
00001 #if defined(__WINE_PSHPACK_H3)
00002
00003 /* Depth > 3 */
00004 #   error "Alignment nesting > 3 is not supported"
00005
00006 #else
```

```

00007
00008 # if !defined(__WINE_PSHPACK_H)
00009 #   define __WINE_PSHPACK_H 2
00010 #   /* Depth == 1 */
00011 # elif !defined(__WINE_PSHPACK_H2)
00012 #   define __WINE_PSHPACK_H2 2
00013 #   /* Depth == 2 */
00014 #   define __WINE_INTERNAL_POPPACK
00015 #   include "poppack.h"
00016 # elif !defined(__WINE_PSHPACK_H3)
00017 #   define __WINE_PSHPACK_H3 2
00018 #   /* Depth == 3 */
00019 #   define __WINE_INTERNAL_POPPACK
00020 #   include "poppack.h"
00021 # endif
00022
00023 # if defined(__GNUC__) || defined(__SUNPRO_C) || defined(__SUNPRO_CC)
00024 #   pragma pack(2)
00025 # elif !defined(RC_INVOKED)
00026 #   error "Adjusting the alignment is not supported with this compiler"
00027 # endif
00028
00029 #endif

```

## 5.6 pshpack4.h

```

00001 #if defined(__WINE_PSHPACK_H3)
00002
00003     /* Depth > 3 */
00004 #   error "Alignment nesting > 3 is not supported"
00005
00006 #else
00007
00008 #   if !defined(__WINE_PSHPACK_H)
00009 #       define __WINE_PSHPACK_H 4
00010 #       /* Depth == 1 */
00011 #   elif !defined(__WINE_PSHPACK_H2)
00012 #       define __WINE_PSHPACK_H2 4
00013 #       /* Depth == 2 */
00014 #       define __WINE_INTERNAL_POPPACK
00015 #       include "poppack.h"
00016 #   elif !defined(__WINE_PSHPACK_H3)
00017 #       define __WINE_PSHPACK_H3 4
00018 #       /* Depth == 3 */
00019 #       define __WINE_INTERNAL_POPPACK
00020 #       include "poppack.h"
00021 #   endif
00022
00023 #   if defined(__GNUC__) || defined(__SUNPRO_C) || defined(__SUNPRO_CC)
00024 #       pragma pack(4)
00025 #   elif !defined(RC_INVOKED)
00026 #       error "Adjusting the alignment is not supported with this compiler"
00027 #   endif
00028
00029 #endif

```

## 5.7 w16.h

```

00001 /*
00002  * These are some left-over definitions which are not supported
00003  * in WINE any more, but still show up in metafiles. They are
00004  * not exposed as API.
00005  */
00006 #ifndef W16_H
00007 #define W16_H
00008
00009 #ifdef __cplusplus
00010 extern "C" {
00011 #endif
00012
00013 /* Standard data types */
00014
00015 typedef short          INT16;
00016 typedef unsigned short UINT16;
00017 typedef unsigned short BOOL16;
00018
00019 typedef HDC            HDC16;
00020
00021 /* The POINT structure */
00022

```

```

00023 typedef struct
00024 {
00025     INT16  x;
00026     INT16  y;
00027 } POINT16, *PPOINT16, *LPPOINT16;
00028
00029 typedef struct {
00030     EMR      emr;
00031     RECTL    rclBounds;
00032     DWORD    cpts;
00033     POINT16  apts[1];
00034 } EMRPOLYLINE16, *PEMRPOLYLINE16,
00035 EMRPOLYBEZIER16, *PEMRPOLYBEZIER16,
00036 EMRPOLYGON16, *PEMRPOLYGON16,
00037 EMRPOLYBEZIERTO16, *PEMRPOLYBEZIERTO16,
00038 EMRPOLYLINETO16, *PEMRPOLYLINETO16;
00039
00040 typedef struct {
00041     EMR      emr;
00042     RECTL    rclBounds;
00043     DWORD    nPolys;
00044     DWORD    cpts;
00045     DWORD    aPolyCounts[1];
00046     POINT16  apts[1];
00047 } EMRPOLYPOLYLINE16, *PEMRPOLYPOLYLINE16,
00048 EMRPOLYPOLYGON16, *PEMRPOLYPOLYGON16;
00049
00050 BOOL      WINAPI PolyBezier16(HDC16, const POINT16*, INT16);
00051 BOOL      WINAPI PolyBezierTo16(HDC16, const POINT16*, INT16);
00052 BOOL      WINAPI Polyline16(HDC16, const POINT16*, INT16);
00053 BOOL      WINAPI PolylineTo16(HDC16, const POINT16*, INT16);
00054 BOOL      WINAPI Polygon16(HDC16, const POINT16*, INT16);
00055 BOOL      WINAPI PolyPolygon16(HDC16, const POINT16*, const INT*, UINT16);
00056 #ifdef __cplusplus
00057 }
00058 #endif
00059
00060
00061 #endif /* W16_H */

```

## 5.8 winbase.h

```

00001 #ifndef __WINE_WINBASE_H
00002 #define __WINE_WINBASE_H
00003
00004 #ifndef RC_INVOKED
00005 #include <stdarg.h>
00006 #endif
00007
00008 #include "basetsd.h"
00009 #include "windef.h"
00010
00011 #ifndef RC_INVOKED
00012 #include <stdarg.h>
00013 #endif
00014
00015 #ifdef __cplusplus
00016 extern "C" {
00017 #endif
00018
00019     /* Windows Exit Procedure flag values */
00020     #define WEP_FREE_DLL      0
00021     #define WEP_SYSTEM_EXIT   1
00022
00023     typedef DWORD CALLBACK (*LPTHREAD_START_ROUTINE) (LPVOID);
00024
00025     typedef VOID /*WINAPI*/ (*PFIBER_START_ROUTINE) ( LPVOID lpFiberParameter );
00026     typedef PFIBER_START_ROUTINE LPFIBER_START_ROUTINE;
00027
00028     typedef RTL_CRITICAL_SECTION CRITICAL_SECTION;
00029     typedef PRTL_CRITICAL_SECTION PCRITICAL_SECTION;
00030     typedef PRTL_CRITICAL_SECTION LPCRITICAL_SECTION;
00031
00032     typedef RTL_CRITICAL_SECTION_DEBUG CRITICAL_SECTION_DEBUG;
00033     typedef PRTL_CRITICAL_SECTION_DEBUG PCRITICAL_SECTION_DEBUG;
00034     typedef PRTL_CRITICAL_SECTION_DEBUG LPCRITICAL_SECTION_DEBUG;
00035
00036
00037     #define EXCEPTION_DEBUG_EVENT      1
00038     #define CREATE_THREAD_DEBUG_EVENT  2
00039     #define CREATE_PROCESS_DEBUG_EVENT 3
00040     #define EXIT_THREAD_DEBUG_EVENT    4
00041     #define EXIT_PROCESS_DEBUG_EVENT   5
00042     #define LOAD_DLL_DEBUG_EVENT       6

```

```

00043 #define UNLOAD_DLL_DEBUG_EVENT 7
00044 #define OUTPUT_DEBUG_STRING_EVENT 8
00045 #define RIP_EVENT 9
00046
00047 typedef struct _EXCEPTION_DEBUG_INFO {
00048     EXCEPTION_RECORD ExceptionRecord;
00049     DWORD dwFirstChance;
00050 } EXCEPTION_DEBUG_INFO;
00051
00052 typedef struct _CREATE_THREAD_DEBUG_INFO {
00053     HANDLE hThread;
00054     LPVOID lpThreadLocalBase;
00055     LPTHREAD_START_ROUTINE lpStartAddress;
00056 } CREATE_THREAD_DEBUG_INFO;
00057
00058 typedef struct _CREATE_PROCESS_DEBUG_INFO {
00059     HANDLE hFile;
00060     HANDLE hProcess;
00061     HANDLE hThread;
00062     LPVOID lpBaseOfImage;
00063     DWORD dwDebugInfoFileOffset;
00064     DWORD nDebugInfoSize;
00065     LPVOID lpThreadLocalBase;
00066     LPTHREAD_START_ROUTINE lpStartAddress;
00067     LPVOID lpImageName;
00068     WORD fUnicode;
00069 } CREATE_PROCESS_DEBUG_INFO;
00070
00071 typedef struct _EXIT_THREAD_DEBUG_INFO {
00072     DWORD dwExitCode;
00073 } EXIT_THREAD_DEBUG_INFO;
00074
00075 typedef struct _EXIT_PROCESS_DEBUG_INFO {
00076     DWORD dwExitCode;
00077 } EXIT_PROCESS_DEBUG_INFO;
00078
00079 typedef struct _LOAD_DLL_DEBUG_INFO {
00080     HANDLE hFile;
00081     LPVOID lpBaseOfDll;
00082     DWORD dwDebugInfoFileOffset;
00083     DWORD nDebugInfoSize;
00084     LPVOID lpImageName;
00085     WORD fUnicode;
00086 } LOAD_DLL_DEBUG_INFO;
00087
00088 typedef struct _UNLOAD_DLL_DEBUG_INFO {
00089     LPVOID lpBaseOfDll;
00090 } UNLOAD_DLL_DEBUG_INFO;
00091
00092 typedef struct _OUTPUT_DEBUG_STRING_INFO {
00093     LPSTR lpDebugStringData;
00094     WORD fUnicode;
00095     WORD nDebugStringLength;
00096 } OUTPUT_DEBUG_STRING_INFO;
00097
00098 typedef struct _RIP_INFO {
00099     DWORD dwError;
00100     DWORD dwType;
00101 } RIP_INFO;
00102
00103 typedef struct _DEBUG_EVENT {
00104     DWORD dwDebugEventCode;
00105     DWORD dwProcessId;
00106     DWORD dwThreadId;
00107     union ull {
00108         EXCEPTION_DEBUG_INFO Exception;
00109         CREATE_THREAD_DEBUG_INFO CreateThread;
00110         CREATE_PROCESS_DEBUG_INFO CreateProcessInfo;
00111         EXIT_THREAD_DEBUG_INFO ExitThread;
00112         EXIT_PROCESS_DEBUG_INFO ExitProcess;
00113         LOAD_DLL_DEBUG_INFO LoadDll;
00114         UNLOAD_DLL_DEBUG_INFO UnloadDll;
00115         OUTPUT_DEBUG_STRING_INFO DebugString;
00116         RIP_INFO RipInfo;
00117     } u;
00118 } DEBUG_EVENT, *LPDEBUG_EVENT;
00119
00120 typedef PCONTEXT LPCONTEXT;
00121 typedef PEXCEPTION_RECORD LPEXCEPTION_RECORD;
00122 typedef PEXCEPTION_POINTERS LPEXCEPTION_POINTERS;
00123
00124 #define OFS_MAXPATHNAME 128
00125 typedef struct
00126 {
00127     BYTE cBytes;
00128     BYTE fFixedDisk;
00129     WORD nErrCode;

```

```

00130     BYTE reserved[4];
00131     BYTE szPathName[OFS_MAXPATHNAME];
00132 } OFSTRUCT, *POFSTRUCT, *LPOFSTRUCT;
00133
00134 #define OF_READ                0x0000
00135 #define OF_WRITE               0x0001
00136 #define OF_READWRITE          0x0002
00137 #define OF_SHARE_COMPAT       0x0000
00138 #define OF_SHARE_EXCLUSIVE    0x0010
00139 #define OF_SHARE_DENY_WRITE   0x0020
00140 #define OF_SHARE_DENY_READ    0x0030
00141 #define OF_SHARE_DENY_NONE    0x0040
00142 #define OF_PARSE               0x0100
00143 #define OF_DELETE              0x0200
00144 #define OF_VERIFY              0x0400    /* Used with OF_REOPEN */
00145 #define OF_SEARCH              0x0400    /* Used without OF_REOPEN */
00146 #define OF_CANCEL              0x0800
00147 #define OF_CREATE              0x1000
00148 #define OF_PROMPT              0x2000
00149 #define OF_EXIST               0x4000
00150 #define OF_REOPEN              0x8000
00151
00152 /* SetErrorMode values */
00153 #define SEM_FAILCRITICALERRORS 0x0001
00154 #define SEM_NOGPFAULTERRORBOX  0x0002
00155 #define SEM_NOALIGNMENTFAULTEXCEPT 0x0004
00156 #define SEM_NOOPENFILEERRORBOX 0x8000
00157
00158 /* CopyFileEx flags */
00159 #define COPY_FILE_FAIL_IF_EXISTS 0x00000001
00160 #define COPY_FILE_RESTARTABLE   0x00000002
00161 #define COPY_FILE_OPEN_SOURCE_FOR_WRITE 0x00000004
00162
00163 /* GetTempFileName() Flags */
00164 #define TF_FORCEDRIVE           0x80
00165
00166 #define DRIVE_UNKNOWN           0
00167 #define DRIVE_NO_ROOT_DIR      1
00168 #define DRIVE_REMOVABLE        2
00169 #define DRIVE_FIXED            3
00170 #define DRIVE_REMOTE           4
00171 /* Win32 additions */
00172 #define DRIVE_CDROM            5
00173 #define DRIVE_RAMDISK          6
00174
00175 /* The security attributes structure */
00176 typedef struct _SECURITY_ATTRIBUTES
00177 {
00178     DWORD    nLength;
00179     LPVOID   lpSecurityDescriptor;
00180     BOOL     bInheritHandle;
00181 } SECURITY_ATTRIBUTES, *PSECURITY_ATTRIBUTES, *LPSECURITY_ATTRIBUTES;
00182
00183 #ifndef _FILETIME_
00184 #define _FILETIME_
00185 /* 64 bit number of 100 nanoseconds intervals since January 1, 1601 */
00186 typedef struct
00187 {
00188     DWORD    dwLowDateTime;
00189     DWORD    dwHighDateTime;
00190 } FILETIME, *PFILETIME, *LPFILETIME;
00191 #endif /* _FILETIME_ */
00192
00193 /* Find* structures */
00194 typedef struct
00195 {
00196     DWORD    dwFileAttributes;
00197     FILETIME ftCreationTime;
00198     FILETIME ftLastAccessTime;
00199     FILETIME ftLastWriteTime;
00200     DWORD    nFileSizeHigh;
00201     DWORD    nFileSizeLow;
00202     DWORD    dwReserved0;
00203     DWORD    dwReserved1;
00204     CHAR     cFileName[260];
00205     CHAR     cAlternateFileName[14];
00206 } WIN32_FIND_DATA, *PWIN32_FIND_DATA, *LPWIN32_FIND_DATA;
00207
00208 typedef struct
00209 {
00210     DWORD    dwFileAttributes;
00211     FILETIME ftCreationTime;
00212     FILETIME ftLastAccessTime;
00213     FILETIME ftLastWriteTime;
00214     DWORD    nFileSizeHigh;
00215     DWORD    nFileSizeLow;
00216     DWORD    dwReserved0;

```

```
00217     DWORD        dwReserved1;
00218     WCHAR         cFileName[260];
00219     WCHAR         cAlternateFileName[14];
00220 } WIN32_FIND_DATAW, *PWIN32_FIND_DATAW, *LPWIN32_FIND_DATAW;
00221
00222 DECL_WINELIB_TYPE_AW(WIN32_FIND_DATA)
00223 DECL_WINELIB_TYPE_AW(PWIN32_FIND_DATA)
00224 DECL_WINELIB_TYPE_AW(LPWIN32_FIND_DATA)
00225
00226 typedef enum _FINDEX_INFO_LEVELS
00227 {
00228     FindExInfoStandard,
00229     FindExInfoMaxInfoLevel
00230 } FINDEX_INFO_LEVELS;
00231
00232 typedef enum _FINDEX_SEARCH_OPS
00233 {
00234     FindExSearchNameMatch,
00235     FindExSearchLimitToDirectories,
00236     FindExSearchLimitToDevices,
00237     FindExSearchMaxSearchOp
00238 } FINDEX_SEARCH_OPS;
00239
00240 typedef struct
00241 {
00242     LPVOID lpData;
00243     DWORD cbData;
00244     BYTE cbOverhead;
00245     BYTE iRegionIndex;
00246     WORD wFlags;
00247     union u21 {
00248         struct {
00249             HANDLE hMem;
00250             DWORD dwReserved[3];
00251         } Block;
00252         struct {
00253             DWORD dwCommittedSize;
00254             DWORD dwUnCommittedSize;
00255             LPVOID lpFirstBlock;
00256             LPVOID lpLastBlock;
00257         } Region;
00258     } DUMMYUNIONNAME;
00259 } PROCESS_HEAP_ENTRY, *PPROCESS_HEAP_ENTRY, *LPPROCESS_HEAP_ENTRY;
00260
00261 #define PROCESS_HEAP_REGION 0x0001
00262 #define PROCESS_HEAP_UNCOMMITTED_RANGE 0x0002
00263 #define PROCESS_HEAP_ENTRY_BUSY 0x0004
00264 #define PROCESS_HEAP_ENTRY_MOVEABLE 0x0010
00265 #define PROCESS_HEAP_ENTRY_DDESHARE 0x0020
00266
00267 #define INVALID_HANDLE_VALUE ((HANDLE) -1)
00268
00269 #define TLS_OUT_OF_INDEXES ((DWORD) 0xFFFFFFFF)
00270
00271 #define SHUTDOWN_NORETRY 1
00272
00273 /* comm */
00274
00275 #define CBR_110 0xFF10
00276 #define CBR_300 0xFF11
00277 #define CBR_600 0xFF12
00278 #define CBR_1200 0xFF13
00279 #define CBR_2400 0xFF14
00280 #define CBR_4800 0xFF15
00281 #define CBR_9600 0xFF16
00282 #define CBR_14400 0xFF17
00283 #define CBR_19200 0xFF18
00284 #define CBR_38400 0xFF1B
00285 #define CBR_56000 0xFF1F
00286 #define CBR_57600 0xFF20
00287 #define CBR_115200 0xFF21
00288 #define CBR_128000 0xFF23
00289 #define CBR_256000 0xFF27
00290
00291 #define NOPARITY 0
00292 #define ODDPARITY 1
00293 #define EVENPARITY 2
00294 #define MARKPARITY 3
00295 #define SPACEPARITY 4
00296 #define ONESTOPBIT 0
00297 #define ONE5STOPBITS 1
00298 #define TWOSTOPBITS 2
00299
00300 #define IGNORE 0
00301 #define INFINITE 0xFFFFFFFF
00302
00303 #define CE_RXOVER 0x0001
```

```
00304 #define CE_OVERRUN 0x0002
00305 #define CE_RXPARITY 0x0004
00306 #define CE_FRAME 0x0008
00307 #define CE_BREAK 0x0010
00308 #define CE_CTSTO 0x0020
00309 #define CE_DSRTO 0x0040
00310 #define CE_RLSDTO 0x0080
00311 #define CE_TXFULL 0x0100
00312 #define CE_PTO 0x0200
00313 #define CE_IOE 0x0400
00314 #define CE_DNS 0x0800
00315 #define CE_OOP 0x1000
00316 #define CE_MODE 0x8000
00317
00318 #define IE_BADID -1
00319 #define IE_OPEN -2
00320 #define IE_NOPEN -3
00321 #define IE_MEMORY -4
00322 #define IE_DEFAULT -5
00323 #define IE_HARDWARE -10
00324 #define IE_BYTESIZE -11
00325 #define IE_BAUDRATE -12
00326
00327 #define EV_RXCHAR 0x0001
00328 #define EV_RXFLAG 0x0002
00329 #define EV_TXEMPT 0x0004
00330 #define EV_CTS 0x0008
00331 #define EV_DSR 0x0010
00332 #define EV_RLSD 0x0020
00333 #define EV_BREAK 0x0040
00334 #define EV_ERR 0x0080
00335 #define EV_RING 0x0100
00336 #define EV_PERR 0x0200
00337 #define EV_RX80FULL 0x0400
00338 #define EV_EVENT1 0x0800
00339 #define EV_EVENT2 0x1000
00340
00341 #define SETXOFF 1
00342 #define SETXON 2
00343 #define SETRTS 3
00344 #define CLRRTS 4
00345 #define SETDTR 5
00346 #define CLRDTR 6
00347 #define RESETDEV 7
00348 #define SETBREAK 8
00349 #define CLRBREAK 9
00350
00351 /* Purge functions for Comm Port */
00352 #define PURGE_TXABORT 0x0001 /* Kill the pending/current writes to the
00353                                comm port */
00354 #define PURGE_RXABORT 0x0002 /*Kill the pending/current reads to
00355                                the comm port */
00356 #define PURGE_TXCLEAR 0x0004 /* Kill the transmit queue if there*/
00357 #define PURGE_RXCLEAR 0x0008 /* Kill the typeahead buffer if there*/
00358
00359
00360 /* Modem Status Flags */
00361 #define MS_CTS_ON ((DWORD)0x0010)
00362 #define MS_DSR_ON ((DWORD)0x0020)
00363 #define MS_RING_ON ((DWORD)0x0040)
00364 #define MS_RLSD_ON ((DWORD)0x0080)
00365
00366 #define RTS_CONTROL_DISABLE 0
00367 #define RTS_CONTROL_ENABLE 1
00368 #define RTS_CONTROL_HANDSHAKE 2
00369 #define RTS_CONTROL_TOGGLE 3
00370
00371 #define DTR_CONTROL_DISABLE 0
00372 #define DTR_CONTROL_ENABLE 1
00373 #define DTR_CONTROL_HANDSHAKE 2
00374
00375
00376 #define LMEM_FIXED 0
00377 #define LMEM_MOVEABLE 0x0002
00378 #define LMEM_NOCOMPACT 0x0010
00379 #define LMEM_NODISCARD 0x0020
00380 #define LMEM_ZEROINIT 0x0040
00381 #define LMEM_MODIFY 0x0080
00382 #define LMEM_DISCARDABLE 0x0F00
00383 #define LMEM_DISCARDED 0x4000
00384 #define LMEM_LOCKCOUNT 0x00FF
00385
00386 #define LPTR (LMEM_FIXED | LMEM_ZEROINIT)
00387 #define LHND (LMEM_MOVEABLE | LMEM_ZEROINIT)
00388
00389 #define NONZEROLHND (LMEM_MOVEABLE)
00390 #define NONZEROLPTR (LMEM_FIXED)
```

```

00391
00392 #define GMEM_FIXED            0x0000
00393 #define GMEM_MOVEABLE        0x0002
00394 #define GMEM_NOCOMPACT       0x0010
00395 #define GMEM_NODISCARD       0x0020
00396 #define GMEM_ZEROINIT        0x0040
00397 #define GMEM_MODIFY          0x0080
00398 #define GMEM_DISCARDABLE     0x0100
00399 #define GMEM_NOT_BANKED      0x1000
00400 #define GMEM_SHARE            0x2000
00401 #define GMEM_DDESHARE        0x2000
00402 #define GMEM_NOTIFY          0x4000
00403 #define GMEM_LOWER            GMEM_NOT_BANKED
00404 #define GMEM_DISCARDED       0x4000
00405 #define GMEM_LOCKCOUNT      0x00ff
00406 #define GMEM_INVALID_HANDLE  0x8000
00407
00408 #define GHND                   (GMEM_MOVEABLE | GMEM_ZEROINIT)
00409 #define GPTR                   (GMEM_FIXED | GMEM_ZEROINIT)
00410
00411 #define INVALID_ATOM           ((ATOM) 0)
00412 #define MAXINTATOM            0xc000
00413 #define MAKEINTATOMA(atom)    ((LPCSTR) ((ULONG_PTR) ((WORD) (atom))))
00414 #define MAKEINTATOMW(atom)    ((LPCWSTR) ((ULONG_PTR) ((WORD) (atom))))
00415 #define MAKEINTATOM WINELIB_NAME_AW(MAKEINTATOM)
00416
00417 typedef struct tagMEMORYSTATUS
00418 {
00419     DWORD    dwLength;
00420     DWORD    dwMemoryLoad;
00421     DWORD    dwTotalPhys;
00422     DWORD    dwAvailPhys;
00423     DWORD    dwTotalPageFile;
00424     DWORD    dwAvailPageFile;
00425     DWORD    dwTotalVirtual;
00426     DWORD    dwAvailVirtual;
00427 } MEMORYSTATUS, *LPMEMORYSTATUS;
00428
00429
00430 typedef struct {
00431     WORD wYear;
00432     WORD wMonth;
00433     WORD wDayOfWeek;
00434     WORD wDay;
00435     WORD wHour;
00436     WORD wMinute;
00437     WORD wSecond;
00438     WORD wMilliseconds;
00439 } SYSTEMTIME, *PSYSTEMTIME, *LPSYSTEMTIME;
00440
00441 /* The 'overlapped' data structure used by async I/O functions.
00442 */
00443 typedef struct {
00444     DWORD Internal;
00445     DWORD InternalHigh;
00446     DWORD Offset;
00447     DWORD OffsetHigh;
00448     HANDLE hEvent;
00449 } OVERLAPPED, *LPOVERLAPPED;
00450
00451 typedef VOID CALLBACK (*LPOVERLAPPED_COMPLETION_ROUTINE) (DWORD dwErrorCode, DWORD
dwNumberOfBytesTransferred, LPOVERLAPPED lpOverlapped);
00452
00453 /* Process startup information.
00454 */
00455
00456 /* STARTUPINFO.dwFlags */
00457 #define STARTF_USESHOWWINDOW    0x00000001
00458 #define STARTF_USESIZE          0x00000002
00459 #define STARTF_USEPOSITION      0x00000004
00460 #define STARTF_USECOUNTCHARS  0x00000008
00461 #define STARTF_USEFILLATTRIBUTE 0x00000010
00462 #define STARTF_RUNFULLSCREEN    0x00000020
00463 #define STARTF_FORCEONFEEDBACK  0x00000040
00464 #define STARTF_FORCEOFFFEEDBACK 0x00000080
00465 #define STARTF_USESTDHANDLES    0x00000100
00466 #define STARTF_USEHOTKEY        0x00000200
00467
00468 typedef struct {
00469     DWORD cb; /* 00: size of struct */
00470     LPSTR lpReserved; /* 04: */
00471     LPSTR lpDesktop; /* 08: */
00472     LPSTR lpTitle; /* 0c: */
00473     DWORD dwX; /* 10: */
00474     DWORD dwY; /* 14: */
00475     DWORD dwXSize; /* 18: */
00476     DWORD dwYSize; /* 1c: */

```

```

00477     DWORD dwXCountChars;    /* 20: */
00478     DWORD dwYCountChars;    /* 24: */
00479     DWORD dwFillAttribute;   /* 28: */
00480     DWORD dwFlags;           /* 2c: */
00481     WORD wShowWindow;        /* 30: */
00482     WORD cbReserved2;        /* 32: */
00483     BYTE *lpReserved2;       /* 34: */
00484     HANDLE hStdInput;         /* 38: */
00485     HANDLE hStdOutput;        /* 3c: */
00486     HANDLE hStdError;         /* 40: */
00487 } STARTUPINFOA, *LPSTARTUPINFOA;
00488
00489 typedef struct {
00490     DWORD cb;
00491     LPWSTR lpReserved;
00492     LPWSTR lpDesktop;
00493     LPWSTR lpTitle;
00494     DWORD dwX;
00495     DWORD dwY;
00496     DWORD dwXSize;
00497     DWORD dwYSize;
00498     DWORD dwXCountChars;
00499     DWORD dwYCountChars;
00500     DWORD dwFillAttribute;
00501     DWORD dwFlags;
00502     WORD wShowWindow;
00503     WORD cbReserved2;
00504     BYTE *lpReserved2;
00505     HANDLE hStdInput;
00506     HANDLE hStdOutput;
00507     HANDLE hStdError;
00508 } STARTUPINFOW, *LPSTARTUPINFOW;
00509
00510 DECL_WINELIB_TYPE_AW(STARTUPINFO)
00511 DECL_WINELIB_TYPE_AW(LPSTARTUPINFO)
00512
00513 typedef struct {
00514     HANDLE hProcess;
00515     HANDLE hThread;
00516     DWORD dwProcessId;
00517     DWORD dwThreadId;
00518 } PROCESS_INFORMATION, *PPROCESS_INFORMATION, *LPPROCESS_INFORMATION;
00519
00520 typedef struct {
00521     LONG Bias;
00522     WCHAR StandardName[32];
00523     SYSTEMTIME StandardDate;
00524     LONG StandardBias;
00525     WCHAR DaylightName[32];
00526     SYSTEMTIME DaylightDate;
00527     LONG DaylightBias;
00528 } TIME_ZONE_INFORMATION, *PTIME_ZONE_INFORMATION, *LPTIME_ZONE_INFORMATION;
00529
00530 #define TIME_ZONE_ID_INVALID ((DWORD) 0xFFFFFFFF)
00531 #define TIME_ZONE_ID_UNKNOWN 0
00532 #define TIME_ZONE_ID_STANDARD 1
00533 #define TIME_ZONE_ID_DAYLIGHT 2
00534
00535 /* CreateProcess: dwCreationFlag values
00536 */
00537 #define DEBUG_PROCESS 0x00000001
00538 #define DEBUG_ONLY_THIS_PROCESS 0x00000002
00539 #define CREATE_SUSPENDED 0x00000004
00540 #define DETACHED_PROCESS 0x00000008
00541 #define CREATE_NEW_CONSOLE 0x00000010
00542 #define NORMAL_PRIORITY_CLASS 0x00000020
00543 #define IDLE_PRIORITY_CLASS 0x00000040
00544 #define HIGH_PRIORITY_CLASS 0x00000080
00545 #define REALTIME_PRIORITY_CLASS 0x00000100
00546 #define CREATE_NEW_PROCESS_GROUP 0x00000200
00547 #define CREATE_UNICODE_ENVIRONMENT 0x00000400
00548 #define CREATE_SEPARATE_WOW_VDM 0x00000800
00549 #define CREATE_SHARED_WOW_VDM 0x00001000
00550 #define CREATE_DEFAULT_ERROR_MODE 0x04000000
00551 #define CREATE_NO_WINDOW 0x08000000
00552 #define PROFILE_USER 0x10000000
00553 #define PROFILE_KERNEL 0x20000000
00554 #define PROFILE_SERVER 0x40000000
00555
00556
00557 /* File object type definitions
00558 */
00559 #define FILE_TYPE_UNKNOWN 0
00560 #define FILE_TYPE_DISK 1
00561 #define FILE_TYPE_CHAR 2
00562 #define FILE_TYPE_PIPE 3
00563 #define FILE_TYPE_REMOTE 32768

```

```

00564
00565 /* File creation flags
00566 */
00567 #define FILE_FLAG_WRITE_THROUGH 0x80000000UL
00568 #define FILE_FLAG_OVERLAPPED 0x40000000L
00569 #define FILE_FLAG_NO_BUFFERING 0x20000000L
00570 #define FILE_FLAG_RANDOM_ACCESS 0x10000000L
00571 #define FILE_FLAG_SEQUENTIAL_SCAN 0x08000000L
00572 #define FILE_FLAG_DELETE_ON_CLOSE 0x04000000L
00573 #define FILE_FLAG_BACKUP_SEMANTICS 0x02000000L
00574 #define FILE_FLAG_POSIX_SEMANTICS 0x01000000L
00575 #define CREATE_NEW 1
00576 #define CREATE_ALWAYS 2
00577 #define OPEN_EXISTING 3
00578 #define OPEN_ALWAYS 4
00579 #define TRUNCATE_EXISTING 5
00580
00581 /* Standard handle identifiers
00582 */
00583 #define STD_INPUT_HANDLE ((DWORD) -10)
00584 #define STD_OUTPUT_HANDLE ((DWORD) -11)
00585 #define STD_ERROR_HANDLE ((DWORD) -12)
00586
00587 typedef struct
00588 {
00589     DWORD dwFileAttributes;
00590     FILETIME ftCreationTime;
00591     FILETIME ftLastAccessTime;
00592     FILETIME ftLastWriteTime;
00593     DWORD dwVolumeSerialNumber;
00594     DWORD nFileSizeHigh;
00595     DWORD nFileSizeLow;
00596     DWORD nNumberOfLinks;
00597     DWORD nFileIndexHigh;
00598     DWORD nFileIndexLow;
00599 } BY_HANDLE_FILE_INFORMATION, *PBY_HANDLE_FILE_INFORMATION, *LPBY_HANDLE_FILE_INFORMATION ;
00600
00601 #define PIPE_ACCESS_INBOUND 1
00602 #define PIPE_ACCESS_OUTBOUND 2
00603 #define PIPE_ACCESS_DUPLEX 3
00604
00605 #define PIPE_TYPE_BYTE 0
00606 #define PIPE_TYPE_MESSAGE 4
00607
00608 #define PIPE_READMODE_BYTE 0
00609 #define PIPE_READMODE_MESSAGE 2
00610
00611 #define PIPE_WAIT 0
00612 #define PIPE_NOWAIT 1
00613
00614 #define PIPE_UNLIMITED_INSTANCES 255
00615
00616 #define NMPWAIT_WAIT_FOREVER 0xffffffff
00617 #define NMPWAIT_NOWAIT 0x00000001
00618 #define NMPWAIT_USE_DEFAULT_WAIT 0x00000000
00619
00620 typedef struct _SYSTEM_POWER_STATUS
00621 {
00622     BYTE ACLineStatus;
00623     BYTE BatteryFlag;
00624     BYTE BatteryLifePercent;
00625     BYTE reserved;
00626     DWORD BatteryLifeTime;
00627     DWORD BatteryFullLifeTime;
00628 } SYSTEM_POWER_STATUS, *LPSYSTEM_POWER_STATUS;
00629
00630
00631 typedef struct tagSYSTEM_INFO
00632 {
00633     union u3 {
00634         DWORD dwOemId; /* Obsolete field - do not use */
00635         struct splits {
00636             WORD wProcessorArchitecture;
00637             WORD wReserved;
00638         } DUMMYSTRUCTNAME;
00639     } DUMMYUNIONNAME;
00640     DWORD dwPageSize;
00641     LPVOID lpMinimumApplicationAddress;
00642     LPVOID lpMaximumApplicationAddress;
00643     DWORD dwActiveProcessorMask;
00644     DWORD dwNumberOfProcessors;
00645     DWORD dwProcessorType;
00646     DWORD dwAllocationGranularity;
00647     WORD wProcessorLevel;
00648     WORD wProcessorRevision;
00649 } SYSTEM_INFO, *LPSYSTEM_INFO;
00650

```

```

00651 typedef BOOL CALLBACK (*ENUMRESTYPEPROCA) (HMODULE, LPSTR, LONG);
00652 typedef BOOL CALLBACK (*ENUMRESTYPEPROCW) (HMODULE, LPWSTR, LONG);
00653 typedef BOOL CALLBACK (*ENUMRESNAMEPROCA) (HMODULE, LPCSTR, LPSTR, LONG);
00654 typedef BOOL CALLBACK (*ENUMRESNAMEPROCW) (HMODULE, LPCWSTR, LPWSTR, LONG);
00655 typedef BOOL CALLBACK (*ENUMRESLANGPROCA) (HMODULE, LPCSTR, LPCSTR, WORD, LONG);
00656 typedef BOOL CALLBACK (*ENUMRESLANGPROCW) (HMODULE, LPCWSTR, LPCWSTR, WORD, LONG);
00657
00658 DECL_WINELIB_TYPE_AW(ENUMRESTYPEPROC)
00659 DECL_WINELIB_TYPE_AW(ENUMRESNAMEPROC)
00660 DECL_WINELIB_TYPE_AW(ENUMRESLANGPROC)
00661
00662 /* flags that can be passed to LoadLibraryEx */
00663 #define DONT_RESOLVE_DLL_REFERENCES 0x00000001
00664 #define LOAD_LIBRARY_AS_DATAFILE 0x00000002
00665 #define LOAD_WITH_ALTERED_SEARCH_PATH 0x00000008
00666
00667 /* ifdef _x86_ ... */
00668 typedef struct _LDT_ENTRY {
00669     WORD    LimitLow;
00670     WORD    BaseLow;
00671     union u4 {
00672         struct {
00673             BYTE    BaseMid;
00674             BYTE    Flags1; /*Declare as bytes to avoid alignment problems */
00675             BYTE    Flags2;
00676             BYTE    BaseHi;
00677         } Bytes;
00678         struct {
00679             unsigned BaseMid      : 8;
00680             unsigned Type        : 5;
00681             unsigned Dpl        : 2;
00682             unsigned Pres       : 1;
00683             unsigned LimitHi    : 4;
00684             unsigned Sys        : 1;
00685             unsigned Reserved_0 : 1;
00686             unsigned Default_Big : 1;
00687             unsigned Granularity : 1;
00688             unsigned BaseHi     : 8;
00689         } Bits;
00690     } HighWord;
00691 } LDT_ENTRY, *LPLDT_ENTRY;
00692
00693
00694 typedef enum _GET_FILEEX_INFO_LEVELS {
00695     GetFileExInfoStandard
00696 } GET_FILEEX_INFO_LEVELS;
00697
00698 typedef struct _WIN32_FILE_ATTRIBUTES_DATA {
00699     DWORD    dwFileAttributes;
00700     FILETIME ftCreationTime;
00701     FILETIME ftLastAccessTime;
00702     FILETIME ftLastWriteTime;
00703     DWORD    nFileSizeHigh;
00704     DWORD    nFileSizeLow;
00705 } WIN32_FILE_ATTRIBUTE_DATA, *LPWIN32_FILE_ATTRIBUTE_DATA;
00706
00707 /*
00708 * This one seems to be a Win32 only definition. It also is defined with
00709 * WINAPI instead of CALLBACK in the windows headers.
00710 */
00711 typedef DWORD CALLBACK (*LPPROGRESS_ROUTINE) (LARGE_INTEGER, LARGE_INTEGER, LARGE_INTEGER,
00712                                               LARGE_INTEGER, DWORD, DWORD, HANDLE,
00713                                               HANDLE, LPVOID);
00714
00715
00716 #define WAIT_FAILED 0xffffffff
00717 #define WAIT_OBJECT_0 0
00718 #define WAIT_ABANDONED STATUS_ABANDONED_WAIT_0
00719 #define WAIT_ABANDONED_0 STATUS_ABANDONED_WAIT_0
00720 #define WAIT_IO_COMPLETION STATUS_USER_APC
00721 #define WAIT_TIMEOUT STATUS_TIMEOUT
00722 #define STILL_ACTIVE STATUS_PENDING
00723
00724 #define FILE_BEGIN 0
00725 #define FILE_CURRENT 1
00726 #define FILE_END 2
00727
00728 #define FILE_MAP_COPY 0x00000001
00729 #define FILE_MAP_WRITE 0x00000002
00730 #define FILE_MAP_READ 0x00000004
00731 #define FILE_MAP_ALL_ACCESS 0x000f001f
00732
00733 #define MOVEFILE_REPLACE_EXISTING 0x00000001
00734 #define MOVEFILE_COPY_ALLOWED 0x00000002
00735 #define MOVEFILE_DELAY_UNTIL_REBOOT 0x00000004
00736
00737 #define FS_CASE_SENSITIVE FILE_CASE_SENSITIVE_SEARCH

```

```

00738 #define FS_CASE_IS_PRESERVED          FILE_CASE_PRESERVED_NAMES
00739 #define FS_UNICODE_STORED_ON_DISK      FILE_UNICODE_ON_DISK
00740 #define FS_PERSISTENT_ACLS            FILE_PERSISTENT_ACLS
00741 #define FS_VOL_IS_COMPRESSED          FILE_VOLUME_IS_COMPRESSED
00742 #define FS_FILE_COMPRESSION           FILE_FILE_COMPRESSION
00743
00744 #define EXCEPTION_ACCESS_VIOLATION     STATUS_ACCESS_VIOLATION
00745 #define EXCEPTION_DATATYPE_MISALIGNMENT STATUS_DATATYPE_MISALIGNMENT
00746 #define EXCEPTION_BREAKPOINT           STATUS_BREAKPOINT
00747 #define EXCEPTION_SINGLE_STEP          STATUS_SINGLE_STEP
00748 #define EXCEPTION_ARRAY_BOUNDS_EXCEEDED STATUS_ARRAY_BOUNDS_EXCEEDED
00749 #define EXCEPTION_FLT_DENORMAL_OPERAND STATUS_FLOAT_DENORMAL_OPERAND
00750 #define EXCEPTION_FLT_DIVIDE_BY_ZERO   STATUS_FLOAT_DIVIDE_BY_ZERO
00751 #define EXCEPTION_FLT_INEXACT_RESULT   STATUS_FLOAT_INEXACT_RESULT
00752 #define EXCEPTION_FLT_INVALID_OPERATION STATUS_FLOAT_INVALID_OPERATION
00753 #define EXCEPTION_FLT_OVERFLOW         STATUS_FLOAT_OVERFLOW
00754 #define EXCEPTION_FLT_STACK_CHECK      STATUS_FLOAT_STACK_CHECK
00755 #define EXCEPTION_FLT_UNDERFLOW        STATUS_FLOAT_UNDERFLOW
00756 #define EXCEPTION_INT_DIVIDE_BY_ZERO   STATUS_INTEGER_DIVIDE_BY_ZERO
00757 #define EXCEPTION_INT_OVERFLOW         STATUS_INTEGER_OVERFLOW
00758 #define EXCEPTION_PRIV_INSTRUCTION     STATUS_PRIVILEGED_INSTRUCTION
00759 #define EXCEPTION_IN_PAGE_ERROR        STATUS_IN_PAGE_ERROR
00760 #define EXCEPTION_ILLEGAL_INSTRUCTION  STATUS_ILLEGAL_INSTRUCTION
00761 #define EXCEPTION_NONCONTINUABLE_EXCEPTION STATUS_NONCONTINUABLE_EXCEPTION
00762 #define EXCEPTION_STACK_OVERFLOW        STATUS_STACK_OVERFLOW
00763 #define EXCEPTION_INVALID_DISPOSITION  STATUS_INVALID_DISPOSITION
00764 #define EXCEPTION_GUARD_PAGE           STATUS_GUARD_PAGE_VIOLATION
00765 #define EXCEPTION_INVALID_HANDLE       STATUS_INVALID_HANDLE
00766 #define CONTROL_C_EXIT                 STATUS_CONTROL_C_EXIT
00767
00768 /* Wine extension; Windows doesn't have a name for this code */
00769 #define EXCEPTION_CRITICAL_SECTION_WAIT 0xc0000194
00770
00771 #define DUPLICATE_CLOSE_SOURCE          0x00000001
00772 #define DUPLICATE_SAME_ACCESS           0x00000002
00773
00774 #define HANDLE_FLAG_INHERIT             0x00000001
00775 #define HANDLE_FLAG_PROTECT_FROM_CLOSE 0x00000002
00776
00777 #define HINSTANCE_ERROR 32
00778
00779 #define THREAD_PRIORITY_LOWEST          THREAD_BASE_PRIORITY_MIN
00780 #define THREAD_PRIORITY_BELOW_NORMAL   (THREAD_PRIORITY_LOWEST+1)
00781 #define THREAD_PRIORITY_NORMAL         0
00782 #define THREAD_PRIORITY_HIGHEST        THREAD_BASE_PRIORITY_MAX
00783 #define THREAD_PRIORITY_ABOVE_NORMAL   (THREAD_PRIORITY_HIGHEST-1)
00784 #define THREAD_PRIORITY_ERROR_RETURN    (0x7fffffff)
00785 #define THREAD_PRIORITY_TIME_CRITICAL  THREAD_BASE_PRIORITY_LOWR
00786 #define THREAD_PRIORITY_IDLE            THREAD_BASE_PRIORITY_IDLE
00787
00788 /* flags to FormatMessage */
00789 #define FORMAT_MESSAGE_ALLOCATE_BUFFER 0x00000100
00790 #define FORMAT_MESSAGE_IGNORE_INSERTS 0x00000200
00791 #define FORMAT_MESSAGE_FROM_STRING     0x00000400
00792 #define FORMAT_MESSAGE_FROM_HMODULE    0x00000800
00793 #define FORMAT_MESSAGE_FROM_SYSTEM     0x00001000
00794 #define FORMAT_MESSAGE_ARGUMENT_ARRAY 0x00002000
00795 #define FORMAT_MESSAGE_MAX_WIDTH_MASK 0x000000ff
00796
00797 #ifdef __WINE__
00798 #define CRITICAL_SECTION_INIT(name) { (void *) (__FILE__ ": " name), -1, 0, 0, 0, 0 }
00799 #endif
00800
00801 typedef struct {
00802     DWORD dwOSVersionInfoSize;
00803     DWORD dwMajorVersion;
00804     DWORD dwMinorVersion;
00805     DWORD dwBuildNumber;
00806     DWORD dwPlatformId;
00807     CHAR szCSDVersion[128];
00808 } OSVERSIONINFOA, *POSVERSIONINFOA, *LPOSVERSIONINFOA;
00809
00810 typedef struct {
00811     DWORD dwOSVersionInfoSize;
00812     DWORD dwMajorVersion;
00813     DWORD dwMinorVersion;
00814     DWORD dwBuildNumber;
00815     DWORD dwPlatformId;
00816     WCHAR szCSDVersion[128];
00817 } OSVERSIONINFOW, *POSVERSIONINFOW, *LPOSVERSIONINFOW;
00818
00819 DECL_WINELIB_TYPE_AW(OSVERSIONINFO)
00820 DECL_WINELIB_TYPE_AW(POSVERSIONINFO)
00821 DECL_WINELIB_TYPE_AW(LPOSVERSIONINFO)
00822
00823 #define VER_PLATFORM_WIN32s 0
00824 #define VER_PLATFORM_WIN32_WINDOWS 1

```

```

00825 #define VER_PLATFORM_WIN32_NT                2
00826
00827 typedef struct tagCOMSTAT
00828 {
00829     DWORD status;
00830     DWORD cbInQue;
00831     DWORD cbOutQue;
00832 } COMSTAT, *LPCOMSTAT;
00833
00834 typedef struct tagDCB
00835 {
00836     DWORD DCBlength;
00837     DWORD BaudRate;
00838     unsigned fBinary                :1;
00839     unsigned fParity                :1;
00840     unsigned fOutxCtsFlow          :1;
00841     unsigned fOutxDsrFlow          :1;
00842     unsigned fDtrControl           :2;
00843     unsigned fDsrSensitivity       :1;
00844     unsigned fTXContinueOnXoff     :1;
00845     unsigned fOutX                 :1;
00846     unsigned fInX                  :1;
00847     unsigned fErrorChar            :1;
00848     unsigned fNull                 :1;
00849     unsigned fRtsControl           :2;
00850     unsigned fAbortOnError         :1;
00851     unsigned fDummy2               :17;
00852     WORD wReserved;
00853     WORD XonLim;
00854     WORD XoffLim;
00855     BYTE ByteSize;
00856     BYTE Parity;
00857     BYTE StopBits;
00858     char XonChar;
00859     char XoffChar;
00860     char ErrorChar;
00861     char EofChar;
00862     char EvtChar;
00863 } DCB, *LPDCB;
00864
00865 typedef struct tagCOMMCONFIG {
00866     DWORD dwSize;
00867     WORD wVersion;
00868     WORD wReserved;
00869     DCB dcb;
00870     DWORD dwProviderSubType;
00871     DWORD dwProviderOffset;
00872     DWORD dwProviderSize;
00873     DWORD wcProviderData[1];
00874 } COMMCONFIG, *LPCOMMCONFIG;
00875
00876 typedef struct tagCOMMPROP {
00877     WORD wPacketLength;
00878     WORD wPacketVersion;
00879     DWORD dwServiceMask;
00880     DWORD dwReserved1;
00881     DWORD dwMaxTxQueue;
00882     DWORD dwMaxRxQueue;
00883     DWORD dwMaxBaud;
00884     DWORD dwProvSubType;
00885     DWORD dwProvCapabilities;
00886     DWORD dwSettableParams;
00887     DWORD dwSettableBaud;
00888     WORD wSettableData;
00889     WORD wSettableStopParity;
00890     DWORD dwCurrentTxQueue;
00891     DWORD dwCurrentRxQueue;
00892     DWORD dwProvSpec1;
00893     DWORD dwProvSpec2;
00894     WCHAR wcProvChar[1];
00895 } COMMPROP, *LPCOMMPROP;
00896
00897 #define SP_SERIALCOMM ((DWORD)1)
00898
00899 #define BAUD_075      ((DWORD)0x01)
00900 #define BAUD_110      ((DWORD)0x02)
00901 #define BAUD_134_5    ((DWORD)0x04)
00902 #define BAUD_150      ((DWORD)0x08)
00903 #define BAUD_300      ((DWORD)0x10)
00904 #define BAUD_600      ((DWORD)0x20)
00905 #define BAUD_1200     ((DWORD)0x40)
00906 #define BAUD_1800     ((DWORD)0x80)
00907 #define BAUD_2400     ((DWORD)0x100)
00908 #define BAUD_4800     ((DWORD)0x200)
00909 #define BAUD_7200     ((DWORD)0x400)
00910 #define BAUD_9600     ((DWORD)0x800)
00911 #define BAUD_14400    ((DWORD)0x1000)

```

```

00912 #define BAUD_19200    ((DWORD) 0x2000)
00913 #define BAUD_38400    ((DWORD) 0x4000)
00914 #define BAUD_56K      ((DWORD) 0x8000)
00915 #define BAUD_57600    ((DWORD) 0x40000)
00916 #define BAUD_115200   ((DWORD) 0x200000)
00917 #define BAUD_128K     ((DWORD) 0x10000)
00918 #define BAUD_USER      ((DWORD) 0x10000000)
00919
00920 #define PST_FAX         ((DWORD) 0x21)
00921 #define PST_LAT         ((DWORD) 0x101)
00922 #define PST_MODEM      ((DWORD) 0x06)
00923 #define PST_NETWORK_BRIDGE ((DWORD) 0x100)
00924 #define PST_PARALLEL_PORT ((DWORD) 0x02)
00925 #define PST_RS232      ((DWORD) 0x01)
00926 #define PST_RS442      ((DWORD) 0x03)
00927 #define PST_RS423      ((DWORD) 0x04)
00928 #define PST_RS449      ((DWORD) 0x06)
00929 #define PST_SCANNER    ((DWORD) 0x22)
00930 #define PST_TCPIP_TELNET ((DWORD) 0x102)
00931 #define PST_UNSPECIFIED ((DWORD) 0x00)
00932 #define PST_X25        ((DWORD) 0x103)
00933
00934 #define PCF_16BITMODE   ((DWORD) 0x200)
00935 #define PCF_DTRDSR     ((DWORD) 0x01)
00936 #define PCF_INTTIMEOUTS ((DWORD) 0x80)
00937 #define PCF_PARITY_CHECK ((DWORD) 0x08)
00938 #define PCF_RLSD       ((DWORD) 0x04)
00939 #define PCF_RTSCCTS     ((DWORD) 0x02)
00940 #define PCF_SETXCHAR    ((DWORD) 0x20)
00941 #define PCF_SPECIALCHARS ((DWORD) 0x100)
00942 #define PCF_TOTALTIMEOUTS ((DWORD) 0x40)
00943 #define PCF_XONXOFF     ((DWORD) 0x10)
00944
00945 #define SP_BAUD         ((DWORD) 0x02)
00946 #define SP_DATABITS     ((DWORD) 0x04)
00947 #define SP_HANDSHAKING ((DWORD) 0x10)
00948 #define SP_PARITY       ((DWORD) 0x01)
00949 #define SP_PARITY_CHECK ((DWORD) 0x20)
00950 #define SP_RLSD         ((DWORD) 0x40)
00951 #define SP_STOPBITS     ((DWORD) 0x08)
00952
00953 #define DATABITS_5      ((DWORD) 0x01)
00954 #define DATABITS_6      ((DWORD) 0x02)
00955 #define DATABITS_7      ((DWORD) 0x04)
00956 #define DATABITS_8      ((DWORD) 0x08)
00957 #define DATABITS_16     ((DWORD) 0x10)
00958 #define DATABITS_16X    ((DWORD) 0x20)
00959
00960 #define STOPBITS_10     ((DWORD) 1)
00961 #define STOPBITS_15     ((DWORD) 2)
00962 #define STOPBITS_20     ((DWORD) 4)
00963
00964 #define PARITY_NONE      ((DWORD) 0x100)
00965 #define PARITY_ODD       ((DWORD) 0x200)
00966 #define PARITY_EVEN      ((DWORD) 0x400)
00967 #define PARITY_MARK      ((DWORD) 0x800)
00968 #define PARITY_SPACE     ((DWORD) 0x1000)
00969
00970 typedef struct tagCOMMTIMEOUTS {
00971     DWORD    ReadIntervalTimeout;
00972     DWORD    ReadTotalTimeoutMultiplier;
00973     DWORD    ReadTotalTimeoutConstant;
00974     DWORD    WriteTotalTimeoutMultiplier;
00975     DWORD    WriteTotalTimeoutConstant;
00976 } COMMTIMEOUTS, *LPCOMMTIMEOUTS;
00977
00978 typedef void CALLBACK (*PAPCFUNC) (ULONG_PTR);
00979 typedef void CALLBACK (*PTIMERAPCRoutine) (LPVOID, DWORD, DWORD);
00980
00981 /*DWORD WINAPI GetVersion( void );*/
00982 BOOL WINAPI GetVersionExA(OSVERSIONINFOA*);
00983 BOOL WINAPI GetVersionExW(OSVERSIONINFOW*);
00984 #define GetVersionEx WINELIB_NAME_AW(GetVersionEx)
00985
00986 /*int WinMain(HINSTANCE, HINSTANCE prev, char *cmd, int show);*/
00987
00988 /* FIXME: need to use defines because we don't have proper imports everywhere yet */
00989 #ifndef have_proper_imports
00990 LONG    WINAPI RtlEnterCriticalSection( CRITICAL_SECTION *crit );
00991 LONG    WINAPI RtlLeaveCriticalSection( CRITICAL_SECTION *crit );
00992 LONG    WINAPI RtlDeleteCriticalSection( CRITICAL_SECTION *crit );
00993 BOOL    WINAPI RtlTryEnterCriticalSection( CRITICAL_SECTION *crit );
00994 PVOID   WINAPI RtlAllocateHeap( HANDLE, ULONG, ULONG );
00995 BOOLEAN WINAPI RtlFreeHeap( HANDLE, ULONG, PVOID );
00996 PVOID   WINAPI RtlReAllocateHeap( HANDLE, ULONG, PVOID, ULONG );
00997 ULONG   WINAPI RtlSizeHeap( HANDLE, ULONG, PVOID );
00998 #define HeapAlloc(heap, flags, size) RtlAllocateHeap(heap, flags, size)

```

```

00999 #define      HeapFree(heap, flags, ptr) RtlFreeHeap(heap, flags, ptr)
01000 #define      HeapReAlloc(heap, flags, ptr, size) RtlReAllocateHeap(heap, flags, ptr, size)
01001 #define      HeapSize(heap, flags, ptr) RtlSizeHeap(heap, flags, ptr)
01002 #define      EnterCriticalSection(crit) RtlEnterCriticalSection(crit)
01003 #define      LeaveCriticalSection(crit) RtlLeaveCriticalSection(crit)
01004 #define      DeleteCriticalSection(crit) RtlDeleteCriticalSection(crit)
01005 #define      TryEnterCriticalSection(crit) RtlTryEnterCriticalSection(crit)
01006 #else
01007 LPVOID      WINAPI HeapAlloc(HANDLE, DWORD, DWORD);
01008 BOOL        WINAPI HeapFree(HANDLE, DWORD, LPVOID);
01009 LPVOID      WINAPI HeapReAlloc(HANDLE, DWORD, LPVOID, DWORD);
01010 DWORD      WINAPI HeapSize(HANDLE, DWORD, LPVOID);
01011 void        WINAPI DeleteCriticalSection(CRITICAL_SECTION *lpCrit);
01012 void        WINAPI EnterCriticalSection(CRITICAL_SECTION *lpCrit);
01013 BOOL        WINAPI TryEnterCriticalSection(CRITICAL_SECTION *lpCrit);
01014 void        WINAPI LeaveCriticalSection(CRITICAL_SECTION *lpCrit);
01015 #endif
01016
01017 void        WINAPI InitializeCriticalSection(CRITICAL_SECTION *lpCrit);
01018 BOOL        WINAPI InitializeCriticalSectionAndSpinCount(CRITICAL_SECTION *, DWORD);
01019 void        WINAPI MakeCriticalSectionGlobal(CRITICAL_SECTION *lpCrit);
01020 BOOL        WINAPI GetProcessWorkingSetSize(HANDLE, LPDWORD, LPDWORD);
01021 DWORD      WINAPI QueueUserAPC(PAPCFUNC, HANDLE, ULONG_PTR);
01022 void        WINAPI RaiseException(DWORD, DWORD, DWORD, const LPDWORD);
01023 BOOL        WINAPI SetProcessWorkingSetSize(HANDLE, DWORD, DWORD);
01024 BOOL        WINAPI TerminateProcess(HANDLE, DWORD);
01025 BOOL        WINAPI TerminateThread(HANDLE, DWORD);
01026 BOOL        WINAPI GetExitCodeThread(HANDLE, LPDWORD);
01027
01028 /* GetBinaryType return values.
01029 */
01030
01031 #define SCS_32BIT_BINARY    0
01032 #define SCS_DOS_BINARY     1
01033 #define SCS_WOW_BINARY     2
01034 #define SCS_PIF_BINARY     3
01035 #define SCS_POSIX_BINARY   4
01036 #define SCS_OS216_BINARY   5
01037
01038 BOOL WINAPI GetBinaryTypeA(LPCSTR lpApplicationName, LPDWORD lpBinaryType);
01039 BOOL WINAPI GetBinaryTypeW(LPCWSTR lpApplicationName, LPDWORD lpBinaryType);
01040 #define GetBinaryType WINELIB_NAME_AW(GetBinaryType)
01041
01042 /* Declarations for functions that exist only in Win32 */
01043
01044 BOOL        WINAPI AddAccessAllowedAce(PACL, DWORD, DWORD, PSID);
01045 BOOL        WINAPI AttachThreadInput(DWORD, DWORD, BOOL);
01046 BOOL        WINAPI
    AccessCheck(PSECURITY_DESCRIPTOR, HANDLE, DWORD, PGENERIC_MAPPING, PPRIVILEGE_SET, LPDWORD, LPDWORD, LPBOOL);
01047 BOOL        WINAPI AdjustTokenPrivileges(HANDLE, BOOL, LPVOID, DWORD, LPVOID, LPDWORD);
01048 BOOL        WINAPI
    AllocateAndInitializeSid(PSID_IDENTIFIER_AUTHORITY, BYTE, DWORD, DWORD, DWORD, DWORD, DWORD, DWORD, DWORD, PSID
    *);
01049 BOOL        WINAPI AllocateLocallyUniqueId(PLUID);
01050 BOOL        WINAPI AreFileApisANSI(void);
01051 BOOL        WINAPI BackupEventLogA(HANDLE, LPCSTR);
01052 BOOL        WINAPI BackupEventLogW(HANDLE, LPCWSTR);
01053 #define      BackupEventLog WINELIB_NAME_AW(BackupEventLog)
01054 BOOL        WINAPI BackupRead(HANDLE, LPBYTE, DWORD, LPDWORD, BOOL, BOOL, LPVOID*);
01055 BOOL        WINAPI BackupSeek(HANDLE, DWORD, DWORD, LPDWORD, LPDWORD, LPVOID*);
01056 BOOL        WINAPI BackupWrite(HANDLE, LPBYTE, DWORD, LPDWORD, BOOL, BOOL, LPVOID*);
01057 BOOL        WINAPI Beep(DWORD, DWORD);
01058 BOOL        WINAPI BuildCommDCBA(LPCSTR, LPDCB);
01059 BOOL        WINAPI BuildCommDCBW(LPCWSTR, LPDCB);
01060 #define      BuildCommDCB WINELIB_NAME_AW(BuildCommDCB)
01061 BOOL        WINAPI BuildCommDCBAndTimeoutsA(LPCSTR, LPDCB, LPCOMMTIMEOUTS);
01062 BOOL        WINAPI BuildCommDCBAndTimeoutsW(LPCWSTR, LPDCB, LPCOMMTIMEOUTS);
01063 #define      BuildCommDCBAndTimeouts WINELIB_NAME_AW(BuildCommDCBAndTimeouts)
01064 BOOL        WINAPI CancelIo(HANDLE);
01065 BOOL        WINAPI CancelWaitableTimer(HANDLE);
01066 BOOL        WINAPI ClearCommBreak(HANDLE);
01067 BOOL        WINAPI ClearCommError(HANDLE, LPDWORD, LPCOMSTAT);
01068 BOOL        WINAPI ClearEventLogA(HANDLE, LPCSTR);
01069 BOOL        WINAPI ClearEventLogW(HANDLE, LPCWSTR);
01070 #define      ClearEventLog WINELIB_NAME_AW(ClearEventLog)
01071 BOOL        WINAPI CloseEventLog(HANDLE);
01072 BOOL        WINAPI CloseHandle(HANDLE);
01073 BOOL        WINAPI CommConfigDialogA(LPCSTR, HANDLE, LPCOMMCONFIG);
01074 BOOL        WINAPI CommConfigDialogW(LPCWSTR, HANDLE, LPCOMMCONFIG);
01075 #define      CommConfigDialog WINELIB_NAME_AW(CommConfigDialog)
01076 BOOL        WINAPI ConnectNamedPipe(HANDLE, LPOVERLAPPED);
01077 BOOL        WINAPI ContinueDebugEvent(DWORD, DWORD, DWORD);
01078 HANDLE      WINAPI ConvertToGlobalHandle(HANDLE hSrc);
01079 BOOL        WINAPI CopyFileA(LPCSTR, LPCSTR, BOOL);
01080 BOOL        WINAPI CopyFileW(LPCWSTR, LPCWSTR, BOOL);
01081 #define      CopyFile WINELIB_NAME_AW(CopyFile)
01082 BOOL        WINAPI CopyFileExA(LPCSTR, LPCSTR, LPPROGRESS_ROUTINE, LPVOID, LPBOOL, DWORD);

```

```

01083 BOOL        WINAPI CopyFileExW(LPCWSTR, LPCWSTR, LPPROGRESS_ROUTINE, LPVOID, LPBOOL, DWORD);
01084 #define        CopyFileEx WINELIB_NAME_AW(CopyFileEx)
01085 BOOL        WINAPI CopySid(DWORD, PSID, PSID);
01086 INT         WINAPI CompareFileTime(LPFILETIME, LPFILETIME);
01087 HANDLE      WINAPI CreateEventA(LPSECURITY_ATTRIBUTES, BOOL, BOOL, LPCSTR);
01088 HANDLE      WINAPI CreateEventW(LPSECURITY_ATTRIBUTES, BOOL, BOOL, LPCWSTR);
01089 #define        CreateEvent WINELIB_NAME_AW(CreateEvent)
01090 HANDLE      WINAPI CreateFileA(LPCSTR, DWORD, DWORD, LPSECURITY_ATTRIBUTES,
01091                                DWORD, DWORD, HANDLE);
01092 HANDLE      WINAPI CreateFileW(LPCWSTR, DWORD, DWORD, LPSECURITY_ATTRIBUTES,
01093                                DWORD, DWORD, HANDLE);
01094 #define        CreateFile WINELIB_NAME_AW(CreateFile)
01095 HANDLE      WINAPI CreateFileMappingA(HANDLE, LPSECURITY_ATTRIBUTES, DWORD,
01096                                       DWORD, DWORD, LPCSTR);
01097 HANDLE      WINAPI CreateFileMappingW(HANDLE, LPSECURITY_ATTRIBUTES, DWORD,
01098                                       DWORD, DWORD, LPCWSTR);
01099 #define        CreateFileMapping WINELIB_NAME_AW(CreateFileMapping)
01100 HANDLE      WINAPI CreateMutexA(LPSECURITY_ATTRIBUTES, BOOL, LPCSTR);
01101 HANDLE      WINAPI CreateMutexW(LPSECURITY_ATTRIBUTES, BOOL, LPCWSTR);
01102 #define        CreateMutex WINELIB_NAME_AW(CreateMutex)
01103 HANDLE      WINAPI CreateNamedPipeA(LPCSTR, DWORD, DWORD, DWORD, DWORD, LPSECURITY_ATTRIBUTES);
01104 HANDLE      WINAPI
01105             CreateNamedPipeW(LPCWSTR, DWORD, DWORD, DWORD, DWORD, LPSECURITY_ATTRIBUTES);
01105 #define        CreateNamedPipe WINELIB_NAME_AW(CreateNamedPipe)
01106 BOOL        WINAPI CreatePipe(PHANDLE, PHANDLE, LPSECURITY_ATTRIBUTES, DWORD);
01107 BOOL        WINAPI CreateProcessA(LPCSTR, LPSTR, LPSECURITY_ATTRIBUTES,
01108                                    LPSECURITY_ATTRIBUTES, BOOL, DWORD, LPVOID, LPCSTR,
01109                                    LPSTARTUPINFOA, LPPROCESS_INFORMATION);
01110 BOOL        WINAPI CreateProcessW(LPCWSTR, LPWSTR, LPSECURITY_ATTRIBUTES,
01111                                    LPSECURITY_ATTRIBUTES, BOOL, DWORD, LPVOID, LPCWSTR,
01112                                    LPSTARTUPINFOW, LPPROCESS_INFORMATION);
01113 #define        CreateProcess WINELIB_NAME_AW(CreateProcess)
01114 HANDLE      WINAPI
01115             CreateRemoteThread(HANDLE, LPSECURITY_ATTRIBUTES, DWORD, LPTHREAD_START_ROUTINE, LPVOID, DWORD, LPDWORD);
01115 HANDLE      WINAPI CreateSemaphoreA(LPSECURITY_ATTRIBUTES, LONG, LONG, LPCSTR);
01116 HANDLE      WINAPI CreateSemaphoreW(LPSECURITY_ATTRIBUTES, LONG, LONG, LPCWSTR);
01117 #define        CreateSemaphore WINELIB_NAME_AW(CreateSemaphore)
01118 DWORD       WINAPI CreateTapePartition(HANDLE, DWORD, DWORD, DWORD);
01119 HANDLE      WINAPI
01120             CreateThread(LPSECURITY_ATTRIBUTES, DWORD, LPTHREAD_START_ROUTINE, LPVOID, DWORD, LPDWORD);
01120 HANDLE      WINAPI CreateWaitableTimerA(LPSECURITY_ATTRIBUTES, BOOL, LPCSTR);
01121 HANDLE      WINAPI CreateWaitableTimerW(LPSECURITY_ATTRIBUTES, BOOL, LPCWSTR);
01122 #define        CreateWaitableTimer WINELIB_NAME_AW(CreateWaitableTimer)
01123 BOOL        WINAPI DebugActiveProcess(DWORD);
01124 void        WINAPI DebugBreak(void);
01125 BOOL        WINAPI DeregisterEventSource(HANDLE);
01126 BOOL        WINAPI DeviceIoControl(HANDLE, DWORD, LPVOID, DWORD, LPVOID, LPDWORD, LPOVERLAPPED);
01127 BOOL        WINAPI DisableThreadLibraryCalls(HMODULE);
01128 BOOL        WINAPI DosDateTimeToFileTime(WORD, WORD, LPFILETIME);
01129 BOOL        WINAPI DuplicateHandle(HANDLE, HANDLE, HANDLE, HANDLE*, DWORD, BOOL, DWORD);
01130 BOOL        WINAPI EscapeCommFunction(HANDLE, UINT);
01131 BOOL        WINAPI EnumResourceLanguagesA(HMODULE, LPCSTR, LPCSTR,
01132                                            ENUMRESLANGPROCA, LONG);
01133 BOOL        WINAPI EnumResourceLanguagesW(HMODULE, LPCWSTR, LPCWSTR,
01134                                            ENUMRESLANGPROCW, LONG);
01135 #define        EnumResourceLanguages WINELIB_NAME_AW(EnumResourceLanguages)
01136 BOOL        WINAPI EnumResourceNamesA(HMODULE, LPCSTR, ENUMRESNAMEPROCA,
01137                                       LONG);
01138 BOOL        WINAPI EnumResourceNamesW(HMODULE, LPCWSTR, ENUMRESNAMEPROCW,
01139                                       LONG);
01140 #define        EnumResourceNames WINELIB_NAME_AW(EnumResourceNames)
01141 BOOL        WINAPI EnumResourceTypesA(HMODULE, ENUMRESTYPEPROCA, LONG);
01142 BOOL        WINAPI EnumResourceTypesW(HMODULE, ENUMRESTYPEPROCW, LONG);
01143 #define        EnumResourceTypes WINELIB_NAME_AW(EnumResourceTypes)
01144 BOOL        WINAPI EqualSid(PSID, PSID);
01145 BOOL        WINAPI EqualPrefixSid(PSID, PSID);
01146 DWORD       WINAPI EraseTape(HANDLE, DWORD, BOOL);
01147 VOID        WINAPI ExitProcess(DWORD) WINE_NORETURN;
01148 VOID        WINAPI ExitThread(DWORD) WINE_NORETURN;
01149 DWORD       WINAPI ExpandEnvironmentStringsA(LPCSTR, LPSTR, DWORD);
01150 DWORD       WINAPI ExpandEnvironmentStringsW(LPCWSTR, LPWSTR, DWORD);
01151 #define        ExpandEnvironmentStrings WINELIB_NAME_AW(ExpandEnvironmentStrings)
01152 BOOL        WINAPI FileTimeToDosDateTime(const FILETIME*, LPWORD, LPWORD);
01153 BOOL        WINAPI FileTimeToLocalFileTime(const FILETIME*, LPFILETIME);
01154 BOOL        WINAPI FileTimeToSystemTime(const FILETIME*, LPSYSTEMTIME);
01155 HANDLE      WINAPI FindFirstChangeNotificationA(LPCSTR, BOOL, DWORD);
01156 HANDLE      WINAPI FindFirstChangeNotificationW(LPCWSTR, BOOL, DWORD);
01157 #define        FindFirstChangeNotification WINELIB_NAME_AW(FindFirstChangeNotification)
01158 BOOL        WINAPI FindNextChangeNotification(HANDLE);
01159 BOOL        WINAPI FindCloseChangeNotification(HANDLE);
01160 HRSRC       WINAPI FindResourceExA(HMODULE, LPCSTR, LPCSTR, WORD);
01161 HRSRC       WINAPI FindResourceExW(HMODULE, LPCWSTR, LPCWSTR, WORD);
01162 #define        FindResourceEx WINELIB_NAME_AW(FindResourceEx)
01163 BOOL        WINAPI FlushFileBuffers(HANDLE);
01164 BOOL        WINAPI FlushViewOfFile(LPCVOID, DWORD);
01165 DWORD       WINAPI FormatMessageA(DWORD, LPCVOID, DWORD, DWORD, LPSTR, DWORD, va_list*);
01166 DWORD       WINAPI FormatMessageW(DWORD, LPCVOID, DWORD, DWORD, LPWSTR, DWORD, va_list*);

```

```
01167 #define      FormatMessage WINELIB_NAME_AW(FormatMessage)
01168 BOOL          WINAPI FreeEnvironmentStringsA(LPSTR);
01169 BOOL          WINAPI FreeEnvironmentStringsW(LPWSTR);
01170 #define      FreeEnvironmentStrings WINELIB_NAME_AW(FreeEnvironmentStrings)
01171 VOID         WINAPI FreeLibraryAndExitThread(HINSTANCE, DWORD);
01172 PVOID        WINAPI FreeSid(PSID);
01173 BOOL         WINAPI GetCommConfig(HANDLE, LPCOMMCONFIG, LPDWORD);
01174 BOOL         WINAPI GetCommMask(HANDLE, LPDWORD);
01175 BOOL         WINAPI GetCommModemStatus(HANDLE, LPDWORD);
01176 BOOL         WINAPI GetCommProperties(HANDLE, LPCOMMPROP);
01177 BOOL         WINAPI GetCommState(HANDLE, LPDCB);
01178 BOOL         WINAPI GetCommTimeouts(HANDLE, LPCOMMTIMEOUTS);
01179 LPSTR        WINAPI GetCommandLineA(void);
01180 LPWSTR       WINAPI GetCommandLineW(void);
01181 #define      GetCommandLine WINELIB_NAME_AW(GetCommandLine)
01182 BOOL         WINAPI GetComputerNameA(LPSTR, LPDWORD);
01183 BOOL         WINAPI GetComputerNameW(LPWSTR, LPDWORD);
01184 #define      GetComputerName WINELIB_NAME_AW(GetComputerName)
01185 HANDLE       WINAPI GetCurrentProcess(void);
01186 HANDLE       WINAPI GetCurrentThread(void);
01187 BOOL         WINAPI GetDefaultCommConfigA(LPCSTR, LPCOMMCONFIG, LPDWORD);
01188 BOOL         WINAPI GetDefaultCommConfigW(LPCWSTR, LPCOMMCONFIG, LPDWORD);
01189 #define      GetDefaultCommConfig WINELIB_NAME_AW(GetDefaultCommConfig)
01190 LPSTR        WINAPI GetEnvironmentStringsA(void);
01191 LPWSTR       WINAPI GetEnvironmentStringsW(void);
01192 #define      GetEnvironmentStrings WINELIB_NAME_AW(GetEnvironmentStrings)
01193 DWORD        WINAPI GetEnvironmentVariableA(LPCSTR, LPSTR, DWORD);
01194 DWORD        WINAPI GetEnvironmentVariableW(LPCWSTR, LPWSTR, DWORD);
01195 #define      GetEnvironmentVariable WINELIB_NAME_AW(GetEnvironmentVariable)
01196 BOOL         WINAPI GetFileAttributesExA(LPCSTR, GET_FILEEX_INFO_LEVELS, LPVOID);
01197 BOOL         WINAPI GetFileAttributesExW(LPCWSTR, GET_FILEEX_INFO_LEVELS, LPVOID);
01198 #define      GetFileAttributesEx WINELIB_NAME_AW(GetFileAttributesEx)
01199 DWORD        WINAPI GetFileInformationByHandle(HANDLE, BY_HANDLE_FILE_INFORMATION*);
01200 BOOL         WINAPI GetFileSecurityA(LPCSTR, SECURITY_INFORMATION, PSECURITY_DESCRIPTOR, DWORD, LPDWORD);
01201 BOOL         WINAPI GetFileSecurityW(LPCWSTR, SECURITY_INFORMATION, PSECURITY_DESCRIPTOR, DWORD, LPDWORD);
01202 #define      GetFileSecurity WINELIB_NAME_AW(GetFileSecurity)
01203 DWORD        WINAPI GetFileSize(HANDLE, LPDWORD);
01204 BOOL         WINAPI GetFileTime(HANDLE, LPFILETIME, LPFILETIME, LPFILETIME);
01205 DWORD        WINAPI GetFileType(HANDLE);
01206 DWORD        WINAPI GetFullPathNameA(LPCSTR, DWORD, LPSTR, LPSTR*);
01207 DWORD        WINAPI GetFullPathNameW(LPCWSTR, DWORD, LPWSTR, LPWSTR*);
01208 #define      GetFullPathName WINELIB_NAME_AW(GetFullPathName)
01209 BOOL         WINAPI GetHandleInformation(HANDLE, LPDWORD);
01210 DWORD        WINAPI GetLengthSid(PSID);
01211 VOID         WINAPI GetLocalTime(LPSYSTEMTIME);
01212 DWORD        WINAPI GetLogicalDrives(void);
01213 DWORD        WINAPI GetLongPathNameA(LPCSTR, LPSTR, DWORD);
01214 DWORD        WINAPI GetLongPathNameW(LPCWSTR, LPWSTR, DWORD);
01215 #define      GetLongPathName WINELIB_NAME_AW(GetLongPathName)
01216 BOOL         WINAPI GetNumberOfEventLogRecords(HANDLE, PDWORD);
01217 BOOL         WINAPI GetOldestEventLogRecord(HANDLE, PDWORD);
01218 DWORD        WINAPI GetPriorityClass(HANDLE);
01219 BOOL         WINAPI GetProcessTimes(HANDLE, LPFILETIME, LPFILETIME, LPFILETIME, LPFILETIME);
01220 DWORD        WINAPI GetProcessVersion(DWORD);
01221 BOOL         WINAPI
GetSecurityDescriptorControl(PSECURITY_DESCRIPTOR, PSECURITY_DESCRIPTOR_CONTROL, LPDWORD);
01222 BOOL         WINAPI GetSecurityDescriptorDacl(PSECURITY_DESCRIPTOR, LPBOOL, PACL *, LPBOOL);
01223 BOOL         WINAPI GetSecurityDescriptorGroup(PSECURITY_DESCRIPTOR, PSID *, LPBOOL);
01224 DWORD        WINAPI GetSecurityDescriptorLength(PSECURITY_DESCRIPTOR);
01225 BOOL         WINAPI GetSecurityDescriptorOwner(PSECURITY_DESCRIPTOR, PSID *, LPBOOL);
01226 BOOL         WINAPI GetSecurityDescriptorSacl(PSECURITY_DESCRIPTOR, LPBOOL, PACL *, LPBOOL);
01227 PSID_IDENTIFIER_AUTHORITY WINAPI GetSidIdentifierAuthority(PSID);
01228 DWORD        WINAPI GetSidLengthRequired(BYTE);
01229 PDWORD       WINAPI GetSidSubAuthority(PSID, DWORD);
01230 PULONG       WINAPI GetSidSubAuthorityCount(PSID);
01231 DWORD        WINAPI GetShortPathNameA(LPCSTR, LPSTR, DWORD);
01232 DWORD        WINAPI GetShortPathNameW(LPCWSTR, LPWSTR, DWORD);
01233 #define      GetShortPathName WINELIB_NAME_AW(GetShortPathName)
01234 HANDLE       WINAPI GetStdHandle(DWORD);
01235 VOID         WINAPI GetSystemInfo(LPSYSTEM_INFO);
01236 BOOL         WINAPI GetSystemPowerStatus(LPSYSTEM_POWER_STATUS);
01237 VOID         WINAPI GetSystemTime(LPSYSTEMTIME);
01238 VOID         WINAPI GetSystemTimeAsFileTime(LPFILETIME);
01239 DWORD        WINAPI GetTapeParameters(HANDLE, DWORD, LPDWORD, LPVOID);
01240 DWORD        WINAPI GetTapePosition(HANDLE, DWORD, LPDWORD, LPDWORD, LPDWORD);
01241 DWORD        WINAPI GetTapeStatus(HANDLE);
01242 DWORD        WINAPI GetTimeZoneInformation(LPTIME_ZONE_INFORMATION);
01243 BOOL         WINAPI GetThreadContext(HANDLE, CONTEXT *);
01244 INT          WINAPI GetThreadPriority(HANDLE);
01245 BOOL         WINAPI GetThreadPriorityBoost(HANDLE, PBOOL);
01246 BOOL         WINAPI GetThreadSelectorEntry(HANDLE, DWORD, LPLDT_ENTRY);
01247 BOOL         WINAPI GetThreadTimes(HANDLE, LPFILETIME, LPFILETIME, LPFILETIME, LPFILETIME);
01248 BOOL         WINAPI GetTokenInformation(HANDLE, TOKEN_INFORMATION_CLASS, LPVOID, DWORD, LPDWORD);
01249 BOOL         WINAPI GetUserNameA(LPSTR, LPDWORD);
01250 BOOL         WINAPI GetUserNameW(LPWSTR, LPDWORD);
01251 #define      GetUserName WINELIB_NAME_AW(GetUserName)
01252 VOID         WINAPI GlobalMemoryStatus(LPMEMORYSTATUS);
```

```

01253 DWORD      WINAPI HeapCompact (HANDLE,DWORD);
01254 HANDLE      WINAPI HeapCreate (DWORD,DWORD,DWORD);
01255 BOOL        WINAPI HeapDestroy (HANDLE);
01256 BOOL        WINAPI HeapLock (HANDLE);
01257 BOOL        WINAPI HeapUnlock (HANDLE);
01258 BOOL        WINAPI HeapValidate (HANDLE,DWORD,LPCVOID);
01259 BOOL        WINAPI HeapWalk (HANDLE,LPPROCESS_HEAP_ENTRY);
01260 DWORD      WINAPI InitializeAcl (PACL,DWORD,DWORD);
01261 BOOL        WINAPI InitializeSecurityDescriptor (PSECURITY_DESCRIPTOR,DWORD);
01262 BOOL        WINAPI InitializeSid (PSID,PSID_IDENTIFIER_AUTHORITY,BYTE);
01263 BOOL        WINAPI IsTextUnicode (CONST LPVOID lpBuffer, int cb, LPINT lpi);
01264 BOOL        WINAPI IsValidSecurityDescriptor (PSECURITY_DESCRIPTOR);
01265 BOOL        WINAPI IsValidSid (PSID);
01266 BOOL        WINAPI ImpersonateSelf (SECURITY_IMPERSONATION_LEVEL);
01267 BOOL        WINAPI IsProcessorFeaturePresent (DWORD);
01268 BOOL        WINAPI LookupAccountSidA (LPCSTR,PSID,LPSTR,LPDWORD,LPSTR,LPDWORD,PSID_NAME_USE);
01269 BOOL        WINAPI LookupAccountSidW (LPCWSTR,PSID,LPWSTR,LPDWORD,LPWSTR,LPDWORD,PSID_NAME_USE);
01270 #define      LookupAccountSid WINELIB_NAME_AW(LookupAccountSid)
01271 BOOL        WINAPI LocalFileTimeToFileTime (const FILETIME*,LPFILETIME);
01272 BOOL        WINAPI LockFile (HANDLE,DWORD,DWORD,DWORD,DWORD);
01273 BOOL        WINAPI LockFileEx (HANDLE, DWORD, DWORD, DWORD, DWORD, LPOVERLAPPED);
01274 BOOL        WINAPI LookupPrivilegeValueA (LPCSTR,LPCSTR,LPVOID);
01275 BOOL        WINAPI LookupPrivilegeValueW (LPCWSTR,LPCWSTR,LPVOID);
01276 #define      LookupPrivilegeValue WINELIB_NAME_AW(LookupPrivilegeValue)
01277 BOOL        WINAPI MakeSelfRelativeSD (PSECURITY_DESCRIPTOR,PSECURITY_DESCRIPTOR,LPDWORD);
01278 HMODULE      WINAPI MapHModuleSL (WORD);
01279 WORD        WINAPI MapHModuleLS (HMODULE);
01280 LPVOID       WINAPI MapViewOfFile (HANDLE,DWORD,DWORD,DWORD,DWORD);
01281 LPVOID       WINAPI MapViewOfFileEx (HANDLE,DWORD,DWORD,DWORD,DWORD,LPVOID);
01282 BOOL        WINAPI MoveFileA (LPCSTR,LPCSTR);
01283 BOOL        WINAPI MoveFileW (LPCWSTR,LPCWSTR);
01284 #define      MoveFile WINELIB_NAME_AW(MoveFile)
01285 BOOL        WINAPI MoveFileExA (LPCSTR,LPCSTR,DWORD);
01286 BOOL        WINAPI MoveFileExW (LPCWSTR,LPCWSTR,DWORD);
01287 #define      MoveFileEx WINELIB_NAME_AW(MoveFileEx)
01288 BOOL        WINAPI NotifyChangeEventLog (HANDLE,HANDLE);
01289 HANDLE       WINAPI OpenBackupEventLogA (LPCSTR,LPCSTR);
01290 HANDLE       WINAPI OpenBackupEventLogW (LPCWSTR,LPCWSTR);
01291 #define      OpenBackupEventLog WINELIB_NAME_AW(OpenBackupEventLog)
01292 HANDLE       WINAPI OpenEventA (DWORD,BOOL,LPCSTR);
01293 HANDLE       WINAPI OpenEventW (DWORD,BOOL,LPCWSTR);
01294 #define      OpenEvent WINELIB_NAME_AW(OpenEvent)
01295 HANDLE       WINAPI OpenEventLogA (LPCSTR,LPCSTR);
01296 HANDLE       WINAPI OpenEventLogW (LPCWSTR,LPCWSTR);
01297 #define      OpenEventLog WINELIB_NAME_AW(OpenEventLog)
01298 HANDLE       WINAPI OpenFileMappingA (DWORD,BOOL,LPCSTR);
01299 HANDLE       WINAPI OpenFileMappingW (DWORD,BOOL,LPCWSTR);
01300 #define      OpenFileMapping WINELIB_NAME_AW(OpenFileMapping)
01301 HANDLE       WINAPI OpenMutexA (DWORD,BOOL,LPCSTR);
01302 HANDLE       WINAPI OpenMutexW (DWORD,BOOL,LPCWSTR);
01303 #define      OpenMutex WINELIB_NAME_AW(OpenMutex)
01304 HANDLE       WINAPI OpenProcess (DWORD,BOOL,DWORD);
01305 BOOL        WINAPI OpenProcessToken (HANDLE,DWORD,PHANDLE);
01306 HANDLE       WINAPI OpenSemaphoreA (DWORD,BOOL,LPCSTR);
01307 HANDLE       WINAPI OpenSemaphoreW (DWORD,BOOL,LPCWSTR);
01308 #define      OpenSemaphore WINELIB_NAME_AW(OpenSemaphore)
01309 BOOL        WINAPI OpenThreadToken (HANDLE,DWORD,BOOL,PHANDLE);
01310 HANDLE       WINAPI OpenWaitableTimerA (DWORD,BOOL,LPCSTR);
01311 HANDLE       WINAPI OpenWaitableTimerW (DWORD,BOOL,LPCWSTR);
01312 #define      OpenWaitableTimer WINELIB_NAME_AW(OpenWaitableTimer)
01313 DWORD       WINAPI PrepareTape (HANDLE,DWORD,BOOL);
01314 BOOL        WINAPI PulseEvent (HANDLE);
01315 BOOL        WINAPI PurgeComm (HANDLE,DWORD);
01316 DWORD       WINAPI QueryDosDeviceA (LPCSTR,LPSTR,DWORD);
01317 DWORD       WINAPI QueryDosDeviceW (LPCWSTR,LPWSTR,DWORD);
01318 #define      QueryDosDevice WINELIB_NAME_AW(QueryDosDevice)
01319 BOOL        WINAPI QueryPerformanceCounter (LARGE_INTEGER*);
01320 BOOL        WINAPI QueryPerformanceFrequency (LARGE_INTEGER*);
01321 BOOL        WINAPI ReadEventLogA (HANDLE,DWORD,DWORD,LPVOID,DWORD,DWORD *,DWORD *);
01322 BOOL        WINAPI ReadEventLogW (HANDLE,DWORD,DWORD,LPVOID,DWORD,DWORD *,DWORD *);
01323 #define      ReadEventLog WINELIB_NAME_AW(ReadEventLog)
01324 BOOL        WINAPI ReadFile (HANDLE,LPVOID,DWORD,LPDWORD,LPOVERLAPPED);
01325 BOOL        WINAPI ReadFileEx (HANDLE,LPVOID,DWORD,LPOVERLAPPED,LPOVERLAPPED_COMPLETION_ROUTINE);
01326 HANDLE       WINAPI RegisterEventSourceA (LPCSTR,LPCSTR);
01327 HANDLE       WINAPI RegisterEventSourceW (LPCWSTR,LPCWSTR);
01328 #define      RegisterEventSource WINELIB_NAME_AW(RegisterEventSource)
01329 BOOL        WINAPI ReleaseMutex (HANDLE);
01330 BOOL        WINAPI ReleaseSemaphore (HANDLE, LONG, LPLONG);
01331 BOOL        WINAPI ReportEventA (HANDLE,WORD,WORD,DWORD,PSID,WORD,DWORD,LPCSTR *,LPVOID);
01332 BOOL        WINAPI ReportEventW (HANDLE,WORD,WORD,DWORD,PSID,WORD,DWORD,LPCWSTR *,LPVOID);
01333 #define      ReportEvent WINELIB_NAME_AW(ReportEvent)
01334 BOOL        WINAPI ResetEvent (HANDLE);
01335 DWORD       WINAPI ResumeThread (HANDLE);
01336 BOOL        WINAPI RevertToSelf (void);
01337 DWORD       WINAPI SearchPathA (LPCSTR,LPCSTR,LPCSTR,DWORD,LPSTR,LPSTR*);
01338 DWORD       WINAPI SearchPathW (LPCWSTR,LPCWSTR,LPCWSTR,DWORD,LPWSTR,LPWSTR*);
01339 #define      SearchPath WINELIB_NAME_AW(SearchPath)

```

```

01340 BOOL        WINAPI SetCommConfig(HANDLE, LPCOMMCONFIG, DWORD);
01341 BOOL        WINAPI SetCommBreak(HANDLE);
01342 BOOL        WINAPI SetCommMask(HANDLE, DWORD);
01343 BOOL        WINAPI SetCommState(HANDLE, LPDCB);
01344 BOOL        WINAPI SetCommTimeouts(HANDLE, LPCOMMTIMEOUTS);
01345 BOOL        WINAPI SetComputerNameA(LPCSTR);
01346 BOOL        WINAPI SetComputerNameW(LPCWSTR);
01347 #define        SetComputerName WINELIB_NAME_AW(SetComputerName)
01348 BOOL        WINAPI SetDefaultCommConfigA(LPCSTR, LPCOMMCONFIG, DWORD);
01349 BOOL        WINAPI SetDefaultCommConfigW(LPCWSTR, LPCOMMCONFIG, DWORD);
01350 #define        SetDefaultCommConfig WINELIB_NAME_AW(SetDefaultCommConfig)
01351 BOOL        WINAPI SetEndOfFile(HANDLE);
01352 BOOL        WINAPI SetEnvironmentVariableA(LPCSTR, LPCSTR);
01353 BOOL        WINAPI SetEnvironmentVariableW(LPCWSTR, LPCWSTR);
01354 #define        SetEnvironmentVariable WINELIB_NAME_AW(SetEnvironmentVariable)
01355 BOOL        WINAPI SetEvent(HANDLE);
01356 VOID        WINAPI SetFileApisToANSI(void);
01357 VOID        WINAPI SetFileApisToOEM(void);
01358 DWORD       WINAPI SetFilePointer(HANDLE, LONG, LPLONG, DWORD);
01359 BOOL        WINAPI SetFileSecurityA(LPCSTR, SECURITY_INFORMATION, PSECURITY_DESCRIPTOR);
01360 BOOL        WINAPI SetFileSecurityW(LPCWSTR, SECURITY_INFORMATION, PSECURITY_DESCRIPTOR);
01361 #define        SetFileSecurity WINELIB_NAME_AW(SetFileSecurity)
01362 BOOL        WINAPI SetFileTime(HANDLE, const FILETIME*, const FILETIME*, const FILETIME*);
01363 BOOL        WINAPI SetHandleInformation(HANDLE, DWORD, DWORD);
01364 BOOL        WINAPI SetKernelObjectSecurity(HANDLE, SECURITY_INFORMATION, PSECURITY_DESCRIPTOR);
01365 BOOL        WINAPI SetPriorityClass(HANDLE, DWORD);
01366 BOOL        WINAPI SetLocalTime(const SYSTEMTIME*);
01367 BOOL        WINAPI SetSecurityDescriptorDacl(PSECURITY_DESCRIPTOR, BOOL, PACL, BOOL);
01368 BOOL        WINAPI SetSecurityDescriptorGroup(PSECURITY_DESCRIPTOR, PSID, BOOL);
01369 BOOL        WINAPI SetSecurityDescriptorOwner(PSECURITY_DESCRIPTOR, PSID, BOOL);
01370 BOOL        WINAPI SetSecurityDescriptorSacl(PSECURITY_DESCRIPTOR, BOOL, PACL, BOOL);
01371 BOOL        WINAPI SetStdHandle(DWORD, HANDLE);
01372 BOOL        WINAPI SetSystemPowerState(BOOL, BOOL);
01373 BOOL        WINAPI SetSystemTime(const SYSTEMTIME*);
01374 DWORD       WINAPI SetTapeParameters(HANDLE, DWORD, LPVOID);
01375 DWORD       WINAPI SetTapePosition(HANDLE, DWORD, DWORD, DWORD, BOOL);
01376 DWORD       WINAPI SetThreadAffinityMask(HANDLE, DWORD);
01377 BOOL        WINAPI SetThreadContext(HANDLE, const CONTEXT *);
01378 DWORD       WINAPI SetThreadExecutionState(EXECUTION_STATE);
01379 BOOL        WINAPI SetThreadPriority(HANDLE, INT);
01380 BOOL        WINAPI SetThreadPriorityBoost(HANDLE, BOOL);
01381 BOOL        WINAPI SetThreadToken(PHANDLE, HANDLE);
01382 BOOL        WINAPI SetTimeZoneInformation(const LPTIME_ZONE_INFORMATION);
01383 BOOL        WINAPI SetWaitableTimer(HANDLE, const LARGE_INTEGER*, LONG, PTIMERAPCROUTINE, LPVOID, BOOL);
01384 BOOL        WINAPI SetupComm(HANDLE, DWORD, DWORD);
01385 VOID        WINAPI Sleep(DWORD);
01386 DWORD       WINAPI SleepEx(DWORD, BOOL);
01387 DWORD       WINAPI SuspendThread(HANDLE);
01388 BOOL        WINAPI SystemTimeToFileTime(const SYSTEMTIME*, LPFILETIME);
01389 DWORD       WINAPI TlsAlloc(void);
01390 BOOL        WINAPI TlsFree(DWORD);
01391 LPVOID       WINAPI TlsGetValue(DWORD);
01392 BOOL        WINAPI TlsSetValue(DWORD, LPVOID);
01393 BOOL        WINAPI TransmitCommChar(HANDLE, CHAR);
01394 BOOL        WINAPI UnlockFile(HANDLE, DWORD, DWORD, DWORD, DWORD);
01395 BOOL        WINAPI UnmapViewOfFile(LPVOID);
01396 LPVOID       WINAPI VirtualAlloc(LPVOID, DWORD, DWORD, DWORD);
01397 LPVOID       WINAPI VirtualAllocEx(HANDLE, LPVOID, DWORD, DWORD, DWORD);
01398 BOOL        WINAPI VirtualFree(LPVOID, DWORD, DWORD);
01399 BOOL        WINAPI VirtualLock(LPVOID, DWORD);
01400 BOOL        WINAPI VirtualProtect(LPVOID, DWORD, DWORD, LPDWORD);
01401 BOOL        WINAPI VirtualProtectEx(HANDLE, LPVOID, DWORD, DWORD, LPDWORD);
01402 DWORD       WINAPI VirtualQuery(LPCVOID, LPMEMORY_BASIC_INFORMATION, DWORD);
01403 DWORD       WINAPI VirtualQueryEx(HANDLE, LPCVOID, LPMEMORY_BASIC_INFORMATION, DWORD);
01404 BOOL        WINAPI VirtualUnlock(LPVOID, DWORD);
01405 BOOL        WINAPI WaitCommEvent(HANDLE, LPDWORD, LPOVERLAPPED);
01406 BOOL        WINAPI WaitForDebugEvent(LPDEBUG_EVENT, DWORD);
01407 DWORD       WINAPI WaitForMultipleObjects(DWORD, const HANDLE*, BOOL, DWORD);
01408 DWORD       WINAPI WaitForMultipleObjectsEx(DWORD, const HANDLE*, BOOL, DWORD, BOOL);
01409 DWORD       WINAPI WaitForSingleObject(HANDLE, DWORD);
01410 DWORD       WINAPI WaitForSingleObjectEx(HANDLE, DWORD, BOOL);
01411 BOOL        WINAPI WaitNamedPipeA(LPCSTR, DWORD);
01412 BOOL        WINAPI WaitNamedPipeW(LPCWSTR, DWORD);
01413 #define        WaitNamedPipe WINELIB_NAME_AW(WaitNamedPipe)
01414 BOOL        WINAPI WriteFile(HANDLE, LPCVOID, DWORD, LPDWORD, LPOVERLAPPED);
01415 BOOL        WINAPI WriteFileEx(HANDLE, LPCVOID, DWORD, LPOVERLAPPED, LPOVERLAPPED_COMPLETION_ROUTINE);
01416 DWORD       WINAPI WriteTapemark(HANDLE, DWORD, DWORD, BOOL);
01417 ATOM        WINAPI AddAtomA(LPCSTR);
01418 ATOM        WINAPI AddAtomW(LPCWSTR);
01419 #define        AddAtom WINELIB_NAME_AW(AddAtom)
01420 BOOL        WINAPI CreateDirectoryA(LPCSTR, LPSECURITY_ATTRIBUTES);
01421 BOOL        WINAPI CreateDirectoryW(LPCWSTR, LPSECURITY_ATTRIBUTES);
01422 #define        CreateDirectory WINELIB_NAME_AW(CreateDirectory)
01423 BOOL        WINAPI CreateDirectoryExA(LPCSTR, LPCSTR, LPSECURITY_ATTRIBUTES);
01424 BOOL        WINAPI CreateDirectoryExW(LPCWSTR, LPCWSTR, LPSECURITY_ATTRIBUTES);
01425 #define        CreateDirectoryEx WINELIB_NAME_AW(CreateDirectoryEx)
01426 BOOL        WINAPI DefineDosDeviceA(DWORD, LPCSTR, LPCSTR);

```

```

01427 #define      DefineHandleTable(w) ((w),TRUE)
01428 ATOM          WINAPI DeleteAtom(ATOM);
01429 BOOL          WINAPI DeleteFileA(LPCSTR);
01430 BOOL          WINAPI DeleteFileW(LPCWSTR);
01431 #define      DeleteFile WINELIB_NAME_AW(DeleteFile)
01432 void          WINAPI FatalAppExitA(UINT,LPCSTR);
01433 void          WINAPI FatalAppExitW(UINT,LPCWSTR);
01434 #define      FatalAppExit WINELIB_NAME_AW(FatalAppExit)
01435 ATOM          WINAPI FindAtomA(LPCSTR);
01436 ATOM          WINAPI FindAtomW(LPCWSTR);
01437 #define      FindAtom WINELIB_NAME_AW(FindAtom)
01438 BOOL          WINAPI FindClose(HANDLE);
01439 HANDLE        WINAPI FindFirstFileA(LPCSTR,LPWIN32_FIND_DATAA);
01440 HANDLE        WINAPI FindFirstFileW(LPCWSTR,LPWIN32_FIND_DATAW);
01441 #define      FindFirstFile WINELIB_NAME_AW(FindFirstFile)
01442 HANDLE        WINAPI FindFirstFileExA(LPCSTR,INDEX_INFO_LEVELS,LPVOID,INDEX_SEARCH_OPS,LPVOID,DWORD);
01443 HANDLE        WINAPI FindFirstFileExW(LPCWSTR,INDEX_INFO_LEVELS,LPVOID,INDEX_SEARCH_OPS,LPVOID,DWORD);
01444 #define      FindFirstFileEx WINELIB_NAME_AW(FindFirstFileEx)
01445 BOOL          WINAPI FindNextFileA(HANDLE,LPWIN32_FIND_DATAA);
01446 BOOL          WINAPI FindNextFileW(HANDLE,LPWIN32_FIND_DATAW);
01447 #define      FindNextFile WINELIB_NAME_AW(FindNextFile)
01448 HRSRC         WINAPI FindResourceA(HMODULE,LPCSTR,LPCSTR);
01449 HRSRC         WINAPI FindResourceW(HMODULE,LPCWSTR,LPCWSTR);
01450 #define      FindResource WINELIB_NAME_AW(FindResource)
01451 BOOL          WINAPI FreeLibrary(HMODULE);
01452 #define      FreeModule(handle) FreeLibrary(handle)
01453 #define      FreeProcInstance(proc) /*nothing*/
01454 BOOL          WINAPI FreeResource(HGLOBAL);
01455 UINT          WINAPI GetAtomNameA(ATOM,LPSTR,INT);
01456 UINT          WINAPI GetAtomNameW(ATOM,LPWSTR,INT);
01457 #define      GetAtomName WINELIB_NAME_AW(GetAtomName)
01458 UINT          WINAPI GetCurrentDirectoryA(UINT,LPSTR);
01459 UINT          WINAPI GetCurrentDirectoryW(UINT,LPWSTR);
01460 #define      GetCurrentDirectory WINELIB_NAME_AW(GetCurrentDirectory)
01461 #define      GetCurrentTime() GetTickCount()
01462 BOOL          WINAPI GetDiskFreeSpaceA(LPCSTR,LPDWORD,LPDWORD,LPDWORD,LPDWORD);
01463 BOOL          WINAPI GetDiskFreeSpaceW(LPCWSTR,LPDWORD,LPDWORD,LPDWORD,LPDWORD);
01464 #define      GetDiskFreeSpace WINELIB_NAME_AW(GetDiskFreeSpace)
01465 BOOL          WINAPI GetDiskFreeSpaceExA(LPCSTR,PULARGE_INTEGER,PULARGE_INTEGER,PULARGE_INTEGER);
01466 BOOL          WINAPI GetDiskFreeSpaceExW(LPCWSTR,PULARGE_INTEGER,PULARGE_INTEGER,PULARGE_INTEGER);
01467 #define      GetDiskFreeSpaceEx WINELIB_NAME_AW(GetDiskFreeSpaceEx)
01468 UNT          WINAPI GetDriveTypeA(LPCSTR);
01469 UNT          WINAPI GetDriveTypeW(LPCWSTR);
01470 #define      GetDriveType WINELIB_NAME_AW(GetDriveType)
01471 BOOL          WINAPI GetExitCodeProcess(HANDLE,LPDWORD);
01472 DWORD         WINAPI GetFileAttributesA(LPCSTR);
01473 DWORD         WINAPI GetFileAttributesW(LPCWSTR);
01474 #define      GetFileAttributes WINELIB_NAME_AW(GetFileAttributes)
01475 #define      GetFreeSpace(w) (0x100000L)
01476 UNT          WINAPI GetLogicalDriveStringsA(UINT,LPSTR);
01477 UNT          WINAPI GetLogicalDriveStringsW(UINT,LPWSTR);
01478 #define      GetLogicalDriveStrings WINELIB_NAME_AW(GetLogicalDriveStrings)
01479 DWORD         WINAPI GetModuleFileNameA(HMODULE,LPSTR,DWORD);
01480 DWORD         WINAPI GetModuleFileNameW(HMODULE,LPWSTR,DWORD);
01481 #define      GetModuleFileName WINELIB_NAME_AW(GetModuleFileName)
01482 HMODULE        WINAPI GetModuleHandleA(LPCSTR);
01483 HMODULE        WINAPI GetModuleHandleW(LPCWSTR);
01484 #define      GetModuleHandle WINELIB_NAME_AW(GetModuleHandle)
01485 BOOL          WINAPI GetOverlappedResult(HANDLE,LPOVERLAPPED,LPDWORD,BOOL);
01486 UNT          WINAPI GetPrivateProfileIntA(LPCSTR,LPCSTR,INT,LPCSTR);
01487 UNT          WINAPI GetPrivateProfileIntW(LPCWSTR,LPCWSTR,INT,LPCWSTR);
01488 #define      GetPrivateProfileInt WINELIB_NAME_AW(GetPrivateProfileInt)
01489 INT           WINAPI GetPrivateProfileSectionA(LPCSTR,LPSTR,DWORD,LPCSTR);
01490 INT           WINAPI GetPrivateProfileSectionW(LPCWSTR,LPWSTR,DWORD,LPCWSTR);
01491 #define      GetPrivateProfileSection WINELIB_NAME_AW(GetPrivateProfileSection)
01492 DWORD         WINAPI GetPrivateProfileSectionNamesA(LPSTR,DWORD,LPCSTR);
01493 DWORD         WINAPI GetPrivateProfileSectionNamesW(LPWSTR,DWORD,LPCWSTR);
01494 #define      GetPrivateProfileSectionNames WINELIB_NAME_AW(GetPrivateProfileSectionNames)
01495 INT           WINAPI GetPrivateProfileStringA(LPCSTR,LPCSTR,LPCSTR,LPSTR,UINT,LPCSTR);
01496 INT           WINAPI GetPrivateProfileStringW(LPCWSTR,LPCWSTR,LPCWSTR,LPWSTR,UINT,LPCWSTR);
01497 #define      GetPrivateProfileString WINELIB_NAME_AW(GetPrivateProfileString)
01498 BOOL          WINAPI GetPrivateProfileStructA(LPCSTR,LPCSTR,LPVOID,UINT,LPCSTR);
01499 BOOL          WINAPI GetPrivateProfileStructW(LPCWSTR,LPCWSTR,LPVOID,UINT,LPCWSTR);
01500 #define      GetPrivateProfileStruct WINELIB_NAME_AW(GetPrivateProfileStruct)
01501 FARPROC        WINAPI GetProcAddress(HMODULE,LPCSTR);
01502 UNT          WINAPI GetProfileIntA(LPCSTR,LPCSTR,INT);
01503 UNT          WINAPI GetProfileIntW(LPCWSTR,LPCWSTR,INT);
01504 #define      GetProfileInt WINELIB_NAME_AW(GetProfileInt)
01505 INT           WINAPI GetProfileSectionA(LPCSTR,LPSTR,DWORD);
01506 INT           WINAPI GetProfileSectionW(LPCWSTR,LPWSTR,DWORD);
01507 #define      GetProfileSection WINELIB_NAME_AW(GetProfileSection)
01508 INT           WINAPI GetProfileStringA(LPCSTR,LPCSTR,LPCSTR,LPSTR,UINT);
01509 INT           WINAPI GetProfileStringW(LPCWSTR,LPCWSTR,LPCWSTR,LPWSTR,UINT);
01510 #define      GetProfileString WINELIB_NAME_AW(GetProfileString)
01511 VOID          WINAPI GetStartupInfoA(LPSTARTUPINFOA);
01512 VOID          WINAPI GetStartupInfoW(LPSTARTUPINFOW);
01513 #define      GetStartupInfo WINELIB_NAME_AW(GetStartupInfo)

```

```

01514 UINT      WINAPI GetSystemDirectoryA(LPSTR,UINT);
01515 UINT      WINAPI GetSystemDirectoryW(LPWSTR,UINT);
01516 #define      GetSystemDirectory WINELIB_NAME_AW(GetSystemDirectory)
01517 DWORD      WINAPI GetTickCount(void);
01518 UINT      WINAPI GetTempFileNameA(LPCSTR,LPCSTR,UINT,LPSTR);
01519 UINT      WINAPI GetTempFileNameW(LPCWSTR,LPCWSTR,UINT,LPWSTR);
01520 #define      GetTempFileName WINELIB_NAME_AW(GetTempFileName)
01521 UINT      WINAPI GetTempPathA(UINT,LPSTR);
01522 UINT      WINAPI GetTempPathW(UINT,LPWSTR);
01523 #define      GetTempPath WINELIB_NAME_AW(GetTempPath)
01524 LONG      WINAPI GetVersion(void);
01525 BOOL      WINAPI GetVolumeInformationA(LPCSTR,LPSTR,DWORD,LPDWORD,LPDWORD,LPDWORD,LPSTR,DWORD);
01526 BOOL      WINAPI GetVolumeInformationW(LPCWSTR,LPWSTR,DWORD,LPDWORD,LPDWORD,LPDWORD,LPWSTR,DWORD);
01527 #define      GetVolumeInformation WINELIB_NAME_AW(GetVolumeInformation)
01528 UINT      WINAPI GetWindowsDirectoryA(LPSTR,UINT);
01529 UINT      WINAPI GetWindowsDirectoryW(LPWSTR,UINT);
01530 #define      GetWindowsDirectory WINELIB_NAME_AW(GetWindowsDirectory)
01531 ATOM      WINAPI GlobalAddAtomA(LPCSTR);
01532 ATOM      WINAPI GlobalAddAtomW(LPCWSTR);
01533 #define      GlobalAddAtom WINELIB_NAME_AW(GlobalAddAtom)
01534 HGLOBAL    WINAPI GlobalAlloc(UINT,DWORD);
01535 DWORD      WINAPI GlobalCompact(DWORD);
01536 ATOM      WINAPI GlobalDeleteAtom(ATOM);
01537 ATOM      WINAPI GlobalFindAtomA(LPCSTR);
01538 ATOM      WINAPI GlobalFindAtomW(LPCWSTR);
01539 #define      GlobalFindAtom WINELIB_NAME_AW(GlobalFindAtom)
01540 UINT      WINAPI GlobalFlags(HGLOBAL);
01541 HGLOBAL    WINAPI GlobalFree(HGLOBAL);
01542 UINT      WINAPI GlobalGetAtomNameA(ATOM,LPSTR,INT);
01543 UINT      WINAPI GlobalGetAtomNameW(ATOM,LPWSTR,INT);
01544 #define      GlobalGetAtomName WINELIB_NAME_AW(GlobalGetAtomName)
01545 HGLOBAL    WINAPI GlobalHandle(LPCVOID);
01546 VOID      WINAPI GlobalFix(HGLOBAL);
01547 LPVOID     WINAPI GlobalLock(HGLOBAL);
01548 HGLOBAL    WINAPI GlobalReAlloc(HGLOBAL,DWORD,UINT);
01549 DWORD      WINAPI GlobalSize(HGLOBAL);
01550 VOID      WINAPI GlobalUnfix(HGLOBAL);
01551 BOOL      WINAPI GlobalUnlock(HGLOBAL);
01552 BOOL      WINAPI GlobalUnWire(HGLOBAL);
01553 LPVOID     WINAPI GlobalWire(HGLOBAL);
01554 #define      HasOverlappedCompleted(lpOverlapped) ((lpOverlapped)->Internal != STATUS_PENDING)
01555 BOOL      WINAPI InitAtomTable(DWORD);
01556 BOOL      WINAPI IsBadCodePtr(FARPROC);
01557 BOOL      WINAPI IsBadHugeReadPtr(LPCVOID,UINT);
01558 BOOL      WINAPI IsBadHugeWritePtr(LPVOID,UINT);
01559 BOOL      WINAPI IsBadReadPtr(LPCVOID,UINT);
01560 BOOL      WINAPI IsBadStringPtrA(LPCSTR,UINT);
01561 BOOL      WINAPI IsBadStringPtrW(LPCWSTR,UINT);
01562 #define      IsBadStringPtr WINELIB_NAME_AW(IsBadStringPtr)
01563 BOOL      WINAPI IsBadWritePtr(LPVOID,UINT);
01564 BOOL      WINAPI IsDebuggerPresent(void);
01565 HMODULE    WINAPI LoadLibraryA(LPCSTR);
01566 HMODULE    WINAPI LoadLibraryW(LPCWSTR);
01567 #define      LoadLibrary WINELIB_NAME_AW(LoadLibrary)
01568 HMODULE    WINAPI LoadLibraryExA(LPCSTR,HANDLE,DWORD);
01569 HMODULE    WINAPI LoadLibraryExW(LPCWSTR,HANDLE,DWORD);
01570 #define      LoadLibraryEx WINELIB_NAME_AW(LoadLibraryEx)
01571 HINSTANCE  WINAPI LoadModule(LPCSTR,LPVOID);
01572 HGLOBAL    WINAPI LoadResource(HMODULE,HRSRC);
01573 HLOCAL     WINAPI LocalAlloc(UINT,DWORD);
01574 UINT      WINAPI LocalCompact(UINT);
01575 UINT      WINAPI LocalFlags(HLOCAL);
01576 HLOCAL     WINAPI LocalFree(HLOCAL);
01577 HLOCAL     WINAPI LocalHandle(LPCVOID);
01578 LPVOID     WINAPI LocalLock(HLOCAL);
01579 HLOCAL     WINAPI LocalReAlloc(HLOCAL,DWORD,UINT);
01580 UINT      WINAPI LocalShrink(HGLOBAL,UINT);
01581 UINT      WINAPI LocalSize(HLOCAL);
01582 BOOL      WINAPI LocalUnlock(HLOCAL);
01583 LPVOID     WINAPI LockResource(HGLOBAL);
01584 #define      LockSegment(handle) GlobalFix((HANDLE)(handle))
01585 #define      MakeProcInstance(proc,inst) (proc)
01586 HFILE      WINAPI OpenFile(LPCSTR,OFSTRUCT*,UINT);
01587 VOID      WINAPI OutputDebugStringA(LPCSTR);
01588 VOID      WINAPI OutputDebugStringW(LPCWSTR);
01589 #define      OutputDebugString WINELIB_NAME_AW(OutputDebugString)
01590 BOOL      WINAPI ReadProcessMemory(HANDLE,LPVOID,DWORD,LPDWORD);
01591 BOOL      WINAPI RemoveDirectoryA(LPCSTR);
01592 BOOL      WINAPI RemoveDirectoryW(LPCWSTR);
01593 #define      RemoveDirectory WINELIB_NAME_AW(RemoveDirectory)
01594 BOOL      WINAPI SetCurrentDirectoryA(LPCSTR);
01595 BOOL      WINAPI SetCurrentDirectoryW(LPCWSTR);
01596 #define      SetCurrentDirectory WINELIB_NAME_AW(SetCurrentDirectory)
01597 UINT      WINAPI SetErrorMode(UINT);
01598 BOOL      WINAPI SetFileAttributesA(LPCSTR,DWORD);
01599 BOOL      WINAPI SetFileAttributesW(LPCWSTR,DWORD);
01600 #define      SetFileAttributes WINELIB_NAME_AW(SetFileAttributes)

```

```

01601 UINT      WINAPI SetHandleCount (UINT);
01602 #define      SetSwapAreaSize(w) (w)
01603 BOOL        WINAPI SetVolumeLabelA (LPCSTR, LPCSTR);
01604 BOOL        WINAPI SetVolumeLabelW (LPCWSTR, LPCWSTR);
01605 #define      SetVolumeLabel WINELIB_NAME_AW(SetVolumeLabel)
01606 DWORD       WINAPI SizeofResource (HMODULE, HRSRC);
01607 BOOL        WINAPI UnlockFileEx (HFILE, DWORD, DWORD, DWORD, LPOVERLAPPED);
01608 #define      UnlockSegment(handle) GlobalUnfix((HANDLE)(handle))
01609 BOOL        WINAPI WritePrivateProfileSectionA (LPCSTR, LPCSTR, LPCSTR);
01610 BOOL        WINAPI WritePrivateProfileSectionW (LPCWSTR, LPCWSTR, LPCWSTR);
01611 #define      WritePrivateProfileSection WINELIB_NAME_AW(WritePrivateProfileSection)
01612 BOOL        WINAPI WritePrivateProfileStringA (LPCSTR, LPCSTR, LPCSTR, LPCSTR);
01613 BOOL        WINAPI WritePrivateProfileStringW (LPCWSTR, LPCWSTR, LPCWSTR, LPCWSTR);
01614 #define      WritePrivateProfileString WINELIB_NAME_AW(WritePrivateProfileString)
01615 BOOL        WINAPI WriteProfileSectionA (LPCSTR, LPCSTR);
01616 BOOL        WINAPI WriteProfileSectionW (LPCWSTR, LPCWSTR);
01617 #define      WritePrivateProfileSection WINELIB_NAME_AW(WritePrivateProfileSection)
01618 BOOL        WINAPI WritePrivateProfileStructA (LPCSTR, LPCSTR, LPVOID, UINT, LPCSTR);
01619 BOOL        WINAPI WritePrivateProfileStructW (LPCWSTR, LPCWSTR, LPVOID, UINT, LPCWSTR);
01620 #define      WritePrivateProfileStruct WINELIB_NAME_AW(WritePrivateProfileStruct)
01621 BOOL        WINAPI WriteProcessMemory (HANDLE, LPVOID, LPCVOID, DWORD, LPDWORD);
01622 BOOL        WINAPI WriteProfileStringA (LPCSTR, LPCSTR, LPCSTR);
01623 BOOL        WINAPI WriteProfileStringW (LPCWSTR, LPCWSTR, LPCWSTR);
01624 #define      WriteProfileString WINELIB_NAME_AW(WriteProfileString)
01625 #define      Yield()
01626 LPSTR        WINAPI lstrcatA (LPSTR, LPCSTR);
01627 LPWSTR        WINAPI lstrcatW (LPWSTR, LPCWSTR);
01628 #define      lstrcat WINELIB_NAME_AW(lstrcat)
01629 LPSTR        WINAPI lstrcpyA (LPSTR, LPCSTR);
01630 LPWSTR        WINAPI lstrcpyW (LPWSTR, LPCWSTR);
01631 #define      lstrcpy WINELIB_NAME_AW(lstrcpy)
01632 LPSTR        WINAPI lstrcpynA (LPSTR, LPCSTR, INT);
01633 LPWSTR        WINAPI lstrcpynW (LPWSTR, LPCWSTR, INT);
01634 #define      lstrcpyn WINELIB_NAME_AW(lstrcpyn)
01635 INT          WINAPI lstrlenA (LPCSTR);
01636 INT          WINAPI lstrlenW (LPCWSTR);
01637 #define      lstrlen WINELIB_NAME_AW(lstrlen)
01638 HINSTANCE     WINAPI WinExec (LPCSTR, UINT);
01639 LONG         WINAPI _hread (HFILE, LPVOID, LONG);
01640 LONG         WINAPI _hwrite (HFILE, LPCSTR, LONG);
01641 HFILE        WINAPI _lcreat (LPCSTR, INT);
01642 HFILE        WINAPI _lclose (HFILE);
01643 LONG         WINAPI _llseek (HFILE, LONG, INT);
01644 HFILE        WINAPI _lopen (LPCSTR, INT);
01645 UINT         WINAPI _hread (HFILE, LPVOID, UINT);
01646 UINT         WINAPI _lwrite (HFILE, LPCSTR, UINT);
01647 INT          WINAPI lstrcmpA (LPCSTR, LPCSTR);
01648 INT          WINAPI lstrcmpW (LPCWSTR, LPCWSTR);
01649 #define      lstrcmp WINELIB_NAME_AW(lstrcmp)
01650 INT          WINAPI lstrcmpiA (LPCSTR, LPCSTR);
01651 INT          WINAPI lstrcmpiW (LPCWSTR, LPCWSTR);
01652 #define      lstrcmpi WINELIB_NAME_AW(lstrcmpi)
01653
01654 /* compatibility macros */
01655 #define      FillMemory RtlFillMemory
01656 #define      MoveMemory RtlMoveMemory
01657 #define      ZeroMemory RtlZeroMemory
01658 #define      CopyMemory RtlCopyMemory
01659
01660 /* undocumented functions */
01661
01662 typedef struct tagSYSLEVEL
01663 {
01664     CRITICAL_SECTION crst;
01665     INT              level;
01666 } SYSLEVEL;
01667
01668 /* [GS]etProcessDword offsets */
01669 #define      GPD_APP_COMPAT_FLAGS      (-56)
01670 #define      GPD_LOAD_DONE_EVENT      (-52)
01671 #define      GPD_HINSTANCE16          (-48)
01672 #define      GPD_WINDOWS_VERSION      (-44)
01673 #define      GPD_THDB                  (-40)
01674 #define      GPD_PDB                   (-36)
01675 #define      GPD_STARTF_SHELLDATA      (-32)
01676 #define      GPD_STARTF_HOTKEY         (-28)
01677 #define      GPD_STARTF_SHOWWINDOW     (-24)
01678 #define      GPD_STARTF_SIZE           (-20)
01679 #define      GPD_STARTF_POSITION       (-16)
01680 #define      GPD_STARTF_FLAGS          (-12)
01681 #define      GPD_PARENT                (- 8)
01682 #define      GPD_FLAGS                 (- 4)
01683 #define      GPD_USERDATA              ( 0)
01684
01685 void        WINAPI DisposeLZ32Handle (HANDLE);
01686 HANDLE      WINAPI DosFileHandleToWin32Handle (HFILE);
01687 DWORD       WINAPI GetProcessDword (DWORD, INT);

```

```

01688 VOID          WINAPI GetpWin16Lock(SYSLEVEL**);
01689 DWORD          WINAPI MapLS(LPCVOID);
01690 DWORD          WINAPI MapProcessHandle(HANDLE);
01691 LPVOID          WINAPI MapSL(DWORD);
01692 VOID          WINAPI ReleaseThunkLock(DWORD*);
01693 VOID          WINAPI RestoreThunkLock(DWORD);
01694 void           WINAPI SetProcessDword(DWORD,INT,DWORD);
01695 VOID          WINAPI UnMapLS(DWORD);
01696 HFILE          WINAPI Win32HandleToDosFileHandle(HANDLE);
01697 VOID          WINAPI _CheckNotSysLevel(SYSLEVEL *lock);
01698 DWORD          WINAPI _ConfirmWin16Lock(void);
01699 DWORD          WINAPI _ConfirmSysLevel(SYSLEVEL*);
01700 VOID          WINAPI _EnterSysLevel(SYSLEVEL*);
01701 VOID          WINAPI _LeaveSysLevel(SYSLEVEL*);
01702
01703
01704 /* Wine internal functions */
01705
01706 BOOL           WINAPI wine_get_unix_file_name( LPCSTR dos, LPSTR buffer, DWORD len );
01707
01708
01709 /* a few optimizations for i386/gcc */
01710
01711 /*#if defined(__i386__) && defined(__GNUC__)*/
01712 /* Deleted these since the real WINE environment is not available */
01713 #if 0
01714
01715 extern inline LONG WINAPI InterlockedCompareExchange( PLONG dest, LONG xchg, LONG compare );
01716 extern inline LONG WINAPI InterlockedCompareExchange( PLONG dest, LONG xchg, LONG compare )
01717 {
01718     LONG ret;
01719     __asm__ __volatile__( "lock; cmpxchgl %2,%1"
01720                          : "=a" (ret) : "r" (dest), "r" (xchg), "0" (compare) : "memory" );
01721     return ret;
01722 }
01723
01724 extern inline LONG WINAPI InterlockedExchange( PLONG dest, LONG val );
01725 extern inline LONG WINAPI InterlockedExchange( PLONG dest, LONG val )
01726 {
01727     LONG ret;
01728     __asm__ __volatile__( "lock; xchgl %0,%1"
01729                          : "=r" (ret) : "r" (dest), "0" (val) : "memory" );
01730     return ret;
01731 }
01732
01733 extern inline LONG WINAPI InterlockedExchangeAdd( PLONG dest, LONG incr );
01734 extern inline LONG WINAPI InterlockedExchangeAdd( PLONG dest, LONG incr )
01735 {
01736     LONG ret;
01737     __asm__ __volatile__( "lock; xaddl %0,%1"
01738                          : "=r" (ret) : "r" (dest), "0" (incr) : "memory" );
01739     return ret;
01740 }
01741
01742 extern inline LONG WINAPI InterlockedIncrement( PLONG dest );
01743 extern inline LONG WINAPI InterlockedIncrement( PLONG dest )
01744 {
01745     return InterlockedExchangeAdd( dest, 1 ) + 1;
01746 }
01747
01748 extern inline LONG WINAPI InterlockedDecrement( PLONG dest );
01749 extern inline LONG WINAPI InterlockedDecrement( PLONG dest )
01750 {
01751     return InterlockedExchangeAdd( dest, -1 ) - 1;
01752 }
01753
01754 extern inline DWORD WINAPI GetLastError(void);
01755 extern inline DWORD WINAPI GetLastError(void)
01756 {
01757     DWORD ret;
01758     __asm__ __volatile__( ".byte 0x64\n\tmovl 0x60,%0" : "=r" (ret) );
01759     return ret;
01760 }
01761
01762 extern inline DWORD WINAPI GetCurrentProcessId(void);
01763 extern inline DWORD WINAPI GetCurrentProcessId(void)
01764 {
01765     DWORD ret;
01766     __asm__ __volatile__( ".byte 0x64\n\tmovl 0x20,%0" : "=r" (ret) );
01767     return ret;
01768 }
01769
01770 extern inline DWORD WINAPI GetCurrentThreadId(void);
01771 extern inline DWORD WINAPI GetCurrentThreadId(void)
01772 {
01773     DWORD ret;
01774     __asm__ __volatile__( ".byte 0x64\n\tmovl 0x24,%0" : "=r" (ret) );

```

```

01775     return ret;
01776 }
01777
01778 extern inline void WINAPI SetLastError( DWORD err );
01779 extern inline void WINAPI SetLastError( DWORD err )
01780 {
01781     __asm__ __volatile__( ".byte 0x64\n\tmovl %0,0x60" : : "r" (err) : "memory" );
01782 }
01783
01784 extern inline HANDLE WINAPI GetProcessHeap(void);
01785 extern inline HANDLE WINAPI GetProcessHeap(void)
01786 {
01787     HANDLE *pdb;
01788     __asm__ __volatile__( ".byte 0x64\n\tmovl 0x30,%0" : "=r" (pdb) );
01789     return pdb[0x18 / sizeof(HANDLE)]; /* get dword at offset 0x18 in pdb */
01790 }
01791
01792 #else /* __i386__ && __GNUC__ */
01793 DWORD WINAPI GetCurrentProcessId(void);
01794 DWORD WINAPI GetCurrentThreadId(void);
01795 DWORD WINAPI GetLastError(void);
01796 HANDLE WINAPI GetProcessHeap(void);
01797 LONG WINAPI InterlockedCompareExchange( LONG*, LONG, LONG );
01798 LONG WINAPI InterlockedDecrement( PLONG );
01799 LONG WINAPI InterlockedExchange( PLONG, LONG );
01800 LONG WINAPI InterlockedExchangeAdd( PLONG, LONG );
01801 LONG WINAPI InterlockedIncrement( PLONG );
01802 VOID WINAPI SetLastError( DWORD );
01803 #endif /* __i386__ && __GNUC__ */
01804
01805 /* FIXME: should handle platforms where sizeof(void*) != sizeof(long) */
01806 #if 0
01807     /* Unused in libEMF */
01808     static inline PVOID WINAPI InterlockedCompareExchangePointer( PVOID *dest, PVOID xchg, PVOID compare )
01809     {
01810         return (PVOID)InterlockedCompareExchange( (PLONG)dest, (LONG)xchg, (LONG)compare );
01811     }
01812
01813     static inline PVOID WINAPI InterlockedExchangePointer( PVOID *dest, PVOID val )
01814     {
01815         return (PVOID)InterlockedExchange( (PLONG)dest, (LONG)val );
01816     }
01817 #endif
01818 #ifdef __WINE__
01819 #define GetCurrentProcess() ((HANDLE)0xffffffff)
01820 #define GetCurrentThread() ((HANDLE)0xfffffffffe)
01821 #endif
01822
01823 /* WinMain(entry point) must be declared in winbase.h. */
01824 /* If this is not declared, we cannot compile many sources written with C++. */
01825 int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int);
01826
01827 #ifdef __cplusplus
01828 }
01829 #endif
01830
01831 #endif /* __WINE_WINBASE_H */

```

## 5.9 windef.h

```

00001 /*
00002  * Basic types definitions
00003  *
00004  * Copyright 1996 Alexandre Julliard
00005  */
00006
00007 #ifndef __WINE_WINDEF_H
00008 #define __WINE_WINDEF_H
00009
00010 #ifdef __WINE__
00011 # undef UNICODE
00012 #endif /* __WINE__ */
00013
00014 #define WINVER 0x0500
00015
00016 #include "winnt.h"
00017
00018
00019 #ifdef __cplusplus
00020 extern "C" {
00021 #endif
00022
00023
00024 /* Macros to map Winelib names to the correct implementation name */

```

```

00025 /* depending on __WINE__ and UNICODE macros. */
00026 /* Note that Winelib is purely Win32. */
00027
00028 #ifdef __WINE__
00029 # define WINELIB_NAME_AW(func) \
00030     func##_must_be_suffixed_with_W_or_A_in_this_context \
00031     func##_must_be_suffixed_with_W_or_A_in_this_context
00032 #else /* __WINE__ */
00033 # ifdef UNICODE
00034 #  define WINELIB_NAME_AW(func) func##W
00035 # else
00036 #  define WINELIB_NAME_AW(func) func##A
00037 # endif /* UNICODE */
00038 #endif /* __WINE__ */
00039
00040 #ifdef __WINE__
00041 # define DECL_WINELIB_TYPE_AW(type) /* nothing */
00042 #else /* __WINE__ */
00043 # define DECL_WINELIB_TYPE_AW(type) typedef WINELIB_NAME_AW(type) type;
00044 #endif /* __WINE__ */
00045
00046
00047 /* Integer types */
00048 typedef UINT WPARAM;
00049 typedef LONG LPARAM;
00050 typedef LONG LRESULT;
00051 typedef WORD ATOM;
00052 typedef WORD CATCHBUF[9];
00053 typedef WORD *LPCATCHBUF;
00054 typedef DWORD COLORREF, *LPCOLORREF;
00055
00056
00057 /* Handle types that exist both in Win16 and Win32. */
00058
00059 typedef int HFILE;
00060 DECLARE_OLD_HANDLE(HACCEL);
00061 DECLARE_OLD_HANDLE(HBITMAP);
00062 DECLARE_OLD_HANDLE(HBRUSH);
00063 DECLARE_HANDLE(HCOLORSPACE);
00064 DECLARE_OLD_HANDLE(HDC);
00065 DECLARE_HANDLE(HDESK);
00066 DECLARE_OLD_HANDLE(HENHMETAFILE);
00067 DECLARE_OLD_HANDLE(HFONT);
00068 DECLARE_OLD_HANDLE(HHOOK);
00069 DECLARE_OLD_HANDLE(HICON);
00070 DECLARE_OLD_HANDLE(HINSTANCE);
00071 DECLARE_OLD_HANDLE(HKEY);
00072 DECLARE_OLD_HANDLE(HKL);
00073 DECLARE_OLD_HANDLE(HMENU);
00074 DECLARE_OLD_HANDLE(HMETAFILE);
00075 DECLARE_OLD_HANDLE(HMONITOR);
00076 DECLARE_OLD_HANDLE(HPALETTE);
00077 DECLARE_OLD_HANDLE(HPEN);
00078 DECLARE_OLD_HANDLE(HRGN);
00079 DECLARE_OLD_HANDLE(HRSRC);
00080 DECLARE_OLD_HANDLE(HTASK);
00081 DECLARE_HANDLE(HWINSTA);
00082 DECLARE_OLD_HANDLE(HWND);
00083
00084 /* Handle types that must remain interchangeable even with strict on */
00085
00086 typedef HINSTANCE HMODULE;
00087 typedef HANDLE HGDIOBJ;
00088 typedef HANDLE HGLOBAL;
00089 typedef HANDLE HLOCAL;
00090 typedef HANDLE GLOBALHANDLE;
00091 typedef HANDLE LOCALHANDLE;
00092 typedef HICON HCURSOR;
00093
00094 /* Callback function pointers types */
00095
00096 typedef INT CALLBACK (*FARPROC)();
00097 typedef INT CALLBACK (*PROC)();
00098
00099
00100 /* Macros to split words and longs. */
00101
00102 #define LOBYTE(w) ((BYTE)(WORD)(w))
00103 #define HIBYTE(w) ((BYTE)((WORD)(w) >> 8))
00104
00105 #define LOWORD(l) ((WORD)(DWORD)(l))
00106 #define HIWORD(l) ((WORD)((DWORD)(l) >> 16))
00107
00108 #define SLOWORD(l) ((SHORT)(LONG)(l))
00109 #define SHIWORD(l) ((SHORT)((LONG)(l) >> 16))
00110
00111 #define MAKEWORD(low,high) ((WORD)((BYTE)(low) | ((WORD)((BYTE)(high))) << 8))

```

```

00112 #define MAKELONG(low,high)      ((LONG)((WORD)(low)) | ((DWORD)((WORD)(high))) << 16))
00113 #define MAKELPARAM(low,high)    ((LPARAM)MAKELONG(low,high))
00114 #define MAKEWPARAM(low,high)    ((WPARAM)MAKELONG(low,high))
00115 #define MAKELRESULT(low,high)   ((LRESULT)MAKELONG(low,high))
00116
00117 #define SELECTOROF(ptr)          (HIWORD(ptr))
00118 #define OFFSETOF(ptr)            (LOWORD(ptr))
00119
00120 #ifdef __WINE__
00121 /* macros to set parts of a DWORD (not in the Windows API) */
00122 #define SET_LOWORD(dw,val)       ((dw) = ((dw) & 0xffff0000) | LOWORD(val))
00123 #define SET_LOBYTE(dw,val)       ((dw) = ((dw) & 0xffffff00) | LOBYTE(val))
00124 #define SET_HIBYTE(dw,val)       ((dw) = ((dw) & 0xffff00ff) | (LOBYTE(val) << 8))
00125 #define ADD_LOWORD(dw,val)       ((dw) = ((dw) & 0xffff0000) | LOWORD((DWORD)(dw)+(val)))
00126 #endif
00127
00128 /* min and max macros */
00129 #ifndef NOMINMAX
00130 #ifndef max
00131 #define max(a,b)      (((a) > (b)) ? (a) : (b))
00132 #endif
00133 #ifndef min
00134 #define min(a,b)      (((a) < (b)) ? (a) : (b))
00135 #endif
00136 #endif /* NOMINMAX */
00137
00138 #ifndef _MAX_PATH
00139 /* FIXME: These are supposed to be in stdlib.h only */
00140 #define _MAX_DRIVE      3
00141 #define _MAX_FNAME      256
00142 #define _MAX_DIR         _MAX_FNAME
00143 #define _MAX_EXT         _MAX_FNAME
00144 #define _MAX_PATH       260
00145 #endif
00146 #define MAX_PATH         _MAX_PATH
00147
00148
00149 #define HFILE_ERROR      ((HFILE)-1)
00150
00151 /* The SIZE structure */
00152 typedef struct tagSIZE
00153 {
00154     LONG cx;
00155     LONG cy;
00156 } SIZE, *PSIZE, *LPSIZE;
00157
00158 typedef SIZE SIZEL, *PSIZEL, *LPSIZEL;
00159
00160 /* The POINT structure */
00161 typedef struct tagPOINT
00162 {
00163     LONG x;
00164     LONG y;
00165 } POINT, *PPOINT, *LPPOINT;
00166
00167 typedef struct _POINTL
00168 {
00169     LONG x;
00170     LONG y;
00171 } POINTL;
00172
00173 /* The POINTS structure */
00174
00175 typedef struct tagPOINTS
00176 {
00177     SHORT x;
00178     SHORT y;
00179 } POINTS, *PPOINTS, *LPPOINTS;
00180
00181 /* The RECT structure */
00182 typedef struct tagRECT
00183 {
00184     INT left;
00185     INT top;
00186     INT right;
00187     INT bottom;
00188 } RECT, *PRECT, *LPRECT;
00189 typedef const RECT *LPCRECT;
00190
00191
00192 typedef struct tagRECTL
00193 {
00194     LONG left;
00195     LONG top;
00196     LONG right;
00197     LONG bottom;
00198 } RECTL, *PRECTL, *LPRECTL;

```

```

00199
00200 typedef const RECTL *LPCRECTL;
00201
00202 #ifdef __cplusplus
00203 }
00204 #endif
00205
00206 #endif /* __WINE_WINDEF_H */

```

## 5.10 winerror.h

```

00001 #ifndef __WINE_WINERROR_H
00002 #define __WINE_WINERROR_H
00003
00004
00005 extern int WIN32_LastError;
00006
00007 #define FACILITY_NULL      0
00008 #define FACILITY_RPC      1
00009 #define FACILITY_DISPATCH 2
00010 #define FACILITY_STORAGE  3
00011 #define FACILITY_ITF      4
00012 #define FACILITY_WIN32    7
00013 #define FACILITY_WINDOWS  8
00014 #define FACILITY_SSPI      9
00015 #define FACILITY_CONTROL  10
00016 #define FACILITY_CERT     11
00017 #define FACILITY_INTERNET 12
00018
00019 #define SEVERITY_SUCCESS   0
00020 #define SEVERITY_ERROR     1
00021
00022
00023 #define MAKE_HRESULT(sev,fac,code) \
00024     ((HRESULT) (((unsigned long)(sev)<31) | ((unsigned long)(fac)<16) | ((unsigned long)(code))) )
00025 #define MAKE_SCODE(sev,fac,code) \
00026     ((SCODE) (((unsigned long)(sev)<31) | ((unsigned long)(fac)<16) | ((unsigned long)(code))) )
00027 #define SUCCEEDED(stat) ((HRESULT)(stat)>=0)
00028 #define FAILED(stat) ((HRESULT)(stat)<0)
00029 #define IS_ERROR(stat) (((unsigned long)(stat)>31) == SEVERITY_ERROR)
00030
00031 #define HRESULT_CODE(hr) ((hr) & 0xFFFF)
00032 #define SCODE_CODE(sc) ((sc) & 0xFFFF)
00033
00034 #define HRESULT_FACILITY(hr) (((hr) >> 16) & 0x1FFF)
00035 #define SCODE_FACILITY(sc) (((sc) >> 16) & 0x1FFF)
00036
00037 #define HRESULT_SEVERITY(hr) (((hr) >> 31) & 0x1)
00038 #define SCODE_SEVERITY(sc) (((sc) >> 31) & 0x1)
00039
00040 #define FACILITY_NT_BIT      0x10000000
00041 #define HRESULT_FROM_WIN32(x) ((x) ? ((HRESULT) (((x) & 0x0000FFFF) | (FACILITY_WIN32 << 16) | 0x80000000)) : 0)
00042 #define HRESULT_FROM_NT(x) ((HRESULT) ((x) | FACILITY_NT_BIT))
00043
00044 /* SCODE <-> HRESULT functions */
00045 /* This macros is obsolete and should not be used in new apps. */
00046 #define GetSCode(hr) ((SCODE)(hr))
00047 /* This macros is obsolete and should not be used in new apps. */
00048 #define ResultFromSCode(sc) ((HRESULT)(sc))
00049
00050 /* ERROR_UNKNOWN is a placeholder for error conditions which haven't
00051  * been tested yet so we're not exactly sure what will be returned.
00052  * All instances of ERROR_UNKNOWN should be tested under Win95/NT
00053  * and replaced.
00054  */
00055 #define ERROR_UNKNOWN          99999
00056
00057 #define NO_ERROR                0
00058 #define ERROR_SUCCESS           0
00059 #define ERROR_INVALID_FUNCTION  1
00060 #define ERROR_FILE_NOT_FOUND    2
00061 #define ERROR_PATH_NOT_FOUND    3
00062 #define ERROR_TOO_MANY_OPEN_FILES 4
00063 #define ERROR_ACCESS_DENIED     5
00064 #define ERROR_INVALID_HANDLE    6
00065 #define ERROR_ARENA_TRASHED     7
00066 #define ERROR_NOT_ENOUGH_MEMORY 8
00067 #define ERROR_INVALID_BLOCK     9
00068 #define ERROR_BAD_ENVIRONMENT  10
00069 #define ERROR_BAD_FORMAT        11
00070 #define ERROR_INVALID_ACCESS    12
00071 #define ERROR_INVALID_DATA      13
00072 #define ERROR_OUTOFMEMORY       14

```

```
00073 #define ERROR_INVALID_DRIVE 15
00074 #define ERROR_CURRENT_DIRECTORY 16
00075 #define ERROR_NOT_SAME_DEVICE 17
00076 #define ERROR_NO_MORE_FILES 18
00077 #define ERROR_WRITE_PROTECT 19
00078 #define ERROR_BAD_UNIT 20
00079 #define ERROR_NOT_READY 21
00080 #define ERROR_BAD_COMMAND 22
00081 #define ERROR_CRC 23
00082 #define ERROR_BAD_LENGTH 24
00083 #define ERROR_SEEK 25
00084 #define ERROR_NOT_DOS_DISK 26
00085 #define ERROR_SECTOR_NOT_FOUND 27
00086 #define ERROR_OUT_OF_PAPER 28
00087 #define ERROR_WRITE_FAULT 29
00088 #define ERROR_READ_FAULT 30
00089 #define ERROR_GEN_FAILURE 31
00090 #define ERROR_SHARING_VIOLATION 32
00091 #define ERROR_LOCK_VIOLATION 33
00092 #define ERROR_WRONG_DISK 34
00093 /* FIXME: 35 gets returned for some unsuccessful DeviceIoControl calls */
00094 #define ERROR_UNKNOWN_NAME_01 35
00095 #define ERROR_SHARING_BUFFER_EXCEEDED 36
00096 #define ERROR_HANDLE_EOF 38
00097 #define ERROR_HANDLE_DISK_FULL 39
00098 #define ERROR_NOT_SUPPORTED 50
00099 #define ERROR_REM_NOT_LIST 51
00100 #define ERROR_DUP_NAME 52
00101 #define ERROR_BAD_NETPATH 53
00102 #define ERROR_NETWORK_BUSY 54
00103 #define ERROR_DEV_NOT_EXIST 55
00104 #define ERROR_TOO_MANY_CMDS 56
00105 #define ERROR_ADAP_HDW_ERR 57
00106 #define ERROR_BAD_NET_RESP 58
00107 #define ERROR_UNEXP_NET_ERR 59
00108 #define ERROR_BAD_REM_ADAP 60
00109 #define ERROR_PRINTQ_FULL 61
00110 #define ERROR_NO_SPOOL_SPACE 62
00111 #define ERROR_PRINT_CANCELLED 63
00112 #define ERROR_NETNAME_DELETED 64
00113 #define ERROR_NETWORK_ACCESS_DENIED 65
00114 #define ERROR_BAD_DEV_TYPE 66
00115 #define ERROR_BAD_NET_NAME 67
00116 #define ERROR_TOO_MANY_NAMES 68
00117 #define ERROR_TOO_MANY_SESS 69
00118 #define ERROR_SHARING_PAUSED 70
00119 #define ERROR_REQ_NOT_ACCEP 71
00120 #define ERROR_REDIR_PAUSED 72
00121 #define ERROR_FILE_EXISTS 80
00122 #define ERROR_CANNOT_MAKE 82
00123 #define ERROR_FAIL_I24 83
00124 #define ERROR_OUT_OF_STRUCTURES 84
00125 #define ERROR_ALREADY_ASSIGNED 85
00126 #define ERROR_INVALID_PASSWORD 86
00127 #define ERROR_INVALID_PARAMETER 87
00128 #define ERROR_NET_WRITE_FAULT 88
00129 #define ERROR_NO_PROC_SLOTS 89
00130 #define ERROR_TOO_MANY_SEMAPHORES 100
00131 #define ERROR_EXCL_SEM_ALREADY_OWNED 101
00132 #define ERROR_SEM_IS_SET 102
00133 #define ERROR_TOO_MANY_SEM_REQUESTS 103
00134 #define ERROR_INVALID_AT_INTERRUPT_TIME 104
00135 #define ERROR_SEM_OWNER_DIED 105
00136 #define ERROR_SEM_USER_LIMIT 106
00137 #define ERROR_DISK_CHANGE 107
00138 #define ERROR_DRIVE_LOCKED 108
00139 #define ERROR_BROKEN_PIPE 109
00140 #define ERROR_OPEN_FAILED 110
00141 #define ERROR_BUFFER_OVERFLOW 111
00142 #define ERROR_DISK_FULL 112
00143 #define ERROR_NO_MORE_SEARCH_HANDLES 113
00144 #define ERROR_INVALID_TARGET_HANDLE 114
00145 #define ERROR_INVALID_CATEGORY 117
00146 #define ERROR_INVALID_VERIFY_SWITCH 118
00147 #define ERROR_BAD_DRIVER_LEVEL 119
00148 #define ERROR_CALL_NOT_IMPLEMENTED 120
00149 #define ERROR_SEM_TIMEOUT 121
00150 #define ERROR_INSUFFICIENT_BUFFER 122
00151 #define ERROR_INVALID_NAME 123
00152 #define ERROR_INVALID_LEVEL 124
00153 #define ERROR_NO_VOLUME_LABEL 125
00154 #define ERROR_MOD_NOT_FOUND 126
00155 #define ERROR_PROC_NOT_FOUND 127
00156 #define ERROR_WAIT_NO_CHILDREN 128
00157 #define ERROR_CHILD_NOT_COMPLETE 129
00158 #define ERROR_DIRECT_ACCESS_HANDLE 130
00159 #define ERROR_NEGATIVE_SEEK 131
```

00160	#define	ERROR_SEEK_ON_DEVICE	132
00161	#define	ERROR_IS_JOIN_TARGET	133
00162	#define	ERROR_IS_JOINED	134
00163	#define	ERROR_IS_SUBSTED	135
00164	#define	ERROR_NOT_JOINED	136
00165	#define	ERROR_NOT_SUBSTED	137
00166	#define	ERROR_JOIN_TO_JOIN	138
00167	#define	ERROR_SUBST_TO_SUBST	139
00168	#define	ERROR_JOIN_TO_SUBST	140
00169	#define	ERROR_SUBST_TO_JOIN	141
00170	#define	ERROR_BUSY_DRIVE	142
00171	#define	ERROR_SAME_DRIVE	143
00172	#define	ERROR_DIR_NOT_ROOT	144
00173	#define	ERROR_DIR_NOT_EMPTY	145
00174	#define	ERROR_IS_SUBST_PATH	146
00175	#define	ERROR_IS_JOIN_PATH	147
00176	#define	ERROR_PATH_BUSY	148
00177	#define	ERROR_IS_SUBST_TARGET	149
00178	#define	ERROR_SYSTEM_TRACE	150
00179	#define	ERROR_INVALID_EVENT_COUNT	151
00180	#define	ERROR_TOO_MANY_MUXWAITERS	152
00181	#define	ERROR_INVALID_LIST_FORMAT	153
00182	#define	ERROR_LABEL_TOO_LONG	154
00183	#define	ERROR_TOO_MANY_TCBS	155
00184	#define	ERROR_SIGNAL_REFUSED	156
00185	#define	ERROR_DISCARDED	157
00186	#define	ERROR_NOT_LOCKED	158
00187	#define	ERROR_BAD_THREADID_ADDR	159
00188	#define	ERROR_BAD_ARGUMENTS	160
00189	#define	ERROR_BAD_PATHNAME	161
00190	#define	ERROR_SIGNAL_PENDING	162
00191	#define	ERROR_MAX_THRDS_REACHED	164
00192	#define	ERROR_LOCK_FAILED	167
00193	#define	ERROR_BUSY	170
00194	#define	ERROR_CANCEL_VIOLATION	173
00195	#define	ERROR_ATOMIC_LOCKS_NOT_SUPPORTED	174
00196	#define	ERROR_INVALID_SEGMENT_NUMBER	180
00197	#define	ERROR_INVALID_ORDINAL	182
00198	#define	ERROR_ALREADY_EXISTS	183
00199	#define	ERROR_INVALID_FLAG_NUMBER	186
00200	#define	ERROR_SEM_NOT_FOUND	187
00201	#define	ERROR_INVALID_STARTING_CODESEG	188
00202	#define	ERROR_INVALID_STACKSEG	189
00203	#define	ERROR_INVALID_MODULETYPE	190
00204	#define	ERROR_INVALID_EXE_SIGNATURE	191
00205	#define	ERROR_EXE_MARKED_INVALID	192
00206	#define	ERROR_BAD_EXE_FORMAT	193
00207	#define	ERROR_ITERATED_DATA_EXCEEDS_64k	194
00208	#define	ERROR_INVALID_MINALLOCSIZE	195
00209	#define	ERROR_DYNLINK_FROM_INVALID_RING	196
00210	#define	ERROR_IOPL_NOT_ENABLED	197
00211	#define	ERROR_INVALID_SEGDP1	198
00212	#define	ERROR_AUTODATASEG_EXCEEDS_64k	199
00213	#define	ERROR_RING2SEG_MUST_BE_MOVABLE	200
00214	#define	ERROR_RELOC_CHAIN_XEEDS_SEGLIM	201
00215	#define	ERROR_INFLOOP_IN_RELOC_CHAIN	202
00216	#define	ERROR_ENVVAR_NOT_FOUND	203
00217	#define	ERROR_NO_SIGNAL_SENT	205
00218	#define	ERROR_FILENAME_EXCED_RANGE	206
00219	#define	ERROR_RING2_STACK_IN_USE	207
00220	#define	ERROR_META_EXPANSION_TOO_LONG	208
00221	#define	ERROR_INVALID_SIGNAL_NUMBER	209
00222	#define	ERROR_THREAD_1_INACTIVE	210
00223	#define	ERROR_LOCKED	212
00224	#define	ERROR_TOO_MANY_MODULES	214
00225	#define	ERROR_NESTING_NOT_ALLOWED	215
00226	#define	ERROR_EXE_MACHINE_TYPE_MISMATCH	216
00227	#define	ERROR_BAD_PIPE	230
00228	#define	ERROR_PIPE_BUSY	231
00229	#define	ERROR_NO_DATA	232
00230	#define	ERROR_PIPE_NOT_CONNECTED	233
00231	#define	ERROR_MORE_DATA	234
00232	#define	ERROR_VC_DISCONNECTED	240
00233	#define	ERROR_INVALID_EA_NAME	254
00234	#define	ERROR_EA_LIST_INCONSISTENT	255
00235	#define	ERROR_NO_MORE_ITEMS	259
00236	#define	ERROR_CANNOT_COPY	266
00237	#define	ERROR_DIRECTORY	267
00238	#define	ERROR_EAS_DIDNT_FIT	275
00239	#define	ERROR_EA_FILE_CORRUPT	276
00240	#define	ERROR_EA_TABLE_FULL	277
00241	#define	ERROR_INVALID_EA_HANDLE	278
00242	#define	ERROR_EAS_NOT_SUPPORTED	282
00243	#define	ERROR_NOT_OWNER	288
00244	#define	ERROR_TOO_MANY_POSTS	298
00245	#define	ERROR_PARTIAL_COPY	299
00246	#define	ERROR_OPLOCK_NOT_GRANTED	300

00247	#define	ERROR_INVALID_OPLock_PROTOCOL	301
00248	#define	ERROR_MR_MID_NOT_FOUND	317
00249	#define	ERROR_INVALID_ADDRESS	487
00250	#define	ERROR_ARITHMETIC_OVERFLOW	534
00251	#define	ERROR_PIPE_CONNECTED	535
00252	#define	ERROR_PIPE_LISTENING	536
00253	#define	ERROR_EA_ACCESS_DENIED	994
00254	#define	ERROR_OPERATION_ABORTED	995
00255	#define	ERROR_IO_INCOMPLETE	996
00256	#define	ERROR_IO_PENDING	997
00257	#define	ERROR_NOACCESS	998
00258	#define	ERROR_SWAPERROR	999
00259	#define	ERROR_STACK_OVERFLOW	1001
00260	#define	ERROR_INVALID_MESSAGE	1002
00261	#define	ERROR_CAN_NOT_COMPLETE	1003
00262	#define	ERROR_INVALID_FLAGS	1004
00263	#define	ERROR_UNRECOGNIZED_VOLUME	1005
00264	#define	ERROR_FILE_INVALID	1006
00265	#define	ERROR_FULLSCREEN_MODE	1007
00266	#define	ERROR_NO_TOKEN	1008
00267	#define	ERROR_BADDB	1009
00268	#define	ERROR_BADKEY	1010
00269	#define	ERROR_CANTOPEN	1011
00270	#define	ERROR_CANTREAD	1012
00271	#define	ERROR_CANTWRITE	1013
00272	#define	ERROR_REGISTRY_RECOVERED	1014
00273	#define	ERROR_REGISTRY_CORRUPT	1015
00274	#define	ERROR_REGISTRY_IO_FAILED	1016
00275	#define	ERROR_NOT_REGISTRY_FILE	1017
00276	#define	ERROR_KEY_DELETED	1018
00277	#define	ERROR_NO_LOG_SPACE	1019
00278	#define	ERROR_KEY_HAS_CHILDREN	1020
00279	#define	ERROR_CHILD_MUST_BE_VOLATILE	1021
00280	#define	ERROR_NOTIFY_ENUM_DIR	1022
00281	#define	ERROR_DEPENDENT_SERVICES_RUNNING	1051
00282	#define	ERROR_INVALID_SERVICE_CONTROL	1052
00283	#define	ERROR_SERVICE_REQUEST_TIMEOUT	1053
00284	#define	ERROR_SERVICE_NO_THREAD	1054
00285	#define	ERROR_SERVICE_DATABASE_LOCKED	1055
00286	#define	ERROR_SERVICE_ALREADY_RUNNING	1056
00287	#define	ERROR_INVALID_SERVICE_ACCOUNT	1057
00288	#define	ERROR_SERVICE_DISABLED	1058
00289	#define	ERROR_CIRCULAR_DEPENDENCY	1059
00290	#define	ERROR_SERVICE_DOES_NOT_EXIST	1060
00291	#define	ERROR_SERVICE_CANNOT_ACCEPT_CTRL	1061
00292	#define	ERROR_SERVICE_NOT_ACTIVE	1062
00293	#define	ERROR_FAILED_SERVICE_CONTROLLER_CONNECT	1063
00294	#define	ERROR_EXCEPTION_IN_SERVICE	1064
00295	#define	ERROR_DATABASE_DOES_NOT_EXIST	1065
00296	#define	ERROR_SERVICE_SPECIFIC_ERROR	1066
00297	#define	ERROR_PROCESS_ABORTED	1067
00298	#define	ERROR_SERVICE_DEPENDENCY_FAIL	1068
00299	#define	ERROR_SERVICE_LOGON_FAILED	1069
00300	#define	ERROR_SERVICE_START_HANG	1070
00301	#define	ERROR_INVALID_SERVICE_LOCK	1071
00302	#define	ERROR_SERVICE_MARKED_FOR_DELETE	1072
00303	#define	ERROR_SERVICE_EXISTS	1073
00304	#define	ERROR_ALREADY_RUNNING_LKG	1074
00305	#define	ERROR_SERVICE_DEPENDENCY_DELETED	1075
00306	#define	ERROR_BOOT_ALREADY_ACCEPTED	1076
00307	#define	ERROR_SERVICE_NEVER_STARTED	1077
00308	#define	ERROR_DUPLICATE_SERVICE_NAME	1078
00309	#define	ERROR_DIFFERENT_SERVICE_ACCOUNT	1079
00310	#define	ERROR_CANNOT_DETECT_DRIVER_FAILURE	1080
00311	#define	ERROR_CANNOT_DETECT_PROCESS_ABORT	1081
00312	#define	ERROR_NO_RECOVERY_PROGRAM	1082
00313	#define	ERROR_SERVICE_NOT_IN_EXE	1083
00314	#define	ERROR_END_OF_MEDIA	1100
00315	#define	ERROR_FILEMARK_DETECTED	1101
00316	#define	ERROR_BEGINNING_OF_MEDIA	1102
00317	#define	ERROR_SETMARK_DETECTED	1103
00318	#define	ERROR_NO_DATA_DETECTED	1104
00319	#define	ERROR_PARTITION_FAILURE	1105
00320	#define	ERROR_INVALID_BLOCK_LENGTH	1106
00321	#define	ERROR_DEVICE_NOT_PARTITIONED	1107
00322	#define	ERROR_UNABLE_TO_LOCK_MEDIA	1108
00323	#define	ERROR_UNABLE_TO_UNLOAD_MEDIA	1109
00324	#define	ERROR_MEDIA_CHANGED	1110
00325	#define	ERROR_BUS_RESET	1111
00326	#define	ERROR_NO_MEDIA_IN_DRIVE	1112
00327	#define	ERROR_NO_UNICODE_TRANSLATION	1113
00328	#define	ERROR_DLL_INIT_FAILED	1114
00329	#define	ERROR_SHUTDOWN_IN_PROGRESS	1115
00330	#define	ERROR_NO_SHUTDOWN_IN_PROGRESS	1116
00331	#define	ERROR_IO_DEVICE	1117
00332	#define	ERROR_SERIAL_NO_DEVICE	1118
00333	#define	ERROR_IRQ_BUSY	1119

00334	#define	ERROR_MORE_WRITES	1120
00335	#define	ERROR_COUNTER_TIMEOUT	1121
00336	#define	ERROR_FLOPPY_ID_MARK_NOT_FOUND	1122
00337	#define	ERROR_FLOPPY_WRONG_CYLINDER	1123
00338	#define	ERROR_FLOPPY_UNKNOWN_ERROR	1124
00339	#define	ERROR_FLOPPY_BAD_REGISTERS	1125
00340	#define	ERROR_DISK_RECALIBRATE_FAILED	1126
00341	#define	ERROR_DISK_OPERATION_FAILED	1127
00342	#define	ERROR_DISK_RESET_FAILED	1128
00343	#define	ERROR_EOM_OVERFLOW	1129
00344	#define	ERROR_NOT_ENOUGH_SERVER_MEMORY	1130
00345	#define	ERROR_POSSIBLE_DEADLOCK	1131
00346	#define	ERROR_MAPPED_ALIGNMENT	1132
00347	#define	ERROR_SET_POWER_STATE_VETOED	1140
00348	#define	ERROR_SET_POWER_STATE_FAILED	1141
00349	#define	ERROR_TOO_MANY_LINKS	1142
00350	#define	ERROR_OLD_WIN_VERSION	1150
00351	#define	ERROR_APP_WRONG_OS	1151
00352	#define	ERROR_SINGLE_INSTANCE_APP	1152
00353	#define	ERROR_RMODE_APP	1153
00354	#define	ERROR_INVALID_DLL	1154
00355	#define	ERROR_NO_ASSOCIATION	1155
00356	#define	ERROR_DDE_FAIL	1156
00357	#define	ERROR_DLL_NOT_FOUND	1157
00358	#define	ERROR_NO_MORE_USER_HANDLES	1158
00359	#define	ERROR_MESSAGE_SYNC_ONLY	1159
00360	#define	ERROR_SOURCE_ELEMENT_EMPTY	1160
00361	#define	ERROR_DESTINATION_ELEMENT_FULL	1161
00362	#define	ERROR_ILLEGAL_ELEMENT_ADDRESS	1162
00363	#define	ERROR_MAGAZINE_NOT_PRESENT	1163
00364	#define	ERROR_DEVICE_REINITIALIZATION_NEEDED	1164
00365	#define	ERROR_DEVICE_REQUIRES_CLEANING	1165
00366	#define	ERROR_DEVICE_DOOR_OPEN	1166
00367	#define	ERROR_DEVICE_NOT_CONNECTED	1167
00368	#define	ERROR_NOT_FOUND	1168
00369	#define	ERROR_NO_MATCH	1169
00370	#define	ERROR_SET_NOT_FOUND	1170
00371	#define	ERROR_POINT_NOT_FOUND	1171
00372	#define	ERROR_NO_TRACKING_SERVICE	1172
00373	#define	ERROR_NO_VOLUME_ID	1173
00374	#define	ERROR_UNABLE_TO_REMOVE_REPLACED	1175
00375	#define	ERROR_UNABLE_TO_MOVE_REPLACEMENT	1176
00376	#define	ERROR_UNABLE_TO_MOVE_REPLACEMENT_2	1177
00377	#define	ERROR_JOURNAL_DELETE_IN_PROGRESS	1178
00378	#define	ERROR_JOURNAL_NOT_ACTIVE	1179
00379	#define	ERROR_POTENTIAL_FILE_FOUND	1180
00380	#define	ERROR_JOURNAL_ENTRY_DELETED	1181
00381	#define	ERROR_BAD_DEVICE	1200
00382	#define	ERROR_CONNECTION_UNAVAIL	1201
00383	#define	ERROR_DEVICE_ALREADY_REMEMBERED	1202
00384	#define	ERROR_NO_NET_OR_BAD_PATH	1203
00385	#define	ERROR_BAD_PROVIDER	1204
00386	#define	ERROR_CANNOT_OPEN_PROFILE	1205
00387	#define	ERROR_BAD_PROFILE	1206
00388	#define	ERROR_NOT_CONTAINER	1207
00389	#define	ERROR_EXTENDED_ERROR	1208
00390	#define	ERROR_INVALID_GROUPNAME	1209
00391	#define	ERROR_INVALID_COMPUTERNAME	1210
00392	#define	ERROR_INVALID_EVENTNAME	1211
00393	#define	ERROR_INVALID_DOMAINNAME	1212
00394	#define	ERROR_INVALID_SERVICENAME	1213
00395	#define	ERROR_INVALID_NETNAME	1214
00396	#define	ERROR_INVALID_SHARENAME	1215
00397	#define	ERROR_INVALID_PASSWORDNAME	1216
00398	#define	ERROR_INVALID_MESSAGE_NAME	1217
00399	#define	ERROR_INVALID_MESSAGEDEST	1218
00400	#define	ERROR_SESSION_CREDENTIAL_CONFLICT	1219
00401	#define	ERROR_REMOTE_SESSION_LIMIT_EXCEEDED	1220
00402	#define	ERROR_DUP_DOMAINNAME	1221
00403	#define	ERROR_NO_NETWORK	1222
00404	#define	ERROR_CANCELLED	1223
00405	#define	ERROR_USER_MAPPED_FILE	1224
00406	#define	ERROR_CONNECTION_REFUSED	1225
00407	#define	ERROR_GRACEFUL_DISCONNECT	1226
00408	#define	ERROR_ADDRESS_ALREADY_ASSOCIATED	1227
00409	#define	ERROR_ADDRESS_NOT_ASSOCIATED	1228
00410	#define	ERROR_CONNECTION_INVALID	1229
00411	#define	ERROR_CONNECTION_ACTIVE	1230
00412	#define	ERROR_NETWORK_UNREACHABLE	1231
00413	#define	ERROR_HOST_UNREACHABLE	1232
00414	#define	ERROR_PROTOCOL_UNREACHABLE	1233
00415	#define	ERROR_PORT_UNREACHABLE	1234
00416	#define	ERROR_REQUEST_ABORTED	1235
00417	#define	ERROR_CONNECTION_ABORTED	1236
00418	#define	ERROR_RETRY	1237
00419	#define	ERROR_CONNECTION_COUNT_LIMIT	1238
00420	#define	ERROR_LOGIN_TIME_RESTRICTION	1239

00421	#define	ERROR_LOGIN_WKSTA_RESTRICTION	1240
00422	#define	ERROR_INCORRECT_ADDRESS	1241
00423	#define	ERROR_ALREADY_REGISTERED	1242
00424	#define	ERROR_SERVICE_NOT_FOUND	1243
00425	#define	ERROR_NOT_AUTHENTICATED	1244
00426	#define	ERROR_NOT_LOGGED_ON	1245
00427	#define	ERROR_CONTINUE	1246
00428	#define	ERROR_ALREADY_INITIALIZED	1247
00429	#define	ERROR_NO_MORE_DEVICES	1248
00430	#define	ERROR_NO_SUCH_SITE	1249
00431	#define	ERROR_DOMAIN_CONTROLLER_EXISTS	1250
00432	#define	ERROR_ONLY_IF_CONNECTED	1251
00433	#define	ERROR_OVERRIDE_NOCHANGES	1252
00434	#define	ERROR_BAD_USER_PROFILE	1253
00435	#define	ERROR_NOT_SUPPORTED_ON_SBS	1254
00436	#define	ERROR_NOT_ALL_ASSIGNED	1300
00437	#define	ERROR_SOME_NOT_MAPPED	1301
00438	#define	ERROR_NO_QUOTAS_FOR_ACCOUNT	1302
00439	#define	ERROR_LOCAL_USER_SESSION_KEY	1303
00440	#define	ERROR_NULL_IM_PASSWORD	1304
00441	#define	ERROR_UNKNOWN_REVISION	1305
00442	#define	ERROR_REVISION_MISMATCH	1306
00443	#define	ERROR_INVALID_OWNER	1307
00444	#define	ERROR_INVALID_PRIMARY_GROUP	1308
00445	#define	ERROR_NO_IMPERSONATION_TOKEN	1309
00446	#define	ERROR_CANT_DISABLE_MANDATORY	1310
00447	#define	ERROR_NO_LOGON_SERVERS	1311
00448	#define	ERROR_NO_SUCH_LOGON_SESSION	1312
00449	#define	ERROR_NO_SUCH_PRIVILEGE	1313
00450	#define	ERROR_PRIVILEGE_NOT_HELD	1314
00451	#define	ERROR_INVALID_ACCOUNT_NAME	1315
00452	#define	ERROR_USER_EXISTS	1316
00453	#define	ERROR_NO_SUCH_USER	1317
00454	#define	ERROR_GROUP_EXISTS	1318
00455	#define	ERROR_NO_SUCH_GROUP	1319
00456	#define	ERROR_MEMBER_IN_GROUP	1320
00457	#define	ERROR_MEMBER_NOT_IN_GROUP	1321
00458	#define	ERROR_LAST_ADMIN	1322
00459	#define	ERROR_WRONG_PASSWORD	1323
00460	#define	ERROR_ILL_FORMED_PASSWORD	1324
00461	#define	ERROR_PASSWORD_RESTRICTION	1325
00462	#define	ERROR_LOGON_FAILURE	1326
00463	#define	ERROR_ACCOUNT_RESTRICTION	1327
00464	#define	ERROR_INVALID_LOGON_HOURS	1328
00465	#define	ERROR_INVALID_WORKSTATION	1329
00466	#define	ERROR_PASSWORD_EXPIRED	1330
00467	#define	ERROR_ACCOUNT_DISABLED	1331
00468	#define	ERROR_NONE_MAPPED	1332
00469	#define	ERROR_TOO_MANY_LUIDS_REQUESTED	1333
00470	#define	ERROR_LUIDS_EXHAUSTED	1334
00471	#define	ERROR_INVALID_SUB_AUTHORITY	1335
00472	#define	ERROR_INVALID_ACL	1336
00473	#define	ERROR_INVALID_SID	1337
00474	#define	ERROR_INVALID_SECURITY_DESCR	1338
00475	#define	ERROR_BAD_INHERITANCE_ACL	1340
00476	#define	ERROR_SERVER_DISABLED	1341
00477	#define	ERROR_SERVER_NOT_DISABLED	1342
00478	#define	ERROR_INVALID_ID_AUTHORITY	1343
00479	#define	ERROR_ALLOTTED_SPACE_EXCEEDED	1344
00480	#define	ERROR_INVALID_GROUP_ATTRIBUTES	1345
00481	#define	ERROR_BAD_IMPERSONATION_LEVEL	1346
00482	#define	ERROR_CANT_OPEN_ANONYMOUS	1347
00483	#define	ERROR_BAD_VALIDATION_CLASS	1348
00484	#define	ERROR_BAD_TOKEN_TYPE	1349
00485	#define	ERROR_NO_SECURITY_ON_OBJECT	1350
00486	#define	ERROR_CANT_ACCESS_DOMAIN_INFO	1351
00487	#define	ERROR_INVALID_SERVER_STATE	1352
00488	#define	ERROR_INVALID_DOMAIN_STATE	1353
00489	#define	ERROR_INVALID_DOMAIN_ROLE	1354
00490	#define	ERROR_NO_SUCH_DOMAIN	1355
00491	#define	ERROR_DOMAIN_EXISTS	1356
00492	#define	ERROR_DOMAIN_LIMIT_EXCEEDED	1357
00493	#define	ERROR_INTERNAL_DB_CORRUPTION	1358
00494	#define	ERROR_INTERNAL_ERROR	1359
00495	#define	ERROR_GENERIC_NOT_MAPPED	1360
00496	#define	ERROR_BAD_DESCRIPTOR_FORMAT	1361
00497	#define	ERROR_NOT_LOGON_PROCESS	1362
00498	#define	ERROR_LOGON_SESSION_EXISTS	1363
00499	#define	ERROR_NO_SUCH_PACKAGE	1364
00500	#define	ERROR_BAD_LOGON_SESSION_STATE	1365
00501	#define	ERROR_LOGON_SESSION_COLLISION	1366
00502	#define	ERROR_INVALID_LOGON_TYPE	1367
00503	#define	ERROR_CANTNOT_IMPERSONATE	1368
00504	#define	ERROR_RXACT_INVALID_STATE	1369
00505	#define	ERROR_RXACT_COMMIT_FAILURE	1370
00506	#define	ERROR_SPECIAL_ACCOUNT	1371
00507	#define	ERROR_SPECIAL_GROUP	1372

```
00508 #define ERROR_SPECIAL_USER 1373
00509 #define ERROR_MEMBERS_PRIMARY_GROUP 1374
00510 #define ERROR_TOKEN_ALREADY_IN_USE 1375
00511 #define ERROR_NO_SUCH_ALIAS 1376
00512 #define ERROR_MEMBER_NOT_IN_ALIAS 1377
00513 #define ERROR_MEMBER_IN_ALIAS 1378
00514 #define ERROR_ALIAS_EXISTS 1379
00515 #define ERROR_LOGON_NOT_GRANTED 1380
00516 #define ERROR_TOO_MANY_SECRETS 1381
00517 #define ERROR_SECRET_TOO_LONG 1382
00518 #define ERROR_INTERNAL_DB_ERROR 1383
00519 #define ERROR_TOO_MANY_CONTEXT_IDS 1384
00520 #define ERROR_LOGON_TYPE_NOT_GRANTED 1385
00521 #define ERROR_NT_CROSS_ENCRYPTION_REQUIRED 1386
00522 #define ERROR_NO_SUCH_MEMBER 1387
00523 #define ERROR_INVALID_MEMBER 1388
00524 #define ERROR_TOO_MANY_SIDS 1389
00525 #define ERROR_LM_CROSS_ENCRYPTION_REQUIRED 1390
00526 #define ERROR_NO_INHERITANCE 1391
00527 #define ERROR_FILE_CORRUPT 1392
00528 #define ERROR_DISK_CORRUPT 1393
00529 #define ERROR_NO_USER_SESSION_KEY 1394
00530 #define ERROR_LICENSE_QUOTA_EXCEEDED 1395
00531 #define ERROR_WRONG_TARGET_NAME 1396
00532 #define ERROR_MUTUAL_AUTH_FAILED 1397
00533 #define ERROR_TIME_SKEW 1398
00534 #define ERROR_INVALID_WINDOW_HANDLE 1400
00535 #define ERROR_INVALID_MENU_HANDLE 1401
00536 #define ERROR_INVALID_CURSOR_HANDLE 1402
00537 #define ERROR_INVALID_ACCEL_HANDLE 1403
00538 #define ERROR_INVALID_HOOK_HANDLE 1404
00539 #define ERROR_INVALID_DWP_HANDLE 1405
00540 #define ERROR_TLW_WITH_WSCHILD 1406
00541 #define ERROR_CANNOT_FIND_WND_CLASS 1407
00542 #define ERROR_WINDOW_OF_OTHER_THREAD 1408
00543 #define ERROR_HOTKEY_ALREADY_REGISTERED 1409
00544 #define ERROR_CLASS_ALREADY_EXISTS 1410
00545 #define ERROR_CLASS_DOES_NOT_EXIST 1411
00546 #define ERROR_CLASS_HAS_WINDOWS 1412
00547 #define ERROR_INVALID_INDEX 1413
00548 #define ERROR_INVALID_ICON_HANDLE 1414
00549 #define ERROR_PRIVATE_DIALOG_INDEX 1415
00550 #define ERROR_LISTBOX_ID_NOT_FOUND 1416
00551 #define ERROR_NO_WILDCARD_CHARACTERS 1417
00552 #define ERROR_CLIPBOARD_NOT_OPEN 1418
00553 #define ERROR_HOTKEY_NOT_REGISTERED 1419
00554 #define ERROR_WINDOW_NOT_DIALOG 1420
00555 #define ERROR_CONTROL_ID_NOT_FOUND 1421
00556 #define ERROR_INVALID_COMBOBOX_MESSAGE 1422
00557 #define ERROR_WINDOW_NOT_COMBOBOX 1423
00558 #define ERROR_INVALID_EDIT_HEIGHT 1424
00559 #define ERROR_DC_NOT_FOUND 1425
00560 #define ERROR_INVALID_HOOK_FILTER 1426
00561 #define ERROR_INVALID_FILTER_PROC 1427
00562 #define ERROR_HOOK_NEEDS_HMOD 1428
00563 #define ERROR_GLOBAL_ONLY_HOOK 1429
00564 #define ERROR_JOURNAL_HOOK_SET 1430
00565 #define ERROR_HOOK_NOT_INSTALLED 1431
00566 #define ERROR_INVALID_LB_MESSAGE 1432
00567 #define ERROR_SETCOUNT_ON_BAD_LB 1433
00568 #define ERROR_LB_WITHOUT_TABSTOPS 1434
00569 #define ERROR_DESTROY_OBJECT_OF_OTHER_THREAD 1435
00570 #define ERROR_CHILD_WINDOW_MENU 1436
00571 #define ERROR_NO_SYSTEM_MENU 1437
00572 #define ERROR_INVALID_MSGBOX_STYLE 1438
00573 #define ERROR_INVALID_SPI_VALUE 1439
00574 #define ERROR_SCREEN_ALREADY_LOCKED 1440
00575 #define ERROR_HWNDS_HAVE_DIFF_PARENT 1441
00576 #define ERROR_NOT_CHILD_WINDOW 1442
00577 #define ERROR_INVALID_GW_COMMAND 1443
00578 #define ERROR_INVALID_THREAD_ID 1444
00579 #define ERROR_NON_MDICHILD_WINDOW 1445
00580 #define ERROR_POPUP_ALREADY_ACTIVE 1446
00581 #define ERROR_NO_SCROLLBARS 1447
00582 #define ERROR_INVALID_SCROLLBAR_RANGE 1448
00583 #define ERROR_INVALID_SHOWWIN_COMMAND 1449
00584 #define ERROR_NO_SYSTEM_RESOURCES 1450
00585 #define ERROR_NONPAGED_SYSTEM_RESOURCES 1451
00586 #define ERROR_PAGED_SYSTEM_RESOURCES 1452
00587 #define ERROR_WORKING_SET_QUOTA 1453
00588 #define ERROR_PAGEFILE_QUOTA 1454
00589 #define ERROR_COMMITMENT_LIMIT 1455
00590 #define ERROR_MENU_ITEM_NOT_FOUND 1456
00591 #define ERROR_INVALID_KEYBOARD_HANDLE 1457
00592 #define ERROR_HOOK_TYPE_NOT_ALLOWED 1458
00593 #define ERROR_REQUIRES_INTERACTIVE_WINDOWSTATION 1459
00594 #define ERROR_TIMEOUT 1460
```

```
00595 #define ERROR_INVALID_MONITOR_HANDLE 1461
00596 #define ERROR_EVENTLOG_FILE_CORRUPT 1500
00597 #define ERROR_EVENTLOG_CANT_START 1501
00598 #define ERROR_LOG_FILE_FULL 1502
00599 #define ERROR_EVENTLOG_FILE_CHANGED 1503
00600 #define ERROR_INSTALL_SERVICE_FAILURE 1601
00601 #define ERROR_INSTALL_USEREXIT 1602
00602 #define ERROR_INSTALL_FAILURE 1603
00603 #define ERROR_INSTALL_SUSPEND 1604
00604 #define ERROR_UNKNOWN_PRODUCT 1605
00605 #define ERROR_UNKNOWN_FEATURE 1606
00606 #define ERROR_UNKNOWN_COMPONENT 1607
00607 #define ERROR_UNKNOWN_PROPERTY 1608
00608 #define ERROR_INVALID_HANDLE_STATE 1609
00609 #define ERROR_BAD_CONFIGURATION 1610
00610 #define ERROR_INDEX_ABSENT 1611
00611 #define ERROR_INSTALL_SOURCE_ABSENT 1612
00612 #define ERROR_INSTALL_PACKAGE_VERSION 1613
00613 #define ERROR_PRODUCT_UNINSTALLED 1614
00614 #define ERROR_BAD_QUERY_SYNTAX 1615
00615 #define ERROR_INVALID_FIELD 1616
00616 #define ERROR_DEVICE_REMOVED 1617
00617 #define ERROR_INSTALL_ALREADY_RUNNING 1618
00618 #define ERROR_INSTALL_PACKAGE_OPEN_FAILED 1619
00619 #define ERROR_INSTALL_PACKAGE_INVALID 1620
00620 #define ERROR_INSTALL_UI_FAILURE 1621
00621 #define ERROR_INSTALL_LOG_FAILURE 1622
00622 #define ERROR_INSTALL_LANGUAGE_UNSUPPORTED 1623
00623 #define ERROR_INSTALL_TRANSFORM_FAILURE 1624
00624 #define ERROR_INSTALL_PACKAGE_REJECTED 1625
00625 #define ERROR_FUNCTION_NOT_CALLED 1626
00626 #define ERROR_FUNCTION_FAILED 1627
00627 #define ERROR_INVALID_TABLE 1628
00628 #define ERROR_DATATYPE_MISMATCH 1629
00629 #define ERROR_UNSUPPORTED_TYPE 1630
00630 #define ERROR_CREATE_FAILED 1631
00631 #define ERROR_INSTALL_TEMP_UNWRITABLE 1632
00632 #define ERROR_INSTALL_PLATFORM_UNSUPPORTED 1633
00633 #define ERROR_INSTALL_NOTUSED 1634
00634 #define ERROR_PATCH_PACKAGE_OPEN_FAILED 1635
00635 #define ERROR_PATCH_PACKAGE_INVALID 1636
00636 #define ERROR_PATCH_PACKAGE_UNSUPPORTED 1637
00637 #define ERROR_PRODUCT_VERSION 1638
00638 #define ERROR_INVALID_COMMAND_LINE 1639
00639 #define ERROR_INSTALL_REMOTE_DISALLOWED 1640
00640 #define ERROR_SUCCESS_REBOOT_INITIATED 1641
00641 #define RPC_S_INVALID_STRING_BINDING 1700
00642 #define RPC_S_WRONG_KIND_OF_BINDING 1701
00643 #define RPC_S_INVALID_BINDING 1702
00644 #define RPC_S_PROTSEQ_NOT_SUPPORTED 1703
00645 #define RPC_S_INVALID_RPC_PROTSEQ 1704
00646 #define RPC_S_INVALID_STRING_UUID 1705
00647 #define RPC_S_INVALID_ENDPOINT_FORMAT 1706
00648 #define RPC_S_INVALID_NET_ADDR 1707
00649 #define RPC_S_NO_ENDPOINT_FOUND 1708
00650 #define RPC_S_INVALID_TIMEOUT 1709
00651 #define RPC_S_OBJECT_NOT_FOUND 1710
00652 #define RPC_S_ALREADY_REGISTERED 1711
00653 #define RPC_S_TYPE_ALREADY_REGISTERED 1712
00654 #define RPC_S_ALREADY_LISTENING 1713
00655 #define RPC_S_NO_PROTSEQS_REGISTERED 1714
00656 #define RPC_S_NOT_LISTENING 1715
00657 #define RPC_S_UNKNOWN_MGR_TYPE 1716
00658 #define RPC_S_UNKNOWN_IF 1717
00659 #define RPC_S_NO_BINDINGS 1718
00660 #define RPC_S_NO_PROTSEQS 1719
00661 #define RPC_S_CANT_CREATE_ENDPOINT 1720
00662 #define RPC_S_OUT_OF_RESOURCES 1721
00663 #define RPC_S_SERVER_UNAVAILABLE 1722
00664 #define RPC_S_SERVER_TOO_BUSY 1723
00665 #define RPC_S_INVALID_NETWORK_OPTIONS 1724
00666 #define RPC_S_NO_CALL_ACTIVE 1725
00667 #define RPC_S_CALL_FAILED 1726
00668 #define RPC_S_CALL_FAILED_DNE 1727
00669 #define RPC_S_PROTOCOL_ERROR 1728
00670 #define RPC_S_UNSUPPORTED_TRANS_SYN 1730
00671 #define RPC_S_UNSUPPORTED_TYPE 1732
00672 #define RPC_S_INVALID_TAG 1733
00673 #define RPC_S_INVALID_BOUND 1734
00674 #define RPC_S_NO_ENTRY_NAME 1735
00675 #define RPC_S_INVALID_NAME_SYNTAX 1736
00676 #define RPC_S_UNSUPPORTED_NAME_SYNTAX 1737
00677 #define RPC_S_UUID_NO_ADDRESS 1739
00678 #define RPC_S_DUPLICATE_ENDPOINT 1740
00679 #define RPC_S_UNKNOWN_AUTHN_TYPE 1741
00680 #define RPC_S_MAX_CALLS_TOO_SMALL 1742
00681 #define RPC_S_STRING_TOO_LONG 1743
```

```
00682 #define RPC_S_PROTSEQ_NOT_FOUND 1744
00683 #define RPC_S_PROCNUM_OUT_OF_RANGE 1745
00684 #define RPC_S_BINDING_HAS_NO_AUTH 1746
00685 #define RPC_S_UNKNOWN_AUTHN_SERVICE 1747
00686 #define RPC_S_UNKNOWN_AUTHN_LEVEL 1748
00687 #define RPC_S_INVALID_AUTH_IDENTITY 1749
00688 #define RPC_S_UNKNOWN_AUTHZ_SERVICE 1750
00689 #define EPT_S_INVALID_ENTRY 1751
00690 #define EPT_S_CANT_PERFORM_OP 1752
00691 #define EPT_S_NOT_REGISTERED 1753
00692 #define RPC_S_NOTHING_TO_EXPORT 1754
00693 #define RPC_S_INCOMPLETE_NAME 1755
00694 #define RPC_S_INVALID_VERS_OPTION 1756
00695 #define RPC_S_NO_MORE_MEMBERS 1757
00696 #define RPC_S_NOT_ALL_OBJS_UNEXPORTED 1758
00697 #define RPC_S_INTERFACE_NOT_FOUND 1759
00698 #define RPC_S_ENTRY_ALREADY_EXISTS 1760
00699 #define RPC_S_ENTRY_NOT_FOUND 1761
00700 #define RPC_S_NAME_SERVICE_UNAVAILABLE 1762
00701 #define RPC_S_INVALID_NAF_ID 1763
00702 #define RPC_S_CANNOT_SUPPORT 1764
00703 #define RPC_S_NO_CONTEXT_AVAILABLE 1765
00704 #define RPC_S_INTERNAL_ERROR 1766
00705 #define RPC_S_ZERO_DIVIDE 1767
00706 #define RPC_S_ADDRESS_ERROR 1768
00707 #define RPC_S_FP_DIV_ZERO 1769
00708 #define RPC_S_FP_UNDERFLOW 1770
00709 #define RPC_S_FP_OVERFLOW 1771
00710 #define RPC_X_NO_MORE_ENTRIES 1772
00711 #define RPC_X_SS_CHAR_TRANS_OPEN_FAIL 1773
00712 #define RPC_X_SS_CHAR_TRANS_SHORT_FILE 1774
00713 #define RPC_X_SS_IN_NULL_CONTEXT 1775
00714 #define RPC_X_SS_CONTEXT_DAMAGED 1777
00715 #define RPC_X_SS_HANDLES_MISMATCH 1778
00716 #define RPC_X_SS_CANNOT_GET_CALL_HANDLE 1779
00717 #define RPC_X_NULL_REF_POINTER 1780
00718 #define RPC_X_ENUM_VALUE_OUT_OF_RANGE 1781
00719 #define RPC_X_BYTE_COUNT_TOO_SMALL 1782
00720 #define RPC_X_BAD_STUB_DATA 1783
00721 #define ERROR_INVALID_USER_BUFFER 1784
00722 #define ERROR_UNRECOGNIZED_MEDIA 1785
00723 #define ERROR_NO_TRUST_LSA_SECRET 1786
00724 #define ERROR_NO_TRUST_SAM_ACCOUNT 1787
00725 #define ERROR_TRUSTED_DOMAIN_FAILURE 1788
00726 #define ERROR_TRUSTED_RELATIONSHIP_FAILURE 1789
00727 #define ERROR_TRUST_FAILURE 1790
00728 #define RPC_S_CALL_IN_PROGRESS 1791
00729 #define ERROR_NETLOGON_NOT_STARTED 1792
00730 #define ERROR_ACCOUNT_EXPIRED 1793
00731 #define ERROR_REDIRECTOR_HAS_OPEN_HANDLES 1794
00732 #define ERROR_PRINTER_DRIVER_ALREADY_INSTALLED 1795
00733 #define ERROR_UNKNOWN_PORT 1796
00734 #define ERROR_UNKNOWN_PRINTER_DRIVER 1797
00735 #define ERROR_UNKNOWN_PRINTPROCESSOR 1798
00736 #define ERROR_INVALID_SEPARATOR_FILE 1799
00737 #define ERROR_INVALID_PRIORITY 1800
00738 #define ERROR_INVALID_PRINTER_NAME 1801
00739 #define ERROR_PRINTER_ALREADY_EXISTS 1802
00740 #define ERROR_INVALID_PRINTER_COMMAND 1803
00741 #define ERROR_INVALID_DATATYPE 1804
00742 #define ERROR_INVALID_ENVIRONMENT 1805
00743 #define RPC_S_NO_MORE_BINDINGS 1806
00744 #define ERROR_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT 1807
00745 #define ERROR_NOLOGON_WORKSTATION_TRUST_ACCOUNT 1808
00746 #define ERROR_NOLOGON_SERVER_TRUST_ACCOUNT 1809
00747 #define ERROR_DOMAIN_TRUST_INCONSISTENT 1810
00748 #define ERROR_SERVER_HAS_OPEN_HANDLES 1811
00749 #define ERROR_RESOURCE_DATA_NOT_FOUND 1812
00750 #define ERROR_RESOURCE_TYPE_NOT_FOUND 1813
00751 #define ERROR_RESOURCE_NAME_NOT_FOUND 1814
00752 #define ERROR_RESOURCE_LANG_NOT_FOUND 1815
00753 #define ERROR_NOT_ENOUGH_QUOTA 1816
00754 #define RPC_S_NO_INTERFACES 1817
00755 #define RPC_S_CALL_CANCELLED 1818
00756 #define RPC_S_BINDING_INCOMPLETE 1819
00757 #define RPC_S_COMM_FAILURE 1820
00758 #define RPC_S_UNSUPPORTED_AUTHN_LEVEL 1821
00759 #define RPC_S_NO_PRINC_NAME 1822
00760 #define RPC_S_NOT_RPC_ERROR 1823
00761 #define RPC_S_UUID_LOCAL_ONLY 1824
00762 #define RPC_S_SEC_PKG_ERROR 1825
00763 #define RPC_S_NOT_CANCELLED 1826
00764 #define RPC_X_INVALID_ES_ACTION 1827
00765 #define RPC_X_WRONG_ES_VERSION 1828
00766 #define RPC_X_WRONG_STUB_VERSION 1829
00767 #define RPC_X_INVALID_PIPE_OBJECT 1830
00768 #define RPC_X_WRONG_PIPE_ORDER 1831
```

```
00769 #define RPC_X_WRONG_PIPE_VERSION 1832
00770 #define RPC_S_GROUP_MEMBER_NOT_FOUND 1898
00771 #define EPT_S_CANT_CREATE 1899
00772 #define RPC_S_INVALID_OBJECT 1900
00773 #define ERROR_INVALID_TIME 1901
00774 #define ERROR_INVALID_FORM_NAME 1902
00775 #define ERROR_INVALID_FORM_SIZE 1903
00776 #define ERROR_ALREADY_WAITING 1904
00777 #define ERROR_PRINTER_DELETED 1905
00778 #define ERROR_INVALID_PRINTER_STATE 1906
00779 #define ERROR_PASSWORD_MUST_CHANGE 1907
00780 #define ERROR_DOMAIN_CONTROLLER_NOT_FOUND 1908
00781 #define ERROR_ACCOUNT_LOCKED_OUT 1909
00782 #define OR_INVALID_OXID 1910
00783 #define OR_INVALID_OID 1911
00784 #define OR_INVALID_SET 1912
00785 #define RPC_S_SEND_INCOMPLETE 1913
00786 #define RPC_S_INVALID_ASYNC_HANDLE 1914
00787 #define RPC_S_INVALID_ASYNC_CALL 1915
00788 #define RPC_X_PIPE_CLOSED 1916
00789 #define RPC_X_PIPE_DISCIPLINE_ERROR 1917
00790 #define RPC_X_PIPE_EMPTY 1918
00791 #define ERROR_NO_SITENAME 1919
00792 #define ERROR_CANT_ACCESS_FILE 1920
00793 #define ERROR_CANT_RESOLVE_FILENAME 1921
00794 #define RPC_S_ENTRY_TYPE_MISMATCH 1922
00795 #define RPC_S_NOT_ALL_OBJS_EXPORTED 1923
00796 #define RPC_S_INTERFACE_NOT_EXPORTED 1924
00797 #define RPC_S_PROFILE_NOT_ADDED 1925
00798 #define RPC_S_PRF_ELT_NOT_ADDED 1926
00799 #define RPC_S_PRF_ELT_NOT_REMOVED 1927
00800 #define RPC_S_GRP_ELT_NOT_ADDED 1928
00801 #define RPC_S_GRP_ELT_NOT_REMOVED 1929
00802 #define ERROR_INVALID_PIXEL_FORMAT 2000
00803 #define ERROR_BAD_DRIVER 2001
00804 #define ERROR_INVALID_WINDOW_STYLE 2002
00805 #define ERROR_METAFILE_NOT_SUPPORTED 2003
00806 #define ERROR_TRANSFORM_NOT_SUPPORTED 2004
00807 #define ERROR_CLIPPING_NOT_SUPPORTED 2005
00808 #define ERROR_INVALID_CMM 2010
00809 #define ERROR_INVALID_PROFILE 2011
00810 #define ERROR_TAG_NOT_FOUND 2012
00811 #define ERROR_TAG_NOT_PRESENT 2013
00812 #define ERROR_DUPLICATE_TAG 2014
00813 #define ERROR_PROFILE_NOT_ASSOCIATED_WITH_DEVICE 2015
00814 #define ERROR_PROFILE_NOT_FOUND 2016
00815 #define ERROR_INVALID_COLORSPACE 2017
00816 #define ERROR_ICM_NOT_ENABLED 2018
00817 #define ERROR_DELETING_ICM_XFORM 2019
00818 #define ERROR_INVALID_TRANSFORM 2020
00819 #define ERROR_COLORSPACE_MISMATCH 2021
00820 #define ERROR_INVALID_COLORINDEX 2022
00821 #define ERROR_CONNECTED_OTHER_PASSWORD 2108
00822 #define ERROR_BAD_USERNAME 2202
00823 #define ERROR_NOT_CONNECTED 2250
00824 #define ERROR_OPEN_FILES 2401
00825 #define ERROR_ACTIVE_CONNECTIONS 2402
00826 #define ERROR_DEVICE_IN_USE 2404
00827 #define ERROR_UNKNOWN_PRINT_MONITOR 3000
00828 #define ERROR_PRINTER_DRIVER_IN_USE 3001
00829 #define ERROR_SPOOL_FILE_NOT_FOUND 3002
00830 #define ERROR_SPL_NO_STARTDOC 3003
00831 #define ERROR_SPL_NO_ADDJOB 3004
00832 #define ERROR_PRINT_PROCESSOR_ALREADY_INSTALLED 3005
00833 #define ERROR_PRINT_MONITOR_ALREADY_INSTALLED 3006
00834 #define ERROR_INVALID_PRINT_MONITOR 3007
00835 #define ERROR_PRINT_MONITOR_IN_USE 3008
00836 #define ERROR_PRINTER_HAS_JOBS_QUEUED 3009
00837 #define ERROR_SUCCESS_REBOOT_REQUIRED 3010
00838 #define ERROR_SUCCESS_RESTART_REQUIRED 3011
00839 #define ERROR_PRINTER_NOT_FOUND 3012
00840 #define ERROR_WINS_INTERNAL 4000
00841 #define ERROR_CAN_NOT_DEL_LOCAL_WINS 4001
00842 #define ERROR_STATIC_INIT 4002
00843 #define ERROR_INC_BACKUP 4003
00844 #define ERROR_FULL_BACKUP 4004
00845 #define ERROR_REC_NON_EXISTENT 4005
00846 #define ERROR_RPL_NOT_ALLOWED 4006
00847 #define ERROR_DHCP_ADDRESS_CONFLICT 4100
00848 #define ERROR_WMI_GUID_NOT_FOUND 4200
00849 #define ERROR_WMI_INSTANCE_NOT_FOUND 4201
00850 #define ERROR_WMI_ITEMID_NOT_FOUND 4202
00851 #define ERROR_WMI_TRY_AGAIN 4203
00852 #define ERROR_WMI_DP_NOT_FOUND 4204
00853 #define ERROR_WMI_UNRESOLVED_INSTANCE_REF 4205
00854 #define ERROR_WMI_ALREADY_ENABLED 4206
00855 #define ERROR_WMI_GUID_DISCONNECTED 4207
```

```
00856 #define ERROR_WMI_SERVER_UNAVAILABLE 4208
00857 #define ERROR_WMI_DP_FAILED 4209
00858 #define ERROR_WMI_INVALID_MOF 4210
00859 #define ERROR_WMI_INVALID_REGINFO 4211
00860 #define ERROR_WMI_ALREADY_DISABLED 4212
00861 #define ERROR_WMI_READ_ONLY 4213
00862 #define ERROR_WMI_SET_FAILURE 4214
00863 #define ERROR_INVALID_MEDIA 4300
00864 #define ERROR_INVALID_LIBRARY 4301
00865 #define ERROR_INVALID_MEDIA_POOL 4302
00866 #define ERROR_DRIVE_MEDIA_MISMATCH 4303
00867 #define ERROR_MEDIA_OFFLINE 4304
00868 #define ERROR_LIBRARY_OFFLINE 4305
00869 #define ERROR_EMPTY 4306
00870 #define ERROR_NOT_EMPTY 4307
00871 #define ERROR_MEDIA_UNAVAILABLE 4308
00872 #define ERROR_RESOURCE_DISABLED 4309
00873 #define ERROR_INVALID_CLEANER 4310
00874 #define ERROR_UNABLE_TO_CLEAN 4311
00875 #define ERROR_OBJECT_NOT_FOUND 4312
00876 #define ERROR_DATABASE_FAILURE 4313
00877 #define ERROR_DATABASE_FULL 4314
00878 #define ERROR_MEDIA_INCOMPATIBLE 4315
00879 #define ERROR_RESOURCE_NOT_PRESENT 4316
00880 #define ERROR_INVALID_OPERATION 4317
00881 #define ERROR_MEDIA_NOT_AVAILABLE 4318
00882 #define ERROR_DEVICE_NOT_AVAILABLE 4319
00883 #define ERROR_REQUEST_REFUSED 4320
00884 #define ERROR_INVALID_DRIVE_OBJECT 4321
00885 #define ERROR_LIBRARY_FULL 4322
00886 #define ERROR_MEDIUM_NOT_ACCESSIBLE 4323
00887 #define ERROR_UNABLE_TO_LOAD_MEDIUM 4324
00888 #define ERROR_UNABLE_TO_INVENTORY_DRIVE 4325
00889 #define ERROR_UNABLE_TO_INVENTORY_SLOT 4326
00890 #define ERROR_UNABLE_TO_INVENTORY_TRANSPORT 4327
00891 #define ERROR_TRANSPORT_FULL 4328
00892 #define ERROR_CONTROLLING_IEPORT 4329
00893 #define ERROR_UNABLE_TO_EJECT_MOUNTED_MEDIA 4330
00894 #define ERROR_CLEANER_SLOT_SET 4331
00895 #define ERROR_CLEANER_SLOT_NOT_SET 4332
00896 #define ERROR_CLEANER_CARTRIDGE_SPENT 4333
00897 #define ERROR_UNEXPECTED_OMID 4334
00898 #define ERROR_CANT_DELETE_LAST_ITEM 4335
00899 #define ERROR_MESSAGE_EXCEEDS_MAX_SIZE 4336
00900 #define ERROR_VOLUME_CONTAINS_SYS_FILES 4337
00901 #define ERROR_INDIGENOUS_TYPE 4338
00902 #define ERROR_NO_SUPPORTING_DRIVES 4339
00903 #define ERROR_FILE_OFFLINE 4350
00904 #define ERROR_REMOTE_STORAGE_NOT_ACTIVE 4351
00905 #define ERROR_REMOTE_STORAGE_MEDIA_ERROR 4352
00906 #define ERROR_NOT_A_REPARSE_POINT 4390
00907 #define ERROR_REPARSE_ATTRIBUTE_CONFLICT 4391
00908 #define ERROR_INVALID_REPARSE_DATA 4392
00909 #define ERROR_REPARSE_TAG_INVALID 4393
00910 #define ERROR_REPARSE_TAG_MISMATCH 4394
00911 #define ERROR_VOLUME_NOT_SIS_ENABLED 4500
00912 #define ERROR_DEPENDENT_RESOURCE_EXISTS 5001
00913 #define ERROR_DEPENDENCY_NOT_FOUND 5002
00914 #define ERROR_DEPENDENCY_ALREADY_EXISTS 5003
00915 #define ERROR_RESOURCE_NOT_ONLINE 5004
00916 #define ERROR_HOST_NODE_NOT_AVAILABLE 5005
00917 #define ERROR_RESOURCE_NOT_AVAILABLE 5006
00918 #define ERROR_RESOURCE_NOT_FOUND 5007
00919 #define ERROR_SHUTDOWN_CLUSTER 5008
00920 #define ERROR_CANT_EVICT_ACTIVE_NODE 5009
00921 #define ERROR_OBJECT_ALREADY_EXISTS 5010
00922 #define ERROR_OBJECT_IN_LIST 5011
00923 #define ERROR_GROUP_NOT_AVAILABLE 5012
00924 #define ERROR_GROUP_NOT_FOUND 5013
00925 #define ERROR_GROUP_NOT_ONLINE 5014
00926 #define ERROR_HOST_NODE_NOT_RESOURCE_OWNER 5015
00927 #define ERROR_HOST_NODE_NOT_GROUP_OWNER 5016
00928 #define ERROR_RESMON_CREATE_FAILED 5017
00929 #define ERROR_RESMON_ONLINE_FAILED 5018
00930 #define ERROR_RESOURCE_ONLINE 5019
00931 #define ERROR_QUORUM_RESOURCE 5020
00932 #define ERROR_NOT_QUORUM_CAPABLE 5021
00933 #define ERROR_CLUSTER_SHUTTING_DOWN 5022
00934 #define ERROR_INVALID_STATE 5023
00935 #define ERROR_RESOURCE_PROPERTIES_STORED 5024
00936 #define ERROR_NOT_QUORUM_CLASS 5025
00937 #define ERROR_CORE_RESOURCE 5026
00938 #define ERROR_QUORUM_RESOURCE_ONLINE_FAILED 5027
00939 #define ERROR_QUORUMLOG_OPEN_FAILED 5028
00940 #define ERROR_CLUSTERLOG_CORRUPT 5029
00941 #define ERROR_CLUSTERLOG_RECORD_EXCEEDS_MAXSIZE 5030
00942 #define ERROR_CLUSTERLOG_EXCEEDS_MAXSIZE 5031
```

00943	#define	ERROR_CLUSTERLOG_CHKPOINT_NOT_FOUND	5032
00944	#define	ERROR_CLUSTERLOG_NOT_ENOUGH_SPACE	5033
00945	#define	ERROR_QUORUM_OWNER_ALIVE	5034
00946	#define	ERROR_NETWORK_NOT_AVAILABLE	5035
00947	#define	ERROR_NODE_NOT_AVAILABLE	5036
00948	#define	ERROR_ALL_NODES_NOT_AVAILABLE	5037
00949	#define	ERROR_RESOURCE_FAILED	5038
00950	#define	ERROR_CLUSTER_INVALID_NODE	5039
00951	#define	ERROR_CLUSTER_NODE_EXISTS	5040
00952	#define	ERROR_CLUSTER_JOIN_IN_PROGRESS	5041
00953	#define	ERROR_CLUSTER_NODE_NOT_FOUND	5042
00954	#define	ERROR_CLUSTER_LOCAL_NODE_NOT_FOUND	5043
00955	#define	ERROR_CLUSTER_NETWORK_EXISTS	5044
00956	#define	ERROR_CLUSTER_NETWORK_NOT_FOUND	5045
00957	#define	ERROR_CLUSTER_NETINTERFACE_EXISTS	5046
00958	#define	ERROR_CLUSTER_NETINTERFACE_NOT_FOUND	5047
00959	#define	ERROR_CLUSTER_INVALID_REQUEST	5048
00960	#define	ERROR_CLUSTER_INVALID_NETWORK_PROVIDER	5049
00961	#define	ERROR_CLUSTER_NODE_DOWN	5050
00962	#define	ERROR_CLUSTER_NODE_UNREACHABLE	5051
00963	#define	ERROR_CLUSTER_NODE_NOT_MEMBER	5052
00964	#define	ERROR_CLUSTER_JOIN_NOT_IN_PROGRESS	5053
00965	#define	ERROR_CLUSTER_INVALID_NETWORK	5054
00966	#define	ERROR_CLUSTER_NODE_UP	5056
00967	#define	ERROR_CLUSTER_IPADDR_IN_USE	5057
00968	#define	ERROR_CLUSTER_NODE_NOT_PAUSED	5058
00969	#define	ERROR_CLUSTER_NO_SECURITY_CONTEXT	5059
00970	#define	ERROR_CLUSTER_NETWORK_NOT_INTERNAL	5060
00971	#define	ERROR_CLUSTER_NODE_ALREADY_UP	5061
00972	#define	ERROR_CLUSTER_NODE_ALREADY_DOWN	5062
00973	#define	ERROR_CLUSTER_NETWORK_ALREADY_ONLINE	5063
00974	#define	ERROR_CLUSTER_NETWORK_ALREADY_OFFLINE	5064
00975	#define	ERROR_CLUSTER_NODE_ALREADY_MEMBER	5065
00976	#define	ERROR_CLUSTER_LAST_INTERNAL_NETWORK	5066
00977	#define	ERROR_CLUSTER_NETWORK_HAS_DEPENDENTS	5067
00978	#define	ERROR_INVALID_OPERATION_ON_QUORUM	5068
00979	#define	ERROR_DEPENDENCY_NOT_ALLOWED	5069
00980	#define	ERROR_CLUSTER_NODE_PAUSED	5070
00981	#define	ERROR_NODE_CANT_HOST_RESOURCE	5071
00982	#define	ERROR_CLUSTER_NODE_NOT_READY	5072
00983	#define	ERROR_CLUSTER_NODE_SHUTTING_DOWN	5073
00984	#define	ERROR_CLUSTER_JOIN_ABORTED	5074
00985	#define	ERROR_CLUSTER_INCOMPATIBLE_VERSIONS	5075
00986	#define	ERROR_CLUSTER_MAXNUM_OF_RESOURCES_EXCEEDED	5076
00987	#define	ERROR_CLUSTER_SYSTEM_CONFIG_CHANGED	5077
00988	#define	ERROR_CLUSTER_RESOURCE_TYPE_NOT_FOUND	5078
00989	#define	ERROR_CLUSTER_RESTYPE_NOT_SUPPORTED	5079
00990	#define	ERROR_CLUSTER_RESNAME_NOT_FOUND	5080
00991	#define	ERROR_CLUSTER_NO_RPC_PACKAGES_REGISTERED	5081
00992	#define	ERROR_CLUSTER_OWNER_NOT_IN_PREFLIST	5082
00993	#define	ERROR_CLUSTER_DATABASE_SEQMISMATCH	5083
00994	#define	ERROR_RESMON_INVALID_STATE	5084
00995	#define	ERROR_CLUSTER_GUM_NOT_LOCKER	5085
00996	#define	ERROR_QUORUM_DISK_NOT_FOUND	5086
00997	#define	ERROR_DATABASE_BACKUP_CORRUPT	5087
00998	#define	ERROR_CLUSTER_NODE_ALREADY_HAS_DFS_ROOT	5088
00999	#define	ERROR_RESOURCE_PROPERTY_UNCHANGEABLE	5089
01000	#define	ERROR_ENCRYPTION_FAILED	6000
01001	#define	ERROR_DECRYPTION_FAILED	6001
01002	#define	ERROR_FILE_ENCRYPTED	6002
01003	#define	ERROR_NO_RECOVERY_POLICY	6003
01004	#define	ERROR_NO_EFS	6004
01005	#define	ERROR_WRONG_EFS	6005
01006	#define	ERROR_NO_USER_KEYS	6006
01007	#define	ERROR_FILE_NOT_ENCRYPTED	6007
01008	#define	ERROR_NOT_EXPORT_FORMAT	6008
01009	#define	ERROR_FILE_READ_ONLY	6009
01010	#define	ERROR_DIR_EFS_DISALLOWED	6010
01011	#define	ERROR_EFS_SERVER_NOT_TRUSTED	6011
01012	#define	ERROR_NO_BROWSER_SERVERS_FOUND	6118
01013	#define	SCHED_E_SERVICE_NOT_LOCALSYSTEM	6200
01014	#define	ERROR_CTX_WINSTATION_NAME_INVALID	7001
01015	#define	ERROR_CTX_INVALID_PD	7002
01016	#define	ERROR_CTX_PD_NOT_FOUND	7003
01017	#define	ERROR_CTX_WD_NOT_FOUND	7004
01018	#define	ERROR_CTX_CANNOT_MAKE_EVENTLOG_ENTRY	7005
01019	#define	ERROR_CTX_SERVICE_NAME_COLLISION	7006
01020	#define	ERROR_CTX_CLOSE_PENDING	7007
01021	#define	ERROR_CTX_NO_OUTBUF	7008
01022	#define	ERROR_CTX_MODEM_INF_NOT_FOUND	7009
01023	#define	ERROR_CTX_INVALID_MODEMNAME	7010
01024	#define	ERROR_CTX_MODEM_RESPONSE_ERROR	7011
01025	#define	ERROR_CTX_MODEM_RESPONSE_TIMEOUT	7012
01026	#define	ERROR_CTX_MODEM_RESPONSE_NO_CARRIER	7013
01027	#define	ERROR_CTX_MODEM_RESPONSE_NO_DIALTONE	7014
01028	#define	ERROR_CTX_MODEM_RESPONSE_BUSY	7015
01029	#define	ERROR_CTX_MODEM_RESPONSE_VOICE	7016

```
01030 #define ERROR_CTX_TD_ERROR 7017
01031 #define ERROR_CTX_WINSTATION_NOT_FOUND 7022
01032 #define ERROR_CTX_WINSTATION_ALREADY_EXISTS 7023
01033 #define ERROR_CTX_WINSTATION_BUSY 7024
01034 #define ERROR_CTX_BAD_VIDEO_MODE 7025
01035 #define ERROR_CTX_GRAPHICS_INVALID 7035
01036 #define ERROR_CTX_LOGON_DISABLED 7037
01037 #define ERROR_CTX_NOT_CONSOLE 7038
01038 #define ERROR_CTX_CLIENT_QUERY_TIMEOUT 7040
01039 #define ERROR_CTX_CONSOLE_DISCONNECT 7041
01040 #define ERROR_CTX_CONSOLE_CONNECT 7042
01041 #define ERROR_CTX_SHADOW_DENIED 7044
01042 #define ERROR_CTX_WINSTATION_ACCESS_DENIED 7045
01043 #define ERROR_CTX_INVALID_WD 7049
01044 #define ERROR_CTX_SHADOW_INVALID 7050
01045 #define ERROR_CTX_SHADOW_DISABLED 7051
01046 #define ERROR_CTX_CLIENT_LICENSE_IN_USE 7052
01047 #define ERROR_CTX_CLIENT_LICENSE_NOT_SET 7053
01048 #define ERROR_CTX_LICENSE_NOT_AVAILABLE 7054
01049 #define ERROR_CTX_LICENSE_CLIENT_INVALID 7055
01050 #define ERROR_CTX_LICENSE_EXPIRED 7056
01051 #define FRS_ERR_INVALID_API_SEQUENCE 8001
01052 #define FRS_ERR_STARTING_SERVICE 8002
01053 #define FRS_ERR_STOPPING_SERVICE 8003
01054 #define FRS_ERR_INTERNAL_API 8004
01055 #define FRS_ERR_INTERNAL 8005
01056 #define FRS_ERR_SERVICE_COMM 8006
01057 #define FRS_ERR_INSUFFICIENT_PRIV 8007
01058 #define FRS_ERR_AUTHENTICATION 8008
01059 #define FRS_ERR_PARENT_INSUFFICIENT_PRIV 8009
01060 #define FRS_ERR_PARENT_AUTHENTICATION 8010
01061 #define FRS_ERR_CHILD_TO_PARENT_COMM 8011
01062 #define FRS_ERR_PARENT_TO_CHILD_COMM 8012
01063 #define FRS_ERR_SYSVOL_POPULATE 8013
01064 #define FRS_ERR_SYSVOL_POPULATE_TIMEOUT 8014
01065 #define FRS_ERR_SYSVOL_IS_BUSY 8015
01066 #define FRS_ERR_SYSVOL_DEMOTE 8016
01067 #define FRS_ERR_INVALID_SERVICE_PARAMETER 8017
01068 #define ERROR_DS_NOT_INSTALLED 8200
01069 #define ERROR_DS_MEMBERSHIP_EVALUATED_LOCALLY 8201
01070 #define ERROR_DS_NO_ATTRIBUTE_OR_VALUE 8202
01071 #define ERROR_DS_INVALID_ATTRIBUTE_SYNTAX 8203
01072 #define ERROR_DS_ATTRIBUTE_TYPE_UNDEFINED 8204
01073 #define ERROR_DS_ATTRIBUTE_OR_VALUE_EXISTS 8205
01074 #define ERROR_DS_BUSY 8206
01075 #define ERROR_DS_UNAVAILABLE 8207
01076 #define ERROR_DS_NO_RIDS_ALLOCATED 8208
01077 #define ERROR_DS_NO_MORE_RIDS 8209
01078 #define ERROR_DS_INCORRECT_ROLE_OWNER 8210
01079 #define ERROR_DS_RIDMGR_INIT_ERROR 8211
01080 #define ERROR_DS_OBJ_CLASS_VIOLATION 8212
01081 #define ERROR_DS_CANT_ON_NON_LEAF 8213
01082 #define ERROR_DS_CANT_ON_RDN 8214
01083 #define ERROR_DS_CANT_MOD_OBJ_CLASS 8215
01084 #define ERROR_DS_CROSS_DOM_MOVE_ERROR 8216
01085 #define ERROR_DS_GC_NOT_AVAILABLE 8217
01086 #define ERROR_SHARED_POLICY 8218
01087 #define ERROR_POLICY_OBJECT_NOT_FOUND 8219
01088 #define ERROR_POLICY_ONLY_IN_DS 8220
01089 #define ERROR_PROMOTION_ACTIVE 8221
01090 #define ERROR_NO_PROMOTION_ACTIVE 8222
01091 #define ERROR_DS_OPERATIONS_ERROR 8224
01092 #define ERROR_DS_PROTOCOL_ERROR 8225
01093 #define ERROR_DS_TIMELIMIT_EXCEEDED 8226
01094 #define ERROR_DS_SIZELIMIT_EXCEEDED 8227
01095 #define ERROR_DS_ADMIN_LIMIT_EXCEEDED 8228
01096 #define ERROR_DS_COMPARE_FALSE 8229
01097 #define ERROR_DS_COMPARE_TRUE 8230
01098 #define ERROR_DS_AUTH_METHOD_NOT_SUPPORTED 8231
01099 #define ERROR_DS_STRONG_AUTH_REQUIRED 8232
01100 #define ERROR_DS_INAPPROPRIATE_AUTH 8233
01101 #define ERROR_DS_AUTH_UNKNOWN 8234
01102 #define ERROR_DS_REFERRAL 8235
01103 #define ERROR_DS_UNAVAILABLE_CRIT_EXTENSION 8236
01104 #define ERROR_DS_CONFIDENTIALITY_REQUIRED 8237
01105 #define ERROR_DS_INAPPROPRIATE_MATCHING 8238
01106 #define ERROR_DS_CONSTRAINT_VIOLATION 8239
01107 #define ERROR_DS_NO_SUCH_OBJECT 8240
01108 #define ERROR_DS_ALIAS_PROBLEM 8241
01109 #define ERROR_DS_INVALID_DN_SYNTAX 8242
01110 #define ERROR_DS_IS_LEAF 8243
01111 #define ERROR_DS_ALIAS_DEREF_PROBLEM 8244
01112 #define ERROR_DS_UNWILLING_TO_PERFORM 8245
01113 #define ERROR_DS_LOOP_DETECT 8246
01114 #define ERROR_DS_NAMING_VIOLATION 8247
01115 #define ERROR_DS_OBJECT_RESULTS_TOO_LARGE 8248
01116 #define ERROR_DS_AFFECTS_MULTIPLE_DSAS 8249
```

```
01117 #define ERROR_DS_SERVER_DOWN 8250
01118 #define ERROR_DS_LOCAL_ERROR 8251
01119 #define ERROR_DS_ENCODING_ERROR 8252
01120 #define ERROR_DS_DECODING_ERROR 8253
01121 #define ERROR_DS_FILTER_UNKNOWN 8254
01122 #define ERROR_DS_PARAM_ERROR 8255
01123 #define ERROR_DS_NOT_SUPPORTED 8256
01124 #define ERROR_DS_NO_RESULTS_RETURNED 8257
01125 #define ERROR_DS_CONTROL_NOT_FOUND 8258
01126 #define ERROR_DS_CLIENT_LOOP 8259
01127 #define ERROR_DS_REFERRAL_LIMIT_EXCEEDED 8260
01128 #define ERROR_DS_ROOT_MUST_BE_NC 8301
01129 #define ERROR_DS_ADD_REPLICA_INHIBITED 8302
01130 #define ERROR_DS_ATT_NOT_DEF_IN_SCHEMA 8303
01131 #define ERROR_DS_MAX_OBJ_SIZE_EXCEEDED 8304
01132 #define ERROR_DS_OBJ_STRING_NAME_EXISTS 8305
01133 #define ERROR_DS_NO_RDN_DEFINED_IN_SCHEMA 8306
01134 #define ERROR_DS_RDN_DOESNT_MATCH_SCHEMA 8307
01135 #define ERROR_DS_NO_REQUESTED_ATTRS_FOUND 8308
01136 #define ERROR_DS_USER_BUFFER_TO_SMALL 8309
01137 #define ERROR_DS_ATT_IS_NOT_ON_OBJ 8310
01138 #define ERROR_DS_ILLEGAL_MOD_OPERATION 8311
01139 #define ERROR_DS_OBJ_TOO_LARGE 8312
01140 #define ERROR_DS_BAD_INSTANCE_TYPE 8313
01141 #define ERROR_DS_MASTERDSA_REQUIRED 8314
01142 #define ERROR_DS_OBJECT_CLASS_REQUIRED 8315
01143 #define ERROR_DS_MISSING_REQUIRED_ATT 8316
01144 #define ERROR_DS_ATT_NOT_DEF_FOR_CLASS 8317
01145 #define ERROR_DS_ATT_ALREADY_EXISTS 8318
01146 #define ERROR_DS_CANT_ADD_ATT_VALUES 8320
01147 #define ERROR_DS_SINGLE_VALUE_CONSTRAINT 8321
01148 #define ERROR_DS_RANGE_CONSTRAINT 8322
01149 #define ERROR_DS_ATT_VAL_ALREADY_EXISTS 8323
01150 #define ERROR_DS_CANT_REM_MISSING_ATT 8324
01151 #define ERROR_DS_CANT_REM_MISSING_ATT_VAL 8325
01152 #define ERROR_DS_ROOT_CANT_BE_SUBREF 8326
01153 #define ERROR_DS_NO_CHAINING 8327
01154 #define ERROR_DS_NO_CHAINED_EVAL 8328
01155 #define ERROR_DS_NO_PARENT_OBJECT 8329
01156 #define ERROR_DS_PARENT_IS_AN_ALIAS 8330
01157 #define ERROR_DS_CANT_MIX_MASTER_AND_REPS 8331
01158 #define ERROR_DS_CHILDREN_EXIST 8332
01159 #define ERROR_DS_OBJ_NOT_FOUND 8333
01160 #define ERROR_DS_ALIASED_OBJ_MISSING 8334
01161 #define ERROR_DS_BAD_NAME_SYNTAX 8335
01162 #define ERROR_DS_ALIAS_POINTS_TO_ALIAS 8336
01163 #define ERROR_DS_CANT_DEREF_ALIAS 8337
01164 #define ERROR_DS_OUT_OF_SCOPE 8338
01165 #define ERROR_DS_CANT_DELETE_DSA_OBJ 8340
01166 #define ERROR_DS_GENERIC_ERROR 8341
01167 #define ERROR_DS_DSA_MUST_BE_INT_MASTER 8342
01168 #define ERROR_DS_CLASS_NOT_DSA 8343
01169 #define ERROR_DS_INSUFF_ACCESS_RIGHTS 8344
01170 #define ERROR_DS_ILLEGAL_SUPERIOR 8345
01171 #define ERROR_DS_ATTRIBUTE_OWNED_BY_SAM 8346
01172 #define ERROR_DS_NAME_TOO_MANY_PARTS 8347
01173 #define ERROR_DS_NAME_TOO_LONG 8348
01174 #define ERROR_DS_NAME_VALUE_TOO_LONG 8349
01175 #define ERROR_DS_NAME_UNPARSEABLE 8350
01176 #define ERROR_DS_NAME_TYPE_UNKNOWN 8351
01177 #define ERROR_DS_NOT_AN_OBJECT 8352
01178 #define ERROR_DS_SEC_DESC_TOO_SHORT 8353
01179 #define ERROR_DS_SEC_DESC_INVALID 8354
01180 #define ERROR_DS_NO_DELETED_NAME 8355
01181 #define ERROR_DS_SUBREF_MUST_HAVE_PARENT 8356
01182 #define ERROR_DS_NCNAME_MUST_BE_NC 8357
01183 #define ERROR_DS_CANT_ADD_SYSTEM_ONLY 8358
01184 #define ERROR_DS_CLASS_MUST_BE_CONCRETE 8359
01185 #define ERROR_DS_INVALID_DMD 8360
01186 #define ERROR_DS_OBJ_GUID_EXISTS 8361
01187 #define ERROR_DS_NOT_ON_BACKLINK 8362
01188 #define ERROR_DS_NO_CROSSREF_FOR_NC 8363
01189 #define ERROR_DS_SHUTTING_DOWN 8364
01190 #define ERROR_DS_UNKNOWN_OPERATION 8365
01191 #define ERROR_DS_INVALID_ROLE_OWNER 8366
01192 #define ERROR_DS_COULDNT_CONTACT_FSMO 8367
01193 #define ERROR_DS_CROSS_NC_DN_RENAME 8368
01194 #define ERROR_DS_CANT_MOD_SYSTEM_ONLY 8369
01195 #define ERROR_DS_REPLICATOR_ONLY 8370
01196 #define ERROR_DS_OBJ_CLASS_NOT_DEFINED 8371
01197 #define ERROR_DS_OBJ_CLASS_NOT_SUBCLASS 8372
01198 #define ERROR_DS_NAME_REFERENCE_INVALID 8373
01199 #define ERROR_DS_CROSS_REF_EXISTS 8374
01200 #define ERROR_DS_CANT_DEL_MASTER_CROSSREF 8375
01201 #define ERROR_DS_SUBTREE_NOTIFY_NOT_NC_HEAD 8376
01202 #define ERROR_DS_NOTIFY_FILTER_TOO_COMPLEX 8377
01203 #define ERROR_DS_DUP_RDN 8378
```

```
01204 #define ERROR_DS_DUP_OID 8379
01205 #define ERROR_DS_DUP_MAPI_ID 8380
01206 #define ERROR_DS_DUP_SCHEMA_ID_GUID 8381
01207 #define ERROR_DS_DUP_LDAP_DISPLAY_NAME 8382
01208 #define ERROR_DS_SEMANTIC_ATT_TEST 8383
01209 #define ERROR_DS_SYNTAX_MISMATCH 8384
01210 #define ERROR_DS_EXISTS_IN_MUST_HAVE 8385
01211 #define ERROR_DS_EXISTS_IN_MAY_HAVE 8386
01212 #define ERROR_DS_NONEXISTENT_MAY_HAVE 8387
01213 #define ERROR_DS_NONEXISTENT_MUST_HAVE 8388
01214 #define ERROR_DS_AUX_CLS_TEST_FAIL 8389
01215 #define ERROR_DS_NONEXISTENT_POSS_SUP 8390
01216 #define ERROR_DS_SUB_CLS_TEST_FAIL 8391
01217 #define ERROR_DS_BAD_RDN_ATT_ID_SYNTAX 8392
01218 #define ERROR_DS_EXISTS_IN_AUX_CLS 8393
01219 #define ERROR_DS_EXISTS_IN_SUB_CLS 8394
01220 #define ERROR_DS_EXISTS_IN_POSS_SUP 8395
01221 #define ERROR_DS_RECALCSHEMA_FAILED 8396
01222 #define ERROR_DS_TREE_DELETE_NOT_FINISHED 8397
01223 #define ERROR_DS_CANT_DELETE 8398
01224 #define ERROR_DS_ATT_SCHEMA_REQ_ID 8399
01225 #define ERROR_DS_BAD_ATT_SCHEMA_SYNTAX 8400
01226 #define ERROR_DS_CANT_CACHE_ATT 8401
01227 #define ERROR_DS_CANT_CACHE_CLASS 8402
01228 #define ERROR_DS_CANT_REMOVE_ATT_CACHE 8403
01229 #define ERROR_DS_CANT_REMOVE_CLASS_CACHE 8404
01230 #define ERROR_DS_CANT_RETRIEVE_DN 8405
01231 #define ERROR_DS_MISSING_SUPREF 8406
01232 #define ERROR_DS_CANT_RETRIEVE_INSTANCE 8407
01233 #define ERROR_DS_CODE_INCONSISTENCY 8408
01234 #define ERROR_DS_DATABASE_ERROR 8409
01235 #define ERROR_DS_GOVERNSID_MISSING 8410
01236 #define ERROR_DS_MISSING_EXPECTED_ATT 8411
01237 #define ERROR_DS_NCNAME_MISSING_CR_REF 8412
01238 #define ERROR_DS_SECURITY_CHECKING_ERROR 8413
01239 #define ERROR_DS_SCHEMA_NOT_LOADED 8414
01240 #define ERROR_DS_SCHEMA_ALLOC_FAILED 8415
01241 #define ERROR_DS_ATT_SCHEMA_REQ_SYNTAX 8416
01242 #define ERROR_DS_GCVERIFY_ERROR 8417
01243 #define ERROR_DS_DRA_SCHEMA_MISMATCH 8418
01244 #define ERROR_DS_CANT_FIND_DSA_OBJ 8419
01245 #define ERROR_DS_CANT_FIND_EXPECTED_NC 8420
01246 #define ERROR_DS_CANT_FIND_NC_IN_CACHE 8421
01247 #define ERROR_DS_CANT_RETRIEVE_CHILD 8422
01248 #define ERROR_DS_SECURITY_ILLEGAL_MODIFY 8423
01249 #define ERROR_DS_CANT_REPLACE_HIDDEN_REC 8424
01250 #define ERROR_DS_BAD_HIERARCHY_FILE 8425
01251 #define ERROR_DS_BUILD_HIERARCHY_TABLE_FAILED 8426
01252 #define ERROR_DS_CONFIG_PARAM_MISSING 8427
01253 #define ERROR_DS_COUNTING_AB_INDICES_FAILED 8428
01254 #define ERROR_DS_HIERARCHY_TABLE_MALLOC_FAILED 8429
01255 #define ERROR_DS_INTERNAL_FAILURE 8430
01256 #define ERROR_DS_UNKNOWN_ERROR 8431
01257 #define ERROR_DS_ROOT_REQUIRES_CLASS_TOP 8432
01258 #define ERROR_DS_REFUSING_FSMO_ROLES 8433
01259 #define ERROR_DS_MISSING_FSMO_SETTINGS 8434
01260 #define ERROR_DS_UNABLE_TO_SURRENDER_ROLES 8435
01261 #define ERROR_DS_DRA_GENERIC 8436
01262 #define ERROR_DS_DRA_INVALID_PARAMETER 8437
01263 #define ERROR_DS_DRA_BUSY 8438
01264 #define ERROR_DS_DRA_BAD_DN 8439
01265 #define ERROR_DS_DRA_BAD_NC 8440
01266 #define ERROR_DS_DRA_DN_EXISTS 8441
01267 #define ERROR_DS_DRA_INTERNAL_ERROR 8442
01268 #define ERROR_DS_DRA_INCONSISTENT_DIT 8443
01269 #define ERROR_DS_DRA_CONNECTION_FAILED 8444
01270 #define ERROR_DS_DRA_BAD_INSTANCE_TYPE 8445
01271 #define ERROR_DS_DRA_OUT_OF_MEM 8446
01272 #define ERROR_DS_DRA_MAIL_PROBLEM 8447
01273 #define ERROR_DS_DRA_REF_ALREADY_EXISTS 8448
01274 #define ERROR_DS_DRA_REF_NOT_FOUND 8449
01275 #define ERROR_DS_DRA_OBJ_IS_REP_SOURCE 8450
01276 #define ERROR_DS_DRA_DB_ERROR 8451
01277 #define ERROR_DS_DRA_NO_REPLICA 8452
01278 #define ERROR_DS_DRA_ACCESS_DENIED 8453
01279 #define ERROR_DS_DRA_NOT_SUPPORTED 8454
01280 #define ERROR_DS_DRA_RPC_CANCELLED 8455
01281 #define ERROR_DS_DRA_SOURCE_DISABLED 8456
01282 #define ERROR_DS_DRA_SINK_DISABLED 8457
01283 #define ERROR_DS_DRA_NAME_COLLISION 8458
01284 #define ERROR_DS_DRA_SOURCE_REINSTALLED 8459
01285 #define ERROR_DS_DRA_MISSING_PARENT 8460
01286 #define ERROR_DS_DRA_PREEMPTED 8461
01287 #define ERROR_DS_DRA_ABANDON_SYNC 8462
01288 #define ERROR_DS_DRA_SHUTDOWN 8463
01289 #define ERROR_DS_DRA_INCOMPATIBLE_PARTIAL_SET 8464
01290 #define ERROR_DS_DRA_SOURCE_IS_PARTIAL_REPLICA 8465
```

```
01291 #define ERROR_DS_DRA_EXTN_CONNECTION_FAILED 8466
01292 #define ERROR_DS_INSTALL_SCHEMA_MISMATCH 8467
01293 #define ERROR_DS_DUP_LINK_ID 8468
01294 #define ERROR_DS_NAME_ERROR_RESOLVING 8469
01295 #define ERROR_DS_NAME_ERROR_NOT_FOUND 8470
01296 #define ERROR_DS_NAME_ERROR_NOT_UNIQUE 8471
01297 #define ERROR_DS_NAME_ERROR_NO_MAPPING 8472
01298 #define ERROR_DS_NAME_ERROR_DOMAIN_ONLY 8473
01299 #define ERROR_DS_NAME_ERROR_NO_SYNTACTICAL_MAPPING 8474
01300 #define ERROR_DS_CONSTRUCTED_ATT_MOD 8475
01301 #define ERROR_DS_WRONG_OM_OBJ_CLASS 8476
01302 #define ERROR_DS_DRA_REPL_PENDING 8477
01303 #define ERROR_DS_DS_REQUIRED 8478
01304 #define ERROR_DS_INVALID_LDAP_DISPLAY_NAME 8479
01305 #define ERROR_DS_NON_BASE_SEARCH 8480
01306 #define ERROR_DS_CANT_RETRIEVE_ATTS 8481
01307 #define ERROR_DS_BACKLINK_WITHOUT_LINK 8482
01308 #define ERROR_DS_EPOCH_MISMATCH 8483
01309 #define ERROR_DS_SRC_NAME_MISMATCH 8484
01310 #define ERROR_DS_SRC_AND_DST_NC_IDENTICAL 8485
01311 #define ERROR_DS_DST_NC_MISMATCH 8486
01312 #define ERROR_DS_NOT_AUTHORITIVE_FOR_DST_NC 8487
01313 #define ERROR_DS_SRC_GUID_MISMATCH 8488
01314 #define ERROR_DS_CANT_MOVE_DELETED_OBJECT 8489
01315 #define ERROR_DS_PDC_OPERATION_IN_PROGRESS 8490
01316 #define ERROR_DS_CROSS_DOMAIN_CLEANUP_REQD 8491
01317 #define ERROR_DS_ILLEGAL_XDOM_MOVE_OPERATION 8492
01318 #define ERROR_DS_CANT_WITH_ACCT_GROUP_MEMBERSHPS 8493
01319 #define ERROR_DS_NC_MUST_HAVE_NC_PARENT 8494
01320 #define ERROR_DS_CR_IMPOSSIBLE_TO_VALIDATE 8495
01321 #define ERROR_DS_DST_DOMAIN_NOT_NATIVE 8496
01322 #define ERROR_DS_MISSING_INFRASTRUCTURE_CONTAINER 8497
01323 #define ERROR_DS_CANT_MOVE_ACCOUNT_GROUP 8498
01324 #define ERROR_DS_CANT_MOVE_RESOURCE_GROUP 8499
01325 #define ERROR_DS_INVALID_SEARCH_FLAG 8500
01326 #define ERROR_DS_NO_TREE_DELETE_ABOVE_NC 8501
01327 #define ERROR_DS_COULDNT_LOCK_TREE_FOR_DELETE 8502
01328 #define ERROR_DS_COULDNT_IDENTIFY_OBJECTS_FOR_TREE_DELETE 8503
01329 #define ERROR_DS_SAM_INIT_FAILURE 8504
01330 #define ERROR_DS_SENSITIVE_GROUP_VIOLATION 8505
01331 #define ERROR_DS_CANT_MOD_PRIMARYGROUPID 8506
01332 #define ERROR_DS_ILLEGAL_BASE_SCHEMA_MOD 8507
01333 #define ERROR_DS_NONSAFE_SCHEMA_CHANGE 8508
01334 #define ERROR_DS_SCHEMA_UPDATE_DISALLOWED 8509
01335 #define ERROR_DS_CANT_CREATE_UNDER_SCHEMA 8510
01336 #define ERROR_DS_INSTALL_NO_SRC_SCH_VERSION 8511
01337 #define ERROR_DS_INSTALL_NO_SCH_VERSION_IN_INIFILE 8512
01338 #define ERROR_DS_INVALID_GROUP_TYPE 8513
01339 #define ERROR_DS_NO_NEST_GLOBALGROUP_IN_MIXEDDOMAIN 8514
01340 #define ERROR_DS_NO_NEST_LOCALGROUP_IN_MIXEDDOMAIN 8515
01341 #define ERROR_DS_GLOBAL_CANT_HAVE_LOCAL_MEMBER 8516
01342 #define ERROR_DS_GLOBAL_CANT_HAVE_UNIVERSAL_MEMBER 8517
01343 #define ERROR_DS_UNIVERSAL_CANT_HAVE_LOCAL_MEMBER 8518
01344 #define ERROR_DS_GLOBAL_CANT_HAVE_CROSSDOMAIN_MEMBER 8519
01345 #define ERROR_DS_LOCAL_CANT_HAVE_CROSSDOMAIN_LOCAL_MEMBER 8520
01346 #define ERROR_DS_HAVE_PRIMARY_MEMBERS 8521
01347 #define ERROR_DS_STRING_SD_CONVERSION_FAILED 8522
01348 #define ERROR_DS_NAMING_MASTER_GC 8523
01349 #define ERROR_DS_LOOKUP_FAILURE 8524
01350 #define ERROR_DS_COULDNT_UPDATE_SPNS 8525
01351 #define ERROR_DS_CANT_RETRIEVE_SD 8526
01352 #define ERROR_DS_KEY_NOT_UNIQUE 8527
01353 #define ERROR_DS_WRONG_LINKED_ATT_SYNTAX 8528
01354 #define ERROR_DS_SAM_NEED_BOOTKEY_PASSWORD 8529
01355 #define ERROR_DS_SAM_NEED_BOOTKEY_FLOPPY 8530
01356 #define ERROR_DS_CANT_START 8531
01357 #define ERROR_DS_INIT_FAILURE 8532
01358 #define ERROR_DS_NO_PKT_PRIVACY_ON_CONNECTION 8533
01359 #define ERROR_DS_SOURCE_DOMAIN_IN_FOREST 8534
01360 #define ERROR_DS_DESTINATION_DOMAIN_NOT_IN_FOREST 8535
01361 #define ERROR_DS_DESTINATION_AUDITING_NOT_ENABLED 8536
01362 #define ERROR_DS_CANT_FIND_DC_FOR_SRC_DOMAIN 8537
01363 #define ERROR_DS_SRC_OBJ_NOT_GROUP_OR_USER 8538
01364 #define ERROR_DS_SRC_SID_EXISTS_IN_FOREST 8539
01365 #define ERROR_DS_SRC_AND_DST_OBJECT_CLASS_MISMATCH 8540
01366 #define ERROR_SAM_INIT_FAILURE 8541
01367 #define ERROR_DS_DRA_SCHEMA_INFO_SHIP 8542
01368 #define ERROR_DS_DRA_SCHEMA_CONFLICT 8543
01369 #define ERROR_DS_DRA_EARLIER_SCHEMA_CONFLICT 8544
01370 #define ERROR_DS_DRA_OBJ_NC_MISMATCH 8545
01371 #define ERROR_DS_NC_STILL_HAS_DSAS 8546
01372 #define ERROR_DS_GC_REQUIRED 8547
01373 #define ERROR_DS_LOCAL_MEMBER_OF_LOCAL_ONLY 8548
01374 #define ERROR_DS_NO_FPO_IN_UNIVERSAL_GROUPS 8549
01375 #define ERROR_DS_CANT_ADD_TO_GC 8550
01376 #define ERROR_DS_NO_CHECKPOINT_WITH_PDC 8551
01377 #define ERROR_DS_SOURCE_AUDITING_NOT_ENABLED 8552
```

```
01378 #define ERROR_DS_CANT_CREATE_IN_NONDOMAIN_NC 8553
01379 #define ERROR_DS_INVALID_NAME_FOR_SPN 8554
01380 #define ERROR_DS_FILTER_USES_CONSTRUCTED_ATTRS 8555
01381 #define ERROR_DS_UNICODEPWD_NOT_IN_QUOTES 8556
01382 #define ERROR_DS_MACHINE_ACCOUNT_QUOTA_EXCEEDED 8557
01383 #define ERROR_DS_MUST_BE_RUN_ON_DST_DC 8558
01384 #define ERROR_DS_SRC_DC_MUST_BE_SP4_OR_GREATER 8559
01385 #define ERROR_DS_CANT_TREE_DELETE_CRITICAL_OBJ 8560
01386 #define DNS_ERROR_RCODE_FORMAT_ERROR 9001
01387 #define DNS_ERROR_RCODE_SERVER_FAILURE 9002
01388 #define DNS_ERROR_RCODE_NAME_ERROR 9003
01389 #define DNS_ERROR_RCODE_NOT_IMPLEMENTED 9004
01390 #define DNS_ERROR_RCODE_REFUSED 9005
01391 #define DNS_ERROR_RCODE_YXDOMAIN 9006
01392 #define DNS_ERROR_RCODE_YXRRSET 9007
01393 #define DNS_ERROR_RCODE_NXRRSET 9008
01394 #define DNS_ERROR_RCODE_NOTAUTH 9009
01395 #define DNS_ERROR_RCODE_NOTZONE 9010
01396 #define DNS_ERROR_RCODE_BADSIG 9016
01397 #define DNS_ERROR_RCODE_BADKEY 9017
01398 #define DNS_ERROR_RCODE_BADTIME 9018
01399 #define DNS_INFO_NO_RECORDS 9501
01400 #define DNS_ERROR_BAD_PACKET 9502
01401 #define DNS_ERROR_NO_PACKET 9503
01402 #define DNS_ERROR_RCODE 9504
01403 #define DNS_ERROR_UNSECURE_PACKET 9505
01404 #define DNS_ERROR_INVALID_TYPE 9551
01405 #define DNS_ERROR_INVALID_IP_ADDRESS 9552
01406 #define DNS_ERROR_INVALID_PROPERTY 9553
01407 #define DNS_ERROR_TRY_AGAIN_LATER 9554
01408 #define DNS_ERROR_NOT_UNIQUE 9555
01409 #define DNS_ERROR_NON_RFC_NAME 9556
01410 #define DNS_STATUS_FQDN 9557
01411 #define DNS_STATUS_DOTTED_NAME 9558
01412 #define DNS_STATUS_SINGLE_PART_NAME 9559
01413 #define DNS_ERROR_INVALID_NAME_CHAR 9560
01414 #define DNS_ERROR_NUMERIC_NAME 9561
01415 #define DNS_ERROR_ZONE_DOES_NOT_EXIST 9601
01416 #define DNS_ERROR_NO_ZONE_INFO 9602
01417 #define DNS_ERROR_INVALID_ZONE_OPERATION 9603
01418 #define DNS_ERROR_ZONE_CONFIGURATION_ERROR 9604
01419 #define DNS_ERROR_ZONE_HAS_NO_SOA_RECORD 9605
01420 #define DNS_ERROR_ZONE_HAS_NO_NS_RECORDS 9606
01421 #define DNS_ERROR_ZONE_LOCKED 9607
01422 #define DNS_ERROR_ZONE_CREATION_FAILED 9608
01423 #define DNS_ERROR_ZONE_ALREADY_EXISTS 9609
01424 #define DNS_ERROR_AUTOZONE_ALREADY_EXISTS 9610
01425 #define DNS_ERROR_INVALID_ZONE_TYPE 9611
01426 #define DNS_ERROR_SECONDARY_REQUIRES_MASTER_IP 9612
01427 #define DNS_ERROR_ZONE_NOT_SECONDARY 9613
01428 #define DNS_ERROR_NEED_SECONDARY_ADDRESSES 9614
01429 #define DNS_ERROR_WINS_INIT_FAILED 9615
01430 #define DNS_ERROR_NEED_WINS_SERVERS 9616
01431 #define DNS_ERROR_NBSTAT_INIT_FAILED 9617
01432 #define DNS_ERROR_SOA_DELETE_INVALID 9618
01433 #define DNS_ERROR_PRIMARY_REQUIRES_DATAFILE 9651
01434 #define DNS_ERROR_INVALID_DATAFILE_NAME 9652
01435 #define DNS_ERROR_DATAFILE_OPEN_FAILURE 9653
01436 #define DNS_ERROR_FILE_WRITEBACK_FAILED 9654
01437 #define DNS_ERROR_DATAFILE_PARSING 9655
01438 #define DNS_ERROR_RECORD_DOES_NOT_EXIST 9701
01439 #define DNS_ERROR_RECORD_FORMAT 9702
01440 #define DNS_ERROR_NODE_CREATION_FAILED 9703
01441 #define DNS_ERROR_UNKNOWN_RECORD_TYPE 9704
01442 #define DNS_ERROR_RECORD_TIMED_OUT 9705
01443 #define DNS_ERROR_NAME_NOT_IN_ZONE 9706
01444 #define DNS_ERROR_CNAME_LOOP 9707
01445 #define DNS_ERROR_NODE_IS_CNAME 9708
01446 #define DNS_ERROR_CNAME_COLLISION 9709
01447 #define DNS_ERROR_RECORD_ONLY_AT_ZONE_ROOT 9710
01448 #define DNS_ERROR_RECORD_ALREADY_EXISTS 9711
01449 #define DNS_ERROR_SECONDARY_DATA 9712
01450 #define DNS_ERROR_NO_CREATE_CACHE_DATA 9713
01451 #define DNS_ERROR_NAME_DOES_NOT_EXIST 9714
01452 #define DNS_WARNING_PTR_CREATE_FAILED 9715
01453 #define DNS_WARNING_DOMAIN_UNDELETED 9716
01454 #define DNS_ERROR_DS_UNAVAILABLE 9717
01455 #define DNS_ERROR_DS_ZONE_ALREADY_EXISTS 9718
01456 #define DNS_ERROR_NO_BOOTFILE_IF_DS_ZONE 9719
01457 #define DNS_INFO_AXFR_COMPLETE 9751
01458 #define DNS_ERROR_AXFR 9752
01459 #define DNS_INFO_ADDED_LOCAL_WINS 9753
01460 #define DNS_STATUS_CONTINUE_NEEDED 9801
01461 #define DNS_ERROR_NO_TCPIP 9851
01462 #define DNS_ERROR_NO_DNS_SERVERS 9852
01463
01464 /* HRESULT values for OLE, SHELL and other Interface stuff */
```

```
01465 /* the codes 4000-40ff are reserved for OLE */
01466 #define NOERROR 0L
01467 #define S_OK ((HRESULT)0L)
01468 #define S_FALSE ((HRESULT)1L)
01469
01470 #define E_PENDING 0x8000000AL
01471
01472
01473 #define E_NOTIMPL 0x80004001L
01474 #define E_NOINTERFACE 0x80004002L
01475 #define E_POINTER 0x80004003L
01476 #define E_ABORT 0x80004004L
01477 #define E_FAIL 0x80004005L
01478 /* FIXME: E_UNSPEC is not a standard value but it is used by
01479  * FileMoniker, IOleLink and DoDragDrop as a return value.
01480  */
01481 #define E_UNSPEC E_FAIL
01482
01483
01484 #define CO_E_INIT_TLS 0x80004006L
01485 #define CO_E_INIT_SHARED_ALLOCATOR 0x80004007L
01486 #define CO_E_INIT_MEMORY_ALLOCATOR 0x80004008L
01487 #define CO_E_INIT_CLASS_CACHE 0x80004009L
01488 #define CO_E_INIT_RPC_CHANNEL 0x8000400AL
01489 #define CO_E_INIT_TLS_SET_CHANNEL_CONTROL 0x8000400BL
01490 #define CO_E_INIT_TLS_CHANNEL_CONTROL 0x8000400CL
01491 #define CO_E_INIT_UNACCEPTED_USER_ALLOCATOR 0x8000400DL
01492 #define CO_E_INIT_SCM_MUTEX_EXISTS 0x8000400EL
01493 #define CO_E_INIT_SCM_FILE_MAPPING_EXISTS 0x8000400FL
01494 #define CO_E_INIT_SCM_MAP_VIEW_OF_FILE 0x80004010L
01495 #define CO_E_INIT_SCM_EXEC_FAILURE 0x80004011L
01496 #define CO_E_INIT_ONLY_SINGLE_THREADED 0x80004012L
01497
01498 #define E_UNEXPECTED 0x8000FFFFL
01499
01500 #define RPC_E_CALL_REJECTED 0x80010001L
01501 #define RPC_E_CALL_CANCELED 0x80010002L
01502 #define RPC_E_CANTPOST_INSENCALL 0x80010003L
01503 #define RPC_E_CANTCALLOUT_INASYNCALL 0x80010004L
01504 #define RPC_E_CANTCALLOUT_INEXTERNALCALL 0x80010005L
01505 #define RPC_E_CONNECTION_TERMINATED 0x80010006L
01506 #define RPC_E_SERVER_DIED 0x80010007L
01507 #define RPC_E_CLIENT_DIED 0x80010008L
01508 #define RPC_E_INVALID_DATAPACKET 0x80010009L
01509 #define RPC_E_CANTTRANSMIT_CALL 0x8001000AL
01510 #define RPC_E_CLIENT_CANTMARSHAL_DATA 0x8001000BL
01511 #define RPC_E_CLIENT_CANTUNMARSHAL_DATA 0x8001000CL
01512 #define RPC_E_SERVER_CANTMARSHAL_DATA 0x8001000DL
01513 #define RPC_E_SERVER_CANTUNMARSHAL_DATA 0x8001000EL
01514 #define RPC_E_INVALID_DATA 0x8001000FL
01515 #define RPC_E_INVALID_PARAMETER 0x80010010L
01516 #define RPC_E_CANTCALLOUT_AGAIN 0x80010011L
01517 #define RPC_E_SERVER_DIED_DNE 0x80010012L
01518 #define RPC_E_SYS_CALL_FAILED 0x80010100L
01519 #define RPC_E_OUT_OF_RESOURCES 0x80010101L
01520 #define RPC_E_ATTEMPTED_MULTITHREAD 0x80010102L
01521 #define RPC_E_NOT_REGISTERED 0x80010103L
01522 #define RPC_E_FAULT 0x80010104L
01523 #define RPC_E_SERVERFAULT 0x80010105L
01524 #define RPC_E_CHANGED_MODE 0x80010106L
01525 #define RPC_E_INVALIDMETHOD 0x80010107L
01526 #define RPC_E_DISCONNECTED 0x80010108L
01527 #define RPC_E_RETRY 0x80010109L
01528 #define RPC_E_SERVERCALL_RETRYLATER 0x8001010AL
01529 #define RPC_E_SERVERCALL_REJECTED 0x8001010BL
01530 #define RPC_E_INVALID_CALldata 0x8001010CL
01531 #define RPC_E_CANTCALLOUT_ININPUTSYNCCALL 0x8001010DL
01532 #define RPC_E_WRONG_THREAD 0x8001010EL
01533 #define RPC_E_THREAD_NOT_INIT 0x8001010FL
01534 #define RPC_E_VERSION_MISMATCH 0x80010110L
01535 #define RPC_E_INVALID_HEADER 0x80010111L
01536 #define RPC_E_INVALID_EXTENSION 0x80010112L
01537 #define RPC_E_INVALID_IPID 0x80010113L
01538 #define RPC_E_INVALID_OBJECT 0x80010114L
01539 #define RPC_S_CALLPENDING 0x80010115L
01540 #define RPC_S_WAITONTIMER 0x80010116L
01541 #define RPC_E_CALL_COMPLETE 0x80010117L
01542 #define RPC_E_UNSECURE_CALL 0x80010118L
01543 #define RPC_E_TOO_LATE 0x80010119L
01544 #define RPC_E_NO_GOOD_SECURITY_PACKAGES 0x8001011AL
01545 #define RPC_E_ACCESS_DENIED 0x8001011BL
01546 #define RPC_E_REMOTE_DISABLED 0x8001011CL
01547 #define RPC_E_INVALID_OBJREF 0x8001011DL
01548 #define RPC_E_NO_CONTEXT 0x8001011EL
01549 #define RPC_E_TIMEOUT 0x8001011FL
01550 #define RPC_E_NO_SYNC 0x80010120L
01551 #define RPC_E_UNEXPECTED 0x8001FFFFL
```

```
01552
01553 #define DISP_E_UNKNOWNINTERFACE 0x80020001L
01554 #define DISP_E_MEMBERNOTFOUND 0x80020003L
01555 #define DISP_E_PARAMNOTFOUND 0x80020004L
01556 #define DISP_E_TYPEMISMATCH 0x80020005L
01557 #define DISP_E_UNKNOWNNAME 0x80020006L
01558 #define DISP_E_NONAMEDARGS 0x80020007L
01559 #define DISP_E_BADVARTYPE 0x80020008L
01560 #define DISP_E_EXCEPTION 0x80020009L
01561 #define DISP_E_OVERFLOW 0x8002000AL
01562 #define DISP_E_BADINDEX 0x8002000BL
01563 #define DISP_E_UNKNOWNLCID 0x8002000CL
01564 #define DISP_E_ARRAYISLOCKED 0x8002000DL
01565 #define DISP_E_BADPARAMCOUNT 0x8002000EL
01566 #define DISP_E_PARAMNOTOPTIONAL 0x8002000FL
01567 #define DISP_E_BADCALLEE 0x80020010L
01568 #define DISP_E_NOTACCOLLECTION 0x80020011L
01569 #define DISP_E_DIVBYZERO 0x80020012L
01570
01571 #define TYPE_E_BUFFERTOOSMALL 0x80028016L
01572 #define TYPE_E_FIELDNOTFOUND 0x80028017L
01573 #define TYPE_E_INVDATAREAD 0x80028018L
01574 #define TYPE_E_UNSUPFORMAT 0x80028019L
01575 #define TYPE_E_REGISTRYACCESS 0x8002801CL
01576 #define TYPE_E_LIBNOTREGISTERED 0x8002801DL
01577 #define TYPE_E_UNDEFINEDTYPE 0x80028027L
01578 #define TYPE_E_QUALIFIEDNAMEDISALLOWED 0x80028028L
01579 #define TYPE_E_INVALIDSTATE 0x80028029L
01580 #define TYPE_E_WRONGTYPEKIND 0x8002802AL
01581 #define TYPE_E_ELEMENTNOTFOUND 0x8002802BL
01582 #define TYPE_E_AMBIGUOUSNAME 0x8002802CL
01583 #define TYPE_E_NAMECONFLICT 0x8002802DL
01584 #define TYPE_E_UNKNOWNLCID 0x8002802EL
01585 #define TYPE_E_DLLFUNCTIONNOTFOUND 0x8002802FL
01586 #define TYPE_E_BADMODULEKIND 0x800288BDL
01587 #define TYPE_E_SIZETOOBIG 0x800288C5L
01588 #define TYPE_E_DUPLICATEID 0x800288C6L
01589 #define TYPE_E_INVALIDID 0x800288CFL
01590 #define TYPE_E_TYPEMISMATCH 0x80028CA0L
01591 #define TYPE_E_OUTOFBOUNDS 0x80028CA1L
01592 #define TYPE_E_IOERROR 0x80028CA2L
01593 #define TYPE_E_CANTCREATETMPFILE 0x80028CA3L
01594 #define TYPE_E_CANTLOADLIBRARY 0x80029C4AL
01595 #define TYPE_E_INCONSISTENTPROPFUNCS 0x80029C83L
01596 #define TYPE_E_CIRCULARTYPE 0x80029C84L
01597
01598 #define STG_S_CONVERTED 0x00030200L
01599 #define STG_S_BLOCK 0x00030201L
01600 #define STG_S_RETRYNOW 0x00030202L
01601 #define STG_S_MONITORING 0x00030203L
01602 #define STG_S_MULTIPLEOPENS 0x00030204L
01603 #define STG_S_CONSOLIDATIONFAILED 0x00030205L
01604 #define STG_S_CANNOTCONSOLIDATE 0x00030206L
01605
01606 #define STG_E_INVALIDFUNCTION 0x80030001L
01607 #define STG_E_FILENOTFOUND 0x80030002L
01608 #define STG_E_PATHNOTFOUND 0x80030003L
01609 #define STG_E_TOOMANYOPENFILES 0x80030004L
01610 #define STG_E_ACCESSDENIED 0x80030005L
01611 #define STG_E_INVALIDHANDLE 0x80030006L
01612 #define STG_E_INSUFFICIENTMEMORY 0x80030008L
01613 #define STG_E_INVALIDPOINTER 0x80030009L
01614 #define STG_E_NOMOREFILES 0x80030012L
01615 #define STG_E_DISKISWRITEPROTECTED 0x80030013L
01616 #define STG_E_SEEKERROR 0x80030019L
01617 #define STG_E_WRITEFAULT 0x8003001DL
01618 #define STG_E_READFAULT 0x8003001EL
01619 #define STG_E_SHAREVIOLATION 0x80030020L
01620 #define STG_E_LOCKVIOLATION 0x80030021L
01621 #define STG_E_FILEALREADYEXISTS 0x80030050L
01622 #define STG_E_INVALIDPARAMETER 0x80030057L
01623 #define STG_E_MEDIUMFULL 0x80030070L
01624 #define STG_E_ABNORMALAPIEXIT 0x800300FAL
01625 #define STG_E_INVALIDHEADER 0x800300FBL
01626 #define STG_E_INVALIDNAME 0x800300FCL
01627 #define STG_E_UNKNOWN 0x800300FDL
01628 #define STG_E_UNIMPLEMENTEDFUNCTION 0x800300FEL
01629 #define STG_E_INVALIDFLAG 0x800300FFL
01630 #define STG_E_INUSE 0x80030100L
01631 #define STG_E_NOTCURRENT 0x80030101L
01632 #define STG_E_REVERTED 0x80030102L
01633 #define STG_E_CANTSAVE 0x80030103L
01634 #define STG_E_OLDFORMAT 0x80030104L
01635 #define STG_E OLDDLL 0x80030105L
01636 #define STG_E_SHAREREQUIRED 0x80030106L
01637 #define STG_E_NOTFILEBASEDSTORAGE 0x80030107L
01638 #define STG_E_EXTANTMARSHALLINGS 0x80030108L
```

```
01639
01640 #define OLE_S_FIRST 0x00040000L
01641 #define OLE_S_USEREG 0x00040000L
01642 #define OLE_S_STATIC 0x00040001L
01643 #define OLE_S_MAC_CLIPFORMAT 0x00040002L
01644 #define OLE_S_LAST 0x000400FFL
01645
01646 #define OLE_E_FIRST 0x80040000L
01647 #define OLE_E_OLEVERB 0x80040000L
01648 #define OLE_E_ADVFE 0x80040001L
01649 #define OLE_E_ENUM_NOMORE 0x80040002L
01650 #define OLE_E_ADVISENOTSUPPORTED 0x80040003L
01651 #define OLE_E_NOCONNECTION 0x80040004L
01652 #define OLE_E_NOTRUNNING 0x80040005L
01653 #define OLE_E_NOCACHE 0x80040006L
01654 #define OLE_E_BLANK 0x80040007L
01655 #define OLE_E_CLASSDIFF 0x80040008L
01656 #define OLE_E_CANT_GETMONIKER 0x80040009L
01657 #define OLE_E_CANT_BINDTOSOURCE 0x8004000AL
01658 #define OLE_E_STATIC 0x8004000BL
01659 #define OLE_E_PROMPTSAVECANCELLED 0x8004000CL
01660 #define OLE_E_INVALIDDIRECT 0x8004000DL
01661 #define OLE_E_WRONGCOMPOBJ 0x8004000EL
01662 #define OLE_E_INVALIDHWND 0x8004000FL
01663 #define OLE_E_NOT_INPLACEACTIVE 0x80040010L
01664 #define OLE_E_CANTCONVERT 0x80040011L
01665 #define OLE_E_NOSTORAGE 0x80040012L
01666 #define DV_E_FORMATETC 0x80040064L
01667 #define DV_E_DVTARGETDEVICE 0x80040065L
01668 #define DV_E_STGMEDIUM 0x80040066L
01669 #define DV_E_STATDATA 0x80040067L
01670 #define DV_E_LINDEX 0x80040068L
01671 #define DV_E_TYMED 0x80040069L
01672 #define DV_E_CLIPFORMAT 0x8004006AL
01673 #define DV_E_DVASPECT 0x8004006BL
01674 #define DV_E_DVTARGETDEVICE_SIZE 0x8004006CL
01675 #define DV_E_NOVIEWOBJECT 0x8004006DL
01676 #define OLE_E_LAST 0x800400FFL
01677
01678 #define DRAGDROP_S_FIRST 0x00040100L
01679 #define DRAGDROP_S_DROP 0x00040100L
01680 #define DRAGDROP_S_CANCEL 0x00040101L
01681 #define DRAGDROP_S_USEDEFAULTCURSORS 0x00040102L
01682 #define DRAGDROP_S_LAST 0x0004010FL
01683
01684 #define DRAGDROP_E_FIRST 0x80040100L
01685 #define DRAGDROP_E_NOTREGISTERED 0x80040100L
01686 #define DRAGDROP_E_ALREADYREGISTERED 0x80040101L
01687 #define DRAGDROP_E_INVALIDHWND 0x80040102L
01688 #define DRAGDROP_E_LAST 0x8004010FL
01689
01690
01691 #define CLASSFACTORY_S_FIRST 0x00040110L
01692 #define CLASSFACTORY_S_LAST 0x0004011FL
01693
01694 #define CLASSFACTORY_E_FIRST 0x80040110L
01695 #define CLASS_E_NOAGGREGATION 0x80040110L
01696 #define CLASS_E_CLASSNOTAVAILABLE 0x80040111L
01697 #define CLASS_E_NOTLICENSED 0x80040112L
01698 #define CLASSFACTORY_E_LAST 0x8004011FL
01699
01700 #define MARSHAL_S_FIRST 0x00040120L
01701 #define MARSHAL_S_LAST 0x0004012FL
01702
01703 #define MARSHAL_E_FIRST 0x80040120L
01704 #define MARSHAL_E_LAST 0x8004012FL
01705
01706 #define DATA_S_FIRST 0x00040130L
01707 #define DATA_S_SAMEFORMATETC 0x00040130L
01708 #define DATA_S_LAST 0x0004013FL
01709
01710 #define DATA_E_FIRST 0x80040130L
01711 #define DATA_E_LAST 0x8004013FL
01712
01713 #define VIEW_S_FIRST 0x00040140L
01714 #define VIEW_S_ALREADY_FROZEN 0x00040140L
01715 #define VIEW_S_LAST 0x0004014FL
01716
01717 #define VIEW_E_FIRST 0x80040140L
01718 #define VIEW_E_DRAW 0x80040140L
01719 #define VIEW_E_LAST 0x8004014FL
01720
01721 #define REGDB_S_FIRST 0x00040150L
01722 #define REGDB_S_LAST 0x0004015FL
01723
01724 #define REGDB_E_FIRST 0x80040150L
01725 #define REGDB_E_READREGDB 0x80040150L
```

```
01726 #define REGDB_E_WRITEREGDB 0x80040151L
01727 #define REGDB_E_KEYMISSING 0x80040152L
01728 #define REGDB_E_INVALIDVALUE 0x80040153L
01729 #define REGDB_E_CLASSNOTREG 0x80040154L
01730 #define REGDB_E_IIDNOTREG 0x80040155L
01731 #define REGDB_E_LAST 0x8004015FL
01732
01733 #define CACHE_S_FIRST 0x00040170L
01734 #define CACHE_S_FORMATETC_NOTSUPPORTED 0x00040170L
01735 #define CACHE_S_SAMECACHE 0x00040171L
01736 #define CACHE_S_SOMECACHES_NOTUPDATED 0x00040172L
01737 #define CACHE_S_LAST 0x0004017FL
01738
01739 #define CACHE_E_FIRST 0x80040170L
01740 #define CACHE_E_NOCACHE_UPDATED 0x80040170L
01741 #define CACHE_E_LAST 0x8004017FL
01742
01743 #define OLEOBJ_S_FIRST 0x00040180L
01744 #define OLEOBJ_S_INVALIDVERB 0x00040180L
01745 #define OLEOBJ_S_CANNOT_DOVERB_NOW 0x00040181L
01746 #define OLEOBJ_S_INVALIDHWN 0x00040182L
01747 #define OLEOBJ_S_LAST 0x0004018FL
01748
01749 #define OLEOBJ_E_FIRST 0x80040180L
01750 #define OLEOBJ_E_NOVERBS 0x80040180L
01751 #define OLEOBJ_E_INVALIDVERB 0x80040181L
01752 #define OLEOBJ_E_LAST 0x8004018FL
01753
01754 #define CLIENTSITE_S_FIRST 0x00040190L
01755 #define CLIENTSITE_S_LAST 0x0004019FL
01756
01757 #define CLIENTSITE_E_FIRST 0x80040190L
01758 #define CLIENTSITE_E_LAST 0x8004019FL
01759
01760 #define INPLACE_S_FIRST 0x000401A0L
01761 #define INPLACE_S_TRUNCATED 0x000401A0L
01762 #define INPLACE_S_LAST 0x000401AFL
01763
01764 #define INPLACE_E_FIRST 0x800401A0L
01765 #define INPLACE_E_NOTUNDOABLE 0x800401A0L
01766 #define INPLACE_E_NOTOOLSPACE 0x800401A1L
01767 #define INPLACE_E_LAST 0x800401AFL
01768
01769 #define ENUM_S_FIRST 0x000401B0L
01770 #define ENUM_S_LAST 0x000401BFL
01771
01772 #define ENUM_E_FIRST 0x800401B0L
01773 #define ENUM_E_LAST 0x800401BFL
01774
01775 #define CONVERT10_S_FIRST 0x000401C0L
01776 #define CONVERT10_S_NO_PRESENTATION 0x000401C0L
01777 #define CONVERT10_S_LAST 0x000401CFL
01778
01779 #define CONVERT10_E_FIRST 0x800401C0L
01780 #define CONVERT10_E_OLESTREAM_GET 0x800401C0L
01781 #define CONVERT10_E_OLESTREAM_PUT 0x800401C1L
01782 #define CONVERT10_E_OLESTREAM_FMT 0x800401C2L
01783 #define CONVERT10_E_OLESTREAM_BITMAP_TO_DIB 0x800401C3L
01784 #define CONVERT10_E_STG_FMT 0x800401C4L
01785 #define CONVERT10_E_STG_NO_STD_STREAM 0x800401C5L
01786 #define CONVERT10_E_STG_DIB_TO_BITMAP 0x800401C6L
01787 #define CONVERT10_E_LAST 0x800401CFL
01788
01789 #define CLIPBRD_S_FIRST 0x000401D0L
01790 #define CLIPBRD_S_LAST 0x000401DFL
01791
01792 #define CLIPBRD_E_FIRST 0x800401D0L
01793 #define CLIPBRD_E_LAST 0x800401DFL
01794 #define CLIPBRD_E_CANT_OPEN 0x800401D0L
01795 #define CLIPBRD_E_CANT_EMPTY 0x800401D1L
01796 #define CLIPBRD_E_CANT_SET 0x800401D2L
01797 #define CLIPBRD_E_BAD_DATA 0x800401D3L
01798 #define CLIPBRD_E_CANT_CLOSE 0x800401D4L
01799
01800 #define MK_S_FIRST 0x000401E0L
01801 #define MK_S_REDUCE_TO_SELF 0x000401E2L
01802 #define MK_S_ME 0x000401E4L
01803 #define MK_S_HIM 0x000401E5L
01804 #define MK_S_US 0x000401E6L
01805 #define MK_S_MONIKERALREADYREGISTERED 0x000401E7L
01806 #define MK_S_LAST 0x000401EFL
01807
01808 #define MK_E_FIRST 0x800401E0L
01809 #define MK_E_CONNECTMANUALLY 0x800401E0L
01810 #define MK_E_EXCEEDEDDEADLINE 0x800401E1L
01811 #define MK_E_NEEDGENERIC 0x800401E2L
01812 #define MK_E_UNAVAILABLE 0x800401E3L
```

```
01813 #define MK_E_SYNTAX 0x800401E4L
01814 #define MK_E_NOOBJECT 0x800401E5L
01815 #define MK_E_INVALIDEXTENSION 0x800401E6L
01816 #define MK_E_INTERMEDIATEINTERFACENOTSUPPORTED 0x800401E7L
01817 #define MK_E_NOTBINDABLE 0x800401E8L
01818 #define MK_E_NOTBOUND 0x800401E9L
01819 #define MK_E_CANTOPENFILE 0x800401EAL
01820 #define MK_E_MUSTBOTHERUSER 0x800401EBL
01821 #define MK_E_NOINVERSE 0x800401ECL
01822 #define MK_E_NOSTORAGE 0x800401EDL
01823 #define MK_E_NOPREFIX 0x800401EEL
01824 #define MK_E_ENUMERATION_FAILED 0x800401EFL
01825 #define MK_E_LAST 0x800401EFL
01826
01827 #define CO_S_FIRST 0x000401F0L
01828 #define CO_S_LAST 0x000401FFL
01829
01830 #define CO_E_FIRST 0x800401F0L
01831 #define CO_E_NOTINITIALIZED 0x800401F0L
01832 #define CO_E_ALREADYINITIALIZED 0x800401F1L
01833 #define CO_E_CANTDETERMINECLASS 0x800401F2L
01834 #define CO_E_CLASSSTRING 0x800401F3L
01835 #define CO_E_IIDSTRING 0x800401F4L
01836 #define CO_E_APPNOTFOUND 0x800401F5L
01837 #define CO_E_APPSINGLEUSE 0x800401F6L
01838 #define CO_E_ERRORINAPP 0x800401F7L
01839 #define CO_E_DLLNOTFOUND 0x800401F8L
01840 #define CO_E_ERRORINDLL 0x800401F9L
01841 #define CO_E_WRONGSFORAPP 0x800401FAL
01842 #define CO_E_OBJNOTREG 0x800401FBL
01843 #define CO_E_OBJISREG 0x800401FCL
01844 #define CO_E_OBJNOTCONNECTED 0x800401FDL
01845 #define CO_E_APPDIDNTREG 0x800401FEL
01846 #define CO_E_RELEASED 0x800401FFL
01847 #define CO_E_LAST 0x800401FFL
01848 #define CO_E_FAILEDTOIMPERSONATE 0x80040200L
01849 #define CO_E_FAILEDTOGETSECCTX 0x80040201L
01850 #define CO_E_FAILEDTOOPENTHREADTOKEN 0x80040202L
01851 #define CO_E_FAILEDTOGETTOKENINFO 0x80040203L
01852 #define CO_E_TRUSTEEDOESNTMATCHCLIENT 0x80040204L
01853 #define CO_E_FAILEDTOQUERYCLIENTBLANKET 0x80040205L
01854 #define CO_E_FAILEDTOSETDACL 0x80040206L
01855 #define CO_E_ACCESSCHECKFAILED 0x80040207L
01856 #define CO_E_NETACCESSAPIFAILED 0x80040208L
01857 #define CO_E_WRONGTRUSTEENAMESYNTAX 0x80040209L
01858 #define CO_E_INVALIDSID 0x8004020AL
01859 #define CO_E_CONVERSIONFAILED 0x8004020BL
01860 #define CO_E_NOMATCHINGSIDFOUND 0x8004020CL
01861 #define CO_E_LOOKUPACCSIDFAILED 0x8004020DL
01862 #define CO_E_NOMATCHINGNAMEFOUND 0x8004020EL
01863 #define CO_E_LOOKUPACNAMEFAILED 0x8004020FL
01864 #define CO_E_SETSERLHNDLFAILED 0x80040210L
01865 #define CO_E_FAILEDTOGETWINDIR 0x80040211L
01866 #define CO_E_PATHTOOLONG 0x80040212L
01867 #define CO_E_FAILEDTOGENUUID 0x80040213L
01868 #define CO_E_FAILEDTOCREATEFILE 0x80040214L
01869 #define CO_E_FAILEDTOCLOSEHANDLE 0x80040215L
01870 #define CO_E_EXCEEDSYSACLLIMIT 0x80040216L
01871 #define CO_E_ACESINWRONGORDER 0x80040217L
01872 #define CO_E_INCOMPATIBLESTREAMVERSION 0x80040218L
01873 #define CO_E_FAILEDTOOPENPROCESSTOKEN 0x80040219L
01874 #define CO_E_DECODEFAILED 0x8004021AL
01875 #define CO_E_ACNOTINITIALIZED 0x8004021BL
01876
01877 #define E_ACCESSDENIED 0x80070005L
01878 #define E_HANDLE 0x80070006L
01879 #define E_OUTOFMEMORY 0x8007000EL
01880 #define E_INVALIDARG 0x80070057L
01881
01882 /* For IksPropertySets */
01883 #define E_PROP_ID_UNSUPPORTED 0x80070490L
01884 #define E_PROP_SET_UNSUPPORTED 0x80070492L
01885
01886 #define CO_S_NOTALLINTERFACES 0x00080012L
01887
01888 #define CO_E_CLASS_CREATE_FAILED 0x80080001L
01889 #define CO_E_SCM_ERROR 0x80080002L
01890 #define CO_E_SCM_RPC_FAILURE 0x80080003L
01891 #define CO_E_BAD_PATH 0x80080004L
01892 #define CO_E_SERVER_EXEC_FAILURE 0x80080005L
01893 #define CO_E_OBJSRV_RPC_FAILURE 0x80080006L
01894 #define MK_E_NO_NORMALIZED 0x80080007L
01895 #define CO_E_SERVER_STOPPING 0x80080008L
01896 #define MEM_E_INVALID_ROOT 0x80080009L
01897 #define MEM_E_INVALID_LINK 0x80080010L
01898 #define MEM_E_INVALID_SIZE 0x80080011L
01899
```

```

01900
01901 #endif /* __WINE_WINERROR_H */

```

## 5.11 wingdi.h

```

00001 #ifndef _WINGDI_
00002 #define _WINGDI_
00003 #ifndef NOGDI
00004
00005 #ifdef __cplusplus
00006 extern "C" {
00007 #endif
00008
00009 typedef struct _ABCFLOAT {
00010     FLOAT    abcfA;
00011     FLOAT    abcfB;
00012     FLOAT    abcfC;
00013 } ABCFLOAT, *PABCFLOAT, *LPABCFLOAT;
00014
00015 #define FONTMAPPER_MAX 10
00016
00017 typedef struct
00018 {
00019     WORD    wFirst;
00020     WORD    wSecond;
00021     INT     iKernAmount;
00022 } KERNINGPAIR, *LPKERNINGPAIR;
00023
00024 typedef struct tagPIXELFORMATDESCRIPTOR {
00025     WORD    nSize;
00026     WORD    nVersion;
00027     DWORD   dwFlags;
00028     BYTE    iPixelFormat;
00029     BYTE    cColorBits;
00030     BYTE    cRedBits;
00031     BYTE    cRedShift;
00032     BYTE    cGreenBits;
00033     BYTE    cGreenShift;
00034     BYTE    cBlueBits;
00035     BYTE    cBlueShift;
00036     BYTE    cAlphaBits;
00037     BYTE    cAlphaShift;
00038     BYTE    cAccumBits;
00039     BYTE    cAccumRedBits;
00040     BYTE    cAccumGreenBits;
00041     BYTE    cAccumBlueBits;
00042     BYTE    cAccumAlphaBits;
00043     BYTE    cDepthBits;
00044     BYTE    cStencilBits;
00045     BYTE    cAuxBuffers;
00046     BYTE    iLayerType;
00047     BYTE    bReserved;
00048     DWORD   dwLayerMask;
00049     DWORD   dwVisibleMask;
00050     DWORD   dwDamageMask;
00051 } PIXELFORMATDESCRIPTOR, *PPIXELFORMATDESCRIPTOR, *LPPIXELFORMATDESCRIPTOR;
00052
00053 #define PFD_TYPE_RGBA          0
00054 #define PFD_TYPE_COLORINDEX    1
00055
00056 #define PFD_MAIN_PLANE         0
00057 #define PFD_OVERLAY_PLANE      1
00058 #define PFD_UNDERLAY_PLANE     (-1)
00059
00060 #define PFD_DOUBLEBUFFER       0x00000001
00061 #define PFD_STEREO             0x00000002
00062 #define PFD_DRAW_TO_WINDOW     0x00000004
00063 #define PFD_DRAW_TO_BITMAP     0x00000008
00064 #define PFD_SUPPORT_GDI        0x00000010
00065 #define PFD_SUPPORT_OPENGL     0x00000020
00066 #define PFD_GENERIC_FORMAT     0x00000040
00067 #define PFD_NEED_PALETTE       0x00000080
00068 #define PFD_NEED_SYSTEM_PALETTE 0x00000100
00069 #define PFD_SWAP_EXCHANGE      0x00000200
00070 #define PFD_SWAP_COPY          0x00000400
00071 #define PFD_SWAP_LAYER_BUFFERS 0x00000800
00072 #define PFD_GENERIC_ACCELERATED 0x00001000
00073
00074 #define PFD_DEPTH_DONTCARE     0x20000000
00075 #define PFD_DOUBLEBUFFER_DONTCARE 0x40000000
00076 #define PFD_STEREO_DONTCARE    0x80000000
00077
00078 typedef struct tagCOLORADJUSTMENT
00079 {

```

```

00080     WORD    caSize;
00081     WORD    caFlags;
00082     WORD    caIlluminantIndex;
00083     WORD    caRedGamma;
00084     WORD    caGreenGamma;
00085     WORD    caBlueGamma;
00086     WORD    caReferenceBlack;
00087     WORD    caReferenceWhite;
00088     SHORT   caContrast;
00089     SHORT   caBrightness;
00090     SHORT   caColorfulness;
00091     SHORT   caRedGreenTint;
00092 } COLORADJUSTMENT, *PCOLORADJUSTMENT, *LPCOLORADJUSTMENT;
00093
00094 #define CA_NEGATIVE          0x0001
00095 #define CA_LOG_FILTER        0x0002
00096
00097 #define ILLUMINANT_DEVICE_DEFAULT  0
00098 #define ILLUMINANT_A                1
00099 #define ILLUMINANT_B                2
00100 #define ILLUMINANT_C                3
00101 #define ILLUMINANT_D50              4
00102 #define ILLUMINANT_D55              5
00103 #define ILLUMINANT_D65              6
00104 #define ILLUMINANT_D75              7
00105 #define ILLUMINANT_F2               8
00106 #define ILLUMINANT_MAX_INDEX        ILLUMINANT_F2
00107
00108 #define ILLUMINANT_TUNGSTEN          ILLUMINANT_A
00109 #define ILLUMINANT_DAYLIGHT          ILLUMINANT_C
00110 #define ILLUMINANT_FLUORESCENT       ILLUMINANT_F2
00111 #define ILLUMINANT_NTSC              ILLUMINANT_C
00112
00113 #define RGB_GAMMA_MIN                (WORD) 02500
00114 #define RGB_GAMMA_MAX                (WORD) 65000
00115
00116 #define REFERENCE_WHITE_MIN          (WORD) 6000
00117 #define REFERENCE_WHITE_MAX          (WORD) 10000
00118 #define REFERENCE_BLACK_MIN          (WORD) 0
00119 #define REFERENCE_BLACK_MAX          (WORD) 4000
00120
00121 #define COLOR_ADJ_MIN                ((SHORT) -100)
00122 #define COLOR_ADJ_MAX                ((SHORT) 100)
00123
00124 typedef LONG FXPT16DOT16, *LPFXPT16DOT16;
00125 typedef LONG FXPT2DOT30, *LPFXPT2DOT30;
00126 typedef LONG LCSCSTYPE;
00127 typedef LONG LCSGAMUTMATCH;
00128
00129 #define LCS_CALIBRATED_RGB            0x00000000L
00130 #define LCS_DEVICE_RGB                0x00000001L
00131 #define LCS_DEVICE_CMYK              0x00000002L
00132
00133 #define LCS_GM_BUSINESS               0x00000001L
00134 #define LCS_GM_GRAPHICS              0x00000002L
00135 #define LCS_GM_IMAGES                0x00000004L
00136
00137 #define CM_OUT_OF_GAMUT               255
00138 #define CM_IN_GAMUT                  0
00139
00140 typedef struct tagCIEXYZ
00141 {
00142     FXPT2DOT30 ciexyzX;
00143     FXPT2DOT30 ciexyzY;
00144     FXPT2DOT30 ciexyzZ;
00145 } CIEXYZ, *LPCIEXYZ;
00146
00147 typedef struct tagCIEXYZTRIPLE
00148 {
00149     CIEXYZ ciexyzRed;
00150     CIEXYZ ciexyzGreen;
00151     CIEXYZ ciexyzBlue;
00152 } CIEXYZTRIPLE, *LPCIEXYZTRIPLE;
00153
00154 typedef struct tagLOGCOLORSPACEA
00155 {
00156     DWORD lcsSignature;
00157     DWORD lcsVersion;
00158     DWORD lcsSize;
00159     LCSCSTYPE lcsCSType;
00160     LCSGAMUTMATCH lcsIntent;
00161     CIEXYZTRIPLE lcsEndpoints;
00162     DWORD lcsGammaRed;
00163     DWORD lcsGammaGreen;
00164     DWORD lcsGammaBlue;
00165     CHAR lcsFilename[MAX_PATH];
00166 } LOGCOLORSPACEA, *LPLOGCOLORSPACEA;

```

```

00167
00168 typedef struct tagLOGCOLORSPACEW
00169 {
00170     DWORD lcsSignature;
00171     DWORD lcsVersion;
00172     DWORD lcsSize;
00173     LCSCSTYPE lcsCSType;
00174     LCSGAMUTMATCH lcsIntent;
00175     CIEXYZTRIPLE lcsEndpoints;
00176     DWORD lcsGammaRed;
00177     DWORD lcsGammaGreen;
00178     DWORD lcsGammaBlue;
00179     WCHAR lcsFilename[MAX_PATH];
00180 } LOGCOLORSPACEW, *LPLOGCOLORSPACEW;
00181
00182 DECL_WINELIB_TYPE_AW(LPLOGCOLORSPACE)
00183 DECL_WINELIB_TYPE_AW(LOGCOLORSPACE)
00184
00185 #define DC_FIELDS 1
00186 #define DC_PAPERS 2
00187 #define DC_PAPERSIZE 3
00188 #define DC_MINEXTENT 4
00189 #define DC_MAXEXTENT 5
00190 #define DC_BINS 6
00191 #define DC_DUPLEX 7
00192 #define DC_SIZE 8
00193 #define DC_EXTRA 9
00194 #define DC_VERSION 10
00195 #define DC_DRIVER 11
00196 #define DC_BINNAMES 12
00197 #define DC_ENUMRESOLUTIONS 13
00198 #define DC_FILEDEPENDENCIES 14
00199 #define DC_TRUETYPE 15
00200 #define DC_PAPERNAME 16
00201 #define DC_ORIENTATION 17
00202 #define DC_COPIES 18
00203 #define DC_BINADJUST 19
00204 #define DC_EMF_COMPLIANT 20
00205 #define DC_DATATYPE_PRODUCED 21
00206 #define DC_COLLATE 22
00207 #define DC_MANUFACTURER 23
00208 #define DC_MODEL 24
00209 #define DC_PERSONALITY 25
00210 #define DC_PRINTRATE 26
00211 #define DC_PRINTRATEUNIT 27
00212 #define DC_PRINTERMEM 28
00213 #define DC_MEDIAREADY 29
00214 #define DC_STAPLE 30
00215 #define DC_PRINTRATEPPM 31
00216 #define DC_COLORDEVICE 32
00217 #define DC_NUP 33
00218
00219 #define DCTT_BITMAP 0x00000001L
00220 #define DCTT_DOWNLOAD 0x00000002L
00221 #define DCTT_SUBDEV 0x00000004L
00222 #define DCTT_DOWNLOAD_OUTLINE 0x00000008L
00223
00224 #define DCBA_FACEUPNONE 0x0000
00225 #define DCBA_FACEUPCENTER 0x0001
00226 #define DCBA_FACEUPLLEFT 0x0002
00227 #define DCBA_FACEUPRIGHT 0x0003
00228 #define DCBA_FACEDOWNNONE 0x0100
00229 #define DCBA_FACEDOWNCENTER 0x0101
00230 #define DCBA_FACEDOWNLEFT 0x0102
00231 #define DCBA_FACEDOWNRIGHT 0x0103
00232
00233 #define PRINTRATEUNIT_PPM 1
00234 #define PRINTRATEUNIT_CPS 2
00235 #define PRINTRATEUNIT_LPM 3
00236 #define PRINTRATEUNIT_IPM 4
00237
00238 /* Flag returned from Escape QUERYDIBSUPPORT */
00239 #define QDI_SETDIBITS 1
00240 #define QDI_GETDIBITS 2
00241 #define QDI_DIBTOSCREEN 4
00242 #define QDI_STRETCHDIB 8
00243
00244
00245 /* GDI Escape commands */
00246 #define NEWFRAME 1
00247 #define ABORTDOC 2
00248 #define NEXTBAND 3
00249 #define SETCOLORTABLE 4
00250 #define GETCOLORTABLE 5
00251 #define FLUSHOUTPUT 6
00252 #define DRAFTMODE 7
00253 #define QUERYESCSUPPORT 8

```

```

00254 #define SETABORTPROC          9
00255 #define STARTDOC              10
00256 #define ENDDOC                11
00257 #define GETPHYSPAGE_SIZE      12
00258 #define GETPRINTINGOFFSET     13
00259 #define GETSCALINGFACTOR      14
00260 #define MFCOMMENT              15
00261 #define GETPENWIDTH           16
00262 #define SETCOPYCOUNT         17
00263 #define SELECTPAPERSOURCE      18
00264 #define DEVICEDATA             19
00265 #define PASSTHROUGH           19
00266 #define GETTECHNOLGY           20
00267 #define GETTECHNOLOGY          20 /* yes, both of them */
00268 #define SETLINECAP             21
00269 #define SETLINEJOIN            22
00270 #define SETMITERLIMIT          23
00271 #define BANDINFO               24
00272 #define DRAWPATTERNRECT        25
00273 #define GETVECTORPEN_SIZE      26
00274 #define GETVECTORBRUSH_SIZE    27
00275 #define ENABLEDUPLEX           28
00276 #define GETSETPAPERBINS        29
00277 #define GETSETPRINTORIENT      30
00278 #define ENUMPAPERBINS          31
00279 #define SETDIBSCALING          32
00280 #define EPSPRINTING            33
00281 #define ENUMPAPERMETRICS        34
00282 #define GETSETPAPERMETRICS      35
00283 #define POSTSCRIPT_DATA        37
00284 #define POSTSCRIPT_IGNORE      38
00285 #define MOUSETRAILS            39
00286 #define GETDEVICEUNITS         42
00287
00288 #define DESKTOPVERTRES          117
00289 #define DESKTOPHORZRES         118
00290
00291 #define GETEXTENDEDTEXTMETRICS 256
00292 #define GETEXTENTTABLE          257
00293 #define GETPAIRKERNTABLE        258
00294 #define GETTRACKKERNTABLE       259
00295 #define EXTTEXTOUT              512
00296 #define GETFACENAME            513
00297 #define DOWNLOADFACE            514
00298 #define ENABLERELATIVEWIDTHS    768
00299 #define ENABLEPAIRKERNING       769
00300 #define SETKERNTRACK            770
00301 #define SETALLJUSTVALUES        771
00302 #define SETCHARSET              772
00303
00304 #define STRETCHBLT              2048
00305 #define GETSETSCREENPARAMS      3072
00306 #define QUERYDIBSUPPORT         3073
00307 #define BEGIN_PATH              4096
00308 #define CLIP_TO_PATH            4097
00309 #define END_PATH                4098
00310 #define EXT_DEVICE_CAPS         4099
00311 #define RESTORE_CTM             4100
00312 #define SAVE_CTM                4101
00313 #define SET_ARC_DIRECTION        4102
00314 #define SET_BACKGROUND_COLOR    4103
00315 #define SET_POLY_MODE            4104
00316 #define SET_SCREEN_ANGLE        4105
00317 #define SET_SPREAD              4106
00318 #define TRANSFORM_CTM           4107
00319 #define SET_CLIP_BOX            4108
00320 #define SET_BOUNDS              4109
00321 #define SET_MIRROR_MODE         4110
00322 #define OPENCHANNEL             4110
00323 #define DOWNLOADHEADER          4111
00324 #define CLOSECHANNEL            4112
00325 #define POSTSCRIPT_PASSTHROUGH  4115
00326 #define ENCAPSULATED_POSTSCRIPT 4116
00327 #define POSTSCRIPT_IDENTIFY     4117
00328 #define POSTSCRIPT_INJECTION    4118
00329
00330 /* for POSTSCRIPT_IDENTIFY */
00331 #define PSIDENT_GDICENTRIC      0
00332 #define PSIDENT_PSCENTRIC       1
00333
00334
00335 #define QDI_SETDIBITS           1
00336 #define QDI_GETDIBITS           2
00337 #define QDI_DIBTOSCREEN         4
00338 #define QDI_STRETCHDIB          8
00339
00340 /* Spooler Error Codes */

```

```

00341 #define SP_NOTREPORTED 0x4000
00342 #define SP_ERROR (-1)
00343 #define SP_APPABORT (-2)
00344 #define SP_USERABORT (-3)
00345 #define SP_OUTOFDISK (-4)
00346 #define SP_OUTOFMEMORY (-5)
00347
00348 #define PR_JOBSTATUS 0
00349
00350 /* Raster operations */
00351
00352 #define R2_BLACK 1
00353 #define R2_NOTMERGEPEN 2
00354 #define R2_MASKNOTPEN 3
00355 #define R2_NOTCOPYPEN 4
00356 #define R2_MASKPENNOT 5
00357 #define R2_NOT 6
00358 #define R2_XORPEN 7
00359 #define R2_NOTMASKPEN 8
00360 #define R2_MASKPEN 9
00361 #define R2_NOTXORPEN 10
00362 #define R2_NOP 11
00363 #define R2_MERGENOTPEN 12
00364 #define R2_COPYPEN 13
00365 #define R2_MERGEENNOT 14
00366 #define R2_MERGEEN 15
00367 #define R2_WHITE 16
00368
00369 #define SRCCOPY 0xcc0020
00370 #define SRCPAINT 0xee0086
00371 #define SRCAND 0x8800c6
00372 #define SRCINVERT 0x660046
00373 #define SRCERASE 0x440328
00374 #define NOTSRCCOPY 0x330008
00375 #define NOTSRCERASE 0x1100a6
00376 #define MERGECOPY 0xc000ca
00377 #define MERGEPAINT 0xbb0226
00378 #define PATCOPY 0xf00021
00379 #define PATPAINT 0xfb0a09
00380 #define PATINVERT 0x5a0049
00381 #define DSTINVERT 0x550009
00382 #define BLACKNESS 0x000042
00383 #define WHITENESS 0xff0062
00384
00385 /* StretchBlt() modes */
00386 #define BLACKONWHITE 1
00387 #define WHITEONBLACK 2
00388 #define COLORONCOLOR 3
00389 #define HALFTONE 4
00390 #define MAXSTRETCHBLTMODE 4
00391
00392 #define STRETCH_ANDSCANS BLACKONWHITE
00393 #define STRETCH_ORSCANS WHITEONBLACK
00394 #define STRETCH_DELETESCANS COLORONCOLOR
00395 #define STRETCH_HALFTONE HALFTONE
00396
00397 /* Colors */
00398
00399 #define RGB(r,g,b) ((COLORREF)((r) | ((g) << 8) | ((b) << 16)))
00400 #define PALETTE_RGB(r,g,b) (0x02000000 | RGB(r,g,b))
00401 #define PALETTE_INDEX(i) ((COLORREF)(0x01000000 | (WORD)(i)))
00402
00403 #define GetRValue(rgb) ((rgb) & 0xff)
00404 #define GetGValue(rgb) (((rgb) >> 8) & 0xff)
00405 #define GetBValue(rgb) (((rgb) >> 16) & 0xff)
00406
00407 #define GetKValue(cmyk) ((BYTE) (cmyk) )
00408 #define GetYValue(cmyk) ((BYTE) ((cymk) >> 8))
00409 #define GetMValue(cmyk) ((BYTE) ((cymk) >> 16))
00410 #define GetCValue(cmyk) ((BYTE) ((cymk) >> 24))
00411
00412 #define CMYK(c,m,y,k) ((COLORREF)(( (BYTE)(k) | ((WORD)((BYTE)(y)<<8)) | ((DWORD)(BYTE)(m)<<16)) | ((DWORD)(BYTE)(c)<<24)))
00413
00414
00415 #define ICM_OFF 1
00416 #define ICM_ON 2
00417 #define ICM_QUERY 3
00418
00419 /* Bounds Accumulation APIs */
00420 #define DCB_RESET 0x0001
00421 #define DCB_ACCUMULATE 0x0002
00422 #define DCB_DIRTY DCB_ACCUMULATE
00423 #define DCB_SET (DCB_RESET | DCB_ACCUMULATE)
00424 #define DCB_ENABLE 0x0004
00425 #define DCB_DISABLE 0x0008
00426

```

```

00427 typedef struct
00428 {
00429     LONG paXCount;
00430     LONG paYCount;
00431     LONG paXExt;
00432     LONG paYExt;
00433     BYTE paRGBs;
00434 } PELARRAY, *PPELARRAY, *LPPELARRAY;
00435
00436 /* Bitmaps */
00437
00438 typedef struct
00439 {
00440     INT bmType;
00441     INT bmWidth;
00442     INT bmHeight;
00443     INT bmWidthBytes;
00444     WORD bmPlanes;
00445     WORD bmBitsPixel;
00446     LPVOID bmBits;
00447 } BITMAP, *PBITMAP, *LPBITMAP;
00448
00449
00450 /* Brushes */
00451
00452 typedef struct
00453 {
00454     UINT lbStyle;
00455     COLORREF lbColor;
00456     INT lbHatch;
00457 } LOGBRUSH, *PLOGBRUSH, *LPLOGBRUSH;
00458
00459 typedef LOGBRUSH PATTERN, *PPATTERN, *LPPATTERN;
00460
00461
00462 /* Brush styles */
00463 #define BS_SOLID 0
00464 #define BS_NULL 1
00465 #define BS_HOLLOW 1
00466 #define BS_HATCHED 2
00467 #define BS_PATTERN 3
00468 #define BS_INDEXED 4
00469 #define BS_DIBPATTERN 5
00470 #define BS_DIBPATTERNPT 6
00471 #define BS_PATTERN8X8 7
00472 #define BS_DIBPATTERN8X8 8
00473 #define BS_MONOPATTERN 9
00474
00475 /* Hatch styles */
00476 #define HS_HORIZONTAL 0
00477 #define HS_VERTICAL 1
00478 #define HS_FDIAGONAL 2
00479 #define HS_BDIAGONAL 3
00480 #define HS_CROSS 4
00481 #define HS_DIAGCROSS 5
00482
00483 /* Fonts */
00484
00485 #define LF_FACESIZE 32
00486 #define LF_FULLFACESIZE 64
00487
00488 #define RASTER_FONTTYPE 0x0001
00489 #define DEVICE_FONTTYPE 0x0002
00490 #define TRUETYPE_FONTTYPE 0x0004
00491
00492 typedef struct
00493 {
00494     LONG lfHeight;
00495     LONG lfWidth;
00496     LONG lfEscapement;
00497     LONG lfOrientation;
00498     LONG lfWeight;
00499     BYTE lfItalic;
00500     BYTE lfUnderline;
00501     BYTE lfStrikeOut;
00502     BYTE lfCharSet;
00503     BYTE lfOutPrecision;
00504     BYTE lfClipPrecision;
00505     BYTE lfQuality;
00506     BYTE lfPitchAndFamily;
00507     CHAR lfFaceName[LF_FACESIZE];
00508 } LOGFONTA, *PLOGFONTA, *LPLOGFONTA;
00509
00510 typedef struct
00511 {
00512     LONG lfHeight;
00513     LONG lfWidth;

```

```

00514     LONG     lfEscapement;
00515     LONG     lfOrientation;
00516     LONG     lfWeight;
00517     BYTE     lfItalic;
00518     BYTE     lfUnderline;
00519     BYTE     lfStrikeOut;
00520     BYTE     lfCharSet;
00521     BYTE     lfOutPrecision;
00522     BYTE     lfClipPrecision;
00523     BYTE     lfQuality;
00524     BYTE     lfPitchAndFamily;
00525     WCHAR     lfFaceName[LF_FACESIZE];
00526 } LOGFONTW, *PLOGFONTW, *LPLOGFONTW;
00527
00528 DECL_WINELIB_TYPE_AW(LOGFONT)
00529 DECL_WINELIB_TYPE_AW(PLOGFONT)
00530 DECL_WINELIB_TYPE_AW(LPLOGFONT)
00531
00532 typedef struct
00533 {
00534     LOGFONTA elfLogFont;
00535     BYTE     elfFullName[LF_FULLFACESIZE];
00536     BYTE     elfStyle[LF_FACESIZE];
00537 } ENUMLOGFONTA, *LPENUMLOGFONTA;
00538
00539 typedef struct
00540 {
00541     LOGFONTW elfLogFont;
00542     WCHAR     elfFullName[LF_FULLFACESIZE];
00543     WCHAR     elfStyle[LF_FACESIZE];
00544 } ENUMLOGFONTW, *LPENUMLOGFONTW;
00545
00546 DECL_WINELIB_TYPE_AW(ENUMLOGFONT)
00547 DECL_WINELIB_TYPE_AW(LPENUMLOGFONT)
00548
00549 typedef struct
00550 {
00551     LOGFONTA elfLogFont;
00552     BYTE     elfFullName[LF_FULLFACESIZE];
00553     BYTE     elfStyle[LF_FACESIZE];
00554     BYTE     elfScript[LF_FACESIZE];
00555 } ENUMLOGFONTEXA, *LPENUMLOGFONTEXA;
00556
00557 typedef struct
00558 {
00559     LOGFONTW elfLogFont;
00560     WCHAR     elfFullName[LF_FULLFACESIZE];
00561     WCHAR     elfStyle[LF_FACESIZE];
00562     WCHAR     elfScript[LF_FACESIZE];
00563 } ENUMLOGFONTEXW, *LPENUMLOGFONTEXW;
00564
00565 DECL_WINELIB_TYPE_AW(ENUMLOGFONTEX)
00566 DECL_WINELIB_TYPE_AW(LPENUMLOGFONTEX)
00567
00568 /*
00569  * The FONTSIGNATURE tells which Unicode ranges and which code pages
00570  * have glyphs in a font.
00571  *
00572  * fsUsb 128-bit bitmap. The most significant bits are 10 (magic number).
00573  *       The remaining 126 bits map the Unicode ISO 10646 subranges
00574  *       for which the font provides glyphs.
00575  *
00576  * fsCsb 64-bit bitmap. The low 32 bits map the Windows codepages for
00577  *       which the font provides glyphs. The high 32 bits are for
00578  *       non Windows codepages.
00579  */
00580 typedef struct
00581 {
00582     DWORD fsUsb[4];
00583     DWORD fsCsb[2];
00584 } FONTSIGNATURE, *PFONTSIGNATURE, *LPFONTSIGNATURE;
00585
00586 typedef struct
00587 {
00588     UINT ciCharset; /* character set */
00589     UINT ciACP; /* ANSI code page */
00590     FONTSIGNATURE fs;
00591 } CHARSETINFO, *PCHARSETINFO, *LPCHARSETINFO;
00592
00593 /* Flags for TranslateCharsetInfo */
00594 #define TCI_SRCCHARSET 1
00595 #define TCI_SRCCODEPAGE 2
00596 #define TCI_SRCFONTSIG 3
00597
00598 typedef struct
00599 {
00600     DWORD lsUsb[4];

```

```

00601     DWORD  lsCsbDefault[2];
00602     DWORD  lsCsbSupported[2];
00603 } LOCALESIGNATURE, *PLOCALESIGNATUR, *LPLOCALESIGNATUREE;
00604
00605
00606 /* Flags for ModifyWorldTransform */
00607 #define MWT_IDENTITY      1
00608 #define MWT_LEFTMULTIPLY  2
00609 #define MWT_RIGHTMULTIPLY 3
00610 #define MWT_MIN           MWT_IDENTITY
00611 #define MWT_MAX           MWT_RIGHTMULTIPLY
00612
00613 /* Object Definitions for EnumObjects() */
00614 #define OBJ_PEN           1
00615 #define OBJ_BRUSH         2
00616 #define OBJ_DC            3
00617 #define OBJ_METADC       4
00618 #define OBJ_PAL          5
00619 #define OBJ_FONT         6
00620 #define OBJ_BITMAP       7
00621 #define OBJ_REGION       8
00622 #define OBJ_METAFILE     9
00623 #define OBJ_MEMDC        10
00624 #define OBJ_EXTPEN       11
00625 #define OBJ_ENHMETADC    12
00626 #define OBJ_ENHMETAFILE  13
00627
00628 typedef struct
00629 {
00630     FLOAT    eM11;
00631     FLOAT    eM12;
00632     FLOAT    eM21;
00633     FLOAT    eM22;
00634     FLOAT    eDx;
00635     FLOAT    eDy;
00636 } XFORM, *PXFORM, *LPXFORM;
00637
00638 /* lfWeight values */
00639 #define FW_DONTCARE      0
00640 #define FW_THIN         100
00641 #define FW_EXTRALIGHT   200
00642 #define FW_ULTRALIGHT   200
00643 #define FW_LIGHT        300
00644 #define FW_NORMAL       400
00645 #define FW_REGULAR      400
00646 #define FW_MEDIUM       500
00647 #define FW_SEMIBOLD     600
00648 #define FW_DEMIBOLD     600
00649 #define FW_BOLD         700
00650 #define FW_EXTRABOLD    800
00651 #define FW_ULTRABOLD    800
00652 #define FW_HEAVY        900
00653 #define FW_BLACK        900
00654
00655 /* lfCharSet values */
00656 #define ANSI_CHARSET      (BYTE)0 /* CP1252, ansi-0, iso8859-{1,15} */
00657 #define DEFAULT_CHARSET   (BYTE)1
00658 #define SYMBOL_CHARSET    (BYTE)2
00659 #define SHIFTJIS_CHARSET  (BYTE)128 /* CP932 */
00660 #define HANGEUL_CHARSET   (BYTE)129 /* CP949, ksc5601.1987-0 */
00661 #define HANGUL_CHARSET    HANGEUL_CHARSET
00662 #define GB2312_CHARSET    (BYTE)134 /* CP936, gb2312.1980-0 */
00663 #define CHINESEBIG5_CHARSET (BYTE)136 /* CP950, big5.et-0 */
00664 #define GREEK_CHARSET     (BYTE)161 /* CP1253 */
00665 #define TURKISH_CHARSET    (BYTE)162 /* CP1254, -iso8859-9 */
00666 #define HEBREW_CHARSET     (BYTE)177 /* CP1255, -iso8859-8 */
00667 #define ARABIC_CHARSET     (BYTE)178 /* CP1256, -iso8859-6 */
00668 #define BALTIC_CHARSET     (BYTE)186 /* CP1257, -iso8859-13 */
00669 #define RUSSIAN_CHARSET    (BYTE)204 /* CP1251, -iso8859-5 */
00670 #define EE_CHARSET        (BYTE)238 /* CP1250, -iso8859-2 */
00671 #define EASTEUROPE_CHARSET EE_CHARSET
00672 #define THAI_CHARSET       (BYTE)222 /* CP874, iso8859-11, tis620 */
00673 #define JOHAB_CHARSET      (BYTE)130 /* korean (johab) CP1361 */
00674 #define MAC_CHARSET        (BYTE)77
00675 #define OEM_CHARSET        (BYTE)255
00676 /* I don't know if the values of *_CHARSET macros are defined in Windows
00677  * or if we can choose them as we want. -- srtxg
00678  */
00679 #define VISCII_CHARSET      (BYTE)240 /* viscii1.1-1 */
00680 #define TCVN_CHARSET        (BYTE)241 /* tcvn-0 */
00681 #define KOI8_CHARSET        (BYTE)242 /* koi8-{r,u,ru} */
00682 #define ISO3_CHARSET        (BYTE)243 /* iso8859-3 */
00683 #define ISO4_CHARSET        (BYTE)244 /* iso8859-4 */
00684 #define ISO10_CHARSET       (BYTE)245 /* iso8859-10 */
00685 #define CELTIC_CHARSET      (BYTE)246 /* iso8859-14 */
00686
00687 #define FS_LATIN1           0x00000001L

```

```

00688 #define FS_LATIN2                0x00000002L
00689 #define FS_CYRILLIC                0x00000004L
00690 #define FS_GREEK                   0x00000008L
00691 #define FS_TURKISH                 0x00000010L
00692 #define FS_HEBREW                  0x00000020L
00693 #define FS_ARABIC                   0x00000040L
00694 #define FS_BALTIC                  0x00000080L
00695 #define FS_VIETNAMESE              0x00000100L
00696 #define FS_THAI                    0x00010000L
00697 #define FS_JISJAPAN                0x00020000L
00698 #define FS_CHINESESIMP              0x00040000L
00699 #define FS_WANSUNG                 0x00080000L
00700 #define FS_CHINESETRAD             0x00100000L
00701 #define FS_JOHAB                   0x00200000L
00702 #define FS_SYMBOL                  0x80000000L
00703
00704 /* lfOutPrecision values */
00705 #define OUT_DEFAULT_PRECIS  0
00706 #define OUT_STRING_PRECIS  1
00707 #define OUT_CHARACTER_PRECIS  2
00708 #define OUT_STROKE_PRECIS  3
00709 #define OUT_TT_PRECIS      4
00710 #define OUT_DEVICE_PRECIS  5
00711 #define OUT_RASTER_PRECIS  6
00712 #define OUT_TT_ONLY_PRECIS  7
00713 #define OUT_OUTLINE_PRECIS  8
00714
00715 /* lfClipPrecision values */
00716 #define CLIP_DEFAULT_PRECIS  0x00
00717 #define CLIP_CHARACTER_PRECIS  0x01
00718 #define CLIP_STROKE_PRECIS  0x02
00719 #define CLIP_MASK            0x0F
00720 #define CLIP_LH_ANGLES       0x10
00721 #define CLIP_TT_ALWAYS       0x20
00722 #define CLIP_EMBEDDED        0x80
00723
00724 /* lfQuality values */
00725 #define DEFAULT_QUALITY      0
00726 #define DRAFT_QUALITY       1
00727 #define PROOF_QUALITY        2
00728 #define NONANTIALIASED_QUALITY 3
00729 #define ANTIALIASED_QUALITY  4
00730
00731 /* lfPitchAndFamily pitch values */
00732 #define DEFAULT_PITCH        0x00
00733 #define FIXED_PITCH          0x01
00734 #define VARIABLE_PITCH       0x02
00735 #define MONO_FONT            0x08
00736
00737 #define FF_DONTCARE           0x00
00738 #define FF_ROMAN              0x10
00739 #define FF_SWISS              0x20
00740 #define FF_MODERN             0x30
00741 #define FF_SCRIPT             0x40
00742 #define FF_DECORATIVE         0x50
00743
00744 typedef struct
00745 {
00746     LONG        tmHeight;
00747     LONG        tmAscent;
00748     LONG        tmDescent;
00749     LONG        tmInternalLeading;
00750     LONG        tmExternalLeading;
00751     LONG        tmAveCharWidth;
00752     LONG        tmMaxCharWidth;
00753     LONG        tmWeight;
00754     LONG        tmOverhang;
00755     LONG        tmDigitizedAspectX;
00756     LONG        tmDigitizedAspectY;
00757     BYTE        tmFirstChar;
00758     BYTE        tmLastChar;
00759     BYTE        tmDefaultChar;
00760     BYTE        tmBreakChar;
00761     BYTE        tmItalic;
00762     BYTE        tmUnderlined;
00763     BYTE        tmStruckOut;
00764     BYTE        tmPitchAndFamily;
00765     BYTE        tmCharSet;
00766 } TEXTMETRICA, *LPTEXTMETRICA, *PTEXTMETRICA;
00767
00768 typedef struct
00769 {
00770     LONG        tmHeight;
00771     LONG        tmAscent;
00772     LONG        tmDescent;
00773     LONG        tmInternalLeading;
00774     LONG        tmExternalLeading;

```

```

00775     LONG        tmAveCharWidth;
00776     LONG        tmMaxCharWidth;
00777     LONG        tmWeight;
00778     LONG        tmOverhang;
00779     LONG        tmDigitizedAspectX;
00780     LONG        tmDigitizedAspectY;
00781     WCHAR        tmFirstChar;
00782     WCHAR        tmLastChar;
00783     WCHAR        tmDefaultChar;
00784     WCHAR        tmBreakChar;
00785     BYTE         tmItalic;
00786     BYTE         tmUnderlined;
00787     BYTE         tmStruckOut;
00788     BYTE         tmPitchAndFamily;
00789     BYTE         tmCharSet;
00790 } TEXTMETRICW, *LPTEXTMETRICW, *PTEXTMETRICW;
00791
00792 DECL_WINELIB_TYPE_AW(TEXTMETRIC)
00793 DECL_WINELIB_TYPE_AW(PTEXTMETRIC)
00794 DECL_WINELIB_TYPE_AW(LPTEXTMETRIC)
00795
00796
00797 typedef struct tagPANOSE
00798 {
00799     BYTE bFamilyType;
00800     BYTE bSerifStyle;
00801     BYTE bWeight;
00802     BYTE bProportion;
00803     BYTE bContrast;
00804     BYTE bStrokeVariation;
00805     BYTE bArmStyle;
00806     BYTE bLetterform;
00807     BYTE bMidline;
00808     BYTE bXHeight;
00809 } PANOSE, *LPPANOSE;
00810
00811 #define PANOSE_COUNT 10
00812
00813 #define PANOSE_FAMILYTYPE_INDEX 0
00814 #define PAN_SERIFSTYLE_INDEX 1
00815 #define PAN_WEIGHT_INDEX 2
00816 #define PAN_PROPORTION_INDEX 3
00817 #define PAN_CONTRAST_INDEX 4
00818 #define PAN_STROKEVARIATION_INDEX 5
00819 #define PAN_ARMSTYLE_INDEX 6
00820 #define PAN_LETTERFORM_INDEX 7
00821 #define PAN_MIDLINE_INDEX 8
00822 #define PAN_XHEIGHT_INDEX 9
00823
00824 #define PAN_CULTURE_LATIN 0
00825
00826 #define PAN_ANY 0
00827 #define PAN_NO_FIT 1
00828
00829 #define PAN_FAMILY_TEXT_DISPLAY 2
00830 #define PAN_FAMILY_SCRIPT 3
00831 #define PAN_FAMILY_DECORATIVE 4
00832 #define PAN_FAMILY_PICTORIAL 5
00833
00834 #define PAN_SERIF_COVE 2
00835 #define PAN_SERIF_OBTUSE_COVE 3
00836 #define PAN_SERIF_SQUARE_COVE 4
00837 #define PAN_SERIF_OBTUSE_SQUARE_COVE 5
00838 #define PAN_SERIF_SQUARE 6
00839 #define PAN_SERIF_THIN 7
00840 #define PAN_SERIF_BONE 8
00841 #define PAN_SERIF_EXAGGERATED 9
00842 #define PAN_SERIF_TRIANGLE 10
00843 #define PAN_SERIF_NORMAL_SANS 11
00844 #define PAN_SERIF_OBTUSE_SANS 12
00845 #define PAN_SERIF_PERP_SANS 13
00846 #define PAN_SERIF_FLARED 14
00847 #define PAN_SERIF_ROUNDED 15
00848
00849 #define PAN_WEIGHT_VERY_LIGHT 2
00850 #define PAN_WEIGHT_LIGHT 3
00851 #define PAN_WEIGHT_THIN 4
00852 #define PAN_WEIGHT_BOOK 5
00853 #define PAN_WEIGHT_MEDIUM 6
00854 #define PAN_WEIGHT_DEMI 7
00855 #define PAN_WEIGHT_BOLD 8
00856 #define PAN_WEIGHT_HEAVY 9
00857 #define PAN_WEIGHT_BLACK 10
00858 #define PAN_WEIGHT_NORD 11
00859
00860 #define PAN_PROP_OLD_STYLE 2
00861 #define PAN_PROP_MODERN 3

```

```

00862 #define PAN_PROP_EVEN_WIDTH 4
00863 #define PAN_PROP_EXPANDED 5
00864 #define PAN_PROP_CONDENSED 6
00865 #define PAN_PROP_VERY_EXPANDED 7
00866 #define PAN_PROP_VERY_CONDENSED 8
00867 #define PAN_PROP_MONOSPACED 9
00868
00869 #define PAN_CONTRAST_NONE 2
00870 #define PAN_CONTRAST_VERY_LOW 3
00871 #define PAN_CONTRAST_LOW 4
00872 #define PAN_CONTRAST_MEDIUM_LOW 5
00873 #define PAN_CONTRAST_MEDIUM 6
00874 #define PAN_CONTRAST_MEDIUM_HIGH 7
00875 #define PAN_CONTRAST_HIGH 8
00876 #define PAN_CONTRAST_VERY_HIGH 9
00877
00878 #define PAN_STROKE_GRADUAL_DIAG 2
00879 #define PAN_STROKE_GRADUAL_TRAN 3
00880 #define PAN_STROKE_GRADUAL_VERT 4
00881 #define PAN_STROKE_GRADUAL_HORZ 5
00882 #define PAN_STROKE_RAPID_VERT 6
00883 #define PAN_STROKE_RAPID_HORZ 7
00884 #define PAN_STROKE_INSTANT_VERT 8
00885
00886 #define PAN_STRAIGHT_ARMS_HORZ 2
00887 #define PAN_STRAIGHT_ARMS_WEDGE 3
00888 #define PAN_STRAIGHT_ARMS_VERT 4
00889 #define PAN_STRAIGHT_ARMS_SINGLE_SERIF 5
00890 #define PAN_STRAIGHT_ARMS_DOUBLE_SERIF 6
00891 #define PAN_BENT_ARMS_HORZ 7
00892 #define PAN_BENT_ARMS_WEDGE 8
00893 #define PAN_BENT_ARMS_VERT 9
00894 #define PAN_BENT_ARMS_SINGLE_SERIF 10
00895 #define PAN_BENT_ARMS_DOUBLE_SERIF 11
00896
00897 #define PAN_LETT_NORMAL_COMPACT 2
00898 #define PAN_LETT_NORMAL_WEIGHTED 3
00899 #define PAN_LETT_NORMAL_BOXED 4
00900 #define PAN_LETT_NORMAL_FLATTENED 5
00901 #define PAN_LETT_NORMAL_ROUNDED 6
00902 #define PAN_LETT_NORMAL_OFF_CENTER 7
00903 #define PAN_LETT_NORMAL_SQUARE 8
00904 #define PAN_LETT_OBLIQUE_COMPACT 9
00905 #define PAN_LETT_OBLIQUE_WEIGHTED 10
00906 #define PAN_LETT_OBLIQUE_BOXED 11
00907 #define PAN_LETT_OBLIQUE_FLATTENED 12
00908 #define PAN_LETT_OBLIQUE_ROUNDED 13
00909 #define PAN_LETT_OBLIQUE_OFF_CENTER 14
00910 #define PAN_LETT_OBLIQUE_SQUARE 15
00911
00912 #define PAN_MIDLINE_STANDARD_TRIMMED 2
00913 #define PAN_MIDLINE_STANDARD_POINTED 3
00914 #define PAN_MIDLINE_STANDARD_SERIFED 4
00915 #define PAN_MIDLINE_HIGH_TRIMMED 5
00916 #define PAN_MIDLINE_HIGH_POINTED 6
00917 #define PAN_MIDLINE_HIGH_SERIFED 7
00918 #define PAN_MIDLINE_CONSTANT_TRIMMED 8
00919 #define PAN_MIDLINE_CONSTANT_POINTED 9
00920 #define PAN_MIDLINE_CONSTANT_SERIFED 10
00921 #define PAN_MIDLINE_LOW_TRIMMED 11
00922 #define PAN_MIDLINE_LOW_POINTED 12
00923 #define PAN_MIDLINE_LOW_SERIFED 13
00924
00925 #define PAN_XHEIGHT_CONSTANT_SMALL 2
00926 #define PAN_XHEIGHT_CONSTANT_STANDARD 3
00927 #define PAN_XHEIGHT_CONSTANT_LARGE 4
00928 #define PAN_XHEIGHT_DUCKING_SMALL 5
00929 #define PAN_XHEIGHT_DUCKING_STANDARD 6
00930 #define PAN_XHEIGHT_DUCKING_LARGE 7
00931
00932 #define ELF_VENDOR_SIZE 4
00933 typedef struct
00934 {
00935     LOGFONTA elfLogFont;
00936     BYTE elfFullName[LF_FULLFACESIZE];
00937     BYTE elfStyle[LF_FACESIZE];
00938     DWORD elfVersion;
00939     DWORD elfStyleSize;
00940     DWORD elfMatch;
00941     DWORD elfReserved;
00942     BYTE elfVendorId[ELF_VENDOR_SIZE];
00943     DWORD elfCulture;
00944     PANOSE elfPanose;
00945 } EXTLOGFONTA, *PEXTLOGFONTA, *LPEXTLOGFONTA;
00946
00947 typedef struct
00948 {

```

```

00949 LOGFONTW      elfLogFont;
00950 WCHAR          elfFullName[LF_FULLFACESIZE];
00951 WCHAR          elfStyle[LF_FACESIZE];
00952 DWORD          elfVersion;
00953 DWORD          elfStyleSize;
00954 DWORD          elfMatch;
00955 DWORD          elfReserved;
00956 BYTE           elfVendorId[ELF_VENDOR_SIZE];
00957 DWORD          elfCulture;
00958 PANOSE         elfPanose;
00959 } EXTLOGFONTW, *PEXTLOGFONTW, *LPEXTLOGFONTW;
00960
00961 DECL_WINELIB_TYPE_AW(EXTLOGFONT)
00962 DECL_WINELIB_TYPE_AW(PEXTLOGFONT)
00963 DECL_WINELIB_TYPE_AW(LPEXTLOGFONT)
00964
00965 #define ELF_VERSION      0
00966 #define ELF_CULTURE_LATIN 0
00967
00968 typedef struct _OUTLINETEXTMETRICA
00969 {
00970     UINT          otmSize;
00971     TEXTMETRICA   otmTextMetrics;
00972     BYTE          otmFiller;
00973     PANOSE        otmPanoseNumber;
00974     UINT          otmfsSelection;
00975     UINT          otmfsType;
00976     INT           otmsCharSlopeRise;
00977     INT           otmsCharSlopeRun;
00978     INT           otmItalicAngle;
00979     UINT          otmEMSquare;
00980     INT           otmAscent;
00981     INT           otmDescent;
00982     UINT          otmLineGap;
00983     UINT          otmsCapEmHeight;
00984     UINT          otmsXHeight;
00985     RECT          otmrcFontBox;
00986     INT           otmMacAscent;
00987     INT           otmMacDescent;
00988     UINT          otmMacLineGap;
00989     UINT          otmusMinimumPPEM;
00990     POINT         otmptSubscriptSize;
00991     POINT         otmptSubscriptOffset;
00992     POINT         otmptSuperscriptSize;
00993     POINT         otmptSuperscriptOffset;
00994     UINT          otmsStrikeoutSize;
00995     INT           otmsStrikeoutPosition;
00996     INT           otmsUnderscoreSize;
00997     INT           otmsUnderscorePosition;
00998     LPSTR         otmpFamilyName;
00999     LPSTR         otmpFaceName;
01000     LPSTR         otmpStyleName;
01001     LPSTR         otmpFullName;
01002 } OUTLINETEXTMETRICA, *POUTLINETEXTMETRICA, *LPOUTLINETEXTMETRICA;
01003
01004 typedef struct _OUTLINETEXTMETRICW
01005 {
01006     UINT          otmSize;
01007     TEXTMETRICW   otmTextMetrics;
01008     BYTE          otmFiller;
01009     PANOSE        otmPanoseNumber;
01010     UINT          otmfsSelection;
01011     UINT          otmfsType;
01012     INT           otmsCharSlopeRise;
01013     INT           otmsCharSlopeRun;
01014     INT           otmItalicAngle;
01015     UINT          otmEMSquare;
01016     INT           otmAscent;
01017     INT           otmDescent;
01018     UINT          otmLineGap;
01019     UINT          otmsCapEmHeight;
01020     UINT          otmsXHeight;
01021     RECT          otmrcFontBox;
01022     INT           otmMacAscent;
01023     INT           otmMacDescent;
01024     UINT          otmMacLineGap;
01025     UINT          otmusMinimumPPEM;
01026     POINT         otmptSubscriptSize;
01027     POINT         otmptSubscriptOffset;
01028     POINT         otmptSuperscriptSize;
01029     POINT         otmptSuperscriptOffset;
01030     UINT          otmsStrikeoutSize;
01031     INT           otmsStrikeoutPosition;
01032     INT           otmsUnderscoreSize;
01033     INT           otmsUnderscorePosition;
01034     LPSTR         otmpFamilyName;
01035     LPSTR         otmpFaceName;

```

```

01036     LPSTR          otmpStyleName;
01037     LPSTR          otmpFullName;
01038 } OUTLINETEXTMETRICW, *POUTLINETEXTMETRICW, *LPOUTLINETEXTMETRICW;
01039
01040 DECL_WINELIB_TYPE_AW(OUTLINETEXTMETRIC)
01041 DECL_WINELIB_TYPE_AW(POUTLINETEXTMETRIC)
01042 DECL_WINELIB_TYPE_AW(LPOUTLINETEXTMETRIC)
01043
01044 typedef struct
01045 {
01046     INT          x;
01047     INT          y;
01048     UINT         n;
01049     LPCSTR       lpstr;
01050     UINT         uiFlags;
01051     RECT         rcl;
01052     INT          *pdx;
01053 } POLYTEXTA, *PPOLYTEXTA, *LPPOLYTEXTA;
01054
01055 typedef struct
01056 {
01057     INT          x;
01058     INT          y;
01059     UINT         n;
01060     LPCWSTR      lpstr;
01061     UINT         uiFlags;
01062     RECT         rcl;
01063     INT          *pdx;
01064 } POLYTEXTW, *PPOLYTEXTW, *LPPOLYTEXTW;
01065
01066 DECL_WINELIB_TYPE_AW(POLYTEXT)
01067 DECL_WINELIB_TYPE_AW(PPOLYTEXT)
01068 DECL_WINELIB_TYPE_AW(LPPOLYTEXT)
01069
01070
01071 /* ntmFlags field flags */
01072 #define NTM_REGULAR      0x00000040L
01073 #define NTM_BOLD         0x00000020L
01074 #define NTM_ITALIC       0x00000001L
01075
01076 typedef struct
01077 {
01078     LONG         tmHeight;
01079     LONG         tmAscent;
01080     LONG         tmDescent;
01081     LONG         tmInternalLeading;
01082     LONG         tmExternalLeading;
01083     LONG         tmAveCharWidth;
01084     LONG         tmMaxCharWidth;
01085     LONG         tmWeight;
01086     LONG         tmOverhang;
01087     LONG         tmDigitizedAspectX;
01088     LONG         tmDigitizedAspectY;
01089     BYTE         tmFirstChar;
01090     BYTE         tmLastChar;
01091     BYTE         tmDefaultChar;
01092     BYTE         tmBreakChar;
01093     BYTE         tmItalic;
01094     BYTE         tmUnderlined;
01095     BYTE         tmStruckOut;
01096     BYTE         tmPitchAndFamily;
01097     BYTE         tmCharSet;
01098     DWORD        ntmFlags;
01099     UINT         ntmSizeEM;
01100     UINT         ntmCellHeight;
01101     UINT         ntmAvgWidth;
01102 } NEWTEXTMETRICA, *PNEWTEXTMETRICA, *LPNEWTEXTMETRICA;
01103
01104 typedef struct
01105 {
01106     LONG         tmHeight;
01107     LONG         tmAscent;
01108     LONG         tmDescent;
01109     LONG         tmInternalLeading;
01110     LONG         tmExternalLeading;
01111     LONG         tmAveCharWidth;
01112     LONG         tmMaxCharWidth;
01113     LONG         tmWeight;
01114     LONG         tmOverhang;
01115     LONG         tmDigitizedAspectX;
01116     LONG         tmDigitizedAspectY;
01117     WCHAR        tmFirstChar;
01118     WCHAR        tmLastChar;
01119     WCHAR        tmDefaultChar;
01120     WCHAR        tmBreakChar;
01121     BYTE         tmItalic;
01122     BYTE         tmUnderlined;

```

```

01123     BYTE        tmStruckOut;
01124     BYTE        tmPitchAndFamily;
01125     BYTE        tmCharSet;
01126     DWORD       ntmFlags;
01127     UINT        ntmSizeEM;
01128     UINT        ntmCellHeight;
01129     UINT        ntmAvgWidth;
01130 } NEWTEXTMETRICW, *PNEWTEXTMETRICW, *LPNEWTEXTMETRICW;
01131
01132 DECL_WINELIB_TYPE_AW(NEWTEXTMETRIC)
01133 DECL_WINELIB_TYPE_AW(PNEWTEXTMETRIC)
01134 DECL_WINELIB_TYPE_AW(LPNEWTEXTMETRIC)
01135
01136 typedef struct
01137 {
01138     NEWTEXTMETRICA ntmTm;
01139     FONTSIGNATURE ntmFontSig;
01140 } NEWTEXTMETRICEXA, *LPNEWTEXTMETRICEXA;
01141
01142 typedef struct
01143 {
01144     NEWTEXTMETRICW ntmTm;
01145     FONTSIGNATURE ntmFontSig;
01146 } NEWTEXTMETRICEXW, *LPNEWTEXTMETRICEXW;
01147
01148 DECL_WINELIB_TYPE_AW(NEWTEXTMETRICEX)
01149 DECL_WINELIB_TYPE_AW(LPNEWTEXTMETRICEX)
01150
01151 typedef int CALLBACK (*OLDFONTENUMPROCA) (const LOGFONTA*, const TEXTMETRICA*,
01152                                           DWORD, LPARAM);
01153 typedef int CALLBACK (*OLDFONTENUMPROCW) (const LOGFONTW*, const TEXTMETRICW*,
01154                                           DWORD, LPARAM);
01155 DECL_WINELIB_TYPE_AW(OLDFONTENUMPROC)
01156
01157 typedef OLDFONTENUMPROCA FONTENUMPROCA;
01158 typedef OLDFONTENUMPROCW FONTENUMPROCW;
01159 DECL_WINELIB_TYPE_AW(FONTENUMPROC)
01160
01161 typedef int CALLBACK (*FONTENUMPROCEXA) (LPENUMLOGFONTEXA, LPNEWTEXTMETRICEXA, DWORD, LPARAM);
01162 typedef int CALLBACK (*FONTENUMPROCEXW) (LPENUMLOGFONTEXW, LPNEWTEXTMETRICEXW, DWORD, LPARAM);
01163 DECL_WINELIB_TYPE_AW(FONTENUMPROCEX)
01164
01165 typedef INT CALLBACK (*GOBJENUMPROC) (LPVOID, LPARAM);
01166 typedef VOID CALLBACK (*LINEDDAPROC) (INT, INT, LPARAM);
01167
01168 /* tmPitchAndFamily bits */
01169 #define TMPF_FIXED_PITCH 1 /* means variable pitch */
01170 #define TMPF_VECTOR 2
01171 #define TMPF_TRUETYPE 4
01172 #define TMPF_DEVICE 8
01173
01174 /* Text alignment */
01175 #define TA_NOUPDATECP 0x00
01176 #define TA_UPDATECP 0x01
01177 #define TA_LEFT 0x00
01178 #define TA_RIGHT 0x02
01179 #define TA_CENTER 0x06
01180 #define TA_TOP 0x00
01181 #define TA_BOTTOM 0x08
01182 #define TA_BASELINE 0x18
01183 #define TA_RTLREADING 0x100
01184 #define TA_MASK TA_BASELINE+TA_CENTER+TA_UPDATECP+TA_RTLREADING
01185
01186 #define VTA_BASELINE TA_BASELINE
01187 #define VTA_LEFT TA_BOTTOM
01188 #define VTA_RIGHT TA_TOP
01189 #define VTA_CENTER TA_CENTER
01190 #define VTA_BOTTOM TA_RIGHT
01191 #define VTA_TOP TA_LEFT
01192
01193
01194 /* ExtTextOut() parameters */
01195 #define ETO_GRAYED 0x0001
01196 #define ETO_OPAQUE 0x0002
01197 #define ETO_CLIPPED 0x0004
01198 #define ETO_GLYPH_INDEX 0x0010
01199 #define ETO_RTLREADING 0x0080
01200 #define ETO_IGNORELANGUAGE 0x1000
01201
01202 #define ASPECT_FILTERING 0x0001
01203
01204 typedef struct
01205 {
01206     UINT gmBlackBoxX;
01207     UINT gmBlackBoxY;
01208     POINT gmptGlyphOrigin;
01209     SHORT gmCellIncX;

```

```

01210     SHORT    gmCellIncY;
01211 } GLYPHMETRICS, *LPGLYPHMETRICS;
01212
01213
01214 #define GGO_METRICS      0
01215 #define GGO_BITMAP      1
01216 #define GGO_NATIVE      2
01217 #define GGO_GRAY2_BITMAP 4
01218 #define GGO_GRAY4_BITMAP 5
01219 #define GGO_GRAY8_BITMAP 6
01220 #define GGO_GLYPH_INDEX 0x80
01221
01222 typedef struct
01223 {
01224     WORD    fract;
01225     SHORT   value;
01226 } FIXED;
01227
01228 typedef struct tagPOINTFX
01229 {
01230     FIXED x;
01231     FIXED y;
01232 } POINTFX, *LPPOINTFX;
01233
01234 typedef struct tagTTPOLYCURVE
01235 {
01236     WORD wType;
01237     WORD cpfx;
01238     POINTFX apfx[1];
01239 } TTPOLYCURVE, *LPTTPOLYCURVE;
01240
01241 typedef struct tagTTPOLYGONHEADER
01242 {
01243     DWORD cb;
01244     DWORD dwType;
01245     POINTFX pfxStart;
01246 } TTPOLYGONHEADER, *LPTTPOLYGONHEADER;
01247
01248 typedef struct
01249 {
01250     FIXED eM11;
01251     FIXED eM12;
01252     FIXED eM21;
01253     FIXED eM22;
01254 } MAT2, *LPMAT2;
01255
01256 /* for GetCharABCWidths() */
01257 typedef struct
01258 {
01259     INT    abcA;
01260     UINT   abcB;
01261     INT    abcC;
01262 } ABC, *PABC, *LPABC;
01263
01264
01265 /* for GetCharacterPlacement () */
01266
01267 #define GCP_DBCS      0x0001
01268 #define GCP_REORDER   0x0002
01269 #define GCP_USEKERNING 0x0008
01270 #define GCP_GLYPHSHAPE 0x0010
01271 #define GCP_LIGATE     0x0020
01272 #define GCP_DIACRITIC  0x0100
01273 #define GCP_KASHIDA    0x0200
01274 #define GCP_ERROR      0x8000
01275 #define FLI_MASK       0x103b
01276 #define GCP_JUSTIFY    0x00010000L
01277 #define FLI_GLYPHS     0x00040000L
01278 #define GCP_CLASSIN    0x00080000L
01279 #define GCP_MAXEXTENT   0x00100000L
01280 #define GCP_JUSTIFYIN  0x00200000L
01281 #define GCP_DISPLAYZWG 0x00400000L
01282 #define GCP_SYMSWAPOFF 0x00800000L
01283 #define GCP_NUMERICOVERRIDE 0x01000000L
01284 #define GCP_NEUTRALOVERRIDE 0x02000000L
01285 #define GCP_NUMERICSLATIN 0x04000000L
01286 #define GCP_NUMERICSLOCAL 0x08000000L
01287
01288 #define GCPCLASS_LATIN      1
01289 #define GCPCLASS_HEBREW    2
01290 #define GCPCLASS_ARABIC    3
01291 #define GCPCLASS_NEUTRAL   4
01292 #define GCPCLASS_LOCALNUMBER 5
01293 #define GCPCLASS_LATINNUMBER 6
01294 #define GCPCLASS_LATINNUMERICTERMINATOR 7
01295 #define GCPCLASS_LATINNUMERICSEPARATOR 8
01296 #define GCPCLASS_NUMERICSEPARATOR 9

```

```

01297 #define GCPCLASS_PREBOUNDLTR          0x80
01298 #define GCPCLASS_PREBOUNDRLT          0x40
01299 #define GCPCLASS_POSTBOUNDLTR         0x20
01300 #define GCPCLASS_POSTBOUNDRTL         0x10
01301
01302 #define GCPGLYPH_LINKBEFORE           0x8000
01303 #define GCPGLYPH_LINKAFTER            0x4000
01304
01305
01306 typedef struct tagGCP_RESULTS{
01307     DWORD    lStructSize;
01308     LPSTR     lpOutString;
01309     UINT      *lpOrder;
01310     INT       *lpDx;
01311     INT       *lpCaretPos;
01312     LPSTR     lpClass;
01313     LPWSTR    lpGlyphs;
01314     UINT      nGlyphs;
01315     UINT      nMaxFit;
01316 } GCP_RESULTS, *LPGCP_RESULTS;
01317
01318 typedef struct tagGCP_RESULTSW
01319 {
01320     DWORD    lStructSize;
01321     LPWSTR    lpOutString;
01322     UINT      *lpOrder;
01323     INT       *lpDx;
01324     INT       *lpCaretPos;
01325     LPWSTR    lpClass;
01326     LPWSTR    lpGlyphs;
01327     UINT      nGlyphs;
01328     UINT      nMaxFit;
01329 } GCP_RESULTSW, *LPGCP_RESULTSW;
01330
01331 DECL_WINELIB_TYPE_AW(GCP_RESULTS)
01332 DECL_WINELIB_TYPE_AW(LPGCP_RESULTS)
01333
01334 /* Rasterizer status */
01335 typedef struct
01336 {
01337     SHORT nSize;
01338     SHORT wFlags;
01339     SHORT nLanguageID;
01340 } RASTERIZER_STATUS, *LPRASTERIZER_STATUS;
01341
01342 #define TT_AVAILABLE          0x0001
01343 #define TT_ENABLED            0x0002
01344
01345 #define TT_PRIM_LINE          1
01346 #define TT_PRIM_QSPLINE       2
01347 #define TT_POLYGON_TYPE       24
01348
01349 /* Get/SetSystemPaletteUse() values */
01350 #define SYSPAL_ERROR          0
01351 #define SYSPAL_STATIC         1
01352 #define SYSPAL_NOSTATIC       2
01353
01354 typedef struct tagPALETTEENTRY
01355 {
01356     BYTE peRed, peGreen, peBlue, peFlags;
01357 } PALETTEENTRY, *PPALETTEENTRY, *LPPALETTEENTRY;
01358
01359 /* Logical palette entry flags */
01360 #define PC_RESERVED           0x01
01361 #define PC_EXPLICIT            0x02
01362 #define PC_NOCOLLAPSE         0x04
01363
01364 typedef struct tagLOGPALETTE
01365 {
01366     WORD        palVersion;
01367     WORD        palNumEntries;
01368     PALETTEENTRY palPalEntry[1];
01369 } LOGPALETTE, *PLOGPALETTE, *LPLOGPALETTE, *NPLOGPALETTE;
01370
01371 /* Pens */
01372
01373 typedef struct
01374 {
01375     UINT        lopnStyle;
01376     POINT        lopnWidth;
01377     COLORREF     lopnColor;
01378 } LOGPEN, *LPLOGPEN;
01379
01380
01381 typedef struct tagEXTLOGPEN
01382 {
01383     DWORD        elpPenStyle;

```

```
01384     DWORD    elpWidth;
01385     UINT      elpBrushStyle;
01386     COLORREF   elpColor;
01387     LONG       elpHatch;
01388     DWORD      elpNumEntries;
01389     DWORD      elpStyleEntry[1];
01390 } EXTLOGPEN, *PEXTLOGPEN, *NPEXTLOGPEN, *LPEXTLOGPEN;
01391
01392 #define PS_SOLID          0x00000000
01393 #define PS_DASH           0x00000001
01394 #define PS_DOT            0x00000002
01395 #define PS_DASHDOT       0x00000003
01396 #define PS_DASHDOTDOT    0x00000004
01397 #define PS_NULL           0x00000005
01398 #define PS_INSIDEFRAME    0x00000006
01399 #define PS_USERSTYLE      0x00000007
01400 #define PS_ALTERNATE      0x00000008
01401 #define PS_STYLE_MASK     0x0000000f
01402
01403 #define PS_ENDCAP_ROUND   0x00000000
01404 #define PS_ENDCAP_SQUARE 0x00000100
01405 #define PS_ENDCAP_FLAT   0x00000200
01406 #define PS_ENDCAP_MASK   0x00000f00
01407
01408 #define PS_JOIN_ROUND     0x00000000
01409 #define PS_JOIN_BEVEL     0x00000100
01410 #define PS_JOIN_MITER     0x00000200
01411 #define PS_JOIN_MASK      0x0000f000
01412
01413 #define PS_COSMETIC       0x00000000
01414 #define PS_GEOMETRIC      0x00001000
01415 #define PS_TYPE_MASK      0x0000f000
01416
01417 /* Regions */
01418
01419 #define ERROR              0
01420 #define NULLREGION         1
01421 #define SIMPLEREGION       2
01422 #define COMPLEXREGION      3
01423 #define RGN_ERROR          ERROR
01424
01425 #define RGN_AND             1
01426 #define RGN_OR              2
01427 #define RGN_XOR             3
01428 #define RGN_DIFF            4
01429 #define RGN_COPY            5
01430 #define RGN_MIN             RGN_AND
01431 #define RGN_MAX             RGN_COPY
01432 /* Device contexts */
01433
01434 /* Polygon modes */
01435 #define ALTERNATE           1
01436 #define WINDING             2
01437 #define POLYFILL_LAST      2
01438
01439 /* Background modes */
01440 /* Apparently some broken svr4 includes define TRANSPARENT */
01441 #undef TRANSPARENT
01442 #define TRANSPARENT         1
01443 #define OPAQUE              2
01444 #define BKMODE_LAST        2
01445
01446 /* Graphics Modes */
01447 #define GM_COMPATIBLE       1
01448 #define GM_ADVANCED         2
01449 #define GM_LAST            2
01450
01451 /* Arc direction modes */
01452 #define AD_COUNTERCLOCKWISE 1
01453 #define AD_CLOCKWISE        2
01454
01455 /* Map modes */
01456 #define MM_TEXT             1
01457 #define MM_LOMETRIC         2
01458 #define MM_HIMETRIC         3
01459 #define MM_LOENGLISH        4
01460 #define MM_HIENGLISH        5
01461 #define MM_TWIPS            6
01462 #define MM_ISOTROPIC        7
01463 #define MM_ANISOTROPIC      8
01464
01465 #define MM_MIN              MM_TEXT
01466 #define MM_MAX              MM_ANISOTROPIC
01467 #define MM_MAX_FIXEDSCALE   MM_TWIPS
01468
01469 /* Coordinate modes */
01470 #define ABSOLUTE            1
```

```
01471 #define RELATIVE          2
01472
01473 /* Flood fill modes */
01474 #define FLOODFILLBORDER    0
01475 #define FLOODFILLSURFACE  1
01476
01477 /* Device parameters for GetDeviceCaps() */
01478 #define DRIVERVERSION      0
01479 #define TECHNOLOGY         2
01480 #define HORZSIZE           4
01481 #define VERTSIZE           6
01482 #define HORZRES            8
01483 #define VERTRES           10
01484 #define BITSPIXEL         12
01485 #define PLANES            14
01486 #define NUMBRUSHES        16
01487 #define NUMPENS            18
01488 #define NUMMARKERS        20
01489 #define NUMFONTS          22
01490 #define NUMCOLORS         24
01491 #define PDEVICESIZE       26
01492 #define CURVECAPS         28
01493 #define LINECAPS          30
01494 #define POLYGONALCAPS    32
01495 #define TEXTCAPS          34
01496 #define CLIPCAPS          36
01497 #define RASTERCAPS       38
01498 #define ASPECTX           40
01499 #define ASPECTY           42
01500 #define ASPECTXY          44
01501 #define LOGPIXELSX        88
01502 #define LOGPIXELSY        90
01503 #define CAPS1             94
01504 #define SIZEPALETTE       104
01505 #define NUMRESERVED       106
01506 #define COLORRES          108
01507
01508 #define PHYSICALWIDTH      110
01509 #define PHYSICALHEIGHT     111
01510 #define PHYSICALOFFSETX   112
01511 #define PHYSICALOFFSETY   113
01512 #define SCALINGFACTORX    114
01513 #define SCALINGFACTORY    115
01514 #define VREFRESH          116
01515 #define DESKTOPVERTRES    117
01516 #define DESKTOPHORZRES    118
01517 #define BTLALIGNMENT      119
01518
01519 /* TECHNOLOGY */
01520 #define DT_PLOTTER         0
01521 #define DT_RASDISPLAY     1
01522 #define DT_RASPRINTER     2
01523 #define DT_RASCAMERA      3
01524 #define DT_CHARSTREAM     4
01525 #define DT_METAFILE       5
01526 #define DT_DISPFILE       6
01527
01528 /* CURVECAPS */
01529 #define CC_NONE            0x0000
01530 #define CC_CIRCLES         0x0001
01531 #define CC_PIE             0x0002
01532 #define CC_CHORD           0x0004
01533 #define CC_ELLIPSES        0x0008
01534 #define CC_WIDE            0x0010
01535 #define CC_STYLED          0x0020
01536 #define CC_WIDESTYLED      0x0040
01537 #define CC_INTERIORS       0x0080
01538 #define CC_ROUNDRECT       0x0100
01539
01540 /* LINECAPS */
01541 #define LC_NONE            0x0000
01542 #define LC_POLYLINE        0x0002
01543 #define LC_MARKER          0x0004
01544 #define LC_POLYMARKER      0x0008
01545 #define LC_WIDE            0x0010
01546 #define LC_STYLED          0x0020
01547 #define LC_WIDESTYLED      0x0040
01548 #define LC_INTERIORS       0x0080
01549
01550 /* POLYGONALCAPS */
01551 #define PC_NONE            0x0000
01552 #define PC_POLYGON         0x0001
01553 #define PC_RECTANGLE       0x0002
01554 #define PC_WINDPOLYGON     0x0004
01555 #define PC_TRAPEZOID       0x0004
01556 #define PC_SCANLINE        0x0008
01557 #define PC_WIDE            0x0010
```

```
01558 #define PC_STYLED 0x0020
01559 #define PC_WIDESTYLED 0x0040
01560 #define PC_INTERIORS 0x0080
01561 #define PC_POLYPOLYGON 0x0100
01562 #define PC_PATHS 0x0200
01563
01564 /* TEXTCAPS */
01565 #define TC_OP_CHARACTER 0x0001
01566 #define TC_OP_STROKE 0x0002
01567 #define TC_CP_STROKE 0x0004
01568 #define TC_CR_90 0x0008
01569 #define TC_CR_ANY 0x0010
01570 #define TC_SF_X_YINDEP 0x0020
01571 #define TC_SA_DOUBLE 0x0040
01572 #define TC_SA_INTEGER 0x0080
01573 #define TC_SA_CONTIN 0x0100
01574 #define TC_EA_DOUBLE 0x0200
01575 #define TC_IA_ABLE 0x0400
01576 #define TC_UA_ABLE 0x0800
01577 #define TC_SO_ABLE 0x1000
01578 #define TC_RA_ABLE 0x2000
01579 #define TC_VA_ABLE 0x4000
01580 #define TC_RESERVED 0x8000
01581 #define TC_SCROLLBLT 0x00010000
01582
01583 /* CLIPCAPS */
01584 #define CP_NONE 0x0000
01585 #define CP_RECTANGLE 0x0001
01586 #define CP_REGION 0x0002
01587
01588 /* RASTERCAPS */
01589 #define RC_NONE 0x0000
01590 #define RC_BITBLT 0x0001
01591 #define RC_BANDING 0x0002
01592 #define RC_SCALING 0x0004
01593 #define RC_BITMAP64 0x0008
01594 #define RC_GDI20_OUTPUT 0x0010
01595 #define RC_GDI20_STATE 0x0020
01596 #define RC_SAVEBITMAP 0x0040
01597 #define RC_DI_BITMAP 0x0080
01598 #define RC_PALETTE 0x0100
01599 #define RC_DIBTODEV 0x0200
01600 #define RC_BIGFONT 0x0400
01601 #define RC_STRETCHBLT 0x0800
01602 #define RC_FLOODFILL 0x1000
01603 #define RC_STRETCHDIB 0x2000
01604 #define RC_OP_DX_OUTPUT 0x4000
01605 #define RC_DEVBITS 0x8000
01606
01607 /* CAPS1 */
01608
01609 #define C1_TRANSPARENT 0x0001
01610 #define TC_TT_ABLE 0x0002
01611 #define C1_TT_CR_ANY 0x0004
01612 #define C1_EMF_COMPLIANT 0x0008
01613 #define C1_DIBENGINE 0x0010
01614 #define C1_GAMMA_RAMP 0x0040
01615 #define C1_REINIT_ABLE 0x0080
01616 #define C1_GLYPH_INDEX 0x0100
01617 #define C1_BIT_PACKED 0x0200
01618 #define C1_BYTE_PACKED 0x0400
01619 #define C1_COLORCURSOR 0x0800
01620 #define C1_CMYK_ABLE 0x1000
01621 #define C1_SLOW_CARD 0x2000
01622
01623 /* Device-independent bitmaps */
01624
01625 typedef struct {
01626     BYTE rgbBlue;
01627     BYTE rgbGreen;
01628     BYTE rgbRed;
01629     BYTE rgbReserved;
01630 } RGBQUAD, *LPRGBQUAD;
01631
01632 typedef struct {
01633     BYTE rgbtBlue;
01634     BYTE rgbtGreen;
01635     BYTE rgbtRed;
01636 } RGBTRIPLE;
01637
01638 #include "pshpack2.h"
01639 typedef struct
01640 {
01641     WORD bfType;
01642     DWORD bfSize;
01643     WORD bfReserved1;
01644     WORD bfReserved2;
```

```

01645     DWORD    bOffBits;
01646 } BITMAPFILEHEADER, *PBITMAPFILEHEADER, *LPBITMAPFILEHEADER;
01647 #include "poppack.h"
01648
01649 #define MAKEPOINTS(l)  (((POINTS *) &(l)))
01650
01651 typedef struct
01652 {
01653     DWORD    biSize;
01654     LONG     biWidth;
01655     LONG     biHeight;
01656     WORD     biPlanes;
01657     WORD     biBitCount;
01658     DWORD    biCompression;
01659     DWORD    biSizeImage;
01660     LONG     biXPelsPerMeter;
01661     LONG     biYPelsPerMeter;
01662     DWORD    biClrUsed;
01663     DWORD    biClrImportant;
01664 } BITMAPINFOHEADER, *PBITMAPINFOHEADER, *LPBITMAPINFOHEADER;
01665
01666 typedef struct
01667 {
01668     DWORD    bV4Size;
01669     LONG     bV4Width;
01670     LONG     bV4Height;
01671     WORD     bV4Planes;
01672     WORD     bV4BitCount;
01673     DWORD    bV4Compression;
01674     DWORD    bV4SizeImage;
01675     LONG     bV4XPelsPerMeter;
01676     LONG     bV4YPelsPerMeter;
01677     DWORD    bV4ClrUsed;
01678     DWORD    bV4ClrImportant;
01679     DWORD    bV4RedMask;
01680     DWORD    bV4GreenMask;
01681     DWORD    bV4BlueMask;
01682     DWORD    bV4AlphaMask;
01683     DWORD    bV4CSType;
01684     CIEXYZTRIPLE bV4Endpoints;
01685     DWORD    bV4GammaRed;
01686     DWORD    bV4GammaGreen;
01687     DWORD    bV4GammaBlue;
01688 } BITMAPV4HEADER, *PBITMAPV4HEADER;
01689
01690 typedef struct {
01691     DWORD    bV5Size;
01692     LONG     bV5Width;
01693     LONG     bV5Height;
01694     WORD     bV5Planes;
01695     WORD     bV5BitCount;
01696     DWORD    bV5Compression;
01697     DWORD    bV5SizeImage;
01698     LONG     bV5XPelsPerMeter;
01699     LONG     bV5YPelsPerMeter;
01700     DWORD    bV5ClrUsed;
01701     DWORD    bV5ClrImportant;
01702     DWORD    bV5RedMask;
01703     DWORD    bV5GreenMask;
01704     DWORD    bV5BlueMask;
01705     DWORD    bV5AlphaMask;
01706     DWORD    bV5CSType;
01707     CIEXYZTRIPLE bV5Endpoints;
01708     DWORD    bV5GammaRed;
01709     DWORD    bV5GammaGreen;
01710     DWORD    bV5GammaBlue;
01711     DWORD    bV5Intent;
01712     DWORD    bV5ProfileData;
01713     DWORD    bV5ProfileSize;
01714     DWORD    bV5Reserved;
01715 } BITMAPV5HEADER, *LPBITMAPV5HEADER, *PBITMAPV5HEADER;
01716
01717 #define PROFILE_LINKED    'LINK'
01718 #define PROFILE_EMBEDDED 'MBED'
01719
01720
01721 /* biCompression */
01722 #define BI_RGB            0
01723 #define BI_RLE8           1
01724 #define BI_RLE4           2
01725 #define BI_BITFIELDS      3
01726
01727 typedef struct {
01728     BITMAPINFOHEADER bmiHeader;
01729     RGBQUAD bmiColors[1];
01730 } BITMAPINFO, *PBITMAPINFO, *LPBITMAPINFO;
01731

```

```

01732 typedef struct
01733 {
01734     DWORD    bcSize;
01735     WORD     bcWidth;
01736     WORD     bcHeight;
01737     WORD     bcPlanes;
01738     WORD     bcBitCount;
01739 } BITMAPCOREHEADER, *PBITMAPCOREHEADER, *LPBITMAPCOREHEADER;
01740
01741 typedef struct
01742 {
01743     BITMAPCOREHEADER bmciHeader;
01744     RGBTRIPLE        bmciColors[1];
01745 } BITMAPCOREINFO, *PBITMAPCOREINFO, *LPBITMAPCOREINFO;
01746
01747 #define DIB_RGB_COLORS    0
01748 #define DIB_PAL_COLORS    1
01749 #define CBM_INIT          4
01750
01751 typedef struct
01752 {
01753     BITMAP        dsBm;
01754     BITMAPINFOHEADER dsBmih;
01755     DWORD         dsBitFields[3];
01756     HANDLE        dshSection;
01757     DWORD         dsOffset;
01758 } DIBSECTION, *PDIBSECTION, *LPDIBSECTION;
01759
01760 /* Stock GDI objects for GetStockObject() */
01761
01762 #define WHITE_BRUSH        0
01763 #define LTGRAY_BRUSH      1
01764 #define GRAY_BRUSH        2
01765 #define DKGRAY_BRUSH      3
01766 #define BLACK_BRUSH       4
01767 #define NULL_BRUSH        5
01768 #define HOLLOW_BRUSH      5
01769 #define WHITE_PEN         6
01770 #define BLACK_PEN         7
01771 #define NULL_PEN          8
01772 #define OEM_FIXED_FONT    10
01773 #define ANSI_FIXED_FONT   11
01774 #define ANSI_VAR_FONT     12
01775 #define SYSTEM_FONT       13
01776 #define DEVICE_DEFAULT_FONT 14
01777 #define DEFAULT_PALETTE   15
01778 #define SYSTEM_FIXED_FONT 16
01779 #define DEFAULT_GUI_FONT  17
01780
01781 #define STOCK_LAST        17
01782
01783 #define CLR_INVALID        0xffffffff
01784 /* Metafile header structure */
01785 #include "pshpack2.h"
01786 typedef struct
01787 {
01788     WORD        mtType;
01789     WORD        mtHeaderSize;
01790     WORD        mtVersion;
01791     DWORD       mtSize;
01792     WORD        mtNoObjects;
01793     DWORD       mtMaxRecord;
01794     WORD        mtNoParameters;
01795 } METAHEADER, *PMETAHEADER, *LPMETAHEADER;
01796 #include "poppack.h"
01797
01798 /* Metafile typical record structure */
01799 typedef struct
01800 {
01801     DWORD       rdSize;
01802     WORD        rdFunction;
01803     WORD        rdParm[1];
01804 } METARECORD, *PMETARECORD, *LPMETARECORD;
01805
01806 /* Handle table structure */
01807
01808 typedef struct
01809 {
01810     HGDIOBJ objectHandle[1];
01811 } HANDLETABLE, *PHANDLETABLE, *LPHANDLETABLE;
01812
01813
01814 /* Clipboard metafile picture structure */
01815 typedef struct
01816 {
01817     LONG        mm;
01818     LONG        xExt;

```

```
01819     LONG        yExt;
01820     HMETAFILE    hMF;
01821 } METAFILEPICT, *LPMETAFILEPICT;
01822
01823
01824 /* Metafile functions */
01825 #define META_SETBKCOLOR          0x0201
01826 #define META_SETBKMODE          0x0102
01827 #define META_SETMAPMODE         0x0103
01828 #define META_SETROP2            0x0104
01829 #define META_SETRELABS         0x0105
01830 #define META_SETPOLYFILLMODE    0x0106
01831 #define META_SETSTRETCHBLTMODE  0x0107
01832 #define META_SETTEXTCHAREXTRA   0x0108
01833 #define META_SETTEXTCOLOR      0x0209
01834 #define META_SETTEXTJUSTIFICATION 0x020A
01835 #define META_SETWINDOWORG       0x020B
01836 #define META_SETWINDOWEXT      0x020C
01837 #define META_SETVIEWPORTORG     0x020D
01838 #define META_SETVIEWPORTEXT    0x020E
01839 #define META_OFFSETWINDOWORG    0x020F
01840 #define META_SCALEWINDOWEXT     0x0410
01841 #define META_OFFSETVIEWPORTORG  0x0211
01842 #define META_SCALEVIEWPORTEXT   0x0412
01843 #define META_LINETO             0x0213
01844 #define META_MOVETO             0x0214
01845 #define META_EXCLUDECLIPRECT    0x0415
01846 #define META_INTERSECTCLIPRECT  0x0416
01847 #define META_ARC                0x0817
01848 #define META_ELLIPSE            0x0418
01849 #define META_FLOODFILL          0x0419
01850 #define META_PIE                0x081A
01851 #define META_RECTANGLE          0x041B
01852 #define META_ROUNDRECT          0x061C
01853 #define META_PATBLT             0x061D
01854 #define META_SAVEDC             0x001E
01855 #define META_SETPIXEL           0x041F
01856 #define META_OFFSETCLIPRGN      0x0220
01857 #define META_TEXTOUT            0x0521
01858 #define META_BITBLT             0x0922
01859 #define META_STRETCHBLT         0x0B23
01860 #define META_POLYGON            0x0324
01861 #define META_POLYLINE           0x0325
01862 #define META_ESCAPE             0x0626
01863 #define META_RESTOREDC          0x0127
01864 #define META_FILLREGION         0x0228
01865 #define META_FRAMEREGION        0x0429
01866 #define META_INVERTREGION       0x012A
01867 #define META_PAINTREGION        0x012B
01868 #define META_SELECTCLIPREGION   0x012C
01869 #define META_SELECTOBJECT       0x012D
01870 #define META_SETTEXTALIGN       0x012E
01871 #define META_DRAWTEXT           0x062F
01872 #define META_CHORD              0x0830
01873 #define META_SETMAPPERFLAGS     0x0231
01874 #define META_EXTTEXTOUT         0x0A32
01875 #define META_SETDIBTODEV        0x0D33
01876 #define META_SELECTPALETTE      0x0234
01877 #define META_REALIZEPALETTE     0x0035
01878 #define META_ANIMATEPALETTE     0x0436
01879 #define META_SETPALENTRIES      0x0037
01880 #define META_POLYPOLYGON        0x0538
01881 #define META_RESIZEPALETTE      0x0139
01882 #define META_DIBBITBLT         0x0940
01883 #define META_DIBSTRETCHBLT      0x0B41
01884 #define META_DIBCREATEPATTERNBRUSH 0x0142
01885 #define META_STRETCHDIB         0x0F43
01886 #define META_EXTFLOODFILL       0x0548
01887 #define META_RESETDC            0x014C
01888 #define META_STARTDOC           0x014D
01889 #define META_STARTPAGE          0x004F
01890 #define META_ENDPAGE            0x0050
01891 #define META_ABORTDOC           0x0052
01892 #define META_ENDDOC             0x005E
01893 #define META_DELETEOBJECT       0x01F0
01894 #define META_CREATEPALETTE      0x00F7
01895 #define META_CREATEBRUSH        0x00F8
01896 #define META_CREATEPATTERNBRUSH 0x01F9
01897 #define META_CREATEPENINDIRECT  0x02FA
01898 #define META_CREATEFONTINDIRECT 0x02FB
01899 #define META_CREATEBRUSHINDIRECT 0x02FC
01900 #define META_CREATEBITMAPINDIRECT 0x02FD
01901 #define META_CREATEBITMAP       0x06FE
01902 #define META_CREATEREGION       0x06FF
01903 #define META_UNKNOWN            0x0529 /* FIXME: unknown meta magic */
01904
01905 typedef INT CALLBACK (*MFENUMPROC) (HDC, HANDLETABLE*, METARECORD*,
```

```

01906                                     INT, LPARAM);
01907
01908 /* enhanced metafile structures and functions */
01909
01910 /* note that ENHMETAHEADER is just a particular kind of ENHMETARECORD,
01911    ie. the header is just the first record in the metafile */
01912 typedef struct {
01913     DWORD iType;
01914     DWORD nSize;
01915     RECTL rclBounds;
01916     RECTL rclFrame;
01917     DWORD dSignature;
01918     DWORD nVersion;
01919     DWORD nBytes;
01920     DWORD nRecords;
01921     WORD nHandles;
01922     WORD sReserved;
01923     DWORD nDescription;
01924     DWORD offDescription;
01925     DWORD nPalEntries;
01926     SIZEL szlDevice;
01927     SIZEL szlMillimeters;
01928
01929     /* Fields for winver >= win95 */
01930     DWORD cbPixelFormat;
01931     DWORD offPixelFormat;
01932     DWORD bOpenGL;
01933 #if 1
01934     /* Fields for winver >= win98 */
01935     SIZEL szlMicrometers;
01936 #endif
01937 } ENHMETAHEADER, *PENHMETAHEADER, *LPENHMETAHEADER;
01938
01939 typedef struct {
01940     DWORD iType;
01941     DWORD nSize;
01942     DWORD dParm[1];
01943 } ENHMETARECORD, *LPENHMETARECORD;
01944
01945 typedef struct {
01946     DWORD iType;
01947     DWORD nSize;
01948 } EMR, *PEMR;
01949
01950 typedef struct {
01951     POINTL ptlReference;
01952     DWORD nChars;
01953     DWORD offString;
01954     DWORD fOptions;
01955     RECTL rcl;
01956     DWORD offDx;
01957 } EMRTEXT, *PEMRTEXT;
01958
01959 typedef struct {
01960     EMR emr;
01961 } EMRABORTPATH, *PEMRABORTPATH,
01962     EMRBEGINPATH, *PEMRBEGINPATH,
01963     EMRENDPATH, *PEMRENDPATH,
01964     EMRCLOSEFIGURE, *PEMRCLOSEFIGURE,
01965     EMRFLATTENPATH, *PEMRFLATTENPATH,
01966     EMRWIDENPATH, *PEMRWIDENPATH,
01967     EMRSETMETARGN, *PEMRSETMETARGN,
01968     EMRSAVEDC, *PEMRSAVEDC,
01969     EMRREALIZEPALETTE, *PEMRREALIZEPALETTE;
01970
01971 typedef struct {
01972     EMR emr;
01973     POINTL ptlCenter;
01974     DWORD nRadius;
01975     FLOAT eStartAngle;
01976     FLOAT eSweepAngle;
01977 } EMRANGLEARC, *PEMRANGLEARC;
01978
01979 typedef struct {
01980     EMR emr;
01981     RECTL rclBox;
01982     POINTL ptlStart;
01983     POINTL ptlEnd;
01984 } EMRARC, *PEMRARC,
01985     EMRARCTO, *PEMRARCTO,
01986     EMRCHORD, *PEMRCHORD,
01987     EMRPIE, *PEMRPIE;
01988
01989 typedef struct {
01990     EMR emr;
01991     RECTL rclBounds;
01992     LONG xDest;

```

```

01993     LONG        yDest;
01994     LONG        cxDest;
01995     LONG        cyDest;
01996     DWORD       dwRop;
01997     LONG        xSrc;
01998     LONG        ySrc;
01999     XFORM       xformSrc;
02000     COLORREF    crBkColorSrc;
02001     DWORD       iUsageSrc;
02002     DWORD       offBmiSrc;
02003     DWORD       cbBmiSrc;
02004     DWORD       offBitsSrc;
02005     DWORD       cbBitsSrc;
02006 } EMRBITBLT, *PEMRBITBLT;
02007
02008 typedef struct {
02009     EMR          emr;
02010     DWORD        ihBrush;
02011     LOGBRUSH     lb;
02012 } EMRCREATEBRUSHINDIRECT, *PEMRCREATEBRUSHINDIRECT;
02013
02014 typedef struct {
02015     EMR          emr;
02016     DWORD        ihCS;
02017     LOGCOLORSPACEA lcs;
02018 } EMRCREATECOLORSPACE, *PEMRCREATECOLORSPACE;
02019
02020 typedef struct {
02021     EMR          emr;
02022     DWORD        ihCS;
02023     LOGCOLORSPACEW lcs;
02024     DWORD        dwFlags;
02025     DWORD        cbData;
02026     BYTE         Data[1];
02027 } EMRCREATECOLORSPACEW, *PEMRCREATECOLORSPACEW;
02028
02029 typedef struct {
02030     EMR          emr;
02031     DWORD        ihBrush;
02032     DWORD        iUsage;
02033     DWORD        offBmi;
02034     DWORD        cbBmi;
02035     DWORD        offBits;
02036     DWORD        cbBits;
02037 } EMRCREATEDIBPATTERNBRUSHPT, *PEMRCREATEDIBPATTERNBRUSHPT;
02038
02039 typedef struct {
02040     EMR          emr;
02041     DWORD        ihBrush;
02042     DWORD        iUsage;
02043     DWORD        offBmi;
02044     DWORD        cbBmi;
02045     DWORD        offBits;
02046     DWORD        cbBits;
02047 } EMRCREATEMONOBRUSH, *PEMRCREATEMONOBRUSH;
02048
02049 typedef struct {
02050     EMR          emr;
02051     DWORD        ihPal;
02052     LOGPALETTE   lgpl;
02053 } EMRCREATEPALETTE, *PEMRCREATEPALETTE;
02054
02055 typedef struct {
02056     EMR          emr;
02057     DWORD        ihPen;
02058     LOGPEN       lopn;
02059 } EMRCREATEPEN, *PEMRCREATEPEN;
02060
02061 typedef struct {
02062     EMR          emr;
02063     DWORD        ihCS;
02064 } EMRDELETECOLORSPACE, *PEMRDELETECOLORSPACE;
02065 EMRSELECTCOLORSPACE, *PEMRSELECTCOLORSPACE;
02066 EMRSETCOLORSPACE, *PEMRSETCOLORSPACE;
02067
02068 typedef struct {
02069     EMR          emr;
02070     DWORD        ihObject;
02071 } EMRDELETEOBJECT, *PEMRDELETEOBJECT;
02072 EMRSELECTOBJECT, *PEMRSELECTOBJECT;
02073
02074 typedef struct {
02075     EMR          emr;
02076     RECT         rcBox;
02077 } EMRELLIPSE, *PEMRELLIPSE;
02078 EMRRECTANGLE, *PEMRRECTANGLE;
02079

```

```

02080 typedef struct {
02081     EMR    emr;
02082     DWORD  nPalEntries;
02083     DWORD  offPalEntries;
02084     DWORD  nSizeLast;
02085 } EMREOF, *PEMREOF;
02086
02087 typedef struct {
02088     EMR    emr;
02089     RECTL  rclClip;
02090 } EMREXCLUDECLIPRECT, *PEMREXCLUDECLIPRECT,
02091   EMRINTERSECTCLIPRECT, *PEMRINTERSECTCLIPRECT;
02092
02093 typedef struct {
02094     EMR    emr;
02095     DWORD  ihFont;
02096     EXTLOGFONTW elfw;
02097 } EMREXTCREATEFONTINDIRECTW, *PEMREXTCREATEFONTINDIRECTW;
02098
02099 typedef struct {
02100     EMR    emr;
02101     DWORD  ihPen;
02102     DWORD  offBmi;
02103     DWORD  cbBmi;
02104     DWORD  offBits;
02105     DWORD  cbBits;
02106     EXTLOGPEN elp;
02107 } EMREXTCREATEPEN, *PEMREXTCREATEPEN;
02108
02109 typedef struct {
02110     EMR    emr;
02111     POINTL  ptlStart;
02112     COLORREF crColor;
02113     DWORD  iMode;
02114 } EMREXTFLOODFILL, *PEMREXTFLOODFILL;
02115
02116 typedef struct {
02117     EMR    emr;
02118     DWORD  cbRgnData;
02119     DWORD  iMode;
02120     BYTE  RgnData[1];
02121 } EMREXTSELECTCLIPRGN, *PEMREXTSELECTCLIPRGN;
02122
02123 typedef struct {
02124     EMR    emr;
02125     RECTL  rclBounds;
02126     DWORD  iGraphicsMode;
02127     FLOAT  exScale;
02128     FLOAT  eyScale;
02129     EMRTEXT emrtext;
02130 } EMREXTTEXTOUTA, *PEMREXTTEXTOUTA,
02131   EMREXTTEXTOUTW, *PEMREXTTEXTOUTW;
02132
02133 typedef struct {
02134     EMR    emr;
02135     RECTL  rclBounds;
02136 } EMRFILLPATH, *PEMRFILLPATH,
02137   EMRSTROKEANDFILLPATH, *PEMRSTROKEANDFILLPATH,
02138   EMRSTROKEPATH, *PEMRSTROKEPATH;
02139
02140 typedef struct {
02141     EMR    emr;
02142     RECTL  rclBounds;
02143     DWORD  cbRgnData;
02144     DWORD  ihBrush;
02145     BYTE  RgnData[1];
02146 } EMRFILLRGN, *PEMRFILLRGN;
02147
02148 typedef struct {
02149     DWORD  signature;
02150     DWORD  nVersion;
02151     DWORD  cbData;
02152     DWORD  offData;
02153 } EMRFORMAT, *PEMRFORMAT;
02154
02155 typedef struct {
02156     EMR    emr;
02157     RECTL  rclBounds;
02158     DWORD  cbRgnData;
02159     DWORD  ihBrush;
02160     SIZEL  szlStroke;
02161     BYTE  RgnData[1];
02162 } EMRFRAMERGN, *PEMRFRAMERGN;
02163
02164 typedef struct {
02165     EMR    emr;
02166     DWORD  cbData;

```

```
02167     BYTE    Data[1];
02168 } EMRGDICOMMENT, *PEMRGDICOMMENT;
02169
02170 #if 0
02171 typedef struct {
02172     EMR      emr;
02173     RECTL    rclBounds;
02174     DWORD    nVer;
02175     DWORD    nTri;
02176     ULONG    ulMode;
02177     TRIVERTEX Ver[1];
02178 } EMRGRADIENTFILL, *PEMRGRADIENTFILL;
02179 #endif
02180
02181 typedef struct {
02182     EMR      emr;
02183     RECTL    rclBounds;
02184     DWORD    cbRgnData;
02185     BYTE     RgnData[1];
02186 } EMRINVERTTRGN, *PEMRINVERTTRGN,
02187 EMRPAINTTRGN, *PEMRPAINTTRGN;
02188
02189 typedef struct {
02190     EMR      emr;
02191     POINTL   ptl;
02192 } EMRLINETO, *PEMRLINETO,
02193 EMRMOVETOEX, *PEMRMOVETOEX;
02194
02195 typedef struct {
02196     EMR      emr;
02197     RECTL    rclBounds;
02198     LONG     xDest;
02199     LONG     yDest;
02200     LONG     cxDest;
02201     LONG     cyDest;
02202     DWORD    dwRop;
02203     LONG     xSrc;
02204     LONG     ySrc;
02205     XFORM     xformSrc;
02206     COLORREF  crBkColorSrc;
02207     DWORD     iUsageSrc;
02208     DWORD     offBmiSrc;
02209     DWORD     cbBmiSrc;
02210     DWORD     offBitsSrc;
02211     DWORD     cbBitsSrc;
02212     LONG     xMask;
02213     LONG     yMask;
02214     DWORD     iUsageMask;
02215     DWORD     offBmiMask;
02216     DWORD     cbBmiMask;
02217     DWORD     offBitsMask;
02218     DWORD     cbBitsMask;
02219 } EMRMASKBLT, *PEMRMASKBLT;
02220
02221 typedef struct {
02222     EMR      emr;
02223     XFORM     xform;
02224     DWORD     iMode;
02225 } EMRMODIFYWORLDTRANSFORM, *PEMRMODIFYWORLDTRANSFORM;
02226
02227 typedef struct {
02228     EMR      emr;
02229     POINTL   ptlOffset;
02230 } EMROFFSETCLIPRGN, *PEMROFFSETCLIPRGN;
02231
02232 typedef struct {
02233     EMR      emr;
02234     RECTL    rclBounds;
02235     POINTL   aptlDst[3];
02236     LONG     xSrc;
02237     LONG     ySrc;
02238     LONG     cxSrc;
02239     LONG     cySrc;
02240     XFORM     xformSrc;
02241     COLORREF  crBkColorSrc;
02242     DWORD     iUsageSrc;
02243     DWORD     offBmiSrc;
02244     DWORD     cbBmiSrc;
02245     DWORD     offBitsSrc;
02246     DWORD     cbBitsSrc;
02247     LONG     xMask;
02248     LONG     yMask;
02249     DWORD     iUsageMask;
02250     DWORD     offBmiMask;
02251     DWORD     cbBmiMask;
02252     DWORD     offBitsMask;
02253     DWORD     cbBitsMask;
```

```
02254 } EMRPLGBLT, *PEMRPLGBLT;
02255
02256 typedef struct {
02257     EMR      emr;
02258     RECTL    rclBounds;
02259     DWORD    cptl;
02260     POINTL   aptl[1];
02261 } EMRPOLYLINE, *PEMRPOLYLINE,
02262 EMRPOLYBEZIER, *PEMRPOLYBEZIER,
02263 EMRPOLYGON, *PEMRPOLYGON,
02264 EMRPOLYBEZIERTO, *PEMRPOLYBEZIERTO,
02265 EMRPOLYLINETO, *PEMRPOLYLINETO;
02266
02267 typedef struct {
02268     EMR      emr;
02269     RECTL    rclBounds;
02270     DWORD    cptl;
02271     POINTL   aptl[1];
02272     BYTE     abTypes[1];
02273 } EMRPOLYDRAW, *PEMRPOLYDRAW;
02274
02275 typedef struct {
02276     EMR      emr;
02277     RECTL    rclBounds;
02278     DWORD    nPolys;
02279     DWORD    cptl;
02280     DWORD    aPolyCounts[1];
02281     POINTL   aptl[1];
02282 } EMRPOLYPOLYLINE, *PEMRPOLYPOLYLINE,
02283 EMRPOLYPOLYGON, *PEMRPOLYPOLYGON;
02284
02285 typedef struct {
02286     EMR      emr;
02287     RECTL    rclBounds;
02288     DWORD    iGraphicsMode;
02289     FLOAT    exScale;
02290     FLOAT    eyScale;
02291     LONG     cStrings;
02292     EMRTEXT  aemrtext[1];
02293 } EMRPOLYTEXTOUTA, *PEMRPOLYTEXTOUTA,
02294 EMRPOLYTEXTOUTW, *PEMRPOLYTEXTOUTW;
02295
02296 typedef struct {
02297     EMR      emr;
02298     DWORD    ihPal;
02299     DWORD    cEntries;
02300 } EMRRESIZEPALETTE, *PEMRRESIZEPALETTE;
02301
02302 typedef struct {
02303     EMR      emr;
02304     LONG     iRelative;
02305 } EMRRESTOREDC, *PEMRRESTOREDC;
02306
02307 typedef struct {
02308     EMR      emr;
02309     RECTL    rclBox;
02310     SIZEL    szlCorner;
02311 } EMRROUNDRECT, *PEMRROUNDRECT;
02312
02313 typedef struct {
02314     EMR      emr;
02315     LONG     xNum;
02316     LONG     xDenom;
02317     LONG     yNum;
02318     LONG     yDenom;
02319 } EMRSCALEVIEWPORTEXT, *PEMRSCALEVIEWPORTEXT,
02320 EMRSCALEWINDOWTEXT, *PEMRSCALEWINDOWTEXT;
02321
02322 typedef struct {
02323     EMR      emr;
02324     DWORD    iMode;
02325 } EMRSELECTCLIPPATH, *PEMRSELECTCLIPPATH,
02326 EMRSETBKMODE, *PEMRSETBKMODE,
02327 EMRSETMAPMODE, *PEMRSETMAPMODE,
02328 EMRSETPOLYFILLMODE, *PEMRSETPOLYFILLMODE,
02329 EMRSETROP2, *PEMRSETROP2,
02330 EMRSETSTRETCHBLTMODE, *PEMRSETSTRETCHBLTMODE,
02331 EMRSETTEXTALIGN, *PEMRSETTEXTALIGN,
02332 EMRSETICMMODE, *PEMRSETICMMODE,
02333 EMRSETLAYOUT, *PEMRSETLAYOUT;
02334
02335 typedef struct {
02336     EMR      emr;
02337     DWORD    ihPal;
02338 } EMRSELECTPALETTE, *PEMRSELECTPALETTE;
02339
02340 typedef struct {
```

```
02341     EMR     emr;
02342     DWORD   iArcDirection;
02343 } EMRSETARCDIRECTION, *PEMRSETARCDIRECTION;
02344
02345 typedef struct {
02346     EMR     emr;
02347     COLORREF crColor;
02348 } EMRSETBKCOLOR, *PEMRSETBKCOLOR,
02349 EMRSETTEXTCOLOR, *PEMRSETTEXTCOLOR;
02350
02351 typedef struct {
02352     EMR     emr;
02353     POINTL  ptlOrigin;
02354 } EMRSETBRUSHORGE, *PEMRSETBRUSHORGE,
02355 EMRSETVIEWPORTORGE, *PEMRSETVIEWPORTORGE,
02356 EMRSETWINDOWORGE, *PEMRSETWINDOWORGE;
02357
02358 typedef struct {
02359     EMR     emr;
02360     COLORADJUSTMENT ColorAdjustment;
02361 } EMRSETCOLORADJUSTMENT, *PEMRSETCOLORADJUSTMENT;
02362
02363 typedef struct {
02364     EMR     emr;
02365     RECTL   rclBounds;
02366     LONG    xDest;
02367     LONG    yDest;
02368     LONG    xSrc;
02369     LONG    ySrc;
02370     LONG    cxSrc;
02371     LONG    cySrc;
02372     DWORD   offBmiSrc;
02373     DWORD   cbBmiSrc;
02374     DWORD   offBitsSrc;
02375     DWORD   cbBitsSrc;
02376     DWORD   iUsageSrc;
02377     DWORD   iStartScan;
02378     DWORD   cScans;
02379 } EMRSETDIBITSTODEVICE, *PEMRSETDIBITSTODEVICE;
02380
02381 typedef struct {
02382     EMR     emr;
02383     DWORD   dwFlags;
02384 } EMRSETMAPPERFLAGS, *PEMRSETMAPPERFLAGS;
02385
02386 typedef struct {
02387     EMR     emr;
02388     FLOAT   eMiterLimit;
02389 } EMRSETMITERLIMIT, *PEMRSETMITERLIMIT;
02390
02391 typedef struct {
02392     EMR     emr;
02393     DWORD   ihPal;
02394     DWORD   iStart;
02395     DWORD   cEntries;
02396     PALETTEENTRY aPalEntries[1];
02397 } EMRSETPALETTEENTRIES, *PEMRSETPALETTEENTRIES;
02398
02399 typedef struct {
02400     EMR     emr;
02401     POINTL  ptlPixel;
02402     COLORREF crColor;
02403 } EMRSETPIXELV, *PEMRSETPIXELV;
02404
02405 typedef struct {
02406     EMR     emr;
02407     SIZE    szlExtent;
02408 } EMRSETVIEWPORTTEXT, *PEMRSETVIEWPORTTEXT,
02409 EMRSETWINDOWTEXT, *PEMRSETWINDOWTEXT;
02410
02411 typedef struct {
02412     EMR     emr;
02413     XFORM   xform;
02414 } EMRSETWORLDTRANSFORM, *PEMRSETWORLDTRANSFORM;
02415
02416 typedef struct {
02417     EMR     emr;
02418     RECTL   rclBounds;
02419     LONG    xDest;
02420     LONG    yDest;
02421     LONG    cxDest;
02422     LONG    cyDest;
02423     DWORD   dwRop;
02424     LONG    xSrc;
02425     LONG    ySrc;
02426     XFORM   xformSrc;
02427     COLORREF crBkColorSrc;
```

```
02428     DWORD    iUsageSrc;
02429     DWORD    offBmiSrc;
02430     DWORD    cbBmiSrc;
02431     DWORD    offBitsSrc;
02432     DWORD    cbBitsSrc;
02433     LONG     cxSrc;
02434     LONG     cySrc;
02435 } EMRSTRETCHBLT, *PEMRSTRETCHBLT;
02436
02437 typedef struct {
02438     EMR     emr;
02439     RECTL   rclBounds;
02440     LONG    xDest;
02441     LONG    yDest;
02442     LONG    xSrc;
02443     LONG    ySrc;
02444     LONG    cxSrc;
02445     LONG    cySrc;
02446     DWORD   offBmiSrc;
02447     DWORD   cbBmiSrc;
02448     DWORD   offBitsSrc;
02449     DWORD   cbBitsSrc;
02450     DWORD   iUsageSrc;
02451     DWORD   dwRop;
02452     LONG    cxDest;
02453     LONG    cyDest;
02454 } EMRSTRETCHDIBITS, *PEMRSTRETCHDIBITS;
02455
02456 typedef struct {
02457     EMR             emr;
02458     PIXELFORMATDESCRIPTOR pfd;
02459 } EMRPIXELFORMAT, *PEMRPIXELFORMAT;
02460
02461 typedef struct tagEMRGLSRECORD {
02462     EMR     emr;
02463     DWORD   cbData;
02464     BYTE    Data[1];
02465 } EMRGLSRECORD, *PEMRGLSRECORD;
02466
02467 typedef struct {
02468     EMR     emr;
02469     RECTL   rclBounds;
02470     DWORD   cbData;
02471     BYTE    Data[1];
02472 } EMRGLSBOUNDEDRECORD, *PEMRGLSBOUNDEDRECORD;
02473
02474 typedef INT CALLBACK (*ENHMFENUMPROC) (HDC, LPHANDLETABLE,
02475                                         LPENHMETARECORD, INT, LPVOID);
02476
02477 #define EMR_HEADER 1
02478 #define EMR_POLYBEZIER 2
02479 #define EMR_POLYGON 3
02480 #define EMR_POLYLINE 4
02481 #define EMR_POLYBEZIERTO 5
02482 #define EMR_POLYLINETO 6
02483 #define EMR_POLYPOLYLINE 7
02484 #define EMR_POLYPOLYGON 8
02485 #define EMR_SETWINDOWEXTEX 9
02486 #define EMR_SETWINDOWORGEX 10
02487 #define EMR_SETVIEWPORTEXTEX 11
02488 #define EMR_SETVIEWPORTORGEX 12
02489 #define EMR_SETBRUSHORGEX 13
02490 #define EMR_EOF 14
02491 #define EMR_SETPIXELV 15
02492 #define EMR_SETMAPPERFLAGS 16
02493 #define EMR_SETMAPMODE 17
02494 #define EMR_SETBKMODE 18
02495 #define EMR_SETPOLYFILLMODE 19
02496 #define EMR_SETROP2 20
02497 #define EMR_SETSTRETCHBLTMODE 21
02498 #define EMR_SETTEXTALIGN 22
02499 #define EMR_SETCOLORADJUSTMENT 23
02500 #define EMR_SETTEXTCOLOR 24
02501 #define EMR_SETBKCOLOR 25
02502 #define EMR_OFFSETCLIPRGN 26
02503 #define EMR_MOVETOEX 27
02504 #define EMR_SETMETARGN 28
02505 #define EMR_EXCLUDECLIPRECT 29
02506 #define EMR_INTERSECTCLIPRECT 30
02507 #define EMR_SCALEVIEWPORTEXTEX 31
02508 #define EMR_SCALEWINDOWEXTEX 32
02509 #define EMR_SAVEDC 33
02510 #define EMR_RESTOREDC 34
02511 #define EMR_SETWORLDTRANSFORM 35
02512 #define EMR_MODIFYWORLDTRANSFORM 36
02513 #define EMR_SELECTOBJECT 37
02514 #define EMR_CREATEPEN 38
```

```

02515 #define EMR_CREATEBRUSHINDIRECT 39
02516 #define EMR_DELETEOBJECT 40
02517 #define EMR_ANGLEARC 41
02518 #define EMR_ELLIPSE 42
02519 #define EMR_RECTANGLE 43
02520 #define EMR_ROUNDRECT 44
02521 #define EMR_ARC 45
02522 #define EMR_CHORD 46
02523 #define EMR_PIE 47
02524 #define EMR_SELECTPALETTE 48
02525 #define EMR_CREATEPALETTE 49
02526 #define EMR_SETPALETTEENTRIES 50
02527 #define EMR_RESIZEPALETTE 51
02528 #define EMR_REALIZEPALETTE 52
02529 #define EMR_EXTFLOODFILL 53
02530 #define EMR_LINETO 54
02531 #define EMR_ARCTO 55
02532 #define EMR_POLYDRAW 56
02533 #define EMR_SETARCDIRECTION 57
02534 #define EMR_SETMITERLIMIT 58
02535 #define EMR_BEGINPATH 59
02536 #define EMR_ENDPATH 60
02537 #define EMR_CLOSEFIGURE 61
02538 #define EMR_FILLPATH 62
02539 #define EMR_STROKEANDFILLPATH 63
02540 #define EMR_STROKEPATH 64
02541 #define EMR_FLATTENPATH 65
02542 #define EMR_WIDENPATH 66
02543 #define EMR_SELECTCLIPPATH 67
02544 #define EMR_ABORTPATH 68
02545 #define EMR_GDICOMMENT 70
02546 #define EMR_FILLRGN 71
02547 #define EMR_FRAMERGN 72
02548 #define EMR_INVERTRGN 73
02549 #define EMR_PAINTRGN 74
02550 #define EMR_EXTSELECTCLIPRGN 75
02551 #define EMR_BITBLT 76
02552 #define EMR_STRETCHBLT 77
02553 #define EMR_MASKBLT 78
02554 #define EMR_PLGBLT 79
02555 #define EMR_SETDIBITSTODEVICE 80
02556 #define EMR_STRETCHDIBITS 81
02557 #define EMR_EXTCREATEFONTINDIRECTW 82
02558 #define EMR_EXTTEXTOUTA 83
02559 #define EMR_EXTTEXTOUTW 84
02560 #define EMR_POLYBEZIER16 85
02561 #define EMR_POLYGON16 86
02562 #define EMR_POLYLINE16 87
02563 #define EMR_POLYBEZIER16TO16 88
02564 #define EMR_POLYLINETO16 89
02565 #define EMR_POLYPOLYLINE16 90
02566 #define EMR_POLYPOLYGON16 91
02567 #define EMR_POLYDRAW16 92
02568 #define EMR_CREATEMONOBRUSH 93
02569 #define EMR_CREATEDIBPATTERNBRUSHPT 94
02570 #define EMR_EXTCREATEPEN 95
02571 #define EMR_POLYTEXTOUTA 96
02572 #define EMR_POLYTEXTOUTW 97
02573 #define EMR_SETICMMODE 98
02574 #define EMR_CREATECOLORSPACE 99
02575 #define EMR_SETCOLORSPACE 100
02576 #define EMR_DELETECOLORSPACE 101
02577 #define EMR_GLSRECORD 102
02578 #define EMR_GLSBOUNDEDRECORD 103
02579 #define EMR_PIXELFORMAT 104
02580
02581 #define EMR_MIN 1
02582 #define EMR_MAX 104
02583
02584 #define ENHMETA_SIGNATURE 1179469088
02585 #define ENHMETA_STOCK_OBJECT 0x80000000
02586
02587 #define GDICPMMENT_INDENTIFIER 0x43494447
02588 #define GDICOMMENT_WINDOWS_METAFILE 0x80000000
02589 #define GDICOMMENT_BEGINGROUP 0x80000001
02590 #define GDICOMMENT_ENDGROUP 0x80000002
02591 #define GDICOMMENT_MULTIFORMATS 0x80000003
02592 #define EPS_SIGNATURE 0x46535045
02593
02594 #define CCHDEVICENAME 32
02595 #define CCHFORMNAME 32
02596
02597 typedef struct
02598 {
02599     BYTE dmDeviceName[CCHDEVICENAME];
02600     WORD dmSpecVersion;
02601     WORD dmDriverVersion;

```

```

02602     WORD    dmSize;
02603     WORD    dmDriverExtra;
02604     DWORD    dmFields;
02605     union u10 {
02606         struct snort {
02607             SHORT    dmOrientation;
02608             SHORT    dmPaperSize;
02609             SHORT    dmPaperLength;
02610             SHORT    dmPaperWidth;
02611             } DUMMYSTRUCTNAME1;
02612             POINTL    dmPosition;
02613         } DUMMYUNIONNAME1;
02614         SHORT    dmScale;
02615         SHORT    dmCopies;
02616         SHORT    dmDefaultSource;
02617         SHORT    dmPrintQuality;
02618         SHORT    dmColor;
02619         SHORT    dmDuplex;
02620         SHORT    dmYResolution;
02621         SHORT    dmTTOption;
02622         SHORT    dmCollate;
02623         BYTE    dmFormName[CCHFORMNAME];
02624         WORD    dmLogPixels;
02625         DWORD    dmBitsPerPel;
02626         DWORD    dmPelsWidth;
02627         DWORD    dmPelsHeight;
02628         DWORD    dmDisplayFlags;
02629         DWORD    dmDisplayFrequency;
02630         DWORD    dmICMMethod;
02631         DWORD    dmICMIntent;
02632         DWORD    dmMediaType;
02633         DWORD    dmDitherType;
02634         DWORD    dmReserved1;
02635         DWORD    dmReserved2;
02636         DWORD    dmPanningWidth;
02637         DWORD    dmPanningHeight;
02638     } DEVMODEA, *PDEVMODEA, *LPDEVMODEA;
02639
02640     typedef struct
02641     {
02642         WCHAR    dmDeviceName[CCHDEVICENAME];
02643         WORD    dmSpecVersion;
02644         WORD    dmDriverVersion;
02645         WORD    dmSize;
02646         WORD    dmDriverExtra;
02647         DWORD    dmFields;
02648         union u20 {
02649             struct blorf {
02650                 SHORT    dmOrientation;
02651                 SHORT    dmPaperSize;
02652                 SHORT    dmPaperLength;
02653                 SHORT    dmPaperWidth;
02654                 } DUMMYSTRUCTNAME1;
02655                 POINTL    dmPosition;
02656             } DUMMYUNIONNAME1;
02657             SHORT    dmScale;
02658             SHORT    dmCopies;
02659             SHORT    dmDefaultSource;
02660             SHORT    dmPrintQuality;
02661             SHORT    dmColor;
02662             SHORT    dmDuplex;
02663             SHORT    dmYResolution;
02664             SHORT    dmTTOption;
02665             SHORT    dmCollate;
02666             WCHAR    dmFormName[CCHFORMNAME];
02667             WORD    dmLogPixels;
02668             DWORD    dmBitsPerPel;
02669             DWORD    dmPelsWidth;
02670             DWORD    dmPelsHeight;
02671             DWORD    dmDisplayFlags;
02672             DWORD    dmDisplayFrequency;
02673             DWORD    dmICMMethod;
02674             DWORD    dmICMIntent;
02675             DWORD    dmMediaType;
02676             DWORD    dmDitherType;
02677             DWORD    dmReserved1;
02678             DWORD    dmReserved2;
02679             DWORD    dmPanningWidth;
02680             DWORD    dmPanningHeight;
02681         } DEVMODEW, *PDEVMODEW, *LPDEVMODEW;
02682
02683     DECL_WINELIB_TYPE_AW(DEVMODE)
02684     DECL_WINELIB_TYPE_AW(PDEVMODE)
02685     DECL_WINELIB_TYPE_AW(LPDEVMODE)
02686
02687     #define DM_SPECVERSION    0x401
02688     #define DM_UPDATE        1

```

```
02689 #define DM_COPY      2
02690 #define DM_PROMPT     4
02691 #define DM_MODIFY      8
02692
02693 #define DM_IN_BUFFER    DM_MODIFY
02694 #define DM_IN_PROMPT   DM_PROMPT
02695 #define DM_OUT_BUFFER   DM_COPY
02696 #define DM_OUT_DEFAULT  DM_UPDATE
02697
02698 #define DM_ORIENTATION  0x00000001L
02699 #define DM_PAPERSIZE    0x00000002L
02700 #define DM_PAPERLENGTH 0x00000004L
02701 #define DM_PAPERWIDTH   0x00000008L
02702 #define DM_SCALE        0x00000010L
02703 #define DM_POSITION     0x00000020L
02704 #define DM_COPIES       0x00000100L
02705 #define DM_DEFAULTSOURCE 0x00000200L
02706 #define DM_PRINTQUALITY 0x00000400L
02707 #define DM_COLOR        0x00000800L
02708 #define DM_DUPLEX       0x00001000L
02709 #define DM_YRESOLUTION  0x00002000L
02710 #define DM_TTOPTION     0x00004000L
02711 #define DM_COLLATE      0x00008000L
02712 #define DM_FORMNAME     0x00010000L
02713 #define DM_LOGPIXELS    0x00020000L
02714 #define DM_BITSPERPEL   0x00040000L
02715 #define DM_PELSWIDTH    0x00080000L
02716 #define DM_PELSHEIGHT   0x00100000L
02717 #define DM_DISPLAYFLAGS 0x00200000L
02718 #define DM_DISPLAYFREQUENCY 0x00400000L
02719 #define DM_ICMMETHOD   0x00800000L
02720 #define DM_ICMINTENT     0x01000000L
02721 #define DM_MEDIATYPE     0x02000000L
02722 #define DM_DITHERTYPE   0x04000000L
02723 #define DM_PANNINGWIDTH  0x08000000L
02724 #define DM_PANNINGHEIGHT 0x10000000L
02725
02726 #define DMORIENT_PORTRAIT 1
02727 #define DMORIENT_LANDSCAPE 2
02728
02729 #define DMPAPER_FIRST      DMPAPER_LETTER
02730 #define DMPAPER_LETTER    1
02731 #define DMPAPER_LETTERSMALL 2
02732 #define DMPAPER_TABLOID   3
02733 #define DMPAPER_LEDGER     4
02734 #define DMPAPER_LEGAL     5
02735 #define DMPAPER_STATEMENT 6
02736 #define DMPAPER_EXECUTIVE 7
02737 #define DMPAPER_A3        8
02738 #define DMPAPER_A4        9
02739 #define DMPAPER_A4SMALL  10
02740 #define DMPAPER_A5       11
02741 #define DMPAPER_B4       12
02742 #define DMPAPER_B5       13
02743 #define DMPAPER_FOLIO    14
02744 #define DMPAPER_QUARTO   15
02745 #define DMPAPER_10X14    16
02746 #define DMPAPER_11X17    17
02747 #define DMPAPER_NOTE     18
02748 #define DMPAPER_ENV_9    19
02749 #define DMPAPER_ENV_10   20
02750 #define DMPAPER_ENV_11   21
02751 #define DMPAPER_ENV_12   22
02752 #define DMPAPER_ENV_14   23
02753 #define DMPAPER_CSHEET   24
02754 #define DMPAPER_DSHEET   25
02755 #define DMPAPER_ESHEET   26
02756 #define DMPAPER_ENV_DL    27
02757 #define DMPAPER_ENV_C5    28
02758 #define DMPAPER_ENV_C3    29
02759 #define DMPAPER_ENV_C4    30
02760 #define DMPAPER_ENV_C6    31
02761 #define DMPAPER_ENV_C65   32
02762 #define DMPAPER_ENV_B4    33
02763 #define DMPAPER_ENV_B5    34
02764 #define DMPAPER_ENV_B6    35
02765 #define DMPAPER_ENV_ITALY 36
02766 #define DMPAPER_ENV_MONARCH 37
02767 #define DMPAPER_ENV_PERSONAL 38
02768 #define DMPAPER_FANFOLD_US 39
02769 #define DMPAPER_FANFOLD_STD_GERMAN 40
02770 #define DMPAPER_FANFOLD_LGL_GERMAN 41
02771 #define DMPAPER_ISO_B4    42
02772 #define DMPAPER_JAPANESE_POSTCARD 43
02773 #define DMPAPER_9X11      44
02774 #define DMPAPER_10X11     45
02775 #define DMPAPER_15X11     46
```

```

02776 #define DMPAPER_ENV_INVITE 47
02777 #define DMPAPER_RESERVED_48 48
02778 #define DMPAPER_RESERVED_49 49
02779 #define DMPAPER_LETTER_EXTRA 50
02780 #define DMPAPER_LEGAL_EXTRA 51
02781 #define DMPAPER_TABLOID_EXTRA 52
02782 #define DMPAPER_A4_EXTRA 53
02783 #define DMPAPER_LETTER_TRANSVERSE 54
02784 #define DMPAPER_A4_TRANSVERSE 55
02785 #define DMPAPER_LETTER_EXTRA_TRANSVERSE 56
02786 #define DMPAPER_A_PLUS 57
02787 #define DMPAPER_B_PLUS 58
02788 #define DMPAPER_LETTER_PLUS 59
02789 #define DMPAPER_A4_PLUS 60
02790 #define DMPAPER_A5_TRANSVERSE 61
02791 #define DMPAPER_B5_TRANSVERSE 62
02792 #define DMPAPER_A3_EXTRA 63
02793 #define DMPAPER_A5_EXTRA 64
02794 #define DMPAPER_B5_EXTRA 65
02795 #define DMPAPER_A2 66
02796 #define DMPAPER_A3_TRANSVERSE 67
02797 #define DMPAPER_A3_EXTRA_TRANSVERSE 68
02798 #define DMPAPER_DBL_JAPANESE_POSTCARD 69
02799 #define DMPAPER_A6 70
02800 #define DMPAPER_JENV_KAKU2 71
02801 #define DMPAPER_JENV_KAKU3 72
02802 #define DMPAPER_JENV_CHOU3 73
02803 #define DMPAPER_JENV_CHOU4 74
02804 #define DMPAPER_LETTER_ROTATED 75
02805 #define DMPAPER_A3_ROTATED 76
02806 #define DMPAPER_A4_ROTATED 77
02807 #define DMPAPER_A5_ROTATED 78
02808 #define DMPAPER_B4_JIS_ROTATED 79
02809 #define DMPAPER_B5_JIS_ROTATED 80
02810 #define DMPAPER_JAPANESE_POSTCARD_ROTATED 81
02811 #define DMPAPER_DBL_JAPANESE_POSTCARD_ROTATED 82
02812 #define DMPAPER_A6_ROTATED 83
02813 #define DMPAPER_JENV_KAKU2_ROTATED 84
02814 #define DMPAPER_JENV_KAKU3_ROTATED 85
02815 #define DMPAPER_JENV_CHOU3_ROTATED 86
02816 #define DMPAPER_JENV_CHOU4_ROTATED 87
02817 #define DMPAPER_B6_JIS 88
02818 #define DMPAPER_B6_JIS_ROTATED 89
02819 #define DMPAPER_12X11 90
02820 #define DMPAPER_JENV_YOU4 91
02821 #define DMPAPER_JENV_YOU4_ROTATED 92
02822 #define DMPAPER_P16K 93
02823 #define DMPAPER_P32K 94
02824 #define DMPAPER_P32KBIG 95
02825 #define DMPAPER_PENV_1 96
02826 #define DMPAPER_PENV_2 97
02827 #define DMPAPER_PENV_3 98
02828 #define DMPAPER_PENV_4 99
02829 #define DMPAPER_PENV_5 100
02830 #define DMPAPER_PENV_6 101
02831 #define DMPAPER_PENV_7 102
02832 #define DMPAPER_PENV_8 103
02833 #define DMPAPER_PENV_9 104
02834 #define DMPAPER_PENV_10 105
02835 #define DMPAPER_P16K_ROTATED 106
02836 #define DMPAPER_P32K_ROTATED 107
02837 #define DMPAPER_P32KBIG_ROTATED 108
02838 #define DMPAPER_PENV_1_ROTATED 109
02839 #define DMPAPER_PENV_2_ROTATED 110
02840 #define DMPAPER_PENV_3_ROTATED 111
02841 #define DMPAPER_PENV_4_ROTATED 112
02842 #define DMPAPER_PENV_5_ROTATED 113
02843 #define DMPAPER_PENV_6_ROTATED 114
02844 #define DMPAPER_PENV_7_ROTATED 115
02845 #define DMPAPER_PENV_8_ROTATED 116
02846 #define DMPAPER_PENV_9_ROTATED 117
02847 #define DMPAPER_PENV_10_ROTATED 118
02848
02849 #define DMPAPER_LAST DMPAPER_PENV_10_ROTATED
02850 #define DMPAPER_USER 256
02851
02852 #define DMBIN_FIRST DMBIN_UPPER
02853 #define DMBIN_UPPER 1
02854 #define DMBIN_ONLYONE 1
02855 #define DMBIN_LOWER 2
02856 #define DMBIN_MIDDLE 3
02857 #define DMBIN_MANUAL 4
02858 #define DMBIN_ENVELOPE 5
02859 #define DMBIN_ENVMANUAL 6
02860 #define DMBIN_AUTO 7
02861 #define DMBIN_TRACTOR 8
02862 #define DMBIN_SMALLFMT 9

```

```

02863 #define DMBIN_LARGEFORMAT 10
02864 #define DMBIN_LARGECAPACITY 11
02865 #define DMBIN_CASSETTE 14
02866 #define DMBIN_FORMSOURCE 15
02867 #define DMBIN_LAST DMBIN_FORMSOURCE
02868 #define DMBIN_USER 256
02869
02870 #define DMRES_DRAFT (-1)
02871 #define DMRES_LOW (-2)
02872 #define DMRES_MEDIUM (-3)
02873 #define DMRES_HIGH (-4)
02874
02875 #define DMCOLOR_MONOCHROME 1
02876 #define DMCOLOR_COLOR 2
02877
02878 #define DMDUP_SIMPLEX 1
02879 #define DMDUP_VERTICAL 2
02880 #define DMDUP_HORIZONTAL 3
02881
02882 #define DMTT_BITMAP 1
02883 #define DMTT_DOWNLOAD 2
02884 #define DMTT_SUBDEV 3
02885 #define DMTT_DOWNLOAD_OUTLINE 4
02886
02887 #define DMCOLLATE_FALSE 0
02888 #define DMCOLLATE_TRUE 1
02889
02890 #define DMICMMETHOD_NONE 1
02891 #define DMICMMETHOD_SYSTEM 2
02892 #define DMICMMETHOD_DRIVER 3
02893 #define DMICMMETHOD_DEVICE 4
02894 #define DMICMMETHOD_USER 256
02895
02896 #define DMICM_SATURATE 1
02897 #define DMICM_CONTRAST 2
02898 #define DMICM_COLORMETRIC 3
02899 #define DMICM_USER 256
02900
02901 #define DMMEDIA_STANDARD 1
02902 #define DMMEDIA_TRANSPARENCY 2
02903 #define DMMEDIA_GLOSSY 3
02904 #define DMMEDIA_USER 256
02905
02906 #define DMDITHER_NONE 1
02907 #define DMDITHER_COARSE 2
02908 #define DMDITHER_FINE 3
02909 #define DMDITHER_LINEART 4
02910 #define DMDITHER_GRAYSCALE 5
02911 #define DMDITHER_USER 256
02912
02913 typedef struct
02914 {
02915     INT cbSize;
02916     LPCSTR lpszDocName;
02917     LPCSTR lpszOutput;
02918     LPCSTR lpszDatatype;
02919     DWORD fwType;
02920 } DOCINFOA, *LPDOCINFOA;
02921
02922 typedef struct
02923 {
02924     INT cbSize;
02925     LPCWSTR lpszDocName;
02926     LPCWSTR lpszOutput;
02927     LPCWSTR lpszDatatype;
02928     DWORD fwType;
02929 } DOCINFOW, *LPDOCINFOW;
02930
02931 DECL_WINELIB_TYPE_AW(DOCINFO)
02932 DECL_WINELIB_TYPE_AW(LPDOCINFO)
02933
02934 #define DI_APPBANDING 0x0001
02935
02936 /* Flags for PolyDraw and GetPath */
02937 #define PT_CLOSEFIGURE 0x0001
02938 #define PT_LINETO 0x0002
02939 #define PT_BEZIERTO 0x0004
02940 #define PT_MOVETO 0x0006
02941
02942 #define RDH_RECTANGLES 1
02943
02944 typedef struct _RGNDATAHEADER {
02945     DWORD dwSize;
02946     DWORD iType;
02947     DWORD nCount;
02948     DWORD nRgnSize;
02949     RECT rcBound;

```

```

02950 } RGNDATAHEADER,*PRGNDATAHEADER;
02951
02952 typedef struct _RGNDATA {
02953     RGNDATAHEADER rdh;
02954     char Buffer[1];
02955 } RGNDATA,*PRGNDATA,*LPRGNDATA;
02956
02957 typedef BOOL CALLBACK (*ABORTPROC)(HDC, INT);
02958
02959 typedef struct {
02960     DWORD cb;
02961     CHAR DeviceName[32];
02962     CHAR DeviceString[128];
02963     DWORD StateFlags;
02964     CHAR DeviceID[128];
02965     CHAR DeviceKey[128];
02966 } DISPLAY_DEVICEA,*PDISPLAY_DEVICEA,*LPDISPLAY_DEVICEA;
02967
02968 typedef struct {
02969     DWORD cb;
02970     WCHAR DeviceName[32];
02971     WCHAR DeviceString[128];
02972     DWORD StateFlags;
02973     WCHAR DeviceID[128];
02974     WCHAR DeviceKey[128];
02975 } DISPLAY_DEVICEW,*PDISPLAY_DEVICEW,*LPDISPLAY_DEVICEW;
02976 DECL_WINELIB_TYPE_AW(DISPLAY_DEVICE)
02977 DECL_WINELIB_TYPE_AW(PDISPLAY_DEVICE)
02978 DECL_WINELIB_TYPE_AW(LPDISPLAY_DEVICE)
02979
02980 /* DISPLAY_DEVICE.StateFlags (?) */
02981 #define DISPLAY_DEVICE_ATTACHED_TO_DESKTOP 0x00000001
02982 #define DISPLAY_DEVICE_MULTI_DRIVER 0x00000002
02983 #define DISPLAY_DEVICE_PRIMARY_DEVICE 0x00000004
02984 #define DISPLAY_DEVICE_MIRRORING_DRIVER 0x00000008
02985 #define DISPLAY_DEVICE_VGA_COMPATIBLE 0x00000010
02986
02987 #define GDI_ERROR (0xFFFFFFFFL)
02988 #define HGDI_ERROR ((HANDLE)0xFFFFFFFFL)
02989
02990 INT WINAPI AbortDoc(HDC);
02991 BOOL WINAPI AbortPath(HDC);
02992 INT WINAPI AddFontResourceA(LPCSTR);
02993 INT WINAPI AddFontResourceW(LPCWSTR);
02994 #define AddFontResource WINELIB_NAME_AW(AddFontResource)
02995 BOOL WINAPI AngleArc(HDC, INT, INT, DWORD, FLOAT, FLOAT);
02996 BOOL WINAPI AnimatePalette(HPALETTE,UINT,UINT,const PALETTEENTRY*);
02997 BOOL WINAPI Arc(HDC,INT,INT,INT,INT,INT,INT,INT,INT);
02998 BOOL WINAPI ArcTo(HDC,INT,INT,INT,INT,INT,INT,INT,INT);
02999 BOOL WINAPI BeginPath(HDC);
03000 BOOL WINAPI BitBlt(HDC,INT,INT,INT,INT,HDC,INT,INT,DWORD);
03001 INT WINAPI ChoosePixelFormat(HDC,const LPPIXELFORMATDESCRIPTOR);
03002 BOOL WINAPI Chord(HDC,INT,INT,INT,INT,INT,INT,INT,INT);
03003 HENHMETAFILE WINAPI CloseEnhMetaFile(HDC);
03004 BOOL WINAPI CloseFigure(HDC);
03005 HMETAFILE WINAPI CloseMetaFile(HDC);
03006 INT WINAPI CombineRgn(HRGN,HRGN,HRGN,INT);
03007 BOOL WINAPI CombineTransform(LPXFORM,const XFORM *,const XFORM *);
03008 HENHMETAFILE WINAPI CopyEnhMetaFileA(HENHMETAFILE,LPCSTR);
03009 HENHMETAFILE WINAPI CopyEnhMetaFileW(HENHMETAFILE,LPCWSTR);
03010 #define CopyEnhMetaFile WINELIB_NAME_AW(CopyEnhMetaFile)
03011 HMETAFILE WINAPI CopyMetaFileA(HMETAFILE,LPCSTR);
03012 HMETAFILE WINAPI CopyMetaFileW(HMETAFILE,LPCWSTR);
03013 #define CopyMetaFile WINELIB_NAME_AW(CopyMetaFile)
03014 HBITMAP WINAPI CreateBitmap(INT,INT,UINT,UINT,LPCVOID);
03015 HBITMAP WINAPI CreateBitmapIndirect(const BITMAP*);
03016 HBRUSH WINAPI CreateBrushIndirect(const LOGBRUSH*);
03017 HCOLORSPACE WINAPI CreateColorSpaceA(LPLOGCOLORSPACEA);
03018 HCOLORSPACE WINAPI CreateColorSpaceW(LPLOGCOLORSPACEW);
03019 #define CreateColorSpace WINELIB_NAME_AW(CreateColorSpace)
03020 HBITMAP WINAPI CreateCompatibleBitmap(HDC,INT,INT);
03021 HDC WINAPI CreateCompatibleDC(HDC);
03022 HDC WINAPI CreateDCA(LPCSTR,LPCSTR,LPCSTR,const DEVMODEA*);
03023 HDC WINAPI CreateDCW(LPCWSTR,LPCWSTR,LPCWSTR,const DEVMODEW*);
03024 #define CreateDC WINELIB_NAME_AW(CreateDC)
03025 HBITMAP WINAPI CreateDIBitmap(HDC,const BITMAPINFOHEADER*,DWORD,
03026     LPCVOID,const BITMAPINFO*,UINT);
03027 HBRUSH WINAPI CreateDIBPatternBrush(HGLOBAL,UINT);
03028 HBRUSH WINAPI CreateDIBPatternBrushPt(const void*,UINT);
03029 HBITMAP WINAPI CreateDIBSection(HDC,BITMAPINFO *,UINT,
03030     LPVOID *,HANDLE,DWORD offset);
03031 HBITMAP WINAPI CreateDiscardableBitmap(HDC,INT,INT);
03032 HRGN WINAPI CreateEllipticRgn(INT,INT,INT,INT);
03033 HRGN WINAPI CreateEllipticRgnIndirect(const RECT *);
03034 HDC WINAPI CreateEnhMetaFileA(HDC,LPCSTR,const RECT*,LPCSTR);
03035 HDC WINAPI CreateEnhMetaFileW(HDC,LPCWSTR,const RECT*,LPCWSTR);
03036 #define CreateEnhMetaFile WINELIB_NAME_AW(CreateEnhMetaFile)

```

```

03037 HFONT      WINAPI CreateFontA(INT, INT, INT, INT, INT, DWORD, DWORD,
03038                                DWORD, DWORD, DWORD, DWORD, DWORD, LPCSTR);
03039 HFONT      WINAPI CreateFontW(INT, INT, INT, INT, INT, DWORD, DWORD,
03040                                DWORD, DWORD, DWORD, DWORD, DWORD, LPCWSTR);
03041 #define      CreateFont WINELIB_NAME_AW(CreateFont)
03042 HFONT      WINAPI CreateFontIndirectA(const LOGFONTA*);
03043 HFONT      WINAPI CreateFontIndirectW(const LOGFONTW*);
03044 #define      CreateFontIndirect WINELIB_NAME_AW(CreateFontIndirect)
03045 HPALETTE    WINAPI CreateHalftonePalette(HDC);
03046 HBRUSH      WINAPI CreateHatchBrush(INT, COLORREF);
03047 HDC         WINAPI CreateICA(LPCSTR, LPCSTR, LPCSTR, const DEVMODEA*);
03048 HDC         WINAPI CreateICW(LPCWSTR, LPCWSTR, LPCWSTR, const DEVMODEW*);
03049 #define      CreateIC WINELIB_NAME_AW(CreateIC)
03050 HDC         WINAPI CreateMetaFileA(LPCSTR);
03051 HDC         WINAPI CreateMetaFileW(LPCWSTR);
03052 #define      CreateMetaFile WINELIB_NAME_AW(CreateMetaFile)
03053 HPALETTE    WINAPI CreatePalette(const LOGPALETTE*);
03054 HBRUSH      WINAPI CreatePatternBrush(HBITMAP);
03055 HPEN        WINAPI CreatePen(INT, INT, COLORREF);
03056 HPEN        WINAPI CreatePenIndirect(const LOGPEN*);
03057 HRGN        WINAPI CreatePolyPolygonRgn(const POINT*, const INT*, INT, INT);
03058 HRGN        WINAPI CreatePolygonRgn(const POINT*, INT, INT);
03059 HRGN        WINAPI CreateRectRgn(INT, INT, INT, INT);
03060 HRGN        WINAPI CreateRectRgnIndirect(const RECT*);
03061 HRGN        WINAPI CreateRoundRectRgn(INT, INT, INT, INT, INT, INT);
03062 BOOL        WINAPI CreateScalableFontResourceA(DWORD, LPCSTR, LPCSTR, LPCSTR);
03063 BOOL        WINAPI CreateScalableFontResourceW(DWORD, LPCWSTR, LPCWSTR, LPCWSTR);
03064 #define      CreateScalableFontResource WINELIB_NAME_AW(CreateScalableFontResource)
03065 HBRUSH      WINAPI CreateSolidBrush(COLORREF);
03066 BOOL        WINAPI DPToLP(HDC, LPPOINT, INT);
03067 BOOL        WINAPI DeleteColorSpace(HCOLORSPACE);
03068 BOOL        WINAPI DeleteDC(HDC);
03069 BOOL        WINAPI DeleteEnhMetaFile(HENHMETAFILE);
03070 BOOL        WINAPI DeleteMetaFile(HMETAFILE);
03071 BOOL        WINAPI DeleteObject(HGDIOBJ);
03072 INT         WINAPI DescribePixelFormat(HDC, int, UINT,
03073                                         LPPIXELFORMATDESCRIPTOR);
03074 INT         WINAPI DrawEscape(HDC, INT, INT, LPCSTR);
03075 BOOL        WINAPI Ellipse(HDC, INT, INT, INT, INT);
03076 INT         WINAPI EndDoc(HDC);
03077 BOOL        WINAPI EndPath(HDC);
03078 BOOL        WINAPI EnumEnhMetaFile(HDC, HENHMETAFILE, ENHMFENUMPROC, LPVOID, const RECT*);
03079 INT         WINAPI EnumFontFamiliesA(HDC, LPCSTR, FONTENUMPROCA, LPARAM);
03080 INT         WINAPI EnumFontFamiliesW(HDC, LPCWSTR, FONTENUMPROCW, LPARAM);
03081 #define      EnumFontFamilies WINELIB_NAME_AW(EnumFontFamilies)
03082 INT         WINAPI EnumFontFamiliesExA(HDC, LPLOGFONTA, FONTENUMPROCEXA, LPARAM, DWORD);
03083 INT         WINAPI EnumFontFamiliesExW(HDC, LPLOGFONTW, FONTENUMPROCEXW, LPARAM, DWORD);
03084 #define      EnumFontFamiliesEx WINELIB_NAME_AW(EnumFontFamiliesEx)
03085 INT         WINAPI EnumFontsA(HDC, LPCSTR, FONTENUMPROCA, LPARAM);
03086 INT         WINAPI EnumFontsW(HDC, LPCWSTR, FONTENUMPROCW, LPARAM);
03087 #define      EnumFonts WINELIB_NAME_AW(EnumFonts)
03088 BOOL        WINAPI EnumMetaFile(HDC, HMETAFILE, MFENUMPROC, LPARAM);
03089 INT         WINAPI EnumObjects(HDC, INT, GOBJENUMPROC, LPARAM);
03090 BOOL        WINAPI EqualRgn(HRGN, HRGN);
03091 INT         WINAPI Escape(HDC, INT, INT, LPCSTR, LPVOID);
03092 INT         WINAPI ExcludeClipRect(HDC, INT, INT, INT, INT);
03093 HPEN        WINAPI ExtCreatePen(DWORD, DWORD, const LOGBRUSH*, DWORD, const DWORD*);
03094 HRGN        WINAPI ExtCreateRegion(const XFORM*, DWORD, const RGNDATA*);
03095 INT         WINAPI ExtEscape(HDC, INT, INT, LPCSTR, INT, LPSTR);
03096 BOOL        WINAPI ExtFloodFill(HDC, INT, INT, COLORREF, UINT);
03097 INT         WINAPI ExtSelectClipRgn(HDC, HRGN, INT);
03098 BOOL        WINAPI ExtTextOutA(HDC, INT, INT, UINT, const RECT*,
03099                                LPCSTR, UINT, const INT*);
03100 BOOL        WINAPI ExtTextOutW(HDC, INT, INT, UINT, const RECT*,
03101                                LPCWSTR, UINT, const INT*);
03102 #define      ExtTextOut WINELIB_NAME_AW(ExtTextOut)
03103 BOOL        WINAPI FillPath(HDC);
03104 BOOL        WINAPI FillRgn(HDC, HRGN, HBRUSH);
03105 BOOL        WINAPI FixBrushOrgEx(HDC, INT, INT, LPPOINT);
03106 BOOL        WINAPI FlattenPath(HDC);
03107 BOOL        WINAPI FloodFill(HDC, INT, INT, COLORREF);
03108 BOOL        WINAPI FrameRgn(HDC, HRGN, HBRUSH, INT, INT);
03109 BOOL        WINAPI GdiComment(HDC, UINT, const BYTE *);
03110 BOOL        WINAPI GdiFlush(void);
03111 INT         WINAPI GetArcDirection(HDC);
03112 BOOL        WINAPI GetAspectRatioFilterEx(HDC, LPPOINT);
03113 LONG        WINAPI GetBitmapBits(HBITMAP, LONG, LPVOID);
03114 BOOL        WINAPI GetBitmapDimensionEx(HBITMAP, LPPOINT);
03115 BOOL        WINAPI GetBrushOrgEx(HDC, LPPOINT);
03116 COLORREF    WINAPI GetBkColor(HDC);
03117 INT         WINAPI GetBkMode(HDC);
03118 UINT        WINAPI GetBoundsRect(HDC, LPRECT, UINT);
03119 BOOL        WINAPI GetCharABCWidthsA(HDC, UINT, UINT, LPABC);
03120 BOOL        WINAPI GetCharABCWidthsW(HDC, UINT, UINT, LPABC);
03121 #define      GetCharABCWidths WINELIB_NAME_AW(GetCharABCWidths)
03122 BOOL        WINAPI GetCharABCWidthsFloatA(HDC, UINT, UINT, LPABCFLOAT);
03123 BOOL        WINAPI GetCharABCWidthsFloatW(HDC, UINT, UINT, LPABCFLOAT);

```

```

03124 #define      GetCharABCWidthsFloat WINELIB_NAME_AW(GetCharABCWidthsFloat)
03125 DWORD      WINAPI GetCharacterPlacementA(HDC, LPCSTR, INT, INT, GCP_RESULTS*, DWORD);
03126 WORD      WINAPI GetCharacterPlacementW(HDC, LPCWSTR, INT, INT, GCP_RESULTS*, DWORD);
03127 #define      GetCharacterPlacement WINELIB_NAME_AW(GetCharacterPlacement)
03128 BOOL      WINAPI GetCharWidth32A(HDC, UINT, UINT, LPINT);
03129 BOOL      WINAPI GetCharWidth32W(HDC, UINT, UINT, LPINT);
03130 #define      GetCharWidthA GetCharWidth32A
03131 #define      GetCharWidthW GetCharWidth32W
03132 #define      GetCharWidth32 WINELIB_NAME_AW(GetCharWidth32)
03133 #define      GetCharWidth WINELIB_NAME_AW(GetCharWidth)
03134 BOOL      WINAPI GetCharWidthFloatA(HDC, UINT, UINT, PFLOAT);
03135 BOOL      WINAPI GetCharWidthFloatW(HDC, UINT, UINT, PFLOAT);
03136 #define      GetCharWidthFloat WINELIB_NAME_AW(GetCharWidthFloat)
03137 INT      WINAPI GetClipBox(HDC, LPRECT);
03138 INT      WINAPI GetClipRgn(HDC, HRGN);
03139 BOOL      WINAPI GetColorAdjustment(HDC, LPCOLORADJUSTMENT);
03140 HANDLE     WINAPI GetCurrentObject(HDC, UINT);
03141 BOOL      WINAPI GetCurrentPositionEx(HDC, LPPOINT);
03142 INT      WINAPI GetDeviceCaps(HDC, INT);
03143 BOOL      WINAPI GetDeviceGammaRamp(HDC, LPVOID);
03144 COLORREF  WINAPI GetDCBrushColor(HDC);
03145 BOOL      WINAPI GetDCOrgEx(HDC, LPPOINT);
03146 COLORREF  WINAPI GetDCPenColor(HDC);
03147 UINT      WINAPI GetDIBColorTable(HDC, UINT, UINT, RGBQUAD*);
03148 INT      WINAPI GetDIBits(HDC, HBITMAP, UINT, UINT, LPVOID, LPBITMAPINFO, UINT);
03149 HENHMETAFILE WINAPI GetEnhMetaFileA(LPCSTR);
03150 HENHMETAFILE WINAPI GetEnhMetaFileW(LPCWSTR);
03151 #define      GetEnhMetaFile WINELIB_NAME_AW(GetEnhMetaFile)
03152 UINT      WINAPI GetEnhMetaFileBits(HENHMETAFILE, UINT, LPBYTE);
03153 UINT      WINAPI GetEnhMetaFileDescriptionA(HENHMETAFILE, UINT, LPSTR);
03154 UINT      WINAPI GetEnhMetaFileDescriptionW(HENHMETAFILE, UINT, LPWSTR);
03155 #define      GetEnhMetaFileDescription WINELIB_NAME_AW(GetEnhMetaFileDescription)
03156 UINT      WINAPI GetEnhMetaFileHeader(HENHMETAFILE, UINT, LPENHMETAHEADER);
03157 UINT      WINAPI GetEnhMetaFilePaletteEntries(HENHMETAFILE, UINT, LPPALETTEENTRY);
03158 DWORD     WINAPI GetFontData(HDC, DWORD, DWORD, LPVOID, DWORD);
03159 DWORD     WINAPI GetFontLanguageInfo(HDC);
03160 DWORD     WINAPI GetGlyphOutlineA(HDC, UINT, UINT, LPGLYPHMETRICS, DWORD, LPVOID, const MAT2*);
03161 DWORD     WINAPI GetGlyphOutlineW(HDC, UINT, UINT, LPGLYPHMETRICS, DWORD, LPVOID, const MAT2*);
03162 #define      GetGlyphOutline WINELIB_NAME_AW(GetGlyphOutline)
03163 INT      WINAPI GetGraphicsMode(HDC);
03164 DWORD     WINAPI GetKerningPairsA(HDC, DWORD, LPKERNINGPAIR);
03165 DWORD     WINAPI GetKerningPairsW(HDC, DWORD, LPKERNINGPAIR);
03166 #define      GetKerningPairs WINELIB_NAME_AW(GetKerningPairs)
03167 DWORD     WINAPI GetLayout(HDC);
03168 BOOL      WINAPI GetLogColorSpaceA(HCOLORSPACE, LPLOGCOLORSPACEA, DWORD);
03169 BOOL      WINAPI GetLogColorSpaceW(HCOLORSPACE, LPLOGCOLORSPACEW, DWORD);
03170 #define      GetLogColorSpace WINELIB_NAME_AW(GetLogColorSpace)
03171 INT      WINAPI GetMapMode(HDC);
03172 HMETAFILE WINAPI GetMetaFileA(LPCSTR);
03173 HMETAFILE WINAPI GetMetaFileW(LPCWSTR);
03174 #define      GetMetaFile WINELIB_NAME_AW(GetMetaFile)
03175 UINT      WINAPI GetMetaFileBitsEx(HMETAFILE, UINT, LPVOID);
03176 INT      WINAPI GetMetaRgn(HDC, HRGN);
03177 BOOL      WINAPI GetMiterLimit(HDC, PFLOAT);
03178 DWORD     WINAPI GetNearestColor(HDC, DWORD);
03179 UINT      WINAPI GetNearestPaletteIndex(HPALETTE, COLORREF);
03180 INT      WINAPI GetObjectA(HANDLE, INT, LPVOID);
03181 INT      WINAPI GetObjectW(HANDLE, INT, LPVOID);
03182 #define      GetObject WINELIB_NAME_AW(GetObject)
03183 DWORD     WINAPI GetObjectType(HANDLE);
03184 UINT      WINAPI GetOutlineTextMetricsA(HDC, UINT, LPOUTLINETEXTMETRICA);
03185 UINT      WINAPI GetOutlineTextMetricsW(HDC, UINT, LPOUTLINETEXTMETRICW);
03186 #define      GetOutlineTextMetrics WINELIB_NAME_AW(GetOutlineTextMetrics)
03187 UINT      WINAPI GetPaletteEntries(HPALETTE, UINT, LPPALETTEENTRY);
03188 INT      WINAPI GetPath(HDC, LPPOINT, LPBYTE, INT);
03189 COLORREF  WINAPI GetPixel(HDC, INT, INT);
03190 INT      WINAPI GetPixelFormat(HDC);
03191 INT      WINAPI GetPolyFillMode(HDC);
03192 BOOL      WINAPI GetRasterizerCaps(LPRASTERIZER_STATUS, UINT);
03193 DWORD     WINAPI GetRegionData(HRGN, DWORD, LPRGNDDATA);
03194 INT      WINAPI GetRelAbs(HDC, DWORD);
03195 INT      WINAPI GetRgnBox(HRGN, LPRECT);
03196 INT      WINAPI GetROP2(HDC);
03197 HGDIOBJ  WINAPI GetStockObject(INT);
03198 INT      WINAPI GetStretchBltMode(HDC);
03199 UINT      WINAPI GetSystemPaletteEntries(HDC, UINT, LPPALETTEENTRY);
03200 UINT      WINAPI GetSystemPaletteUse(HDC);
03201 UINT      WINAPI GetTextAlign(HDC);
03202 INT      WINAPI GetTextCharacterExtra(HDC);
03203 UINT      WINAPI GetTextCharset(HDC);
03204 UINT      WINAPI GetTextCharsetInfo(HDC, LPFONTSIGNATURE, DWORD);
03205 COLORREF  WINAPI GetTextColor(HDC);
03206 BOOL      WINAPI GetTextExtentExPointA(HDC, LPCSTR, INT, INT, LPINT, LPINT, LPVOID);
03207                                     LPINT, LPINT, LPVOID);
03208 BOOL      WINAPI GetTextExtentExPointW(HDC, LPCWSTR, INT, INT, LPINT, LPINT, LPVOID);
03209                                     LPINT, LPINT, LPVOID);
03210 BOOL      WINAPI GetTextExtentPointA(HDC, LPCSTR, INT, LPVOID);

```

```

03211 BOOL        WINAPI GetTextExtentPointW(HDC, LPCWSTR, INT, LPCTSTR);
03212 #define        GetTextExtentPoint WINELIB_NAME_AW(GetTextExtentPoint)
03213 BOOL        WINAPI GetTextExtentPoint32A(HDC, LPCSTR, INT, LPCTSTR);
03214 BOOL        WINAPI GetTextExtentPoint32W(HDC, LPCWSTR, INT, LPCTSTR);
03215 #define        GetTextExtentPoint32 WINELIB_NAME_AW(GetTextExtentPoint32)
03216 #define        GetTextExtentExPoint WINELIB_NAME_AW(GetTextExtentExPoint)
03217 INT        WINAPI GetTextFaceA(HDC, INT, LPSTR);
03218 INT        WINAPI GetTextFaceW(HDC, INT, LPWSTR);
03219 #define        GetTextFace WINELIB_NAME_AW(GetTextFace)
03220 BOOL        WINAPI GetTextMetricsA(HDC, LPTEXTMETRICA);
03221 BOOL        WINAPI GetTextMetricsW(HDC, LPTEXTMETRICW);
03222 #define        GetTextMetrics WINELIB_NAME_AW(GetTextMetrics)
03223 BOOL        WINAPI GetViewportExtEx(HDC, LPCTSTR);
03224 BOOL        WINAPI GetViewportOrgEx(HDC, LPPOINT);
03225 BOOL        WINAPI GetWindowExtEx(HDC, LPCTSTR);
03226 BOOL        WINAPI GetWindowOrgEx(HDC, LPPOINT);
03227 BOOL        WINAPI GetWorldTransform(HDC, LPXFORM);
03228 INT        WINAPI IntersectClipRect(HDC, INT, INT, INT, INT);
03229 BOOL        WINAPI InvertRgn(HDC, HRGN);
03230 BOOL        WINAPI LineDDA(INT, INT, INT, INT, LINEDDAPROC, LPARAM);
03231 BOOL        WINAPI LineTo(HDC, INT, INT);
03232 BOOL        WINAPI LPToDP(HDC, LPPOINT, INT);
03233 BOOL        WINAPI MaskBlt(HDC, INT, INT, INT, INT, HDC, INT, INT, HBITMAP, INT, INT, DWORD);
03234 BOOL        WINAPI ModifyWorldTransform(HDC, const XFORM *, DWORD);
03235 BOOL        WINAPI MoveToEx(HDC, INT, INT, LPPOINT);
03236 /* FIXME This is defined in kernel32.spec !*/
03237 INT        WINAPI MulDiv(INT, INT, INT);
03238 INT        WINAPI OffsetClipRgn(HDC, INT, INT);
03239 INT        WINAPI OffsetRgn(HRGN, INT, INT);
03240 BOOL        WINAPI OffsetViewportOrgEx(HDC, INT, INT, LPPOINT);
03241 BOOL        WINAPI OffsetWindowOrgEx(HDC, INT, INT, LPPOINT);
03242 BOOL        WINAPI PaintRgn(HDC, HRGN);
03243 BOOL        WINAPI PatBlt(HDC, INT, INT, INT, DWORD);
03244 HRGN        WINAPI PathToRegion(HDC);
03245 BOOL        WINAPI Pie(HDC, INT, INT, INT, INT, INT, INT, INT);
03246 BOOL        WINAPI PlayEnhMetaFile(HDC, HENHMETAFILE, const RECT*);
03247 BOOL        WINAPI PlayEnhMetaFileRecord(HDC, LPHANDLETABLE, const ENHMETARECORD*, UINT);
03248 BOOL        WINAPI PlayMetaFile(HDC, HMETAFILE);
03249 BOOL        WINAPI PlayMetaFileRecord(HDC, LPHANDLETABLE, LPMETARECORD, UINT);
03250 BOOL        WINAPI PlgBlt(HDC, const POINT*, HDC, INT, INT, INT, INT, HBITMAP, INT, INT);
03251 BOOL        WINAPI PolyBezier(HDC, const POINT*, DWORD);
03252 BOOL        WINAPI PolyBezierTo(HDC, const POINT*, DWORD);
03253 BOOL        WINAPI PolyDraw(HDC, const POINT*, const BYTE*, DWORD);
03254 BOOL        WINAPI PolyPolygon(HDC, const POINT*, const INT*, UINT);
03255 BOOL        WINAPI PolyPolyline(HDC, const POINT*, const DWORD*, DWORD);
03256 BOOL        WINAPI Polygon(HDC, const POINT*, INT);
03257 BOOL        WINAPI Polyline(HDC, const POINT*, INT);
03258 BOOL        WINAPI PolylineTo(HDC, const POINT*, DWORD);
03259 BOOL        WINAPI PtInRegion(HRGN, INT, INT);
03260 BOOL        WINAPI PtVisible(HDC, INT, INT);
03261 UINT        WINAPI RealizePalette(HDC);
03262 BOOL        WINAPI Rectangle(HDC, INT, INT, INT, INT);
03263 BOOL        WINAPI RectInRegion(HRGN, const RECT*);
03264 BOOL        WINAPI RectVisible(HDC, const RECT*);
03265 BOOL        WINAPI RemoveFontResourceA(LPCSTR);
03266 BOOL        WINAPI RemoveFontResourceW(LPCWSTR);
03267 #define        RemoveFontResource WINELIB_NAME_AW(RemoveFontResource)
03268 HDC        WINAPI ResetDCA(HDC, const DEVMODEA*);
03269 HDC        WINAPI ResetDCW(HDC, const DEVMODEW*);
03270 #define        ResetDC WINELIB_NAME_AW(ResetDC)
03271 BOOL        WINAPI ResizePalette(HPALETTE, UINT);
03272 BOOL        WINAPI RestoreDC(HDC, INT);
03273 BOOL        WINAPI RoundRect(HDC, INT, INT, INT, INT, INT, INT);
03274 INT        WINAPI SaveDC(HDC);
03275 BOOL        WINAPI ScaleViewportExtEx(HDC, INT, INT, INT, INT, LPCTSTR);
03276 BOOL        WINAPI ScaleWindowExtEx(HDC, INT, INT, INT, INT, LPCTSTR);
03277 BOOL        WINAPI SelectClipPath(HDC, INT);
03278 INT        WINAPI SelectClipRgn(HDC, HRGN);
03279 HGDIOBJ        WINAPI SelectObject(HDC, HGDIOBJ);
03280 HPALETTE        WINAPI SelectPalette(HDC, HPALETTE, BOOL);
03281 INT        WINAPI SetAbortProc(HDC, ABORTPROC);
03282 INT        WINAPI SetArcDirection(HDC, INT);
03283 LONG        WINAPI SetBitmapBits(HBITMAP, LONG, LPCVOID);
03284 BOOL        WINAPI SetBitmapDimensionEx(HBITMAP, INT, INT, LPCTSTR);
03285 COLORREF        WINAPI SetBkColor(HDC, COLORREF);
03286 INT        WINAPI SetBkMode(HDC, INT);
03287 UINT        WINAPI SetBoundsRect(HDC, const RECT*, UINT);
03288 BOOL        WINAPI SetBrushOrgEx(HDC, INT, INT, LPPOINT);
03289 BOOL        WINAPI SetColorAdjustment(HDC, const COLORADJUSTMENT*);
03290 HCOLORSPACE        WINAPI SetColorSpace(HDC, HCOLORSPACE);
03291 BOOL        WINAPI SetDeviceGammaRamp(HDC, LPVOID);
03292 UINT        WINAPI SetDIBColorTable(HDC, UINT, UINT, RGBQUAD*);
03293 INT        WINAPI SetDIBits(HDC, HBITMAP, UINT, UINT, LPCVOID, const BITMAPINFO*, UINT);
03294 INT        WINAPI SetDIBitsToDevice(HDC, INT, INT, DWORD, DWORD, INT,
03295                                     INT, UINT, UINT, LPCVOID, const BITMAPINFO*, UINT);
03296 HENHMETAFILE        WINAPI SetEnhMetaFileBits(UINT, const BYTE*);
03297 INT        WINAPI SetGraphicsMode(HDC, INT);

```

```

03298 INT          WINAPI SetICMMode (HDC, INT);
03299 DWORD         WINAPI SetLayout (HDC, DWORD);
03300 INT          WINAPI SetMapMode (HDC, INT);
03301 DWORD         WINAPI SetMapperFlags (HDC, DWORD);
03302 HMETAFILE      WINAPI SetMetaFileBitsEx (UINT, const BYTE*);
03303 INT          WINAPI SetMetaRgn (HDC);
03304 BOOL          WINAPI SetMiterLimit (HDC, FLOAT, PFLOAT);
03305 UINT          WINAPI SetPaletteEntries (HPALETTE, UINT, UINT, LPPALETTEENTRY);
03306 COLORREF      WINAPI SetPixel (HDC, INT, INT, COLORREF);
03307 BOOL          WINAPI SetPixelV (HDC, INT, INT, COLORREF);
03308 BOOL          WINAPI SetPixelFormat (HDC, int, const PIXELFORMATDESCRIPTOR*);
03309 INT          WINAPI SetPolyFillMode (HDC, INT);
03310 BOOL          WINAPI SetRectRgn (HRGN, INT, INT, INT, INT);
03311 INT          WINAPI SetRelAbs (HDC, INT);
03312 INT          WINAPI SetROP2 (HDC, INT);
03313 INT          WINAPI SetStretchBltMode (HDC, INT);
03314 UINT          WINAPI SetSystemPaletteUse (HDC, UINT);
03315 UINT          WINAPI SetTextAlign (HDC, UINT);
03316 INT          WINAPI SetTextCharacterExtra (HDC, INT);
03317 COLORREF      WINAPI SetTextColor (HDC, COLORREF);
03318 BOOL          WINAPI SetTextJustification (HDC, INT, INT);
03319 BOOL          WINAPI SetViewportExtEx (HDC, INT, INT, LPPOINT);
03320 BOOL          WINAPI SetViewportOrgEx (HDC, INT, INT, LPPOINT);
03321 BOOL          WINAPI SetWindowExtEx (HDC, INT, INT, LPPOINT);
03322 BOOL          WINAPI SetWindowOrgEx (HDC, INT, INT, LPPOINT);
03323 HENHMETAFILE  WINAPI SetWinMetaFileBits (UINT, CONST BYTE*, HDC, CONST METAFILEPCT *);
03324 BOOL          WINAPI SetWorldTransform (HDC, const XFORM*);
03325 INT          WINAPI StartDocA (HDC, const DOCINFOA*);
03326 INT          WINAPI StartDocW (HDC, const DOCINFOW*);
03327 #define        StartDoc WINELIB_NAME_AW(StartDoc)
03328 INT          WINAPI StartPage (HDC);
03329 INT          WINAPI EndPage (HDC);
03330 BOOL          WINAPI StretchBlt (HDC, INT, INT, INT, INT, HDC, INT,
03331                                INT, INT, INT, DWORD);
03332 INT          WINAPI StretchDIBits (HDC, INT, INT, INT, INT, INT, INT,
03333                                INT, INT, const VOID*, const BITMAPINFO*, UINT, DWORD);
03334 BOOL          WINAPI StrokeAndFillPath (HDC);
03335 BOOL          WINAPI StrokePath (HDC);
03336 BOOL          WINAPI SwapBuffers (HDC);
03337 BOOL          WINAPI TextOutA (HDC, INT, INT, LPCSTR, INT);
03338 BOOL          WINAPI TextOutW (HDC, INT, INT, LPCWSTR, INT);
03339 #define        TextOut WINELIB_NAME_AW(TextOut)
03340 BOOL          WINAPI TranslateCharsetInfo (LPDWORD, LPCHARSETINFO, DWORD);
03341 BOOL          WINAPI UnrealizeObject (HGDIOBJ);
03342 BOOL          WINAPI UpdateColors (HDC);
03343 BOOL          WINAPI WidenPath (HDC);
03344 BOOL          WINAPI PolyTextOutA (HDC, PPOLYTEXTA, INT);
03345 BOOL          WINAPI PolyTextOutW (HDC, PPOLYTEXTW, INT);
03346 #define        PolyTextOut WINELIB_NAME_AW(PolyTextOut)
03347
03348 #ifdef __cplusplus
03349 }
03350 #endif
03351
03352 #endif /* !NOGDI */
03353 #endif /* _WINGDI */

```

## 5.12 winnt.h

```

00001 /*
00002  * Win32 definitions for Windows NT
00003  *
00004  * Copyright 1996 Alexandre Julliard
00005  */
00006
00007 #ifndef __WINE_WINNT_H
00008 #define __WINE_WINNT_H
00009
00010 #include "basetsd.h"
00011
00012 #ifndef RC_INVOKED
00013 #include <ctype.h>
00014 #include <stddef.h>
00015 #include <string.h>
00016 #endif
00017
00018
00019 /* On Windows winnt.h depends on a few windef.h types and macros and thus
00020  * is not self-contained. Furthermore windef.h includes winnt.h so that it
00021  * would be pointless to try to use winnt.h directly.
00022  * But for Wine and Winalib I decided to make winnt.h self-contained by
00023  * moving these definitions to winnt.h. It makes no difference to Winalib
00024  * programs since they are not using winnt.h directly anyway, and it allows
00025  * us to use winnt.h and get a minimal set of definitions.

```

```
00026 */
00027
00028 /**** Some Wine specific definitions *****/
00029
00030 /* Architecture dependent settings. */
00031 /* These are hardcoded to avoid dependencies on config.h in Winelib apps. */
00032 #if defined(__i386__)
00033 # undef WORDS_BIGENDIAN
00034 # undef BITFIELDS_BIGENDIAN
00035 # define ALLOW_UNALIGNED_ACCESS
00036 #elif defined(__x86_64__)
00037 # undef WORDS_BIGENDIAN
00038 # undef BITFIELDS_BIGENDIAN
00039 # define ALLOW_UNALIGNED_ACCESS
00040 #elif defined(__alpha__)
00041 # undef WORDS_BIGENDIAN
00042 # undef BITFIELDS_BIGENDIAN
00043 # undef ALLOW_UNALIGNED_ACCESS
00044 #elif defined(__arm__)
00045 # undef WORDS_BIGENDIAN
00046 # undef BITFIELDS_BIGENDIAN
00047 # undef ALLOW_UNALIGNED_ACCESS
00048 #elif defined(__aarch64__)
00049 # undef WORDS_BIGENDIAN
00050 # undef BITFIELDS_BIGENDIAN
00051 # undef ALLOW_UNALIGNED_ACCESS
00052 #elif defined(__loongarch64)
00053 # undef WORDS_BIGENDIAN
00054 # undef BITFIELDS_BIGENDIAN
00055 # undef ALLOW_UNALIGNED_ACCESS
00056 #elif defined(__sparc__)
00057 # define WORDS_BIGENDIAN
00058 # define BITFIELDS_BIGENDIAN
00059 # undef ALLOW_UNALIGNED_ACCESS
00060 #elif defined(__PPC__)
00061 # define WORDS_BIGENDIAN
00062 # define BITFIELDS_BIGENDIAN
00063 # undef ALLOW_UNALIGNED_ACCESS
00064 #elif defined(__s390__)
00065 # define WORDS_BIGENDIAN
00066 # define BITFIELDS_BIGENDIAN
00067 # undef ALLOW_UNALIGNED_ACCESS
00068 #elif defined(__e2k__)
00069 # undef WORDS_BIGENDIAN
00070 # undef BITFIELDS_BIGENDIAN
00071 # undef ALLOW_UNALIGNED_ACCESS
00072 #elif defined(__MIPSEB__)
00073 # define WORDS_BIGENDIAN
00074 # define BITFIELDS_BIGENDIAN
00075 # undef ALLOW_UNALIGNED_ACCESS
00076 #elif defined(__riscv) && defined(__riscv_xlen) && __riscv_xlen == 64
00077 # undef WORDS_BIGENDIAN
00078 # undef BITFIELDS_BIGENDIAN
00079 # undef ALLOW_UNALIGNED_ACCESS
00080 #elif !defined(RC_INVOKED)
00081 # error Unknown CPU architecture!
00082 #endif
00083
00084
00085 #ifndef DECLSPEC_ALIGN
00086 # if defined(_MSC_VER) && (_MSC_VER >= 1300) && !defined(MIDL_PASS)
00087 #  define DECLSPEC_ALIGN(x) __declspec(align(x))
00088 # elif defined(__GNUC__)
00089 #  define DECLSPEC_ALIGN(x) __attribute__((aligned(x)))
00090 # else
00091 #  define DECLSPEC_ALIGN(x)
00092 # endif
00093 #endif
00094
00095
00096 /* Calling conventions definitions */
00097
00098 #ifdef __i386__
00099 # ifndef _X86_
00100 #  define _X86_
00101 # endif
00102 # if defined(__GNUC__) && ((__GNUC__ > 2) || ((__GNUC__ == 2) && (__GNUC_MINOR__ >= 7)))
00103 #  define __stdcall __attribute__((__stdcall__))
00104 #  define __cdecl __attribute__((__cdecl__))
00105 # else
00106 #  error You need gcc >= 2.7 to build Wine on a 386
00107 # endif /* __GNUC__ */
00108 #else /* __i386__ */
00109 # define __stdcall
00110 # define __cdecl
00111 #endif /* __i386__ */
00112
```

```

00113 #ifndef __WINE__
00114 #define pascal      __stdcall
00115 #define _pascal     __stdcall
00116 #ifndef __stdcall
00117 #define __stdcall   __stdcall
00118 #endif
00119 #ifndef __fastcall
00120 #define __fastcall   __stdcall
00121 #endif
00122 #ifndef __fastcall
00123 #define __fastcall   __stdcall
00124 #endif
00125 #define __export     __stdcall
00126 #define cdecl        __cdecl
00127 #ifndef _cdecl
00128 #define _cdecl       __cdecl
00129 #endif
00130
00131 #define near
00132 #define far
00133 #define _near
00134 #define _far
00135 #define NEAR
00136 #define FAR
00137
00138 #ifndef _declspec
00139 #define _declspec(x)
00140 #endif
00141 #ifndef __declspec
00142 #define __declspec(x)
00143 #endif
00144 #endif /* __WINE__ */
00145
00146 #define CALLBACK     __stdcall
00147 #if 0
00148 #define WINAPI       __stdcall
00149 #else
00150 #define WINAPI       __attribute__((visibility("default")))
00151 #endif
00152 #define APIPRIVATE   __stdcall
00153 #define PASCAL        __stdcall
00154 #define CDECL         __cdecl
00155 #define _CDECL        __cdecl
00156 #define WINAPIV       __cdecl
00157 #define APIENTRY      WINAPI
00158 #define CONST         const
00159
00160 /* Macro for structure packing and more. */
00161
00162 #ifdef __GNUC__
00163 #define WINE_PACKED    __attribute__((packed))
00164 #define WINE_UNUSED    __attribute__((unused))
00165 #define WINE_NORETURN  __attribute__((noreturn))
00166 #else
00167 #define WINE_PACKED    /* nothing */
00168 #define WINE_UNUSED    /* nothing */
00169 #define WINE_NORETURN  /* nothing */
00170 #endif
00171
00172 /* Anonymous union/struct handling */
00173
00174 #ifdef __WINE__
00175 # define NONAMELESSSTRUCT
00176 # define NONAMELESSUNION
00177 #else
00178 #if !defined(__cplusplus)
00179 /* for c we can keep the anonymous version (to avoid compiler warnings) */
00180 #define NONAMELESSSTRUCT
00181 #define NONAMELESSUNION
00182 #else
00183 /* Anonymous struct support starts with gcc/g++ 2.96 */
00184 # if !defined(NONAMELESSSTRUCT) && defined(__GNUC__) && ((__GNUC__ < 2) || ((__GNUC__ == 2) &&
    (__GNUC_MINOR__ < 96)))
00185 /* && !defined(__cplusplus) */
00186 #  define NONAMELESSSTRUCT
00187 #  endif
00188 /* Anonymous unions support starts with gcc 2.96/g++ 2.95 */
00189 # if !defined(NONAMELESSUNION) && defined(__GNUC__) && ((__GNUC__ < 2) || ((__GNUC__ == 2) &&
    (__GNUC_MINOR__ < 95) || ((__GNUC_MINOR__ == 95) && !defined(__cplusplus))))
00190 #  define NONAMELESSUNION
00191 #  endif
00192 #endif
00193 #endif
00194
00195 #ifndef NONAMELESSSTRUCT
00196 #define DUMMYSTRUCTNAME
00197 #define DUMMYSTRUCTNAME1

```

```

00198 #define DUMMYSTRUCTNAME2
00199 #define DUMMYSTRUCTNAME3
00200 #define DUMMYSTRUCTNAME4
00201 #define DUMMYSTRUCTNAME5
00202 #else /* !defined(NONAMELESSSTRUCT) */
00203 #define DUMMYSTRUCTNAME s
00204 #define DUMMYSTRUCTNAME1 s1
00205 #define DUMMYSTRUCTNAME2 s2
00206 #define DUMMYSTRUCTNAME3 s3
00207 #define DUMMYSTRUCTNAME4 s4
00208 #define DUMMYSTRUCTNAME5 s5
00209 #endif /* !defined(NONAMELESSSTRUCT) */
00210
00211 #ifndef NONAMELESSUNION
00212 #define DUMMYUNIONNAME
00213 #define DUMMYUNIONNAME1
00214 #define DUMMYUNIONNAME2
00215 #define DUMMYUNIONNAME3
00216 #define DUMMYUNIONNAME4
00217 #define DUMMYUNIONNAME5
00218 #define DUMMYUNIONNAME6
00219 #define DUMMYUNIONNAME7
00220 #define DUMMYUNIONNAME8
00221 #else /* !defined(NONAMELESSUNION) */
00222 #define DUMMYUNIONNAME u
00223 #define DUMMYUNIONNAME1 u1
00224 #define DUMMYUNIONNAME2 u2
00225 #define DUMMYUNIONNAME3 u3
00226 #define DUMMYUNIONNAME4 u4
00227 #define DUMMYUNIONNAME5 u5
00228 #define DUMMYUNIONNAME6 u6
00229 #define DUMMYUNIONNAME7 u7
00230 #define DUMMYUNIONNAME8 u8
00231 #endif /* !defined(NONAMELESSUNION) */
00232
00233
00234 /**** Parts of windef.h that are needed here *****/
00235
00236 /* Misc. constants. */
00237
00238 #undef NULL
00239 #ifdef __cplusplus
00240 #define NULL 0
00241 #else
00242 #define NULL ((void*)0)
00243 #endif
00244
00245 #ifdef FALSE
00246 #undef FALSE
00247 #endif
00248 #define FALSE 0
00249
00250 #ifdef TRUE
00251 #undef TRUE
00252 #endif
00253 #define TRUE 1
00254
00255 #ifdef IN
00256 #define IN
00257 #endif
00258
00259 #ifndef OUT
00260 #define OUT
00261 #endif
00262
00263 #ifndef OPTIONAL
00264 #define OPTIONAL
00265 #endif
00266
00267 /* Standard data types */
00268 typedef const void *PCVOID, *LPCVOID;
00269 typedef int BOOL, *PBOOL, *LPBOOL;
00270 typedef unsigned char BYTE, *PBYTE, *LPBYTE;
00271 typedef unsigned char UCHAR, *PUCHAR;
00272 typedef unsigned short USHORT, *PUSHORT, *LPUSHORT;
00273 typedef unsigned short WORD, *PWORD, *LPWORD;
00274 typedef int INT, *PINT, *LPINT;
00275 typedef unsigned int UINT, *PUINT, *LPUINT;
00276 /* Not sure this is correct. Probably should depend on the compiler, too. */
00277 #if defined(__LP64__) || defined(__alpha__)
00278 typedef unsigned int DWORD, *PDWORD, *LPDWORD;
00279 typedef unsigned int ULONG, *PULONG, *LPULONG;
00280 #else
00281 typedef unsigned long DWORD, *PDWORD, *LPDWORD;
00282 typedef unsigned long ULONG, *PULONG, *LPULONG;
00283 #endif
00284 typedef float FLOAT, *PFLOAT, *LPFLOAT;

```

```

00285 typedef double          DOUBLE,      *PDOUBLE,  *LPDOUBLE;
00286 typedef double          DATE;
00287
00288
00289 /**** winnt.h proper *****/
00290
00291 /* Microsoft's macros for declaring functions */
00292
00293 #ifdef __cplusplus
00294 # define EXTERN_C      extern "C"
00295 #else
00296 # define EXTERN_C      extern
00297 #endif
00298
00299 #ifndef __WINE__
00300 #define STDMETHODCALLTYPE      __stdcall
00301 #define STDMETHODCALLTYPE      __cdecl
00302 #define STDAPICALLTYPE        __stdcall
00303 #define STDAPICALLTYPE        __cdecl
00304
00305 #define STDAPI                EXTERN_C HRESULT STDMETHODCALLTYPE
00306 #define STDAPI_(type)        EXTERN_C type STDMETHODCALLTYPE
00307 #define STDMETHODCALLTYPE    HRESULT STDMETHODCALLTYPE
00308 #define STDMETHODCALLTYPE    type STDMETHODCALLTYPE
00309 #define STDAPIV              EXTERN_C HRESULT STDMETHODCALLTYPE
00310 #define STDAPIV_(type)      EXTERN_C type STDMETHODCALLTYPE
00311 #define STDMETHODCALLTYPE    HRESULT STDMETHODCALLTYPE
00312 #define STDMETHODCALLTYPE    type STDMETHODCALLTYPE
00313 #endif
00314
00315 /* Define the basic types */
00316 #ifndef VOID
00317 #define VOID void
00318 #endif
00319 typedef VOID          *PVOID,      *LPVOID;
00320 typedef BYTE          BOOLEAN,     *PBOOLEAN;
00321 typedef char          CHAR,        *PCHAR;
00322 typedef short         SHORT,       *PSHORT;
00323 #if defined(__LP64__) || defined(__alpha__)
00324 typedef int           LONG,        *PLONG,   *LPLONG;
00325 #else
00326 typedef long          LONG,        *PLONG,   *LPLONG;
00327 #endif
00328
00329 /* Some systems might have wchar_t, but we really need 16 bit characters */
00330 #ifndef WINE_WCHAR_DEFINED
00331 #ifdef WINE_UNICODE_NATIVE
00332 typedef wchar_t       WCHAR,      *PWCHAR;
00333 #else
00334 typedef unsigned short WCHAR,     *PWCHAR;
00335 #endif
00336 #define WINE_WCHAR_DEFINED
00337 #endif
00338
00339 /* 'Extended/Wide' numerical types */
00340 #ifndef _ULONGLONG_
00341 #define _ULONGLONG_
00342 typedef __int64        LONGLONG,   *PLONGLONG;
00343 typedef __uint64       ULONGLONG,  *PULONGLONG;
00344 #endif
00345
00346 #ifndef _DWORDLONG_
00347 #define _DWORDLONG_
00348 typedef ULONGLONG      DWORDLONG,  *PDWORDLONG;
00349 #endif
00350
00351 /* ANSI string types */
00352 typedef CHAR          *PCH,        *LPCH;
00353 typedef const CHAR    *PCCH,       *LPCCH;
00354 typedef CHAR          *PSTR,       *LPSTR;
00355 typedef const CHAR    *PCSTR,      *LPCSTR;
00356
00357 /* Unicode string types */
00358 typedef WCHAR         *PWCH,       *LPWCH;
00359 typedef const WCHAR   *PCWCH,      *LPCWCH;
00360 typedef WCHAR         *PWSTR,      *LPWSTR;
00361 typedef const WCHAR   *PCWSTR,     *LPCWSTR;
00362
00363 /* Neutral character and string types */
00364 /* These are only defined for Winelib, i.e. _not_ defined for
00365  * the emulator. The reason is they depend on the UNICODE
00366  * macro which only exists in the user's code.
00367  */
00368 #ifndef __WINE__
00369 # ifdef WINE_UNICODE_REWRITE
00370
00371 /* Use this if your compiler does not provide a 16bit wchar_t type.

```

```

00372 * Note that you will need to specify -fwritable-strings or an option
00373 * to this effect.
00374 * In C++ both WINE_UNICODE_TEXT('c') and WINE_UNICODE_TEXT("str") are
00375 * supported, but only the string form can be supported in C.
00376 */
00377 EXTERN_C unsigned short* wine_rewrite_s4tos2(const wchar_t* str4);
00378 # ifdef __cplusplus
00379 inline WCHAR* wine_unicode_text(const wchar_t* str4)
00380 {
00381     return (WCHAR*)wine_rewrite_s4tos2(str4);
00382 }
00383 inline WCHAR wine_unicode_text(wchar_t chr4)
00384 {
00385     return (WCHAR)chr4;
00386 }
00387 #   define WINE_UNICODE_TEXT(x)        wine_unicode_text(L##x)
00388 #   else /* __cplusplus */
00389 #   define WINE_UNICODE_TEXT(x)        ((WCHAR*)wine_rewrite_s4tos2(L##x))
00390 #   endif /* __cplusplus */
00391
00392 # else /* WINE_UNICODE_REWRITE */
00393
00394 /* Define WINE_UNICODE_NATIVE if:
00395  * - your compiler provides a 16bit wchar_t type, e.g. gcc >= 2.96 with
00396  *   -fshort-wchar option
00397  * - or if you decide to use the native 32bit Unix wchar_t type. Be aware
00398  *   though that the Wine APIs only support 16bit WCHAR characters for
00399  *   binary compatibility reasons.
00400  * - or define nothing at all if you don't use Unicode, and blissfully
00401  *   ignore the issue :-)
00402  */
00403 #   define WINE_UNICODE_TEXT(string)   L##string
00404
00405 #   endif /* WINE_UNICODE_REWRITE */
00406
00407 #   ifdef UNICODE
00408     typedef WCHAR          TCHAR,          *PTCHAR;
00409     typedef LPWSTR         PTSTR,          *LPSTR;
00410     typedef LPCWSTR        PCTSTR,        *LPCTSTR;
00411 #   define __TEXT(string)   WINE_UNICODE_TEXT(string)
00412 #   else /* UNICODE */
00413     typedef CHAR           TCHAR,          *PTCHAR;
00414     typedef LPSTR          PTSTR,          *LPSTR;
00415     typedef LPCSTR         PCTSTR,        *LPCTSTR;
00416 #   define __TEXT(string)   string
00417 #   endif /* UNICODE */
00418 #   define TEXT(quote)     __TEXT(quote)
00419 #endif /* __WINE__ */
00420
00421 /* Misc common WIN32 types */
00422 typedef LONG              HRESULT;
00423 typedef DWORD             LCID,          *PLCID;
00424 typedef WORD              LANGID;
00425 typedef DWORD             EXECUTION_STATE;
00426
00427 /* Handle type */
00428
00429 /* FIXME: Wine does not compile with strict on, therefore strict
00430  * handles are presently only usable on machines where sizeof(UINT) ==
00431  * sizeof(void*). HANDLES are supposed to be void* but a large amount
00432  * of WINE code operates on HANDLES as if they are UINTs. So to WINE
00433  * they exist as UINTs but to the Winelib user who turns on strict,
00434  * they exist as void*. If there is a size difference between UINT and
00435  * void* then things get ugly.
00436  *
00437  * Here is the plan to convert Wine to STRICT:
00438  *
00439  * Types will be converted one at a time by volunteers who will compile
00440  * Wine with STRICT turned on. Handles that have not been converted yet
00441  * will be declared with DECLARE_OLD_HANDLE. Converted handles are
00442  * declared with DECLARE_HANDLE.
00443  * See the bug report 90 for more details:
00444  *   http://wine.codeweavers.com/bugzilla/show_bug.cgi?id=90
00445  */
00446 /*
00447  * when compiling Wine we always treat HANDLE as an UINT. Then when
00448  * we're ready we'll remove the 'defined(__WINE__)' (the equivalent
00449  * of converting it from DECLARE_OLD_HANDLE to DECLARE_HANDLE).
00450  */
00451 #if defined(STRICT) && !defined(__WINE__)
00452 typedef VOID*            HANDLE;
00453 #define DECLARE_OLD_HANDLE(a) \
00454     typedef struct a##_ { int unused; } *a; \
00455     typedef a             *P##a,          *LP##a
00456 #else
00457 #define
00458 typedef UINT              HANDLE;

```

```

00459 #define DECLARE_OLD_HANDLE(a) \
00460     typedef HANDLE      a; \
00461     typedef a           *P##a,      *LP##a
00462 #endif
00463 typedef HANDLE          *PHANDLE,    *LPHANDLE;
00464
00465 #ifdef STRICT
00466 #define DECLARE_HANDLE(a) \
00467     typedef struct a##_ { int unused; } *a; \
00468     typedef a           *P##a,      *LP##a
00469 #else /*STRICT*/
00470 #define DECLARE_HANDLE(a) \
00471     typedef HANDLE      a; \
00472     typedef a           *P##a,      *LP##a
00473 #endif /*STRICT*/
00474
00475 /* Defines */
00476
00477 /* Argument 1 passed to the DllEntryProc. */
00478 #define DLL_PROCESS_DETACH 0 /* detach process (unload library) */
00479 #define DLL_PROCESS_ATTACH 1 /* attach process (load library) */
00480 #define DLL_THREAD_ATTACH 2 /* attach new thread */
00481 #define DLL_THREAD_DETACH 3 /* detach thread */
00482
00483
00484 /* u.x.wProcessorArchitecture (NT) */
00485 #define PROCESSOR_ARCHITECTURE_INTEL 0
00486 #define PROCESSOR_ARCHITECTURE_MIPS 1
00487 #define PROCESSOR_ARCHITECTURE_ALPHA 2
00488 #define PROCESSOR_ARCHITECTURE_PPC 3
00489 #define PROCESSOR_ARCHITECTURE_SHX 4
00490 #define PROCESSOR_ARCHITECTURE_ARM 5
00491 #define PROCESSOR_ARCHITECTURE_UNKNOWN 0xFFFF
00492
00493 /* dwProcessorType */
00494 #define PROCESSOR_INTEL_386 386
00495 #define PROCESSOR_INTEL_486 486
00496 #define PROCESSOR_INTEL_PENTIUM 586
00497 #define PROCESSOR_INTEL_860 860
00498 #define PROCESSOR_MIPS_R2000 2000
00499 #define PROCESSOR_MIPS_R3000 3000
00500 #define PROCESSOR_MIPS_R4000 4000
00501 #define PROCESSOR_ALPHA_21064 21064
00502 #define PROCESSOR_PPC_601 601
00503 #define PROCESSOR_PPC_603 603
00504 #define PROCESSOR_PPC_604 604
00505 #define PROCESSOR_PPC_620 620
00506 #define PROCESSOR_HITACHI_SH3 10003
00507 #define PROCESSOR_HITACHI_SH3E 10004
00508 #define PROCESSOR_HITACHI_SH4 10005
00509 #define PROCESSOR_MOTOROLA_821 821
00510 #define PROCESSOR_SHx_SH3 103
00511 #define PROCESSOR_SHx_SH4 104
00512 #define PROCESSOR_STRONGARM 2577
00513 #define PROCESSOR_ARM720 1824 /* 0x720 */
00514 #define PROCESSOR_ARM820 2080 /* 0x820 */
00515 #define PROCESSOR_ARM920 2336 /* 0x920 */
00516 #define PROCESSOR_ARM_7TDMI 70001
00517
00518 typedef struct _MEMORY_BASIC_INFORMATION
00519 {
00520     LPVOID BaseAddress;
00521     LPVOID AllocationBase;
00522     DWORD AllocationProtect;
00523     DWORD RegionSize;
00524     DWORD State;
00525     DWORD Protect;
00526     DWORD Type;
00527 } MEMORY_BASIC_INFORMATION, *LPMEMORY_BASIC_INFORMATION, *PMEMORY_BASIC_INFORMATION;
00528
00529 #define PAGE_NOACCESS 0x01
00530 #define PAGE_READONLY 0x02
00531 #define PAGE_READWRITE 0x04
00532 #define PAGE_WRITECOPY 0x08
00533 #define PAGE_EXECUTE 0x10
00534 #define PAGE_EXECUTE_READ 0x20
00535 #define PAGE_EXECUTE_READWRITE 0x40
00536 #define PAGE_EXECUTE_WRITECOPY 0x80
00537 #define PAGE_GUARD 0x100
00538 #define PAGE_NOCACHE 0x200
00539
00540 #define MEM_COMMIT 0x00001000
00541 #define MEM_RESERVE 0x00002000
00542 #define MEM_DECOMMIT 0x00004000
00543 #define MEM_RELEASE 0x00008000
00544 #define MEM_FREE 0x00010000
00545 #define MEM_PRIVATE 0x00020000

```

```

00546 #define MEM_MAPPED                0x00040000
00547 #define MEM_RESET                   0x00080000
00548 #define MEM_TOP_DOWN                0x00100000
00549 #ifdef __WINE__
00550 #define MEM_SYSTEM                   0x80000000
00551 #endif
00552
00553 #define SEC_FILE                     0x00800000
00554 #define SEC_IMAGE                    0x01000000
00555 #define SEC_RESERVE                  0x04000000
00556 #define SEC_COMMIT                   0x08000000
00557 #define SEC_NOCACHE                  0x10000000
00558 #define MEM_IMAGE                    SEC_IMAGE
00559
00560
00561 #define MINCHAR                      0x80
00562 #define MAXCHAR                      0x7f
00563 #define MINSHORT                     0x8000
00564 #define MAXSHORT                     0x7fff
00565 #define MINLONG                      0x80000000
00566 #define MAXLONG                      0x7fffffff
00567 #define MAXBYTE                      0xff
00568 #define MAXWORD                      0xffff
00569 #define MAXDWORD                     0xffffffff
00570
00571 #define FIELD_OFFSET(type, field) \
00572     ((LONG)(INT)&((type *)0)->field))
00573
00574 #define CONTAINING_RECORD(address, type, field) \
00575     ((type *)((PCHAR)(address) - (PCHAR)&((type *)0)->field))
00576
00577 /* Types */
00578
00579 typedef struct _LIST_ENTRY {
00580     struct _LIST_ENTRY *Flink;
00581     struct _LIST_ENTRY *Blink;
00582 } LIST_ENTRY, *PLIST_ENTRY;
00583
00584 typedef struct _SINGLE_LIST_ENTRY {
00585     struct _SINGLE_LIST_ENTRY *Next;
00586 } SINGLE_LIST_ENTRY, *PSINGLE_LIST_ENTRY;
00587
00588 /* Heap flags */
00589
00590 #define HEAP_NO_SERIALIZE             0x00000001
00591 #define HEAP_GROWABLE                 0x00000002
00592 #define HEAP_GENERATE_EXCEPTIONS     0x00000004
00593 #define HEAP_ZERO_MEMORY              0x00000008
00594 #define HEAP_REALLOC_IN_PLACE_ONLY    0x00000010
00595 #define HEAP_TAIL_CHECKING_ENABLED    0x00000020
00596 #define HEAP_FREE_CHECKING_ENABLED    0x00000040
00597 #define HEAP_DISABLE_COALESCE_ON_FREE 0x00000080
00598 #define HEAP_CREATE_ALIGN_16          0x00010000
00599 #define HEAP_CREATE_ENABLE_TRACING    0x00020000
00600
00601 /* This flag allows it to create heaps shared by all processes under win95,
00602    FIXME: correct name */
00603 #define HEAP_SHARED                    0x04000000
00604
00605 /* Processor feature flags. */
00606 #define PF_FLOATING_POINT_PRECISION_ERRATA 0
00607 #define PF_FLOATING_POINT_EMULATED        1
00608 #define PF_COMPARE_EXCHANGE_DOUBLE        2
00609 #define PF_MMX_INSTRUCTIONS_AVAILABLE      3
00610 #define PF_PPC_MOVEMEM_64BIT_OK           4
00611 #define PF_ALPHA_BYTE_INSTRUCTIONS         5
00612 #define PF_XMMI_INSTRUCTIONS_AVAILABLE     6
00613 #define PF_AMD3D_INSTRUCTIONS_AVAILABLE    7
00614 #define PF_RDTSC_INSTRUCTION_AVAILABLE    8
00615
00616
00617 /* Execution state flags */
00618 #define ES_SYSTEM_REQUIRED              0x00000001
00619 #define ES_DISPLAY_REQUIRED             0x00000002
00620 #define ES_USER_PRESENT                 0x00000004
00621 #define ES_CONTINUOUS                   0x80000000
00622
00623 /* The Win32 register context */
00624
00625 /* CONTEXT is the CPU-dependent context; it should be used
00626 /* wherever a platform-specific context is needed (e.g. exception */
00627 /* handling, Win32 register functions). */
00628
00629 /* CONTEXT86 is the i386-specific context; it should be used
00630 /* wherever only a 386 context makes sense (e.g. DOS interrupts, */
00631 /* Win16 register functions), so that this code can be compiled */
00632 /* on all platforms. */

```

```

00633
00634 #define SIZE_OF_80387_REGISTERS      80
00635
00636 typedef struct _FLOATING_SAVE_AREA
00637 {
00638     DWORD    ControlWord;
00639     DWORD    StatusWord;
00640     DWORD    TagWord;
00641     DWORD    ErrorOffset;
00642     DWORD    ErrorSelector;
00643     DWORD    DataOffset;
00644     DWORD    DataSelector;
00645     BYTE    RegisterArea[SIZE_OF_80387_REGISTERS];
00646     DWORD    Cr0NpxState;
00647 } FLOATING_SAVE_AREA, *PFLOATING_SAVE_AREA;
00648
00649 #define MAXIMUM_SUPPORTED_EXTENSION    512
00650
00651 typedef struct _CONTEXT86
00652 {
00653     DWORD    ContextFlags;
00654
00655     /* These are selected by CONTEXT_DEBUG_REGISTERS */
00656     DWORD    Dr0;
00657     DWORD    Dr1;
00658     DWORD    Dr2;
00659     DWORD    Dr3;
00660     DWORD    Dr6;
00661     DWORD    Dr7;
00662
00663     /* These are selected by CONTEXT_FLOATING_POINT */
00664     FLOATING_SAVE_AREA FloatSave;
00665
00666     /* These are selected by CONTEXT_SEGMENTS */
00667     DWORD    SegGs;
00668     DWORD    SegFs;
00669     DWORD    SegEs;
00670     DWORD    SegDs;
00671
00672     /* These are selected by CONTEXT_INTEGER */
00673     DWORD    Edi;
00674     DWORD    Esi;
00675     DWORD    Ebx;
00676     DWORD    Edx;
00677     DWORD    Ecx;
00678     DWORD    Eax;
00679
00680     /* These are selected by CONTEXT_CONTROL */
00681     DWORD    Ebp;
00682     DWORD    Eip;
00683     DWORD    SegCs;
00684     DWORD    EFlags;
00685     DWORD    Esp;
00686     DWORD    SegSs;
00687
00688     BYTE    ExtendedRegisters[MAXIMUM_SUPPORTED_EXTENSION];
00689 } CONTEXT86;
00690
00691 #define CONTEXT_X86            0x00010000
00692 #define CONTEXT_i386          CONTEXT_X86
00693 #define CONTEXT_i486          CONTEXT_X86
00694
00695 #define CONTEXT86_CONTROL      (CONTEXT_i386 | 0x0001) /* SS:SP, CS:IP, FLAGS, BP */
00696 #define CONTEXT86_INTEGER      (CONTEXT_i386 | 0x0002) /* AX, BX, CX, DX, SI, DI */
00697 #define CONTEXT86_SEGMENTS     (CONTEXT_i386 | 0x0004) /* DS, ES, FS, GS */
00698 #define CONTEXT86_FLOATING_POINT (CONTEXT_i386 | 0x0008L) /* 387 state */
00699 #define CONTEXT86_DEBUG_REGISTERS (CONTEXT_i386 | 0x0010L) /* DB 0-3,6,7 */
00700 #define CONTEXT86_FULL (CONTEXT86_CONTROL | CONTEXT86_INTEGER | CONTEXT86_SEGMENTS)
00701
00702 /* i386 context definitions */
00703 #ifdef __i386__
00704
00705 #define CONTEXT_CONTROL        CONTEXT86_CONTROL
00706 #define CONTEXT_INTEGER        CONTEXT86_INTEGER
00707 #define CONTEXT_SEGMENTS       CONTEXT86_SEGMENTS
00708 #define CONTEXT_FLOATING_POINT CONTEXT86_FLOATING_POINT
00709 #define CONTEXT_DEBUG_REGISTERS CONTEXT86_DEBUG_REGISTERS
00710 #define CONTEXT_FULL           CONTEXT86_FULL
00711
00712 typedef CONTEXT86 CONTEXT;
00713
00714 #endif /* __i386__ */
00715
00716 /* x86-64 context definitions */
00717 #if defined(__x86_64__)
00718
00719 #define CONTEXT_AMD64          0x00100000

```

```

00720
00721 #define CONTEXT_CONTROL      (CONTEXT_AMD64 | 0x0001)
00722 #define CONTEXT_INTEGER      (CONTEXT_AMD64 | 0x0002)
00723 #define CONTEXT_SEGMENTS     (CONTEXT_AMD64 | 0x0004)
00724 #define CONTEXT_FLOATING_POINT (CONTEXT_AMD64 | 0x0008L)
00725 #define CONTEXT_DEBUG_REGISTERS (CONTEXT_AMD64 | 0x0010L)
00726 #define CONTEXT_FULL (CONTEXT_CONTROL | CONTEXT_INTEGER | CONTEXT_FLOATING_POINT)
00727 #define CONTEXT_ALL (CONTEXT_CONTROL | CONTEXT_INTEGER | CONTEXT_SEGMENTS | CONTEXT_FLOATING_POINT |
CONTEXT_DEBUG_REGISTERS)
00728
00729 #define EXCEPTION_READ_FAULT  0
00730 #define EXCEPTION_WRITE_FAULT 1
00731 #define EXCEPTION_EXECUTE_FAULT 8
00732
00733 typedef struct DECLSPEC_ALIGN(16) _M128A {
00734     ULONGLONG Low;
00735     ULONGLONG High;
00736 } M128A, *PM128A;
00737
00738 typedef struct _XMM_SAVE_AREA32 {
00739     WORD ControlWord;          /* 000 */
00740     WORD StatusWord;           /* 002 */
00741     BYTE TagWord;              /* 004 */
00742     BYTE Reserved1;            /* 005 */
00743     WORD ErrorOpcode;           /* 006 */
00744     DWORD ErrorOffset;         /* 008 */
00745     WORD ErrorSelector;        /* 00c */
00746     WORD Reserved2;            /* 00e */
00747     DWORD DataOffset;          /* 010 */
00748     WORD DataSelector;         /* 014 */
00749     WORD Reserved3;            /* 016 */
00750     DWORD MxCsr;               /* 018 */
00751     DWORD MxCsr_Mask;          /* 01c */
00752     M128A FloatRegisters[8];   /* 020 */
00753     M128A XmmRegisters[16];    /* 0a0 */
00754     BYTE Reserved4[96];        /* 1a0 */
00755 } XMM_SAVE_AREA32, *PXMM_SAVE_AREA32;
00756
00757 typedef struct DECLSPEC_ALIGN(16) _CONTEXT {
00758     DWORD64 P1Home;            /* 000 */
00759     DWORD64 P2Home;            /* 008 */
00760     DWORD64 P3Home;            /* 010 */
00761     DWORD64 P4Home;            /* 018 */
00762     DWORD64 P5Home;            /* 020 */
00763     DWORD64 P6Home;            /* 028 */
00764
00765     /* Control flags */
00766     DWORD ContextFlags;         /* 030 */
00767     DWORD MxCsr;                /* 034 */
00768
00769     /* Segment */
00770     WORD SegCs;                 /* 038 */
00771     WORD SegDs;                 /* 03a */
00772     WORD SegEs;                 /* 03c */
00773     WORD SegFs;                 /* 03e */
00774     WORD SegGs;                 /* 040 */
00775     WORD SegSs;                 /* 042 */
00776     DWORD EFlags;              /* 044 */
00777
00778     /* Debug */
00779     DWORD64 Dr0;                /* 048 */
00780     DWORD64 Dr1;                /* 050 */
00781     DWORD64 Dr2;                /* 058 */
00782     DWORD64 Dr3;                /* 060 */
00783     DWORD64 Dr6;                /* 068 */
00784     DWORD64 Dr7;                /* 070 */
00785
00786     /* Integer */
00787     DWORD64 Rax;                /* 078 */
00788     DWORD64 Rcx;                /* 080 */
00789     DWORD64 Rdx;                /* 088 */
00790     DWORD64 Rbx;                /* 090 */
00791     DWORD64 Rsp;                /* 098 */
00792     DWORD64 Rbp;                /* 0a0 */
00793     DWORD64 Rsi;                /* 0a8 */
00794     DWORD64 Rdi;                /* 0b0 */
00795     DWORD64 R8;                 /* 0b8 */
00796     DWORD64 R9;                 /* 0c0 */
00797     DWORD64 R10;                /* 0c8 */
00798     DWORD64 R11;                /* 0d0 */
00799     DWORD64 R12;                /* 0d8 */
00800     DWORD64 R13;                /* 0e0 */
00801     DWORD64 R14;                /* 0e8 */
00802     DWORD64 R15;                /* 0f0 */
00803
00804     /* Counter */
00805     DWORD64 Rip;                /* 0f8 */

```

```

00806
00807     /* Floating point */
00808     union {
00809         XMM_SAVE_AREA32 FltSave; /* 100 */
00810         struct {
00811             M128A Header[2]; /* 100 */
00812             M128A Legacy[8]; /* 120 */
00813             M128A Xmm0; /* 1a0 */
00814             M128A Xmm1; /* 1b0 */
00815             M128A Xmm2; /* 1c0 */
00816             M128A Xmm3; /* 1d0 */
00817             M128A Xmm4; /* 1e0 */
00818             M128A Xmm5; /* 1f0 */
00819             M128A Xmm6; /* 200 */
00820             M128A Xmm7; /* 210 */
00821             M128A Xmm8; /* 220 */
00822             M128A Xmm9; /* 230 */
00823             M128A Xmm10; /* 240 */
00824             M128A Xmm11; /* 250 */
00825             M128A Xmm12; /* 260 */
00826             M128A Xmm13; /* 270 */
00827             M128A Xmm14; /* 280 */
00828             M128A Xmm15; /* 290 */
00829         } DUMMYSTRUCTNAME;
00830     } DUMMYUNIONNAME;
00831
00832     /* Vector */
00833     M128A VectorRegister[26]; /* 300 */
00834     DWORD64 VectorControl; /* 4a0 */
00835
00836     /* Debug control */
00837     DWORD64 DebugControl; /* 4a8 */
00838     DWORD64 LastBranchToRip; /* 4b0 */
00839     DWORD64 LastBranchFromRip; /* 4b8 */
00840     DWORD64 LastExceptionToRip; /* 4c0 */
00841     DWORD64 LastExceptionFromRip; /* 4c8 */
00842 } CONTEXT;
00843
00844 typedef struct _RUNTIME_FUNCTION
00845 {
00846     DWORD BeginAddress;
00847     DWORD EndAddress;
00848     DWORD UnwindData;
00849 } RUNTIME_FUNCTION, *PRUNTIME_FUNCTION;
00850
00851 #define UNWIND_HISTORY_TABLE_SIZE 12
00852
00853 typedef struct _UNWIND_HISTORY_TABLE_ENTRY
00854 {
00855     ULONG64 ImageBase;
00856     PRUNTIME_FUNCTION FunctionEntry;
00857 } UNWIND_HISTORY_TABLE_ENTRY, *PUNWIND_HISTORY_TABLE_ENTRY;
00858
00859 #define UNWIND_HISTORY_TABLE_NONE 0
00860 #define UNWIND_HISTORY_TABLE_GLOBAL 1
00861 #define UNWIND_HISTORY_TABLE_LOCAL 2
00862
00863 typedef struct _UNWIND_HISTORY_TABLE
00864 {
00865     ULONG Count;
00866     UCHAR Search;
00867     ULONG64 LowAddress;
00868     ULONG64 HighAddress;
00869     UNWIND_HISTORY_TABLE_ENTRY Entry[UNWIND_HISTORY_TABLE_SIZE];
00870 } UNWIND_HISTORY_TABLE, *PUNWIND_HISTORY_TABLE;
00871
00872 typedef struct _KNONVOLATILE_CONTEXT_POINTERS
00873 {
00874     union
00875     {
00876         PM128A FloatingContext[16];
00877         struct
00878         {
00879             PM128A Xmm0;
00880             PM128A Xmm1;
00881             PM128A Xmm2;
00882             PM128A Xmm3;
00883             PM128A Xmm4;
00884             PM128A Xmm5;
00885             PM128A Xmm6;
00886             PM128A Xmm7;
00887             PM128A Xmm8;
00888             PM128A Xmm9;
00889             PM128A Xmm10;
00890             PM128A Xmm11;
00891             PM128A Xmm12;
00892             PM128A Xmm13;

```

```

00893         PM128A Xmm14;
00894         PM128A Xmm15;
00895     } DUMMYSTRUCTNAME1;
00896 } DUMMYUNIONNAME1;
00897
00898 union
00899 {
00900     PULONG64 IntegerContext[16];
00901     struct
00902     {
00903         PULONG64 Rax;
00904         PULONG64 Rcx;
00905         PULONG64 Rdx;
00906         PULONG64 Rbx;
00907         PULONG64 Rsp;
00908         PULONG64 Rbp;
00909         PULONG64 Rsi;
00910         PULONG64 Rdi;
00911         PULONG64 R8;
00912         PULONG64 R9;
00913         PULONG64 R10;
00914         PULONG64 R11;
00915         PULONG64 R12;
00916         PULONG64 R13;
00917         PULONG64 R14;
00918         PULONG64 R15;
00919     } DUMMYSTRUCTNAME2;
00920 } DUMMYUNIONNAME2;
00921 } KNONVOLATILE_CONTEXT_POINTERS, *PKNONVOLATILE_CONTEXT_POINTERS;
00922
00923 BOOLEAN CDECL RtlAddFunctionTable(RUNTIME_FUNCTION*, DWORD, DWORD64);
00924 BOOLEAN CDECL RtlDeleteFunctionTable(RUNTIME_FUNCTION*);
00925 PRUNTIME_FUNCTION WINAPI RtlLookupFunctionEntry(DWORD64, DWORD64*, UNWIND_HISTORY_TABLE*);
00926 PVOID WINAPI
00927     RtlVirtualUnwind(ULONG, ULONG64, ULONG64, RUNTIME_FUNCTION*, CONTEXT*, PVOID*, ULONG64*, KNONVOLATILE_CONTEXT_POINTERS*);
00928 #define UNW_FLAG_NHANDLER 0
00929 #define UNW_FLAG_EHANDLER 1
00930 #define UNW_FLAG_UHANDLER 2
00931 #define UNW_FLAG_CHAININFO 4
00932
00933 #endif /* __x86_64__ */
00934
00935 /* Alpha context definitions */
00936 #if defined(__alpha__)
00937
00938 #define CONTEXT_ALPHA 0x00020000
00939
00940 #define CONTEXT_CONTROL (CONTEXT_ALPHA | 0x00000001L)
00941 #define CONTEXT_FLOATING_POINT (CONTEXT_ALPHA | 0x00000002L)
00942 #define CONTEXT_INTEGER (CONTEXT_ALPHA | 0x00000004L)
00943 #define CONTEXT_FULL (CONTEXT_CONTROL | CONTEXT_FLOATING_POINT | CONTEXT_INTEGER)
00944
00945 typedef struct _CONTEXT
00946 {
00947     /* selected by CONTEXT_FLOATING_POINT */
00948     ULONGLONG FltF0;
00949     ULONGLONG FltF1;
00950     ULONGLONG FltF2;
00951     ULONGLONG FltF3;
00952     ULONGLONG FltF4;
00953     ULONGLONG FltF5;
00954     ULONGLONG FltF6;
00955     ULONGLONG FltF7;
00956     ULONGLONG FltF8;
00957     ULONGLONG FltF9;
00958     ULONGLONG FltF10;
00959     ULONGLONG FltF11;
00960     ULONGLONG FltF12;
00961     ULONGLONG FltF13;
00962     ULONGLONG FltF14;
00963     ULONGLONG FltF15;
00964     ULONGLONG FltF16;
00965     ULONGLONG FltF17;
00966     ULONGLONG FltF18;
00967     ULONGLONG FltF19;
00968     ULONGLONG FltF20;
00969     ULONGLONG FltF21;
00970     ULONGLONG FltF22;
00971     ULONGLONG FltF23;
00972     ULONGLONG FltF24;
00973     ULONGLONG FltF25;
00974     ULONGLONG FltF26;
00975     ULONGLONG FltF27;
00976     ULONGLONG FltF28;
00977     ULONGLONG FltF29;
00978     ULONGLONG FltF30;

```

```

00979     ULONGLONG FltF31;
00980
00981     /* selected by CONTEXT_INTEGER */
00982     ULONGLONG IntV0;
00983     ULONGLONG IntT0;
00984     ULONGLONG IntT1;
00985     ULONGLONG IntT2;
00986     ULONGLONG IntT3;
00987     ULONGLONG IntT4;
00988     ULONGLONG IntT5;
00989     ULONGLONG IntT6;
00990     ULONGLONG IntT7;
00991     ULONGLONG IntS0;
00992     ULONGLONG IntS1;
00993     ULONGLONG IntS2;
00994     ULONGLONG IntS3;
00995     ULONGLONG IntS4;
00996     ULONGLONG IntS5;
00997     ULONGLONG IntFp;
00998     ULONGLONG IntA0;
00999     ULONGLONG IntA1;
01000     ULONGLONG IntA2;
01001     ULONGLONG IntA3;
01002     ULONGLONG IntA4;
01003     ULONGLONG IntA5;
01004     ULONGLONG IntT8;
01005     ULONGLONG IntT9;
01006     ULONGLONG IntT10;
01007     ULONGLONG IntT11;
01008     ULONGLONG IntRa;
01009     ULONGLONG IntT12;
01010     ULONGLONG IntAt;
01011     ULONGLONG IntGp;
01012     ULONGLONG IntSp;
01013     ULONGLONG IntZero;
01014
01015     /* selected by CONTEXT_FLOATING_POINT */
01016     ULONGLONG FpCsr;
01017     ULONGLONG SoftFpCsr;
01018
01019     /* selected by CONTEXT_CONTROL */
01020     ULONGLONG Fir;
01021     DWORD Psr;
01022     DWORD ContextFlags;
01023     DWORD Fill[4];
01024 } CONTEXT;
01025
01026 #define _QUAD_PSR_OFFSET    HighSoftFpCsr
01027 #define _QUAD_FLAGS_OFFSET HighFir
01028
01029 #endif /* _ALPHA_ */
01030
01031 #ifndef __loongarch64
01032 /*
01033  * FIXME:
01034  *
01035  * There is not yet an official CONTEXT structure defined for the
01036  * loongarch64 architecture (64-bit LE), so I just made one up.
01037  */
01038 */
01039
01040 #define CONTEXT_LOONGARCH64            0x40000000
01041 #define CONTEXT_CONTROL                (CONTEXT_LOONGARCH64 | 0x00000001)
01042 #define CONTEXT_INTEGER                (CONTEXT_LOONGARCH64 | 0x00000002)
01043 #define CONTEXT_FLOATING_POINT        (CONTEXT_LOONGARCH64 | 0x00000004)
01044 #define CONTEXT_DEBUG_REGISTERS      (CONTEXT_LOONGARCH64 | 0x00000008)
01045
01046 #define CONTEXT_FULL (CONTEXT_CONTROL | CONTEXT_INTEGER)
01047
01048 #define EXCEPTION_READ_FAULT    0
01049 #define EXCEPTION_WRITE_FAULT  1
01050 #define EXCEPTION_EXECUTE_FAULT 8
01051
01052 typedef struct _CONTEXT {
01053     ULONG ContextFlags;
01054
01055     /* This section is specified/returned if the ContextFlags word contains
01056        the flag CONTEXT_INTEGER. */
01057     ULONGLONG X0;
01058     ULONGLONG X1;
01059     ULONGLONG X2;
01060     ULONGLONG X3;
01061     ULONGLONG X4;
01062     ULONGLONG X5;
01063     ULONGLONG X6;
01064     ULONGLONG X7;
01065     ULONGLONG X8;

```

```
01066     ULONGLONG X9;
01067     ULONGLONG X10;
01068     ULONGLONG X11;
01069     ULONGLONG X12;
01070     ULONGLONG X13;
01071     ULONGLONG X14;
01072     ULONGLONG X15;
01073     ULONGLONG X16;
01074     ULONGLONG X17;
01075     ULONGLONG X18;
01076     ULONGLONG X19;
01077     ULONGLONG X20;
01078     ULONGLONG X21;
01079     ULONGLONG X22;
01080     ULONGLONG X23;
01081     ULONGLONG X24;
01082     ULONGLONG X25;
01083     ULONGLONG X26;
01084     ULONGLONG X27;
01085     ULONGLONG X28;
01086     ULONGLONG X29;
01087     ULONGLONG X30;
01088     ULONGLONG X31;
01089
01090     /* These are selected by CONTEXT_CONTROL */
01091     ULONGLONG Sp;
01092     ULONGLONG Pc;
01093     ULONGLONG PState;
01094
01095     /* These are selected by CONTEXT_FLOATING_POINT */
01096     /* FIXME */
01097 } CONTEXT;
01098
01099 #endif /* __loongarch64 */
01100
01101 /* Mips context definitions */
01102 #if defined(_MIPS_) || defined(__MIPS__) || defined(__mips__)
01103
01104 #define CONTEXT_R4000    0x00010000
01105
01106 #define CONTEXT_CONTROL    (CONTEXT_R4000 | 0x00000001)
01107 #define CONTEXT_FLOATING_POINT    (CONTEXT_R4000 | 0x00000002)
01108 #define CONTEXT_INTEGER    (CONTEXT_R4000 | 0x00000004)
01109
01110 #define CONTEXT_FULL    (CONTEXT_CONTROL | CONTEXT_FLOATING_POINT | CONTEXT_INTEGER)
01111
01112 typedef struct _CONTEXT
01113 {
01114     DWORD Argument[4];
01115     /* These are selected by CONTEXT_FLOATING_POINT */
01116     DWORD FltF0;
01117     DWORD FltF1;
01118     DWORD FltF2;
01119     DWORD FltF3;
01120     DWORD FltF4;
01121     DWORD FltF5;
01122     DWORD FltF6;
01123     DWORD FltF7;
01124     DWORD FltF8;
01125     DWORD FltF9;
01126     DWORD FltF10;
01127     DWORD FltF11;
01128     DWORD FltF12;
01129     DWORD FltF13;
01130     DWORD FltF14;
01131     DWORD FltF15;
01132     DWORD FltF16;
01133     DWORD FltF17;
01134     DWORD FltF18;
01135     DWORD FltF19;
01136     DWORD FltF20;
01137     DWORD FltF21;
01138     DWORD FltF22;
01139     DWORD FltF23;
01140     DWORD FltF24;
01141     DWORD FltF25;
01142     DWORD FltF26;
01143     DWORD FltF27;
01144     DWORD FltF28;
01145     DWORD FltF29;
01146     DWORD FltF30;
01147     DWORD FltF31;
01148
01149     /* These are selected by CONTEXT_INTEGER */
01150     DWORD IntZero;
01151     DWORD IntAt;
01152     DWORD IntV0;
```

```

01153     DWORD IntV1;
01154     DWORD IntA0;
01155     DWORD IntA1;
01156     DWORD IntA2;
01157     DWORD IntA3;
01158     DWORD IntT0;
01159     DWORD IntT1;
01160     DWORD IntT2;
01161     DWORD IntT3;
01162     DWORD IntT4;
01163     DWORD IntT5;
01164     DWORD IntT6;
01165     DWORD IntT7;
01166     DWORD IntS0;
01167     DWORD IntS1;
01168     DWORD IntS2;
01169     DWORD IntS3;
01170     DWORD IntS4;
01171     DWORD IntS5;
01172     DWORD IntS6;
01173     DWORD IntS7;
01174     DWORD IntT8;
01175     DWORD IntT9;
01176     DWORD IntK0;
01177     DWORD IntK1;
01178     DWORD IntGp;
01179     DWORD IntSp;
01180     DWORD IntS8;
01181     DWORD IntRa;
01182     DWORD IntLo;
01183     DWORD IntHi;
01184
01185     /* These are selected by CONTEXT_FLOATING_POINT */
01186     DWORD Fsr;
01187
01188     /* These are selected by CONTEXT_CONTROL */
01189     DWORD Fir;
01190     DWORD Psr;
01191
01192     DWORD ContextFlags;
01193     DWORD Fill[2];
01194 } CONTEXT;
01195
01196 #endif /* _MIPS_ */
01197
01198 /* PowerPC context definitions */
01199 #ifdef __PPC__
01200
01201 #define CONTEXT_CONTROL          0x0001
01202 #define CONTEXT_FLOATING_POINT  0x0002
01203 #define CONTEXT_INTEGER          0x0004
01204 #define CONTEXT_DEBUG_REGISTERS 0x0008
01205 #define CONTEXT_FULL (CONTEXT_CONTROL | CONTEXT_FLOATING_POINT | CONTEXT_INTEGER)
01206
01207 typedef struct
01208 {
01209     /* These are selected by CONTEXT_FLOATING_POINT */
01210     double Fpr0;
01211     double Fpr1;
01212     double Fpr2;
01213     double Fpr3;
01214     double Fpr4;
01215     double Fpr5;
01216     double Fpr6;
01217     double Fpr7;
01218     double Fpr8;
01219     double Fpr9;
01220     double Fpr10;
01221     double Fpr11;
01222     double Fpr12;
01223     double Fpr13;
01224     double Fpr14;
01225     double Fpr15;
01226     double Fpr16;
01227     double Fpr17;
01228     double Fpr18;
01229     double Fpr19;
01230     double Fpr20;
01231     double Fpr21;
01232     double Fpr22;
01233     double Fpr23;
01234     double Fpr24;
01235     double Fpr25;
01236     double Fpr26;
01237     double Fpr27;
01238     double Fpr28;
01239     double Fpr29;

```

```
01240     double Fpr30;
01241     double Fpr31;
01242     double Fpscr;
01243
01244     /* These are selected by CONTEXT_INTEGER */
01245     DWORD Gpr0;
01246     DWORD Gpr1;
01247     DWORD Gpr2;
01248     DWORD Gpr3;
01249     DWORD Gpr4;
01250     DWORD Gpr5;
01251     DWORD Gpr6;
01252     DWORD Gpr7;
01253     DWORD Gpr8;
01254     DWORD Gpr9;
01255     DWORD Gpr10;
01256     DWORD Gpr11;
01257     DWORD Gpr12;
01258     DWORD Gpr13;
01259     DWORD Gpr14;
01260     DWORD Gpr15;
01261     DWORD Gpr16;
01262     DWORD Gpr17;
01263     DWORD Gpr18;
01264     DWORD Gpr19;
01265     DWORD Gpr20;
01266     DWORD Gpr21;
01267     DWORD Gpr22;
01268     DWORD Gpr23;
01269     DWORD Gpr24;
01270     DWORD Gpr25;
01271     DWORD Gpr26;
01272     DWORD Gpr27;
01273     DWORD Gpr28;
01274     DWORD Gpr29;
01275     DWORD Gpr30;
01276     DWORD Gpr31;
01277
01278     DWORD Cr;
01279     DWORD Xer;
01280
01281     /* These are selected by CONTEXT_CONTROL */
01282     DWORD Msr;
01283     DWORD Iar;
01284     DWORD Lr;
01285     DWORD Ctr;
01286
01287     DWORD ContextFlags;
01288     DWORD Fill[3];
01289
01290     /* These are selected by CONTEXT_DEBUG_REGISTERS */
01291     DWORD Dr0;
01292     DWORD Dr1;
01293     DWORD Dr2;
01294     DWORD Dr3;
01295     DWORD Dr4;
01296     DWORD Dr5;
01297     DWORD Dr6;
01298     DWORD Dr7;
01299 } CONTEXT;
01300
01301 typedef struct _STACK_FRAME_HEADER
01302 {
01303     DWORD BackChain;
01304     DWORD GlueSaved1;
01305     DWORD GlueSaved2;
01306     DWORD Reserved1;
01307     DWORD Spare1;
01308     DWORD Spare2;
01309
01310     DWORD Parameter0;
01311     DWORD Parameter1;
01312     DWORD Parameter2;
01313     DWORD Parameter3;
01314     DWORD Parameter4;
01315     DWORD Parameter5;
01316     DWORD Parameter6;
01317     DWORD Parameter7;
01318 } STACK_FRAME_HEADER, *PSTACK_FRAME_HEADER;
01319
01320 #endif /* __PPC__ */
01321
01322 #ifdef __sparc__
01323 /*
01324  * FIXME:
01325  *
01326  */
```

```

01327 * There is no official CONTEXT structure defined for the SPARC
01328 * architecture, so I just made one up.
01329 *
01330 * This structure is valid only for 32-bit SPARC architectures,
01331 * not for 64-bit SPARC.
01332 *
01333 * Note that this structure contains only the 'top-level' registers;
01334 * the rest of the register window chain is not visible.
01335 *
01336 * The layout follows the Solaris 'prgregset_t' structure.
01337 *
01338 */
01339
01340 #define CONTEXT_SPARC          0x10000000
01341
01342 #define CONTEXT_CONTROL        (CONTEXT_SPARC | 0x00000001)
01343 #define CONTEXT_FLOATING_POINT (CONTEXT_SPARC | 0x00000002)
01344 #define CONTEXT_INTEGER        (CONTEXT_SPARC | 0x00000004)
01345
01346 #define CONTEXT_FULL (CONTEXT_CONTROL | CONTEXT_FLOATING_POINT | CONTEXT_INTEGER)
01347
01348 typedef struct _CONTEXT
01349 {
01350     DWORD ContextFlags;
01351
01352     /* These are selected by CONTEXT_INTEGER */
01353     DWORD g0;
01354     DWORD g1;
01355     DWORD g2;
01356     DWORD g3;
01357     DWORD g4;
01358     DWORD g5;
01359     DWORD g6;
01360     DWORD g7;
01361     DWORD o0;
01362     DWORD o1;
01363     DWORD o2;
01364     DWORD o3;
01365     DWORD o4;
01366     DWORD o5;
01367     DWORD o6;
01368     DWORD o7;
01369     DWORD i0;
01370     DWORD i1;
01371     DWORD i2;
01372     DWORD i3;
01373     DWORD i4;
01374     DWORD i5;
01375     DWORD i6;
01376     DWORD i7;
01377
01378     /* These are selected by CONTEXT_CONTROL */
01379     DWORD psr;
01380     DWORD pc;
01381     DWORD npc;
01382     DWORD y;
01383     DWORD wim;
01384     DWORD tbr;
01385
01386     /* FIXME: floating point registers missing */
01387 } CONTEXT;
01388
01389 #endif /* __sparc__ */
01390
01391 #ifdef __s390__
01392 /*
01393 * FIXME:
01394 *
01395 * There is no official CONTEXT structure defined for the S390
01396 * architecture, so I just made one up.
01397 *
01398 * Note that this structure contains only the 'top-level' registers;
01399 * the rest of the register window chain is not visible.
01400 *
01401 * The layout is based on the sparc one.
01402 *
01403 */

```

```

01414
01415 #define CONTEXT_S390C          0x20000000
01416
01417 #define CONTEXT_CONTROL        (CONTEXT_S390 | 0x00000001)
01418 #define CONTEXT_FLOATING_POINT (CONTEXT_S390 | 0x00000002)
01419 #define CONTEXT_INTEGER        (CONTEXT_S390 | 0x00000004)
01420
01421 #define CONTEXT_FULL (CONTEXT_CONTROL | CONTEXT_FLOATING_POINT | CONTEXT_INTEGER)
01422
01423 typedef struct _CONTEXT
01424 {
01425     DWORD ContextFlags;
01426
01427     /* These are selected by CONTEXT_INTEGER */
01428     DWORD r0;
01429     DWORD r1;
01430     DWORD r2;
01431     DWORD r3;
01432     DWORD r4;
01433     DWORD r5;
01434     DWORD r6;
01435     DWORD r7;
01436     DWORD r8;
01437     DWORD r9;
01438     DWORD r10;
01439     DWORD r11;
01440     DWORD r12;
01441     DWORD r13;
01442     DWORD r14;
01443     DWORD r15;
01444
01445     /* FIXME: this section is fictional (copied from sparc) */
01446     DWORD psr;
01447     DWORD pc;
01448     DWORD npc;
01449     DWORD y;
01450     DWORD wim;
01451     DWORD tbr;
01452
01453     /* FIXME: floating point registers missing */
01454 } CONTEXT;
01455
01456 #endif /* __s390__ */
01457
01458 #ifdef __arm__
01459
01460 /* These definitions are taken directly from wine
01461 * http://source.winehq.org/git/wine.git/blob\_plain/HEAD:/include/winnt.h */
01462
01463 /* The following flags control the contents of the CONTEXT structure. */
01464
01465 #define CONTEXT_ARM          0x02000000
01466 #define CONTEXT_CONTROL      (CONTEXT_ARM | 0x00000001)
01467 #define CONTEXT_INTEGER      (CONTEXT_ARM | 0x00000002)
01468 #define CONTEXT_FLOATING_POINT (CONTEXT_ARM | 0x00000004)
01469 #define CONTEXT_DEBUG_REGISTERS (CONTEXT_ARM | 0x00000008)
01470
01471 #define CONTEXT_FULL (CONTEXT_CONTROL | CONTEXT_INTEGER)
01472
01473 #define EXCEPTION_READ_FAULT  0
01474 #define EXCEPTION_WRITE_FAULT 1
01475 #define EXCEPTION_EXECUTE_FAULT 8
01476
01477 typedef struct _CONTEXT {
01478     /* The flags values within this flag control the contents of
01479     * a CONTEXT record.
01480     *
01481     * If the context record is used as an input parameter, then
01482     * for each portion of the context record controlled by a flag
01483     * whose value is set, it is assumed that that portion of the
01484     * context record contains valid context. If the context record
01485     * is being used to modify a thread's context, then only that
01486     * portion of the threads context will be modified.
01487     *
01488     * If the context record is used as an IN OUT parameter to capture
01489     * the context of a thread, then only those portions of the thread's
01490     * context corresponding to set flags will be returned.
01491     *
01492     * The context record is never used as an OUT only parameter. */
01493     ULONG ContextFlags;
01494
01495     /* This section is specified/returned if the ContextFlags word contains
01496     * the flag CONTEXT_INTEGER. */
01497     ULONG R0;
01498     ULONG R1;

```

```

01501 ULONG R2;
01502 ULONG R3;
01503 ULONG R4;
01504 ULONG R5;
01505 ULONG R6;
01506 ULONG R7;
01507 ULONG R8;
01508 ULONG R9;
01509 ULONG R10;
01510 ULONG Fp;
01511 ULONG Ip;
01512
01513 /* These are selected by CONTEXT_CONTROL */
01514 ULONG Sp;
01515 ULONG Lr;
01516 ULONG Pc;
01517 ULONG Cpsr;
01518 } CONTEXT;
01519
01520 #endif /* __arm__ */
01521
01522 #ifdef __aarch64__
01523 /*
01524  * FIXME:
01525  *
01526  * There is not yet an official CONTEXT structure defined for the AArch64
01527  * architecture, so I just made one up.
01528  *
01529  */
01530
01531 /* These definitions are taken directly from wine
01532  * http://source.winehq.org/git/wine.git/blob\_plain/HEAD:/include/winnt.h */
01533
01534 #define CONTEXT_ARM64 0x2000000
01535 #define CONTEXT_CONTROL (CONTEXT_ARM64 | 0x00000001)
01536 #define CONTEXT_INTEGER (CONTEXT_ARM64 | 0x00000002)
01537 #define CONTEXT_FLOATING_POINT (CONTEXT_ARM64 | 0x00000004)
01538 #define CONTEXT_DEBUG_REGISTERS (CONTEXT_ARM64 | 0x00000008)
01539
01540 #define CONTEXT_FULL (CONTEXT_CONTROL | CONTEXT_INTEGER)
01541
01542 #define EXCEPTION_READ_FAULT 0
01543 #define EXCEPTION_WRITE_FAULT 1
01544 #define EXCEPTION_EXECUTE_FAULT 8
01545
01546 typedef struct _CONTEXT {
01547     ULONG ContextFlags;
01548
01549     /* This section is specified/returned if the ContextFlags word contains
01550      the flag CONTEXT_INTEGER. */
01551     ULONGLONG X0;
01552     ULONGLONG X1;
01553     ULONGLONG X2;
01554     ULONGLONG X3;
01555     ULONGLONG X4;
01556     ULONGLONG X5;
01557     ULONGLONG X6;
01558     ULONGLONG X7;
01559     ULONGLONG X8;
01560     ULONGLONG X9;
01561     ULONGLONG X10;
01562     ULONGLONG X11;
01563     ULONGLONG X12;
01564     ULONGLONG X13;
01565     ULONGLONG X14;
01566     ULONGLONG X15;
01567     ULONGLONG X16;
01568     ULONGLONG X17;
01569     ULONGLONG X18;
01570     ULONGLONG X19;
01571     ULONGLONG X20;
01572     ULONGLONG X21;
01573     ULONGLONG X22;
01574     ULONGLONG X23;
01575     ULONGLONG X24;
01576     ULONGLONG X25;
01577     ULONGLONG X26;
01578     ULONGLONG X27;
01579     ULONGLONG X28;
01580     ULONGLONG X29;
01581     ULONGLONG X30;
01582
01583     /* These are selected by CONTEXT_CONTROL */
01584     ULONGLONG Sp;
01585     ULONGLONG Pc;
01586     ULONGLONG PState;
01587

```

```

01588     /* These are selected by CONTEXT_FLOATING_POINT */
01589     /* FIXME */
01590 } CONTEXT;
01591
01592 #endif /* __aarch64__ */
01593
01594 #ifdef __e2k__
01595     /*
01596     * FIXME:
01597     *
01598     * There is not yet an official CONTEXT structure defined for the
01599     * Elbrus 2000 architecture (64-bit LE), so I just made one up.
01600     */
01601     /*
01602     #define CONTEXT_E2K                0x40000000
01603     #define CONTEXT_CONTROL            (CONTEXT_E2K | 0x00000001)
01604     #define CONTEXT_INTEGER            (CONTEXT_E2K | 0x00000002)
01605     #define CONTEXT_FLOATING_POINT    (CONTEXT_E2K | 0x00000004)
01606     #define CONTEXT_DEBUG_REGISTERS   (CONTEXT_E2K | 0x00000008)
01607     #define CONTEXT_FULL (CONTEXT_CONTROL | CONTEXT_INTEGER)
01608
01609     #define EXCEPTION_READ_FAULT      0
01610     #define EXCEPTION_WRITE_FAULT     1
01611     #define EXCEPTION_EXECUTE_FAULT   8
01612
01613     typedef struct _CONTEXT {
01614         ULONG ContextFlags;
01615
01616         /* This section is specified/returned if the ContextFlags word contains
01617         the flag CONTEXT_INTEGER. */
01618         ULONGLONG X0;
01619         ULONGLONG X1;
01620         ULONGLONG X2;
01621         ULONGLONG X3;
01622         ULONGLONG X4;
01623         ULONGLONG X5;
01624         ULONGLONG X6;
01625         ULONGLONG X7;
01626         ULONGLONG X8;
01627         ULONGLONG X9;
01628         ULONGLONG X10;
01629         ULONGLONG X11;
01630         ULONGLONG X12;
01631         ULONGLONG X13;
01632         ULONGLONG X14;
01633         ULONGLONG X15;
01634         ULONGLONG X16;
01635         ULONGLONG X17;
01636         ULONGLONG X18;
01637         ULONGLONG X19;
01638         ULONGLONG X20;
01639         ULONGLONG X21;
01640         ULONGLONG X22;
01641         ULONGLONG X23;
01642         ULONGLONG X24;
01643         ULONGLONG X25;
01644         ULONGLONG X26;
01645         ULONGLONG X27;
01646         ULONGLONG X28;
01647         ULONGLONG X29;
01648         ULONGLONG X30;
01649
01650         /* These are selected by CONTEXT_CONTROL */
01651         ULONGLONG Sp;
01652         ULONGLONG Pc;
01653         ULONGLONG PState;
01654
01655         /* These are selected by CONTEXT_FLOATING_POINT */
01656         /* FIXME */
01657     } CONTEXT;
01658
01659 #endif /* __e2k__ */
01660
01661 #ifdef __riscv
01662     #if __riscv_xlen==64
01663         /*
01664         * FIXME:
01665         *
01666         * There is not yet an official CONTEXT structure defined for the
01667         * riscv64 architecture (64-bit LE), so I just made one up.
01668         */
01669         /*
01670         #define CONTEXT_RISCV64                0x40000000
01671         #define CONTEXT_CONTROL            (CONTEXT_RISCV64 | 0x00000001)
01672
01673         #define EXCEPTION_READ_FAULT      0
01674         #define EXCEPTION_WRITE_FAULT     1
01675         #define EXCEPTION_EXECUTE_FAULT   8
01676
01677         typedef struct _CONTEXT {
01678             ULONG ContextFlags;
01679
01680             /* This section is specified/returned if the ContextFlags word contains
01681             the flag CONTEXT_INTEGER. */
01682             ULONGLONG X0;
01683             ULONGLONG X1;
01684             ULONGLONG X2;
01685             ULONGLONG X3;
01686             ULONGLONG X4;
01687             ULONGLONG X5;
01688             ULONGLONG X6;
01689             ULONGLONG X7;
01690             ULONGLONG X8;
01691             ULONGLONG X9;
01692             ULONGLONG X10;
01693             ULONGLONG X11;
01694             ULONGLONG X12;
01695             ULONGLONG X13;
01696             ULONGLONG X14;
01697             ULONGLONG X15;
01698             ULONGLONG X16;
01699             ULONGLONG X17;
01700             ULONGLONG X18;
01701             ULONGLONG X19;
01702             ULONGLONG X20;
01703             ULONGLONG X21;
01704             ULONGLONG X22;
01705             ULONGLONG X23;
01706             ULONGLONG X24;
01707             ULONGLONG X25;
01708             ULONGLONG X26;
01709             ULONGLONG X27;
01710             ULONGLONG X28;
01711             ULONGLONG X29;
01712             ULONGLONG X30;
01713
01714             /* These are selected by CONTEXT_CONTROL */
01715             ULONGLONG Sp;
01716             ULONGLONG Pc;
01717             ULONGLONG PState;
01718
01719             /* These are selected by CONTEXT_FLOATING_POINT */
01720             /* FIXME */
01721         } CONTEXT;
01722     #endif
01723 #endif

```

```

01675 #define CONTEXT_INTEGER          (CONTEXT_RISCV64 | 0x00000002)
01676 #define CONTEXT_FLOATING_POINT    (CONTEXT_RISCV64 | 0x00000004)
01677 #define CONTEXT_DEBUG_REGISTERS   (CONTEXT_RISCV64 | 0x00000008)
01678
01679 #define CONTEXT_FULL (CONTEXT_CONTROL | CONTEXT_INTEGER)
01680
01681 #define EXCEPTION_READ_FAULT      0
01682 #define EXCEPTION_WRITE_FAULT     1
01683 #define EXCEPTION_EXECUTE_FAULT   8
01684
01685 typedef struct _CONTEXT {
01686     ULONG ContextFlags;
01687
01688     /* This section is specified/returned if the ContextFlags word contains
01689        the flag CONTEXT_INTEGER. */
01690     ULONGLONG X0;
01691     ULONGLONG X1;
01692     ULONGLONG X2;
01693     ULONGLONG X3;
01694     ULONGLONG X4;
01695     ULONGLONG X5;
01696     ULONGLONG X6;
01697     ULONGLONG X7;
01698     ULONGLONG X8;
01699     ULONGLONG X9;
01700     ULONGLONG X10;
01701     ULONGLONG X11;
01702     ULONGLONG X12;
01703     ULONGLONG X13;
01704     ULONGLONG X14;
01705     ULONGLONG X15;
01706     ULONGLONG X16;
01707     ULONGLONG X17;
01708     ULONGLONG X18;
01709     ULONGLONG X19;
01710     ULONGLONG X20;
01711     ULONGLONG X21;
01712     ULONGLONG X22;
01713     ULONGLONG X23;
01714     ULONGLONG X24;
01715     ULONGLONG X25;
01716     ULONGLONG X26;
01717     ULONGLONG X27;
01718     ULONGLONG X28;
01719     ULONGLONG X29;
01720     ULONGLONG X30;
01721     ULONGLONG X31;
01722
01723     /* These are selected by CONTEXT_CONTROL */
01724     ULONGLONG Sp;
01725     ULONGLONG Pc;
01726     ULONGLONG PState;
01727
01728     /* These are selected by CONTEXT_FLOATING_POINT */
01729     /* FIXME */
01730 } CONTEXT;
01731
01732 #endif /* __riscv_xlen==64 */
01733 #endif /* __riscv64__ */
01734
01735
01736 #if !defined(CONTEXT_FULL) && !defined(RC_INVOKED)
01737 #error You need to define a CONTEXT for your CPU
01738 #endif
01739
01740 typedef CONTEXT *PCONTEXT;
01741
01742 #ifdef __WINE__
01743
01744 /* Macros for easier access to i386 context registers */
01745
01746 #define AX_reg(context)      (*(WORD*)&(context)->Eax)
01747 #define BX_reg(context)      (*(WORD*)&(context)->Ebx)
01748 #define CX_reg(context)      (*(WORD*)&(context)->Ecx)
01749 #define DX_reg(context)      (*(WORD*)&(context)->Edx)
01750 #define SI_reg(context)      (*(WORD*)&(context)->Esi)
01751 #define DI_reg(context)      (*(WORD*)&(context)->Edi)
01752 #define BP_reg(context)      (*(WORD*)&(context)->Ebp)
01753
01754 #define AL_reg(context)      (*(BYTE*)&(context)->Eax)
01755 #define AH_reg(context)      (*( (BYTE*)&(context)->Eax + 1))
01756 #define BL_reg(context)      (*(BYTE*)&(context)->Ebx)
01757 #define BH_reg(context)      (*( (BYTE*)&(context)->Ebx + 1))
01758 #define CL_reg(context)      (*(BYTE*)&(context)->Ecx)
01759 #define CH_reg(context)      (*( (BYTE*)&(context)->Ecx + 1))
01760 #define DL_reg(context)      (*(BYTE*)&(context)->Edx)
01761 #define DH_reg(context)      (*( (BYTE*)&(context)->Edx + 1))

```

```

01762
01763 #define SET_CFLAG(context) ((context)->EFlags |= 0x0001)
01764 #define RESET_CFLAG(context) ((context)->EFlags &= ~0x0001)
01765 #define SET_ZFLAG(context) ((context)->EFlags |= 0x0040)
01766 #define RESET_ZFLAG(context) ((context)->EFlags &= ~0x0040)
01767 #define ISV86(context) ((context)->EFlags & 0x00020000)
01768
01769
01770 /* Macros to retrieve the current context */
01771
01772 #ifdef NEED_UNDERSCORE_PREFIX
01773 # define __ASM_NAME(name) "_" name
01774 #else
01775 # define __ASM_NAME(name) name
01776 #endif
01777
01778 #ifdef NEED_TYPE_IN_DEF
01779 # define __ASM_FUNC(name) ".def " __ASM_NAME(name) "; .scl 2; .type 32; .endef"
01780 #else
01781 # define __ASM_FUNC(name) ".type " __ASM_NAME(name) ",@function"
01782 #endif
01783
01784 #ifdef __GNUC__
01785 # define __ASM_GLOBAL_FUNC(name,code) \
01786     __asm__( ".align 4\n\t" \
01787             ".globl " __ASM_NAME(#name) "\n\t" \
01788             __ASM_FUNC(#name) "\n" \
01789             __ASM_NAME(#name) ":\n\t" \
01790             code );
01791 #else /* __GNUC__ */
01792 # define __ASM_GLOBAL_FUNC(name,code) \
01793     void __asm_dummy_##name(void) { \
01794         asm( ".align 4\n\t" \
01795             ".globl " __ASM_NAME(#name) "\n\t" \
01796             __ASM_FUNC(#name) "\n" \
01797             __ASM_NAME(#name) ":\n\t" \
01798             code ); \
01799     }
01800 #endif /* __GNUC__ */
01801
01802 #ifdef __i386__
01803
01804 #define _DEFINE_REGS_ENTRYPOINT( name, fn, args ) \
01805     __ASM_GLOBAL_FUNC( name, \
01806         "call " __ASM_NAME("__wine_call_from_32_regs") "\n\t" \
01807         ".long " __ASM_NAME(#fn) "\n\t" \
01808         ".byte " #args ", " #args )
01809 #define DEFINE_REGS_ENTRYPOINT_0( name, fn ) \
01810     extern void WINAPI name(void); \
01811     _DEFINE_REGS_ENTRYPOINT( name, fn, 0 )
01812 #define DEFINE_REGS_ENTRYPOINT_1( name, fn, t1 ) \
01813     extern void WINAPI name( t1 a1 ); \
01814     _DEFINE_REGS_ENTRYPOINT( name, fn, 4 )
01815 #define DEFINE_REGS_ENTRYPOINT_2( name, fn, t1, t2 ) \
01816     extern void WINAPI name( t1 a1, t2 a2 ); \
01817     _DEFINE_REGS_ENTRYPOINT( name, fn, 8 )
01818 #define DEFINE_REGS_ENTRYPOINT_3( name, fn, t1, t2, t3 ) \
01819     extern void WINAPI name( t1 a1, t2 a2, t3 a3 ); \
01820     _DEFINE_REGS_ENTRYPOINT( name, fn, 12 )
01821 #define DEFINE_REGS_ENTRYPOINT_4( name, fn, t1, t2, t3, t4 ) \
01822     extern void WINAPI name( t1 a1, t2 a2, t3 a3, t4 a4 ); \
01823     _DEFINE_REGS_ENTRYPOINT( name, fn, 16 )
01824
01825 #endif /* __i386__ */
01826
01827 #ifdef __sparc__
01828 /* FIXME: use getcontext() to retrieve full context */
01829 #define _GET_CONTEXT \
01830     CONTEXT context; \
01831     do { memset(&context, 0, sizeof(CONTEXT)); \
01832         context.ContextFlags = CONTEXT_CONTROL; \
01833         context.pc = (DWORD)__builtin_return_address(0); \
01834     } while (0)
01835
01836 #define DEFINE_REGS_ENTRYPOINT_0( name, fn ) \
01837     void WINAPI name ( void ) \
01838     { _GET_CONTEXT; fn( &context ); }
01839 #define DEFINE_REGS_ENTRYPOINT_1( name, fn, t1 ) \
01840     void WINAPI name ( t1 a1 ) \
01841     { _GET_CONTEXT; fn( a1, &context ); }
01842 #define DEFINE_REGS_ENTRYPOINT_2( name, fn, t1, t2 ) \
01843     void WINAPI name ( t1 a1, t2 a2 ) \
01844     { _GET_CONTEXT; fn( a1, a2, &context ); }
01845 #define DEFINE_REGS_ENTRYPOINT_3( name, fn, t1, t2, t3 ) \
01846     void WINAPI name ( t1 a1, t2 a2, t3 a3 ) \
01847     { _GET_CONTEXT; fn( a1, a2, a3, &context ); }
01848 #define DEFINE_REGS_ENTRYPOINT_4( name, fn, t1, t2, t3, t4 ) \

```

```

01849 void WINAPI name ( t1 a1, t2 a2, t3 a3, t4 a4 ) \
01850 { _GET_CONTEXT; fn( a1, a2, a3, a4, &context ); }
01851
01852 #endif /* __sparc__ */
01853
01854 #ifdef __s390__
01855 /* FIXME: use getcontext() to retrieve full context */
01856 #define _GET_CONTEXT \
01857     CONTEXT context; \
01858     do { memset(&context, 0, sizeof(CONTEXT)); \
01859         context.ContextFlags = CONTEXT_CONTROL; \
01860         context.pc = (DWORD)__builtin_return_address(0); \
01861     } while (0)
01862
01863 #define DEFINE_REGS_ENTRYPOINT_0( name, fn ) \
01864     void WINAPI name ( void ) \
01865     { _GET_CONTEXT; fn( &context ); }
01866 #define DEFINE_REGS_ENTRYPOINT_1( name, fn, t1 ) \
01867     void WINAPI name ( t1 a1 ) \
01868     { _GET_CONTEXT; fn( a1, &context ); }
01869 #define DEFINE_REGS_ENTRYPOINT_2( name, fn, t1, t2 ) \
01870     void WINAPI name ( t1 a1, t2 a2 ) \
01871     { _GET_CONTEXT; fn( a1, a2, &context ); }
01872 #define DEFINE_REGS_ENTRYPOINT_3( name, fn, t1, t2, t3 ) \
01873     void WINAPI name ( t1 a1, t2 a2, t3 a3 ) \
01874     { _GET_CONTEXT; fn( a1, a2, a3, &context ); }
01875 #define DEFINE_REGS_ENTRYPOINT_4( name, fn, t1, t2, t3, t4 ) \
01876     void WINAPI name ( t1 a1, t2 a2, t3 a3, t4 a4 ) \
01877     { _GET_CONTEXT; fn( a1, a2, a3, a4, &context ); }
01878
01879 #endif /* __s390__ */
01880
01881 #ifdef __PPC__
01882
01883 /* FIXME: use getcontext() to retrieve full context */
01884 #define _GET_CONTEXT \
01885     CONTEXT context; \
01886     do { memset(&context, 0, sizeof(CONTEXT)); \
01887         context.ContextFlags = CONTEXT_CONTROL; \
01888     } while (0)
01889
01890 #define DEFINE_REGS_ENTRYPOINT_0( name, fn ) \
01891     void WINAPI name ( void ) \
01892     { _GET_CONTEXT; fn( &context ); }
01893 #define DEFINE_REGS_ENTRYPOINT_1( name, fn, t1 ) \
01894     void WINAPI name ( t1 a1 ) \
01895     { _GET_CONTEXT; fn( a1, &context ); }
01896 #define DEFINE_REGS_ENTRYPOINT_2( name, fn, t1, t2 ) \
01897     void WINAPI name ( t1 a1, t2 a2 ) \
01898     { _GET_CONTEXT; fn( a1, a2, &context ); }
01899 #define DEFINE_REGS_ENTRYPOINT_3( name, fn, t1, t2, t3 ) \
01900     void WINAPI name ( t1 a1, t2 a2, t3 a3 ) \
01901     { _GET_CONTEXT; fn( a1, a2, a3, &context ); }
01902 #define DEFINE_REGS_ENTRYPOINT_4( name, fn, t1, t2, t3, t4 ) \
01903     void WINAPI name ( t1 a1, t2 a2, t3 a3, t4 a4 ) \
01904     { _GET_CONTEXT; fn( a1, a2, a3, a4, &context ); }
01905
01906 #endif /* __PPC__ */
01907
01908 #ifndef DEFINE_REGS_ENTRYPOINT_0
01909 #error You need to define DEFINE_REGS_ENTRYPOINT macros for your CPU
01910 #endif
01911
01912 /* Constructor functions */
01913
01914 #ifdef __GNUC__
01915 # define DECL_GLOBAL_CONSTRUCTOR(func) \
01916     static void func(void) __attribute__((constructor)); \
01917     static void func(void)
01918 #else /* __GNUC__ */
01919 # ifdef __i386__
01920 #  define DECL_GLOBAL_CONSTRUCTOR(func) \
01921     static void __dummy_init_##func(void) { \
01922         asm(".section .init,\"ax\"\\n\\t\" \\
01923             \"call \" #func \"\\n\\t\" \\
01924             \"previous\"; } \\
01925         static void func(void)
01926 # else /* __i386__ */
01927 #  error You must define the DECL_GLOBAL_CONSTRUCTOR macro for your platform
01928 # endif
01929 # endif /* __GNUC__ */
01930
01931 /* Segment register access */
01932
01933 #ifdef __i386__
01934 # ifdef __GNUC__

```

```

01936 # define __DEFINE_GET_SEG(seg) \
01937     extern inline unsigned short __get_##seg(void) \
01938     { unsigned short res; __asm__("movw %% #seg", %w0 : "=r"(res)); return res; }
01939 # define __DEFINE_SET_SEG(seg) \
01940     extern inline void __set_##seg(int val) { __asm__("movw %w0,%% #seg : : "r" (val)); }
01941 # else /* __GNUC__ */
01942 # define __DEFINE_GET_SEG(seg) extern unsigned short __get_##seg(void);
01943 # define __DEFINE_SET_SEG(seg) extern void __set_##seg(unsigned int);
01944 # endif /* __GNUC__ */
01945 #else /* __i386__ */
01946 # define __DEFINE_GET_SEG(seg) inline static unsigned short __get_##seg(void) { return 0; }
01947 # define __DEFINE_SET_SEG(seg) /* nothing */
01948 #endif /* __i386__ */
01949
01950 __DEFINE_GET_SEG(cs)
01951 __DEFINE_GET_SEG(ds)
01952 __DEFINE_GET_SEG(es)
01953 __DEFINE_GET_SEG(fs)
01954 __DEFINE_GET_SEG(gs)
01955 __DEFINE_GET_SEG(ss)
01956 __DEFINE_SET_SEG(fs)
01957 __DEFINE_SET_SEG(gs)
01958 #undef __DEFINE_GET_SEG
01959 #undef __DEFINE_SET_SEG
01960
01961 #endif /* __WINE__ */
01962
01963
01964
01965 /*
01966 * Language IDs
01967 */
01968
01969 #define MAKELCID(l, s)      (MAKELONG(l, s))
01970
01971 #define MAKELANGID(p, s)    (((WORD)(s)<10) | (WORD)(p))
01972 #define PRIMARYLANGID(l)   ((WORD)(l) & 0x3ff)
01973 #define SUBLANGID(l)       ((WORD)(l) >> 10)
01974
01975 #define LANGIDFROMLCID(lcid) ((WORD)(lcid))
01976 #define SORTIDFROMLCID(lcid) ((WORD)(((DWORD)(lcid)) >> 16) & 0x0f)
01977
01978 #define LANG_SYSTEM_DEFAULT (MAKELANGID(LANG_NEUTRAL, SUBLANG_SYS_DEFAULT))
01979 #define LANG_USER_DEFAULT   (MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT))
01980 #define LOCALE_SYSTEM_DEFAULT (MAKELCID(LANG_SYSTEM_DEFAULT, SORT_DEFAULT))
01981 #define LOCALE_USER_DEFAULT  (MAKELCID(LANG_USER_DEFAULT, SORT_DEFAULT))
01982 #define LOCALE_NEUTRAL       (MAKELCID(MAKELANGID(LANG_NEUTRAL, SUBLANG_NEUTRAL), SORT_DEFAULT))
01983
01984 /* FIXME: are the symbolic names correct for LIDs: 0x17, 0x20, 0x28,
01985 * 0x2a, 0x2b, 0x2c, 0x2f, 0x30, 0x31, 0x32, 0x33, 0x34, 0x35,
01986 * 0x37, 0x39, 0x3a, 0x3b, 0x3c, 0x3e, 0x3f, 0x41, 0x43, 0x44,
01987 * 0x45, 0x46, 0x47, 0x48, 0x49, 0x4a, 0x4b, 0x4c, 0x4d, 0x4e,
01988 * 0x4f, 0x57
01989 */
01990 #define LANG_NEUTRAL          0x00
01991 #define LANG_AFIKAANS         0x36
01992 #define LANG_ALBANIAN         0x1c
01993 #define LANG_ARABIC           0x01
01994 #define LANG_ARMENIAN         0x2b
01995 #define LANG_ASSAMESE         0x4d
01996 #define LANG_AZERI            0x2c
01997 #define LANG_BASQUE           0x2d
01998 #define LANG_BENGALI          0x45
01999 #define LANG_BULGARIAN        0x02
02000 #define LANG_BYELORUSSIAN     0x23
02001 #define LANG_CATALAN          0x03
02002 #define LANG_CHINESE          0x04
02003 #define LANG_SERBO_CROATIAN   0x1a
02004 #define LANG_CROATIAN         LANG_SERBO_CROATIAN
02005 #define LANG_SERBIAN          LANG_SERBO_CROATIAN
02006 #define LANG_CZECH            0x05
02007 #define LANG_DANISH           0x06
02008 #define LANG_DUTCH            0x13
02009 #define LANG_ENGLISH          0x09
02010 #define LANG_ESTONIAN         0x25
02011 #define LANG_FAEROESE         0x38
02012 #define LANG_FARSI            0x29
02013 #define LANG_FINNISH          0x0b
02014 #define LANG_FRENCH           0x0c
02015 #define LANG_GAELIC           0x3c
02016 #define LANG_GEORGIAN         0x37
02017 #define LANG_GERMAN           0x07
02018 #define LANG_GREEK            0x08
02019 #define LANG_GUJARATI         0x47
02020 #define LANG_HEBREW           0x0d
02021 #define LANG_HINDI            0x39
02022 #define LANG_HUNGARIAN        0x0e

```

```

02023 #define LANG_ICELANDIC 0x0f
02024 #define LANG_INDONESIAN 0x21
02025 #define LANG_ITALIAN 0x10
02026 #define LANG_JAPANESE 0x11
02027 #define LANG_KANNADA 0x4b
02028 #define LANG_KAZAKH 0x3f
02029 #define LANG_KONKANI 0x57
02030 #define LANG_KOREAN 0x12
02031 #define LANG_LATVIAN 0x26
02032 #define LANG_LITHUANIAN 0x27
02033 #define LANG_MACEDONIAN 0x2f
02034 #define LANG_MALAY 0x3e
02035 #define LANG_MALAYALAM 0x4c
02036 #define LANG_MALTESE 0x3a
02037 #define LANG_MAORI 0x28
02038 #define LANG_MARATHI 0x4e
02039 #define LANG_NORWEGIAN 0x14
02040 #define LANG_ORIYA 0x48
02041 #define LANG_POLISH 0x15
02042 #define LANG_PORTUGUESE 0x16
02043 #define LANG_PUNJABI 0x46
02044 #define LANG_RHAETO_ROMANCE 0x17
02045 #define LANG_ROMANIAN 0x18
02046 #define LANG_RUSSIAN 0x19
02047 #define LANG_SAAMI 0x3b
02048 #define LANG_SANSKRIT 0x4f
02049 #define LANG_SLOVAK 0x1b
02050 #define LANG_SLOVENIAN 0x24
02051 #define LANG_SORBIAN 0x2e
02052 #define LANG_SPANISH 0x0a
02053 #define LANG_SUTU 0x30
02054 #define LANG_SWAHILI 0x41
02055 #define LANG_SWEDISH 0x1d
02056 #define LANG_TAMIL 0x49
02057 #define LANG_TATAR 0x44
02058 #define LANG_TELUGU 0x4a
02059 #define LANG_THAI 0x1e
02060 #define LANG_TSONGA 0x31
02061 #define LANG_TSWANA 0x32
02062 #define LANG_TURKISH 0x1f
02063 #define LANG_UKRAINIAN 0x22
02064 #define LANG_URDU 0x20
02065 #define LANG_UZBEK 0x43
02066 #define LANG_VENDA 0x33
02067 #define LANG_VIETNAMESE 0x2a
02068 #define LANG_XHOSA 0x34
02069 #define LANG_ZULU 0x35
02070 /* non standard; keep the number high enough (but < 0xff) */
02071 #define LANG_ESPERANTO 0x8f
02072 #define LANG_WALON 0x90
02073 #define LANG_CORNISH 0x91
02074 #define LANG_WELSH 0x92
02075 #define LANG_BRETON 0x93
02076
02077 /* Sublanguage definitions */
02078 #define SUBLANG_NEUTRAL 0x00 /* language neutral */
02079 #define SUBLANG_DEFAULT 0x01 /* user default */
02080 #define SUBLANG_SYS_DEFAULT 0x02 /* system default */
02081
02082 #define SUBLANG_ARABIC 0x01
02083 #define SUBLANG_ARABIC_SAUDI_ARABIA 0x01
02084 #define SUBLANG_ARABIC_IRAQ 0x02
02085 #define SUBLANG_ARABIC_EGYPT 0x03
02086 #define SUBLANG_ARABIC_LIBYA 0x04
02087 #define SUBLANG_ARABIC_ALGERIA 0x05
02088 #define SUBLANG_ARABIC_MOROCCO 0x06
02089 #define SUBLANG_ARABIC_TUNISIA 0x07
02090 #define SUBLANG_ARABIC_OMAN 0x08
02091 #define SUBLANG_ARABIC_YEMEN 0x09
02092 #define SUBLANG_ARABIC_SYRIA 0x0a
02093 #define SUBLANG_ARABIC_JORDAN 0x0b
02094 #define SUBLANG_ARABIC_LEBANON 0x0c
02095 #define SUBLANG_ARABIC_KUWAIT 0x0d
02096 #define SUBLANG_ARABIC_UAE 0x0e
02097 #define SUBLANG_ARABIC_BAHRAIN 0x0f
02098 #define SUBLANG_ARABIC_QATAR 0x10
02099 #define SUBLANG_CHINESE_TRADITIONAL 0x01
02100 #define SUBLANG_CHINESE_SIMPLIFIED 0x02
02101 #define SUBLANG_CHINESE_HONGKONG 0x03
02102 #define SUBLANG_CHINESE_SINGAPORE 0x04
02103 #define SUBLANG_CHINESE_MACAU 0x05
02104 #define SUBLANG_DUTCH 0x01
02105 #define SUBLANG_DUTCH_BELGIAN 0x02
02106 #define SUBLANG_DUTCH_SURINAM 0x03
02107 #define SUBLANG_ENGLISH_US 0x01
02108 #define SUBLANG_ENGLISH_UK 0x02
02109 #define SUBLANG_ENGLISH_AUS 0x03

```

```

02110 #define  SUBLANG_ENGLISH_CAN          0x04
02111 #define  SUBLANG_ENGLISH_NZ            0x05
02112 #define  SUBLANG_ENGLISH_EIRE          0x06
02113 #define  SUBLANG_ENGLISH_SAFRICA       0x07
02114 #define  SUBLANG_ENGLISH_JAMAICA       0x08
02115 #define  SUBLANG_ENGLISH_CARRIBEAN     0x09
02116 #define  SUBLANG_ENGLISH_BELIZE        0x0a
02117 #define  SUBLANG_ENGLISH_TRINIDAD      0x0b
02118 #define  SUBLANG_ENGLISH_ZIMBABWE      0x0c
02119 #define  SUBLANG_ENGLISH_PHILIPPINES   0x0d
02120 #define  SUBLANG_FRENCH                 0x01
02121 #define  SUBLANG_FRENCH_BELGIAN        0x02
02122 #define  SUBLANG_FRENCH_CANADIAN       0x03
02123 #define  SUBLANG_FRENCH_SWISS           0x04
02124 #define  SUBLANG_FRENCH_LUXEMBOURG     0x05
02125 #define  SUBLANG_FRENCH_MONACO         0x06
02126 #define  SUBLANG_GERMAN                 0x01
02127 #define  SUBLANG_GERMAN_SWISS          0x02
02128 #define  SUBLANG_GERMAN_AUSTRIAN       0x03
02129 #define  SUBLANG_GERMAN_LUXEMBOURG     0x04
02130 #define  SUBLANG_GERMAN_LIECHTENSTEIN  0x05
02131 #define  SUBLANG_ITALIAN                0x01
02132 #define  SUBLANG_ITALIAN_SWISS         0x02
02133 #define  SUBLANG_KOREAN                 0x01
02134 #define  SUBLANG_KOREAN_JOHAB          0x02
02135 #define  SUBLANG_NORWEGIAN_BOKMAL       0x01
02136 #define  SUBLANG_NORWEGIAN_NYNORSK     0x02
02137 #define  SUBLANG_PORTUGUESE            0x02
02138 #define  SUBLANG_PORTUGUESE_BRAZILIAN  0x01
02139 #define  SUBLANG_SPANISH                0x01
02140 #define  SUBLANG_SPANISH_MEXICAN        0x02
02141 #define  SUBLANG_SPANISH_MODERN         0x03
02142 #define  SUBLANG_SPANISH_GUATEMALA     0x04
02143 #define  SUBLANG_SPANISH_COSTARICA     0x05
02144 #define  SUBLANG_SPANISH_PANAMA         0x06
02145 #define  SUBLANG_SPANISH_DOMINICAN     0x07
02146 #define  SUBLANG_SPANISH_VENEZUELA     0x08
02147 #define  SUBLANG_SPANISH_COLOMBIA      0x09
02148 #define  SUBLANG_SPANISH_PERU          0x0a
02149 #define  SUBLANG_SPANISH_ARGENTINA      0x0b
02150 #define  SUBLANG_SPANISH_ECUADOR       0x0c
02151 #define  SUBLANG_SPANISH_CHILE          0x0d
02152 #define  SUBLANG_SPANISH_URUGUAY       0x0e
02153 #define  SUBLANG_SPANISH_PARAGUAY       0x0f
02154 #define  SUBLANG_SPANISH_BOLIVIA        0x10
02155 #define  SUBLANG_SPANISH_EL_SALVADOR   0x11
02156 #define  SUBLANG_SPANISH_HONDURAS       0x12
02157 #define  SUBLANG_SPANISH_NICARAGUA     0x13
02158 #define  SUBLANG_SPANISH_PUERTO_RICO   0x14
02159 /* FIXME: I don't know the symbolic names for those */
02160 #define  SUBLANG_ROMANIAN                0x01
02161 #define  SUBLANG_ROMANIAN_MOLDAVIA      0x02
02162 #define  SUBLANG_RUSSIAN                 0x01
02163 #define  SUBLANG_RUSSIAN_MOLDAVIA       0x02
02164 #define  SUBLANG_CROATIAN               0x01
02165 #define  SUBLANG_SERBIAN                 0x02
02166 #define  SUBLANG_SERBIAN_LATIN           0x03
02167 #define  SUBLANG_SWEDISH                 0x01
02168 #define  SUBLANG_SWEDISH_FINLAND         0x02
02169 #define  SUBLANG_LITHUANIAN              0x01
02170 #define  SUBLANG_LITHUANIAN_CLASSIC     0x02
02171 #define  SUBLANG_AZERI                   0x01
02172 #define  SUBLANG_AZERI_CYRILLIC          0x02
02173 #define  SUBLANG_GAELIC                  0x01
02174 #define  SUBLANG_GAELIC_SCOTTISH         0x02
02175 #define  SUBLANG_GAELIC_MANX             0x03
02176 #define  SUBLANG_MALAY                   0x01
02177 #define  SUBLANG_MALAY_BRUNEI_DARUSSALAM 0x02
02178 #define  SUBLANG_UZBEK                   0x01
02179 #define  SUBLANG_UZBEK_CYRILLIC          0x02
02180 #define  SUBLANG_URDU_PAKISTAN           0x01
02181
02182
02183
02184 /*
02185  * Sort definitions
02186  */
02187
02188 #define  SORT_DEFAULT                     0x0
02189 #define  SORT_JAPANESE_XJIS              0x0
02190 #define  SORT_JAPANESE_UNICODE           0x1
02191 #define  SORT_CHINESE_BIG5               0x0
02192 #define  SORT_CHINESE_UNICODE            0x1
02193 #define  SORT_KOREAN_KSC                  0x0
02194 #define  SORT_KOREAN_UNICODE             0x1
02195
02196

```

```
02197
02198 /*
02199  * Definitions for IsTextUnicode()
02200  */
02201
02202 #define IS_TEXT_UNICODE_ASCII16          0x0001
02203 #define IS_TEXT_UNICODE_STATISTICS       0x0002
02204 #define IS_TEXT_UNICODE_CONTROLS        0x0004
02205 #define IS_TEXT_UNICODE_SIGNATURE       0x0008
02206 #define IS_TEXT_UNICODE_UNICODE_MASK    0x000F
02207 #define IS_TEXT_UNICODE_REVERSE_ASCII16 0x0010
02208 #define IS_TEXT_UNICODE_REVERSE_STATISTICS 0x0020
02209 #define IS_TEXT_UNICODE_REVERSE_CONTROLS 0x0040
02210 #define IS_TEXT_UNICODE_REVERSE_SIGNATURE 0x0080
02211 #define IS_TEXT_UNICODE_REVERSE_MASK    0x00F0
02212 #define IS_TEXT_UNICODE_ILLEGAL_CHARS   0x0100
02213 #define IS_TEXT_UNICODE_ODD_LENGTH      0x0200
02214 #define IS_TEXT_UNICODE_DBCS_LEADBYTE  0x0400
02215 #define IS_TEXT_UNICODE_NOT_UNICODE_MASK 0x0F00
02216 #define IS_TEXT_UNICODE_NULL_BYTES      0x1000
02217 #define IS_TEXT_UNICODE_NOT_ASCII_MASK  0xF000
02218
02219
02220
02221 /*
02222  * Exception codes
02223  */
02224
02225 #define STATUS_SUCCESS                   0x00000000
02226 #define STATUS_WAIT_0                   0x00000000
02227 #define STATUS_ABANDONED_WAIT_0         0x00000080
02228 #define STATUS_ABANDONED_WAIT_63       0x000000BF
02229 #define STATUS_USER_APC                 0x000000C0
02230 #define STATUS_ALERTED                  0x00000101
02231 #define STATUS_TIMEOUT                  0x00000102
02232 #define STATUS_PENDING                  0x00000103
02233 #define STATUS_REPARSE                  0x00000104
02234 #define STATUS_MORE_ENTRIES             0x00000105
02235 #define STATUS_NOT_ALL_ASSIGNED         0x00000106
02236 #define STATUS_SOME_NOT_MAPPED         0x00000107
02237 #define STATUS_OPLOCK_BREAK_IN_PROGRESS 0x00000108
02238 #define STATUS_VOLUME_MOUNTED          0x00000109
02239 #define STATUS_RXACT_COMMITTED          0x0000010A
02240 #define STATUS_NOTIFY_CLEANUP           0x0000010B
02241 #define STATUS_NOTIFY_ENUM_DIR         0x0000010C
02242 #define STATUS_NO_QUOTAS_FOR_ACCOUNT    0x0000010D
02243 #define STATUS_PRIMARY_TRANSPORT_CONNECT_FAILED 0x0000010E
02244 #define STATUS_PAGE_FAULT_TRANSITION    0x00000110
02245 #define STATUS_PAGE_FAULT_DEMAND_ZERO  0x00000111
02246 #define STATUS_PAGE_FAULT_COPY_ON_WRITE 0x00000112
02247 #define STATUS_PAGE_FAULT_GUARD_PAGE    0x00000113
02248 #define STATUS_PAGE_FAULT_PAGING_FILE   0x00000114
02249 #define STATUS_CACHE_PAGE_LOCKED        0x00000115
02250 #define STATUS_CRASH_DUMP               0x00000116
02251 #define STATUS_BUFFER_ALL_ZEROS         0x00000117
02252 #define STATUS_REPARSE_OBJECT           0x00000118
02253
02254 #define STATUS_THREAD_WAS_SUSPENDED     0x40000001
02255 #define STATUS_WORKING_SET_LIMIT_RANGE  0x40000002
02256 #define STATUS_IMAGE_NOT_AT_BASE        0x40000003
02257 #define STATUS_RXACT_STATE_CREATED       0x40000004
02258 #define STATUS_SEGMENT_NOTIFICATION     0x40000005
02259 #define STATUS_LOCAL_USER_SESSION_KEY   0x40000006
02260 #define STATUS_BAD_CURRENT_DIRECTORY    0x40000007
02261 #define STATUS_SERIAL_MORE_WRITES       0x40000008
02262 #define STATUS_REGISTRY_RECOVERED        0x40000009
02263 #define STATUS_FT_READ_RECOVERY_FROM_BACKUP 0x4000000A
02264 #define STATUS_FT_WRITE_RECOVERY        0x4000000B
02265 #define STATUS_SERIAL_COUNTER_TIMEOUT    0x4000000C
02266 #define STATUS_NULL_LM_PASSWORD          0x4000000D
02267 #define STATUS_IMAGE_MACHINE_TYPE_MISMATCH 0x4000000E
02268 #define STATUS_RECEIVE_PARTIAL           0x4000000F
02269 #define STATUS_RECEIVE_EXPEDITED         0x40000010
02270 #define STATUS_RECEIVE_PARTIAL_EXPEDITED 0x40000011
02271 #define STATUS_EVENT_DONE               0x40000012
02272 #define STATUS_EVENT_PENDING            0x40000013
02273 #define STATUS_CHECKING_FILE_SYSTEM      0x40000014
02274 #define STATUS_FATAL_APP_EXIT            0x40000015
02275 #define STATUS_PREDEFINED_HANDLE         0x40000016
02276 #define STATUS_WAS_UNLOCKED             0x40000017
02277 #define STATUS_SERVICE_NOTIFICATION      0x40000018
02278 #define STATUS_WAS_LOCKED                0x40000019
02279 #define STATUS_LOG_HARD_ERROR            0x4000001A
02280 #define STATUS_ALREADY_WIN32             0x4000001B
02281 #define STATUS_WX86_UNSIMULATE           0x4000001C
02282 #define STATUS_WX86_CONTINUE            0x4000001D
02283 #define STATUS_WX86_SINGLE_STEP          0x4000001E
```

```
02284 #define STATUS_WX86_BREAKPOINT 0x4000001F
02285 #define STATUS_WX86_EXCEPTION_CONTINUE 0x40000020
02286 #define STATUS_WX86_EXCEPTION_LASTCHANCE 0x40000021
02287 #define STATUS_WX86_EXCEPTION_CHAIN 0x40000022
02288 #define STATUS_IMAGE_MACHINE_TYPE_MISMATCH_EXE 0x40000023
02289 #define STATUS_NO_YIELD_PERFORMED 0x40000024
02290 #define STATUS_TIMER_RESUME_IGNORED 0x40000025
02291
02292 #define STATUS_GUARD_PAGE_VIOLATION 0x80000001
02293 #define STATUS_DATATYPE_MISALIGNMENT 0x80000002
02294 #define STATUS_BREAKPOINT 0x80000003
02295 #define STATUS_SINGLE_STEP 0x80000004
02296 #define STATUS_BUFFER_OVERFLOW 0x80000005
02297 #define STATUS_NO_MORE_FILES 0x80000006
02298 #define STATUS_WAKE_SYSTEM_DEBUGGER 0x80000007
02299
02300 #define STATUS_HANDLES_CLOSED 0x8000000A
02301 #define STATUS_NO_INHERITANCE 0x8000000B
02302 #define STATUS_GUID_SUBSTITUTION_MADE 0x8000000C
02303 #define STATUS_PARTIAL_COPY 0x8000000D
02304 #define STATUS_DEVICE_PAPER_EMPTY 0x8000000E
02305 #define STATUS_DEVICE_POWERED_OFF 0x8000000F
02306 #define STATUS_DEVICE_OFF_LINE 0x80000010
02307 #define STATUS_DEVICE_BUSY 0x80000011
02308 #define STATUS_NO_MORE_EAS 0x80000012
02309 #define STATUS_INVALID_EA_NAME 0x80000013
02310 #define STATUS_EA_LIST_INCONSISTENT 0x80000014
02311 #define STATUS_INVALID_EA_FLAG 0x80000015
02312 #define STATUS_VERIFY_REQUIRED 0x80000016
02313 #define STATUS_EXTRANEIOUS_INFORMATION 0x80000017
02314 #define STATUS_RXACT_COMMIT_NECESSARY 0x80000018
02315 #define STATUS_NO_MORE_ENTRIES 0x8000001A
02316 #define STATUS_FILEMARK_DETECTED 0x8000001B
02317 #define STATUS_MEDIA_CHANGED 0x8000001C
02318 #define STATUS_BUS_RESET 0x8000001D
02319 #define STATUS_END_OF_MEDIA 0x8000001E
02320 #define STATUS_BEGINNING_OF_MEDIA 0x8000001F
02321 #define STATUS_MEDIA_CHECK 0x80000020
02322 #define STATUS_SETMARK_DETECTED 0x80000021
02323 #define STATUS_NO_DATA_DETECTED 0x80000022
02324 #define STATUS_REDIRECTOR_HAS_OPEN_HANDLES 0x80000023
02325 #define STATUS_SERVER_HAS_OPEN_HANDLES 0x80000024
02326 #define STATUS_ALREADY_DISCONNECTED 0x80000025
02327 #define STATUS_LONGJUMP 0x80000026
02328
02329 #define STATUS_UNSUCCESSFUL 0xC0000001
02330 #define STATUS_NOT_IMPLEMENTED 0xC0000002
02331 #define STATUS_INVALID_INFO_CLASS 0xC0000003
02332 #define STATUS_INFO_LENGTH_MISMATCH 0xC0000004
02333 #define STATUS_ACCESS_VIOLATION 0xC0000005
02334 #define STATUS_IN_PAGE_ERROR 0xC0000006
02335 #define STATUS_PAGEFILE_QUOTA 0xC0000007
02336 #define STATUS_INVALID_HANDLE 0xC0000008
02337 #define STATUS_BAD_INITIAL_STACK 0xC0000009
02338 #define STATUS_BAD_INITIAL_PC 0xC000000A
02339 #define STATUS_INVALID_CID 0xC000000B
02340 #define STATUS_TIMER_NOT_CANCELED 0xC000000C
02341 #define STATUS_INVALID_PARAMETER 0xC000000D
02342 #define STATUS_NO_SUCH_DEVICE 0xC000000E
02343 #define STATUS_NO_SUCH_FILE 0xC000000F
02344 #define STATUS_INVALID_DEVICE_REQUEST 0xC0000010
02345 #define STATUS_END_OF_FILE 0xC0000011
02346 #define STATUS_WRONG_VOLUME 0xC0000012
02347 #define STATUS_NO_MEDIA_IN_DEVICE 0xC0000013
02348 #define STATUS_UNRECOGNIZED_MEDIA 0xC0000014
02349 #define STATUS_NONEXISTENT_SECTOR 0xC0000015
02350 #define STATUS_MORE_PROCESSING_REQUIRED 0xC0000016
02351 #define STATUS_NO_MEMORY 0xC0000017
02352 #define STATUS_CONFLICTING_ADDRESSES 0xC0000018
02353 #define STATUS_NOT_MAPPED_VIEW 0xC0000019
02354 #define STATUS_UNABLE_TO_FREE_VM 0xC000001A
02355 #define STATUS_UNABLE_TO_DELETE_SECTION 0xC000001B
02356 #define STATUS_INVALID_SYSTEM_SERVICE 0xC000001C
02357 #define STATUS_ILLEGAL_INSTRUCTION 0xC000001D
02358 #define STATUS_INVALID_LOCK_SEQUENCE 0xC000001E
02359 #define STATUS_INVALID_VIEW_SIZE 0xC000001F
02360 #define STATUS_INVALID_FILE_FOR_SECTION 0xC0000020
02361 #define STATUS_ALREADY_COMMITTED 0xC0000021
02362 #define STATUS_ACCESS_DENIED 0xC0000022
02363 #define STATUS_BUFFER_TOO_SMALL 0xC0000023
02364 #define STATUS_OBJECT_TYPE_MISMATCH 0xC0000024
02365 #define STATUS_NONCONTINUABLE_EXCEPTION 0xC0000025
02366 #define STATUS_INVALID_DISPOSITION 0xC0000026
02367 #define STATUS_UNWIND 0xC0000027
02368 #define STATUS_BAD_STACK 0xC0000028
02369 #define STATUS_INVALID_UNWIND_TARGET 0xC0000029
02370 #define STATUS_NOT_LOCKED 0xC000002A
```

```
02371 #define STATUS_PARITY_ERROR 0xC000002B
02372 #define STATUS_UNABLE_TO_DECOMMIT_VM 0xC000002C
02373 #define STATUS_NOT_COMMITTED 0xC000002D
02374 #define STATUS_INVALID_PORT_ATTRIBUTES 0xC000002E
02375 #define STATUS_PORT_MESSAGE_TOO_LONG 0xC000002F
02376 #define STATUS_INVALID_PARAMETER_MIX 0xC0000030
02377 #define STATUS_INVALID_QUOTA_LOWER 0xC0000031
02378 #define STATUS_DISK_CORRUPT_ERROR 0xC0000032
02379 #define STATUS_OBJECT_NAME_INVALID 0xC0000033
02380 #define STATUS_OBJECT_NAME_NOT_FOUND 0xC0000034
02381 #define STATUS_OBJECT_NAME_COLLISION 0xC0000035
02382 #define STATUS_PORT_DISCONNECTED 0xC0000037
02383 #define STATUS_DEVICE_ALREADY_ATTACHED 0xC0000038
02384 #define STATUS_OBJECT_PATH_INVALID 0xC0000039
02385 #define STATUS_OBJECT_PATH_NOT_FOUND 0xC000003A
02386 #define STATUS_PATH_SYNTAX_BAD 0xC000003B
02387 #define STATUS_DATA_OVERRUN 0xC000003C
02388 #define STATUS_DATA_LATE_ERROR 0xC000003D
02389 #define STATUS_DATA_ERROR 0xC000003E
02390 #define STATUS_CRC_ERROR 0xC000003F
02391 #define STATUS_SECTION_TOO_BIG 0xC0000040
02392 #define STATUS_PORT_CONNECTION_REFUSED 0xC0000041
02393 #define STATUS_INVALID_PORT_HANDLE 0xC0000042
02394 #define STATUS_SHARING_VIOLATION 0xC0000043
02395 #define STATUS_QUOTA_EXCEEDED 0xC0000044
02396 #define STATUS_INVALID_PAGE_PROTECTION 0xC0000045
02397 #define STATUS_MUTANT_NOT_OWNED 0xC0000046
02398 #define STATUS_SEMAPHORE_LIMIT_EXCEEDED 0xC0000047
02399 #define STATUS_PORT_ALREADY_SET 0xC0000048
02400 #define STATUS_SECTION_NOT_IMAGE 0xC0000049
02401 #define STATUS_SUSPEND_COUNT_EXCEEDED 0xC000004A
02402 #define STATUS_THREAD_IS_TERMINATING 0xC000004B
02403 #define STATUS_BAD_WORKING_SET_LIMIT 0xC000004C
02404 #define STATUS_INCOMPATIBLE_FILE_MAP 0xC000004D
02405 #define STATUS_SECTION_PROTECTION 0xC000004E
02406 #define STATUS_EAS_NOT_SUPPORTED 0xC000004F
02407 #define STATUS_EA_TOO_LARGE 0xC0000050
02408 #define STATUS_NONEXISTENT_EA_ENTRY 0xC0000051
02409 #define STATUS_NO_EAS_ON_FILE 0xC0000052
02410 #define STATUS_EA_CORRUPT_ERROR 0xC0000053
02411 #define STATUS_LOCK_NOT_GRANTED 0xC0000054 /* FIXME: not sure */
02412 #define STATUS_FILE_LOCK_CONFLICT 0xC0000055 /* FIXME: not sure */
02413 #define STATUS_DELETE_PENDING 0xC0000056
02414 #define STATUS_CTL_FILE_NOT_SUPPORTED 0xC0000057
02415 #define STATUS_UNKNOWN_REVISION 0xC0000058
02416 #define STATUS_REVISION_MISMATCH 0xC0000059
02417 #define STATUS_INVALID_OWNER 0xC000005A
02418 #define STATUS_INVALID_PRIMARY_GROUP 0xC000005B
02419 #define STATUS_NO_IMPERSONATION_TOKEN 0xC000005C
02420 #define STATUS_CANT_DISABLE_MANDATORY 0xC000005D
02421 #define STATUS_NO_LOGON_SERVERS 0xC000005E
02422 #define STATUS_NO_SUCH_LOGON_SESSION 0xC000005F
02423 #define STATUS_NO_SUCH_PRIVILEGE 0xC0000060
02424 #define STATUS_PRIVILEGE_NOT_HELD 0xC0000061
02425 #define STATUS_INVALID_ACCOUNT_NAME 0xC0000062
02426 #define STATUS_USER_EXISTS 0xC0000063
02427 #define STATUS_NO_SUCH_USER 0xC0000064
02428 #define STATUS_GROUP_EXISTS 0xC0000065
02429 #define STATUS_NO_SUCH_GROUP 0xC0000066
02430 #define STATUS_MEMBER_IN_GROUP 0xC0000067
02431 #define STATUS_MEMBER_NOT_IN_GROUP 0xC0000068
02432 #define STATUS_LAST_ADMIN 0xC0000069
02433 #define STATUS_WRONG_PASSWORD 0xC000006A
02434 #define STATUS_ILL_FORMED_PASSWORD 0xC000006B
02435 #define STATUS_PASSWORD_RESTRICTION 0xC000006C
02436 #define STATUS_LOGON_FAILURE 0xC000006D
02437 #define STATUS_ACCOUNT_RESTRICTION 0xC000006E
02438 #define STATUS_INVALID_LOGON_HOURS 0xC000006F
02439 #define STATUS_INVALID_WORKSTATION 0xC0000070
02440 #define STATUS_PASSWORD_EXPIRED 0xC0000071
02441 #define STATUS_ACCOUNT_DISABLED 0xC0000072
02442 #define STATUS_NONE_MAPPED 0xC0000073
02443 #define STATUS_TOO_MANY_LUIDS_REQUESTED 0xC0000074
02444 #define STATUS_LUIDS_EXHAUSTED 0xC0000075
02445 #define STATUS_INVALID_SUB_AUTHORITY 0xC0000076
02446 #define STATUS_INVALID_ACL 0xC0000077
02447 #define STATUS_INVALID_SID 0xC0000078
02448 #define STATUS_INVALID_SECURITY_DESCR 0xC0000079
02449 #define STATUS_PROCEDURE_NOT_FOUND 0xC000007A
02450 #define STATUS_INVALID_IMAGE_FORMAT 0xC000007B
02451 #define STATUS_NO_TOKEN 0xC000007C
02452 #define STATUS_BAD_INHERITANCE_ACL 0xC000007D
02453 #define STATUS_RANGE_NOT_LOCKED 0xC000007E
02454 #define STATUS_DISK_FULL 0xC000007F
02455 #define STATUS_SERVER_DISABLED 0xC0000080
02456 #define STATUS_SERVER_NOT_DISABLED 0xC0000081
02457 #define STATUS_TOO_MANY_GUIDS_REQUESTED 0xC0000082
```

```
02458 #define STATUS_GUIDS_EXHAUSTED 0xC0000083
02459 #define STATUS_INVALID_ID_AUTHORITY 0xC0000084
02460 #define STATUS_AGENTS_EXHAUSTED 0xC0000085
02461 #define STATUS_INVALID_VOLUME_LABEL 0xC0000086
02462 #define STATUS_SECTION_NOT_EXTENDED 0xC0000087
02463 #define STATUS_NOT_MAPPED_DATA 0xC0000088
02464 #define STATUS_RESOURCE_DATA_NOT_FOUND 0xC0000089
02465 #define STATUS_RESOURCE_TYPE_NOT_FOUND 0xC000008A
02466 #define STATUS_RESOURCE_NAME_NOT_FOUND 0xC000008B
02467 #define STATUS_ARRAY_BOUNDS_EXCEEDED 0xC000008C
02468 #define STATUS_FLOAT_DENORMAL_OPERAND 0xC000008D
02469 #define STATUS_FLOAT_DIVIDE_BY_ZERO 0xC000008E
02470 #define STATUS_FLOAT_INEXACT_RESULT 0xC000008F
02471 #define STATUS_FLOAT_INVALID_OPERATION 0xC0000090
02472 #define STATUS_FLOAT_OVERFLOW 0xC0000091
02473 #define STATUS_FLOAT_STACK_CHECK 0xC0000092
02474 #define STATUS_FLOAT_UNDERFLOW 0xC0000093
02475 #define STATUS_INTEGER_DIVIDE_BY_ZERO 0xC0000094
02476 #define STATUS_INTEGER_OVERFLOW 0xC0000095
02477 #define STATUS_PRIVILEGED_INSTRUCTION 0xC0000096
02478 #define STATUS_TOO_MANY_PAGING_FILES 0xC0000097
02479 #define STATUS_FILE_INVALID 0xC0000098
02480 #define STATUS_ALLOTTED_SPACE_EXCEEDED 0xC0000099
02481 #define STATUS_INSUFFICIENT_RESOURCES 0xC000009A
02482 #define STATUS_DFS_EXIT_PATH_FOUND 0xC000009B
02483 #define STATUS_DEVICE_DATA_ERROR 0xC000009C
02484 #define STATUS_DEVICE_NOT_CONNECTED 0xC000009D
02485 #define STATUS_DEVICE_POWER_FAILURE 0xC000009E
02486 #define STATUS_FREE_VM_NOT_AT_BASE 0xC000009F
02487 #define STATUS_MEMORY_NOT_ALLOCATED 0xC00000A0
02488 #define STATUS_WORKING_SET_QUOTA 0xC00000A1
02489 #define STATUS_MEDIA_WRITE_PROTECTED 0xC00000A2
02490 #define STATUS_DEVICE_NOT_READY 0xC00000A3
02491 #define STATUS_INVALID_GROUP_ATTRIBUTES 0xC00000A4
02492 #define STATUS_BAD_IMPERSONATION_LEVEL 0xC00000A5
02493 #define STATUS_CANT_OPEN_ANONYMOUS 0xC00000A6
02494 #define STATUS_BAD_VALIDATION_CLASS 0xC00000A7
02495 #define STATUS_BAD_TOKEN_TYPE 0xC00000A8
02496 #define STATUS_BAD_MASTER_BOOT_RECORD 0xC00000A9
02497 #define STATUS_INSTRUCTION_MISALIGNMENT 0xC00000AA
02498 #define STATUS_INSTANCE_NOT_AVAILABLE 0xC00000AB
02499 #define STATUS_PIPE_NOT_AVAILABLE 0xC00000AC
02500 #define STATUS_INVALID_PIPE_STATE 0xC00000AD
02501 #define STATUS_PIPE_BUSY 0xC00000AE
02502 #define STATUS_ILLEGAL_FUNCTION 0xC00000AF
02503 #define STATUS_PIPE_DISCONNECTED 0xC00000B0
02504 #define STATUS_PIPE_CLOSING 0xC00000B1
02505 #define STATUS_PIPE_CONNECTED 0xC00000B2
02506 #define STATUS_PIPE_LISTENING 0xC00000B3
02507 #define STATUS_INVALID_READ_MODE 0xC00000B4
02508 #define STATUS_IO_TIMEOUT 0xC00000B5
02509 #define STATUS_FILE_FORCED_CLOSED 0xC00000B6
02510 #define STATUS_PROFILING_NOT_STARTED 0xC00000B7
02511 #define STATUS_PROFILING_NOT_STOPPED 0xC00000B8
02512 #define STATUS_COULD_NOT_INTERPRET 0xC00000B9
02513 #define STATUS_FILE_IS_A_DIRECTORY 0xC00000BA
02514 #define STATUS_NOT_SUPPORTED 0xC00000BB
02515 #define STATUS_REMOTE_NOT_LISTENING 0xC00000BC
02516 #define STATUS_DUPLICATE_NAME 0xC00000BD
02517 #define STATUS_BAD_NETWORK_PATH 0xC00000BE
02518 #define STATUS_NETWORK_BUSY 0xC00000BF
02519 #define STATUS_DEVICE_DOES_NOT_EXIST 0xC00000C0
02520 #define STATUS_TOO_MANY_COMMANDS 0xC00000C1
02521 #define STATUS_ADAPTER_HARDWARE_ERROR 0xC00000C2
02522 #define STATUS_INVALID_NETWORK_RESPONSE 0xC00000C3
02523 #define STATUS_UNEXPECTED_NETWORK_ERROR 0xC00000C4
02524 #define STATUS_BAD_REMOTE_ADAPTER 0xC00000C5
02525 #define STATUS_PRINT_QUEUE_FULL 0xC00000C6
02526 #define STATUS_NO_SPOOL_SPACE 0xC00000C7
02527 #define STATUS_PRINT_CANCELLED 0xC00000C8
02528 #define STATUS_NETWORK_NAME_DELETED 0xC00000C9
02529 #define STATUS_NETWORK_ACCESS_DENIED 0xC00000CA
02530 #define STATUS_BAD_DEVICE_TYPE 0xC00000CB
02531 #define STATUS_BAD_NETWORK_NAME 0xC00000CC
02532 #define STATUS_TOO_MANY_NAMES 0xC00000CD
02533 #define STATUS_TOO_MANY_SESSIONS 0xC00000CE
02534 #define STATUS_SHARING_PAUSED 0xC00000CF
02535 #define STATUS_REQUEST_NOT_ACCEPTED 0xC00000D0
02536 #define STATUS_REDIRECTOR_PAUSED 0xC00000D1
02537 #define STATUS_NET_WRITE_FAULT 0xC00000D2
02538 #define STATUS_PROFILING_AT_LIMIT 0xC00000D3
02539 #define STATUS_NOT_SAME_DEVICE 0xC00000D4
02540 #define STATUS_FILE_RENAMED 0xC00000D5
02541 #define STATUS_VIRTUAL_CIRCUIT_CLOSED 0xC00000D6
02542 #define STATUS_NO_SECURITY_ON_OBJECT 0xC00000D7
02543 #define STATUS_CANT_WAIT 0xC00000D8
02544 #define STATUS_PIPE_EMPTY 0xC00000D9
```

```
02545 #define STATUS_CANT_ACCESS_DOMAIN_INFO 0xC00000DA
02546 #define STATUS_CANT_TERMINATE_SELF 0xC00000DB
02547 #define STATUS_INVALID_SERVER_STATE 0xC00000DC
02548 #define STATUS_INVALID_DOMAIN_STATE 0xC00000DD
02549 #define STATUS_INVALID_DOMAIN_ROLE 0xC00000DE
02550 #define STATUS_NO_SUCH_DOMAIN 0xC00000DF
02551 #define STATUS_DOMAIN_EXISTS 0xC00000E0
02552 #define STATUS_DOMAIN_LIMIT_EXCEEDED 0xC00000E1
02553 #define STATUS_OPLOCK_NOT_GRANTED 0xC00000E2
02554 #define STATUS_INVALID_OPLOCK_PROTOCOL 0xC00000E3
02555 #define STATUS_INTERNAL_DB_CORRUPTION 0xC00000E4
02556 #define STATUS_INTERNAL_ERROR 0xC00000E5
02557 #define STATUS_GENERIC_NOT_MAPPED 0xC00000E6
02558 #define STATUS_BAD_DESCRIPTOR_FORMAT 0xC00000E7
02559 #define STATUS_INVALID_USER_BUFFER 0xC00000E8
02560 #define STATUS_UNEXPECTED_IO_ERROR 0xC00000E9
02561 #define STATUS_UNEXPECTED_MM_CREATE_ERR 0xC00000EA
02562 #define STATUS_UNEXPECTED_MM_MAP_ERROR 0xC00000EB
02563 #define STATUS_UNEXPECTED_MM_EXTEND_ERR 0xC00000EC
02564 #define STATUS_NOT_LOGON_PROCESS 0xC00000ED
02565 #define STATUS_LOGON_SESSION_EXISTS 0xC00000EE
02566 #define STATUS_INVALID_PARAMETER_1 0xC00000EF
02567 #define STATUS_INVALID_PARAMETER_2 0xC00000F0
02568 #define STATUS_INVALID_PARAMETER_3 0xC00000F1
02569 #define STATUS_INVALID_PARAMETER_4 0xC00000F2
02570 #define STATUS_INVALID_PARAMETER_5 0xC00000F3
02571 #define STATUS_INVALID_PARAMETER_6 0xC00000F4
02572 #define STATUS_INVALID_PARAMETER_7 0xC00000F5
02573 #define STATUS_INVALID_PARAMETER_8 0xC00000F6
02574 #define STATUS_INVALID_PARAMETER_9 0xC00000F7
02575 #define STATUS_INVALID_PARAMETER_10 0xC00000F8
02576 #define STATUS_INVALID_PARAMETER_11 0xC00000F9
02577 #define STATUS_INVALID_PARAMETER_12 0xC00000FA
02578 #define STATUS_REDIRECTOR_NOT_STARTED 0xC00000FB
02579 #define STATUS_REDIRECTOR_STARTED 0xC00000FC
02580 #define STATUS_STACK_OVERFLOW 0xC00000FD
02581 #define STATUS_BAD_FUNCTION_TABLE 0xC00000FF
02582 #define STATUS_VARIABLE_NOT_FOUND 0xC0000100
02583 #define STATUS_DIRECTORY_NOT_EMPTY 0xC0000101
02584 #define STATUS_FILE_CORRUPT_ERROR 0xC0000102
02585 #define STATUS_NOT_A_DIRECTORY 0xC0000103
02586 #define STATUS_BAD_LOGON_SESSION_STATE 0xC0000104
02587 #define STATUS_LOGON_SESSION_COLLISION 0xC0000105
02588 #define STATUS_NAME_TOO_LONG 0xC0000106
02589 #define STATUS_FILES_OPEN 0xC0000107
02590 #define STATUS_CONNECTION_IN_USE 0xC0000108
02591 #define STATUS_MESSAGE_NOT_FOUND 0xC0000109
02592 #define STATUS_PROCESS_IS_TERMINATING 0xC000010A
02593 #define STATUS_INVALID_LOGON_TYPE 0xC000010B
02594 #define STATUS_NO_GUID_TRANSLATION 0xC000010C
02595 #define STATUS_CANNOT_IMPERSONATE 0xC000010D
02596 #define STATUS_IMAGE_ALREADY_LOADED 0xC000010E
02597 #define STATUS_ABIOS_NOT_PRESENT 0xC000010F
02598 #define STATUS_ABIOS_LID_NOT_EXIST 0xC0000110
02599 #define STATUS_ABIOS_LID_ALREADY_OWNED 0xC0000111
02600 #define STATUS_ABIOS_NOT_LID_OWNER 0xC0000112
02601 #define STATUS_ABIOS_INVALID_COMMAND 0xC0000113
02602 #define STATUS_ABIOS_INVALID_LID 0xC0000114
02603 #define STATUS_ABIOS_SELECTOR_NOT_AVAILABLE 0xC0000115
02604 #define STATUS_ABIOS_INVALID_SELECTOR 0xC0000116
02605 #define STATUS_NO_LDT 0xC0000117
02606 #define STATUS_INVALID_LDT_SIZE 0xC0000118
02607 #define STATUS_INVALID_LDT_OFFSET 0xC0000119
02608 #define STATUS_INVALID_LDT_DESCRIPTOR 0xC000011A
02609 #define STATUS_INVALID_IMAGE_NE_FORMAT 0xC000011B
02610 #define STATUS_RXACT_INVALID_STATE 0xC000011C
02611 #define STATUS_RXACT_COMMIT_FAILURE 0xC000011D
02612 #define STATUS_MAPPED_FILE_SIZE_ZERO 0xC000011E
02613 #define STATUS_TOO_MANY_OPENED_FILES 0xC000011F
02614 #define STATUS_CANCELLED 0xC0000120
02615 #define STATUS_CANNOT_DELETE 0xC0000121
02616 #define STATUS_INVALID_COMPUTER_NAME 0xC0000122
02617 #define STATUS_FILE_DELETED 0xC0000123
02618 #define STATUS_SPECIAL_ACCOUNT 0xC0000124
02619 #define STATUS_SPECIAL_GROUP 0xC0000125
02620 #define STATUS_SPECIAL_USER 0xC0000126
02621 #define STATUS_MEMBERS_PRIMARY_GROUP 0xC0000127
02622 #define STATUS_FILE_CLOSED 0xC0000128
02623 #define STATUS_TOO_MANY_THREADS 0xC0000129
02624 #define STATUS_THREAD_NOT_IN_PROCESS 0xC000012A
02625 #define STATUS_TOKEN_ALREADY_IN_USE 0xC000012B
02626 #define STATUS_PAGEFILE_QUOTA_EXCEEDED 0xC000012C
02627 #define STATUS_COMMITMENT_LIMIT 0xC000012D
02628 #define STATUS_INVALID_IMAGE_LE_FORMAT 0xC000012E
02629 #define STATUS_INVALID_IMAGE_NOT_MZ 0xC000012F
02630 #define STATUS_INVALID_IMAGE_PROTECT 0xC0000130
02631 #define STATUS_INVALID_IMAGE_WIN_16 0xC0000131
```

```
02632 #define STATUS_LOGON_SERVER_CONFLICT 0xC0000132
02633 #define STATUS_TIME_DIFFERENCE_AT_DC 0xC0000133
02634 #define STATUS_SYNCHRONIZATION_REQUIRED 0xC0000134
02635 #define STATUS_DLL_NOT_FOUND 0xC0000135
02636 #define STATUS_OPEN_FAILED 0xC0000136
02637 #define STATUS_IO_PRIVILEGE_FAILED 0xC0000137
02638 #define STATUS_ORDINAL_NOT_FOUND 0xC0000138
02639 #define STATUS_ENTRYPOINT_NOT_FOUND 0xC0000139
02640 #define STATUS_CONTROL_C_EXIT 0xC000013A
02641 #define STATUS_LOCAL_DISCONNECT 0xC000013B
02642 #define STATUS_REMOTE_DISCONNECT 0xC000013C
02643 #define STATUS_REMOTE_RESOURCES 0xC000013D
02644 #define STATUS_LINK_FAILED 0xC000013E
02645 #define STATUS_LINK_TIMEOUT 0xC000013F
02646 #define STATUS_INVALID_CONNECTION 0xC0000140
02647 #define STATUS_INVALID_ADDRESS 0xC0000141
02648 #define STATUS_DLL_INIT_FAILED 0xC0000142
02649 #define STATUS_MISSING_SYSTEMFILE 0xC0000143
02650 #define STATUS_UNHANDLED_EXCEPTION 0xC0000144
02651 #define STATUS_APP_INIT_FAILURE 0xC0000145
02652 #define STATUS_PAGEFILE_CREATE_FAILED 0xC0000146
02653 #define STATUS_NO_PAGEFILE 0xC0000147
02654 #define STATUS_INVALID_LEVEL 0xC0000148
02655 #define STATUS_WRONG_PASSWORD_CORE 0xC0000149
02656 #define STATUS_ILLEGAL_FLOAT_CONTEXT 0xC000014A
02657 #define STATUS_PIPE_BROKEN 0xC000014B
02658 #define STATUS_REGISTRY_CORRUPT 0xC000014C
02659 #define STATUS_REGISTRY_IO_FAILED 0xC000014D
02660 #define STATUS_NO_EVENT_PAIR 0xC000014E
02661 #define STATUS_UNRECOGNIZED_VOLUME 0xC000014F
02662 #define STATUS_SERIAL_NO_DEVICE_INITED 0xC0000150
02663 #define STATUS_NO_SUCH_ALIAS 0xC0000151
02664 #define STATUS_MEMBER_NOT_IN_ALIAS 0xC0000152
02665 #define STATUS_MEMBER_IN_ALIAS 0xC0000153
02666 #define STATUS_ALIAS_EXISTS 0xC0000154
02667 #define STATUS_LOGON_NOT_GRANTED 0xC0000155
02668 #define STATUS_TOO_MANY_SECRETS 0xC0000156
02669 #define STATUS_SECRET_TOO_LONG 0xC0000157
02670 #define STATUS_INTERNAL_DB_ERROR 0xC0000158
02671 #define STATUS_FULLSCREEN_MODE 0xC0000159
02672 #define STATUS_TOO_MANY_CONTEXT_IDS 0xC000015A
02673 #define STATUS_LOGON_TYPE_NOT_GRANTED 0xC000015B
02674 #define STATUS_NOT_REGISTRY_FILE 0xC000015C
02675 #define STATUS_NT_CROSS_ENCRYPTION_REQUIRED 0xC000015D
02676 #define STATUS_DOMAIN_CTRLR_CONFIG_ERROR 0xC000015E
02677 #define STATUS_FT_MISSING_MEMBER 0xC000015F
02678 #define STATUS_ILL_FORMED_SERVICE_ENTRY 0xC0000160
02679 #define STATUS_ILLEGAL_CHARACTER 0xC0000161
02680 #define STATUS_UNMAPPABLE_CHARACTER 0xC0000162
02681 #define STATUS_UNDEFINED_CHARACTER 0xC0000163
02682 #define STATUS_FLOPPY_VOLUME 0xC0000164
02683 #define STATUS_FLOPPY_ID_MARK_NOT_FOUND 0xC0000165
02684 #define STATUS_FLOPPY_WRONG_CYLINDER 0xC0000166
02685 #define STATUS_FLOPPY_UNKNOWN_ERROR 0xC0000167
02686 #define STATUS_FLOPPY_BAD_REGISTERS 0xC0000168
02687 #define STATUS_DISK_RECALIBRATE_FAILED 0xC0000169
02688 #define STATUS_DISK_OPERATION_FAILED 0xC000016A
02689 #define STATUS_DISK_RESET_FAILED 0xC000016B
02690 #define STATUS_SHARED_IRQ_BUSY 0xC000016C
02691 #define STATUS_FT_ORPHANING 0xC000016D
02692 #define STATUS_BIOS_FAILED_TO_CONNECT_INTERRUPT 0xC000016E
02693
02694 #define STATUS_PARTITION_FAILURE 0xC0000172
02695 #define STATUS_INVALID_BLOCK_LENGTH 0xC0000173
02696 #define STATUS_DEVICE_NOT_PARTITIONED 0xC0000174
02697 #define STATUS_UNABLE_TO_LOCK_MEDIA 0xC0000175
02698 #define STATUS_UNABLE_TO_UNLOAD_MEDIA 0xC0000176
02699 #define STATUS_EOM_OVERFLOW 0xC0000177
02700 #define STATUS_NO_MEDIA 0xC0000178
02701 #define STATUS_NO_SUCH_MEMBER 0xC000017A
02702 #define STATUS_INVALID_MEMBER 0xC000017B
02703 #define STATUS_KEY_DELETED 0xC000017C
02704 #define STATUS_NO_LOG_SPACE 0xC000017D
02705 #define STATUS_TOO_MANY_SIDS 0xC000017E
02706 #define STATUS_LM_CROSS_ENCRYPTION_REQUIRED 0xC000017F
02707 #define STATUS_KEY_HAS_CHILDREN 0xC0000180
02708 #define STATUS_CHILD_MUST_BE_VOLATILE 0xC0000181
02709 #define STATUS_DEVICE_CONFIGURATION_ERROR 0xC0000182
02710 #define STATUS_DRIVER_INTERNAL_ERROR 0xC0000183
02711 #define STATUS_INVALID_DEVICE_STATE 0xC0000184
02712 #define STATUS_IO_DEVICE_ERROR 0xC0000185
02713 #define STATUS_DEVICE_PROTOCOL_ERROR 0xC0000186
02714 #define STATUS_BACKUP_CONTROLLER 0xC0000187
02715 #define STATUS_LOG_FILE_FULL 0xC0000188
02716 #define STATUS_TOO_LATE 0xC0000189
02717 #define STATUS_NO_TRUST_LSA_SECRET 0xC000018A
02718 #define STATUS_NO_TRUST_SAM_ACCOUNT 0xC000018B
```

```
02719 #define STATUS_TRUSTED_DOMAIN_FAILURE 0xC000018C
02720 #define STATUS_TRUSTED_RELATIONSHIP_FAILURE 0xC000018D
02721 #define STATUS_EVENTLOG_FILE_CORRUPT 0xC000018E
02722 #define STATUS_EVENTLOG_CANT_START 0xC000018F
02723 #define STATUS_TRUST_FAILURE 0xC0000190
02724 #define STATUS_MUTANT_LIMIT_EXCEEDED 0xC0000191
02725 #define STATUS_NETLOGON_NOT_STARTED 0xC0000192
02726 #define STATUS_ACCOUNT_EXPIRED 0xC0000193
02727 #define STATUS_POSSIBLE_DEADLOCK 0xC0000194
02728 #define STATUS_NETWORK_CREDENTIAL_CONFLICT 0xC0000195
02729 #define STATUS_REMOTE_SESSION_LIMIT 0xC0000196
02730 #define STATUS_EVENTLOG_FILE_CHANGED 0xC0000197
02731 #define STATUS_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT 0xC0000198
02732 #define STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT 0xC0000199
02733 #define STATUS_NOLOGON_SERVER_TRUST_ACCOUNT 0xC000019A
02734 #define STATUS_DOMAIN_TRUST_INCONSISTENT 0xC000019B
02735 #define STATUS_FS_DRIVER_REQUIRED 0xC000019C
02736
02737 #define STATUS_NO_USER_SESSION_KEY 0xC0000202
02738 #define STATUS_USER_SESSION_DELETED 0xC0000203
02739 #define STATUS_RESOURCE_LANG_NOT_FOUND 0xC0000204
02740 #define STATUS_INSUFF_SERVER_RESOURCES 0xC0000205
02741 #define STATUS_INVALID_BUFFER_SIZE 0xC0000206
02742 #define STATUS_INVALID_ADDRESS_COMPONENT 0xC0000207
02743 #define STATUS_INVALID_ADDRESS_WILDCARD 0xC0000208
02744 #define STATUS_TOO_MANY_ADDRESSES 0xC0000209
02745 #define STATUS_ADDRESS_ALREADY_EXISTS 0xC000020A
02746 #define STATUS_ADDRESS_CLOSED 0xC000020B
02747 #define STATUS_CONNECTION_DISCONNECTED 0xC000020C
02748 #define STATUS_CONNECTION_RESET 0xC000020D
02749 #define STATUS_TOO_MANY_NODES 0xC000020E
02750 #define STATUS_TRANSACTION_ABORTED 0xC000020F
02751 #define STATUS_TRANSACTION_TIMED_OUT 0xC0000210
02752 #define STATUS_TRANSACTION_NO_RELEASE 0xC0000211
02753 #define STATUS_TRANSACTION_NO_MATCH 0xC0000212
02754 #define STATUS_TRANSACTION_RESPONDED 0xC0000213
02755 #define STATUS_TRANSACTION_INVALID_ID 0xC0000214
02756 #define STATUS_TRANSACTION_INVALID_TYPE 0xC0000215
02757 #define STATUS_NOT_SERVER_SESSION 0xC0000216
02758 #define STATUS_NOT_CLIENT_SESSION 0xC0000217
02759 #define STATUS_CANNOT_LOAD_REGISTRY_FILE 0xC0000218
02760 #define STATUS_DEBUG_ATTACH_FAILED 0xC0000219
02761 #define STATUS_SYSTEM_PROCESS_TERMINATED 0xC000021A
02762 #define STATUS_DATA_NOT_ACCEPTED 0xC000021B
02763 #define STATUS_NO_BROWSER_SERVERS_FOUND 0xC000021C
02764 #define STATUS_VDM_HARD_ERROR 0xC000021D
02765 #define STATUS_DRIVER_CANCEL_TIMEOUT 0xC000021E
02766 #define STATUS_REPLY_MESSAGE_MISMATCH 0xC000021F
02767 #define STATUS_MAPPED_ALIGNMENT 0xC0000220
02768 #define STATUS_IMAGE_CHECKSUM_MISMATCH 0xC0000221
02769 #define STATUS_LOST_WRITEBEHIND_DATA 0xC0000222
02770 #define STATUS_CLIENT_SERVER_PARAMETERS_INVALID 0xC0000223
02771 #define STATUS_PASSWORD_MUST_CHANGE 0xC0000224
02772 #define STATUS_NOT_FOUND 0xC0000225
02773 #define STATUS_NOT_TINY_STREAM 0xC0000226
02774 #define STATUS_RECOVERY_FAILURE 0xC0000227
02775 #define STATUS_STACK_OVERFLOW_READ 0xC0000228
02776 #define STATUS_FAIL_CHECK 0xC0000229
02777 #define STATUS_DUPLICATE_OBJECTID 0xC000022A
02778 #define STATUS_OBJECTID_EXISTS 0xC000022B
02779 #define STATUS_CONVERT_TO_LARGE 0xC000022C
02780 #define STATUS_RETRY 0xC000022D
02781 #define STATUS_FOUND_OUT_OF_SCOPE 0xC000022E
02782 #define STATUS_ALLOCATE_BUCKET 0xC000022F
02783 #define STATUS_PROPSET_NOT_FOUND 0xC0000230
02784 #define STATUS_MARSHALL_OVERFLOW 0xC0000231
02785 #define STATUS_INVALID_VARIANT 0xC0000232
02786 #define STATUS_DOMAIN_CONTROLLER_NOT_FOUND 0xC0000233
02787 #define STATUS_ACCOUNT_LOCKED_OUT 0xC0000234
02788 #define STATUS_HANDLE_NOT_CLOSABLE 0xC0000235
02789 #define STATUS_CONNECTION_REFUSED 0xC0000236
02790 #define STATUS_GRACEFUL_DISCONNECT 0xC0000237
02791 #define STATUS_ADDRESS_ALREADY_ASSOCIATED 0xC0000238
02792 #define STATUS_ADDRESS_NOT_ASSOCIATED 0xC0000239
02793 #define STATUS_CONNECTION_INVALID 0xC000023A
02794 #define STATUS_CONNECTION_ACTIVE 0xC000023B
02795 #define STATUS_NETWORK_UNREACHABLE 0xC000023C
02796 #define STATUS_HOST_UNREACHABLE 0xC000023D
02797 #define STATUS_PROTOCOL_UNREACHABLE 0xC000023E
02798 #define STATUS_PORT_UNREACHABLE 0xC000023F
02799 #define STATUS_REQUEST_ABORTED 0xC0000240
02800 #define STATUS_CONNECTION_ABORTED 0xC0000241
02801 #define STATUS_BAD_COMPRESSION_BUFFER 0xC0000242
02802 #define STATUS_USER_MAPPED_FILE 0xC0000243
02803 #define STATUS_AUDIT_FAILED 0xC0000244
02804 #define STATUS_TIMER_RESOLUTION_NOT_SET 0xC0000245
02805 #define STATUS_CONNECTION_COUNT_LIMIT 0xC0000246
```

```
02806 #define STATUS_LOGIN_TIME_RESTRICTION 0xC0000247
02807 #define STATUS_LOGIN_WKSTA_RESTRICTION 0xC0000248
02808 #define STATUS_IMAGE_MP_UP_MISMATCH 0xC0000249
02809 #define STATUS_INSUFFICIENT_LOGON_INFO 0xC0000250
02810 #define STATUS_BAD_DLL_ENTRYPOINT 0xC0000251
02811 #define STATUS_BAD_SERVICE_ENTRYPOINT 0xC0000252
02812 #define STATUS_LPC_REPLY_LOST 0xC0000253
02813 #define STATUS_IP_ADDRESS_CONFLICT1 0xC0000254
02814 #define STATUS_IP_ADDRESS_CONFLICT2 0xC0000255
02815 #define STATUS_REGISTRY_QUOTA_LIMIT 0xC0000256
02816 #define STATUS_PATH_NOT_COVERED 0xC0000257
02817 #define STATUS_NO_CALLBACK_ACTIVE 0xC0000258
02818 #define STATUS_LICENSE_QUOTA_EXCEEDED 0xC0000259
02819 #define STATUS_PWD_TOO_SHORT 0xC000025A
02820 #define STATUS_PWD_TOO_RECENT 0xC000025B
02821 #define STATUS_PWD_HISTORY_CONFLICT 0xC000025C
02822 #define STATUS_PLUGPLAY_NO_DEVICE 0xC000025E
02823 #define STATUS_UNSUPPORTED_COMPRESSION 0xC000025F
02824 #define STATUS_INVALID_HW_PROFILE 0xC0000260
02825 #define STATUS_INVALID_PLUGPLAY_DEVICE_PATH 0xC0000261
02826 #define STATUS_DRIVER_ORDINAL_NOT_FOUND 0xC0000262
02827 #define STATUS_DRIVER_ENTRYPOINT_NOT_FOUND 0xC0000263
02828 #define STATUS_RESOURCE_NOT_OWNED 0xC0000264
02829 #define STATUS_TOO_MANY_LINKS 0xC0000265
02830 #define STATUS_QUOTA_LIST_INCONSISTENT 0xC0000266
02831 #define STATUS_FILE_IS_OFFLINE 0xC0000267
02832 #define STATUS_EVALUATION_EXPIRATION 0xC0000268
02833 #define STATUS_ILLEGAL_DLL_RELOCATION 0xC0000269
02834 #define STATUS_LICENSE_VIOLATION 0xC000026A
02835 #define STATUS_DLL_INIT_FAILED_LOGOFF 0xC000026B
02836 #define STATUS_DRIVER_UNABLE_TO_LOAD 0xC000026C
02837 #define STATUS_DFS_UNAVAILABLE 0xC000026D
02838 #define STATUS_VOLUME_DISMOUNTED 0xC000026E
02839 #define STATUS_WX86_INTERNAL_ERROR 0xC000026F
02840 #define STATUS_WX86_FLOAT_STACK_CHECK 0xC0000270
02841 #define STATUS_WOW_ASSERTION 0xC0009898
02842 #define RPC_NT_INVALID_STRING_BINDING 0xC0020001
02843 #define RPC_NT_WRONG_KIND_OF_BINDING 0xC0020002
02844 #define RPC_NT_INVALID_BINDING 0xC0020003
02845 #define RPC_NT_PROTSEQ_NOT_SUPPORTED 0xC0020004
02846 #define RPC_NT_INVALID_RPC_PROTSEQ 0xC0020005
02847 #define RPC_NT_INVALID_STRING_UUID 0xC0020006
02848 #define RPC_NT_INVALID_ENDPOINT_FORMAT 0xC0020007
02849 #define RPC_NT_INVALID_NET_ADDR 0xC0020008
02850 #define RPC_NT_NO_ENDPOINT_FOUND 0xC0020009
02851 #define RPC_NT_INVALID_TIMEOUT 0xC002000A
02852 #define RPC_NT_OBJECT_NOT_FOUND 0xC002000B
02853 #define RPC_NT_ALREADY_REGISTERED 0xC002000C
02854 #define RPC_NT_TYPE_ALREADY_REGISTERED 0xC002000D
02855 #define RPC_NT_ALREADY_LISTENING 0xC002000E
02856 #define RPC_NT_NO_PROTSEQS_REGISTERED 0xC002000F
02857 #define RPC_NT_NOT_LISTENING 0xC0020010
02858 #define RPC_NT_UNKNOWN_MGR_TYPE 0xC0020011
02859 #define RPC_NT_UNKNOWN_IF 0xC0020012
02860 #define RPC_NT_NO_BINDINGS 0xC0020013
02861 #define RPC_NT_NO_PROTSEQS 0xC0020014
02862 #define RPC_NT_CANT_CREATE_ENDPOINT 0xC0020015
02863 #define RPC_NT_OUT_OF_RESOURCES 0xC0020016
02864 #define RPC_NT_SERVER_UNAVAILABLE 0xC0020017
02865 #define RPC_NT_SERVER_TOO_BUSY 0xC0020018
02866 #define RPC_NT_INVALID_NETWORK_OPTIONS 0xC0020019
02867 #define RPC_NT_NO_CALL_ACTIVE 0xC002001A
02868 #define RPC_NT_CALL_FAILED 0xC002001B
02869 #define RPC_NT_CALL_FAILED_DNE 0xC002001C
02870 #define RPC_NT_PROTOCOL_ERROR 0xC002001D
02871 #define RPC_NT_UNSUPPORTED_TRANS_SYN 0xC002001F
02872 #define RPC_NT_UNSUPPORTED_TYPE 0xC0020021
02873 #define RPC_NT_INVALID_TAG 0xC0020022
02874 #define RPC_NT_INVALID_BOUND 0xC0020023
02875 #define RPC_NT_NO_ENTRY_NAME 0xC0020024
02876 #define RPC_NT_INVALID_NAME_SYNTAX 0xC0020025
02877 #define RPC_NT_UNSUPPORTED_NAME_SYNTAX 0xC0020026
02878 #define RPC_NT_UUID_NO_ADDRESS 0xC0020028
02879 #define RPC_NT_DUPLICATE_ENDPOINT 0xC0020029
02880 #define RPC_NT_UNKNOWN_AUTHN_TYPE 0xC002002A
02881 #define RPC_NT_MAX_CALLS_TOO_SMALL 0xC002002B
02882 #define RPC_NT_STRING_TOO_LONG 0xC002002C
02883 #define RPC_NT_PROTSEQ_NOT_FOUND 0xC002002D
02884 #define RPC_NT_PROCNUM_OUT_OF_RANGE 0xC002002E
02885 #define RPC_NT_BINDING_HAS_NO_AUTH 0xC002002F
02886 #define RPC_NT_UNKNOWN_AUTHN_SERVICE 0xC0020030
02887 #define RPC_NT_UNKNOWN_AUTHN_LEVEL 0xC0020031
02888 #define RPC_NT_INVALID_AUTH_IDENTITY 0xC0020032
02889 #define RPC_NT_UNKNOWN_AUTHZ_SERVICE 0xC0020033
02890 #define EPT_NT_INVALID_ENTRY 0xC0020034
02891 #define EPT_NT_CANT_PERFORM_OP 0xC0020035
02892 #define EPT_NT_NOT_REGISTERED 0xC0020036
```

```

02893 #define RPC_NT_NOTHING_TO_EXPORT          0xC0020037
02894 #define RPC_NT_INCOMPLETE_NAME             0xC0020038
02895 #define RPC_NT_INVALID_VERS_OPTION          0xC0020039
02896 #define RPC_NT_NO_MORE_MEMBERS              0xC002003A
02897 #define RPC_NT_NOT_ALL_OBJS_UNEXPORTED      0xC002003B
02898 #define RPC_NT_INTERFACE_NOT_FOUND          0xC002003C
02899 #define RPC_NT_ENTRY_ALREADY_EXISTS          0xC002003D
02900 #define RPC_NT_ENTRY_NOT_FOUND              0xC002003E
02901 #define RPC_NT_NAME_SERVICE_UNAVAILABLE      0xC002003F
02902 #define RPC_NT_INVALID_NAF_ID                0xC0020040
02903 #define RPC_NT_CANNOT_SUPPORT                0xC0020041
02904 #define RPC_NT_NO_CONTEXT_AVAILABLE          0xC0020042
02905 #define RPC_NT_INTERNAL_ERROR                0xC0020043
02906 #define RPC_NT_ZERO_DIVIDE                   0xC0020044
02907 #define RPC_NT_ADDRESS_ERROR                 0xC0020045
02908 #define RPC_NT_FP_DIV_ZERO                   0xC0020046
02909 #define RPC_NT_FP_UNDERFLOW                 0xC0020047
02910 #define RPC_NT_FP_OVERFLOW                   0xC0020048
02911 #define RPC_NT_NO_MORE_ENTRIES               0xC0030001
02912 #define RPC_NT_SS_CHAR_TRANS_OPEN_FAIL       0xC0030002
02913 #define RPC_NT_SS_CHAR_TRANS_SHORT_FILE      0xC0030003
02914 #define RPC_NT_SS_IN_NULL_CONTEXT            0xC0030004
02915 #define RPC_NT_SS_CONTEXT_MISMATCH           0xC0030005
02916 #define RPC_NT_SS_CONTEXT_DAMAGED            0xC0030006
02917 #define RPC_NT_SS_HANDLES_MISMATCH           0xC0030007
02918 #define RPC_NT_SS_CANNOT_GET_CALL_HANDLE     0xC0030008
02919 #define RPC_NT_NULL_REF_POINTER              0xC0030009
02920 #define RPC_NT_ENUM_VALUE_OUT_OF_RANGE        0xC003000A
02921 #define RPC_NT_BYTE_COUNT_TOO_SMALL          0xC003000B
02922 #define RPC_NT_BAD_STUB_DATA                 0xC003000C
02923 #define RPC_NT_CALL_IN_PROGRESS              0xC0020049
02924 #define RPC_NT_NO_MORE_BINDINGS              0xC002004A
02925 #define RPC_NT_GROUP_MEMBER_NOT_FOUND        0xC002004B
02926 #define EPT_NT_CANT_CREATE                   0xC002004C
02927 #define RPC_NT_INVALID_OBJECT                0xC002004D
02928 #define RPC_NT_NO_INTERFACES                 0xC002004F
02929 #define RPC_NT_CALL_CANCELLED                0xC0020050
02930 #define RPC_NT_BINDING_INCOMPLETE            0xC0020051
02931 #define RPC_NT_COMM_FAILURE                  0xC0020052
02932 #define RPC_NT_UNSUPPORTED_AUTHN_LEVEL        0xC0020053
02933 #define RPC_NT_NO_PRINC_NAME                 0xC0020054
02934 #define RPC_NT_NOT_RPC_ERROR                 0xC0020055
02935 #define RPC_NT_UUID_LOCAL_ONLY               0x40020056
02936 #define RPC_NT_SEC_PKG_ERROR                 0xC0020057
02937 #define RPC_NT_NOT_CANCELLED                 0xC0020058
02938 #define RPC_NT_INVALID_ES_ACTION             0xC0030059
02939 #define RPC_NT_WRONG_ES_VERSION              0xC003005A
02940 #define RPC_NT_WRONG_STUB_VERSION            0xC003005B
02941 #define RPC_NT_INVALID_PIPE_OBJECT           0xC003005C
02942 #define RPC_NT_INVALID_PIPE_OPERATION        0xC003005D
02943 #define RPC_NT_WRONG_PIPE_VERSION            0xC003005E
02944 #define RPC_NT_SEND_INCOMPLETE               0x400200AF
02945
02946 #define MAXIMUM_WAIT_OBJECTS 64
02947 #define MAXIMUM_SUSPEND_COUNT 127
02948
02949
02950 /*
02951  * Return values from the actual exception handlers
02952  */
02953
02954 #define ExceptionContinueExecution 0
02955 #define ExceptionContinueSearch 1
02956 #define ExceptionNestedException 2
02957 #define ExceptionCollidedUnwind 3
02958
02959 /*
02960  * Return values from filters in except() and from UnhandledExceptionFilter
02961  */
02962
02963 #define EXCEPTION_EXECUTE_HANDLER 1
02964 #define EXCEPTION_CONTINUE_SEARCH 0
02965 #define EXCEPTION_CONTINUE_EXECUTION -1
02966
02967 /*
02968  * From OS/2 2.0 exception handling
02969  * Win32 seems to use the same flags as ExceptionFlags in an EXCEPTION_RECORD
02970  */
02971
02972 #define EH_NONCONTINUABLE 0x01
02973 #define EH_UNWINDING 0x02
02974 #define EH_EXIT_UNWIND 0x04
02975 #define EH_STACK_INVALID 0x08
02976 #define EH_NESTED_CALL 0x10
02977
02978 #define EXCEPTION_CONTINUABLE 0
02979 #define EXCEPTION_NONCONTINUABLE EH_NONCONTINUABLE

```

```

02980
02981 /*
02982  * The exception record used by Win32 to give additional information
02983  * about exception to exception handlers.
02984  */
02985
02986 #define EXCEPTION_MAXIMUM_PARAMETERS 15
02987
02988 typedef struct __EXCEPTION_RECORD
02989 {
02990     DWORD      ExceptionCode;
02991     DWORD      ExceptionFlags;
02992     struct __EXCEPTION_RECORD *ExceptionRecord;
02993
02994     LPVOID      ExceptionAddress;
02995     DWORD      NumberParameters;
02996     DWORD      ExceptionInformation[EXCEPTION_MAXIMUM_PARAMETERS];
02997 } EXCEPTION_RECORD, *PEXCEPTION_RECORD;
02998
02999 /*
03000  * The exception pointers structure passed to exception filters
03001  * in except() and the UnhandledExceptionFilter().
03002  */
03003
03004 typedef struct _EXCEPTION_POINTERS
03005 {
03006     PEXCEPTION_RECORD ExceptionRecord;
03007     PCONTEXT          ContextRecord;
03008 } EXCEPTION_POINTERS, *PEXCEPTION_POINTERS;
03009
03010
03011 /*
03012  * The exception frame, used for registering exception handlers
03013  * Win32 cares only about this, but compilers generally emit
03014  * larger exception frames for their own use.
03015  */
03016
03017 struct __EXCEPTION_FRAME;
03018
03019 typedef DWORD (*PEXCEPTION_HANDLER) (PEXCEPTION_RECORD, struct __EXCEPTION_FRAME*,
03020                                     PCONTEXT, struct __EXCEPTION_FRAME **);
03021
03022 typedef struct __EXCEPTION_FRAME
03023 {
03024     struct __EXCEPTION_FRAME *Prev;
03025     PEXCEPTION_HANDLER      Handler;
03026 } EXCEPTION_FRAME, *PEXCEPTION_FRAME;
03027
03028 /*
03029  * function pointer to a exception filter
03030  */
03031
03032 typedef LONG CALLBACK (*PTOP_LEVEL_EXCEPTION_FILTER) (PEXCEPTION_POINTERS ExceptionInfo);
03033 typedef TOP_LEVEL_EXCEPTION_FILTER LPTOP_LEVEL_EXCEPTION_FILTER;
03034
03035 DWORD WINAPI UnhandledExceptionFilter( PEXCEPTION_POINTERS epointers );
03036 LPTOP_LEVEL_EXCEPTION_FILTER
03037 WINAPI SetUnhandledExceptionFilter( LPTOP_LEVEL_EXCEPTION_FILTER filter );
03038
03039 /* status values for ContinueDebugEvent */
03040 #define DBG_CONTINUE          0x00010002
03041 #define DBG_TERMINATE_THREAD  0x40010003
03042 #define DBG_TERMINATE_PROCESS 0x40010004
03043 #define DBG_CONTROL_C         0x40010005
03044 #define DBG_CONTROL_BREAK     0x40010008
03045 #define DBG_EXCEPTION_NOT_HANDLED 0x80010001
03046
03047 typedef struct _NT_TIB
03048 {
03049     struct _EXCEPTION_REGISTRATION_RECORD *ExceptionList;
03050     PVOID StackBase;
03051     PVOID StackLimit;
03052     PVOID SubSystemTib;
03053     union {
03054         PVOID FiberData;
03055         DWORD Version;
03056     } DUMMYUNIONNAME;
03057     PVOID ArbitraryUserPointer;
03058     struct _NT_TIB *Self;
03059 } NT_TIB, *PNT_TIB;
03060
03061 struct _TEB;
03062
03063 #if defined(__i386__) && defined(__GNUC__) && !defined(__CHECKER__)
03064 extern inline struct _TEB WINAPI *NtCurrentTeb(void);
03065 extern inline struct _TEB WINAPI *NtCurrentTeb(void)
03066 {

```

```

03067     struct _TEB *teb;
03068     __asm__("byte 0x64\n\tmovl (0x18),%0" : "=r" (teb));
03069     return teb;
03070 }
03071 #else
03072 extern struct _TEB WINAPI *NtCurrentTeb(void);
03073 #endif
03074
03075
03076 /*
03077  * File formats definitions
03078  */
03079
03080 typedef struct _IMAGE_DOS_HEADER {
03081     WORD    e_magic;        /* 00: MZ Header signature */
03082     WORD    e_cblp;         /* 02: Bytes on last page of file */
03083     WORD    e_cp;           /* 04: Pages in file */
03084     WORD    e_crlc;         /* 06: Relocations */
03085     WORD    e_cparhdr;      /* 08: Size of header in paragraphs */
03086     WORD    e_minalloc;     /* 0a: Minimum extra paragraphs needed */
03087     WORD    e_maxalloc;     /* 0c: Maximum extra paragraphs needed */
03088     WORD    e_ss;           /* 0e: Initial (relative) SS value */
03089     WORD    e_sp;           /* 10: Initial SP value */
03090     WORD    e_csum;         /* 12: Checksum */
03091     WORD    e_ip;           /* 14: Initial IP value */
03092     WORD    e_cs;           /* 16: Initial (relative) CS value */
03093     WORD    e_lfarlc;       /* 18: File address of relocation table */
03094     WORD    e_ovno;         /* 1a: Overlay number */
03095     WORD    e_res[4];       /* 1c: Reserved words */
03096     WORD    e_oemid;        /* 24: OEM identifier (for e_oeminfo) */
03097     WORD    e_oeminfo;      /* 26: OEM information; e_oemid specific */
03098     WORD    e_res2[10];     /* 28: Reserved words */
03099     DWORD    e_lfanew;      /* 3c: Offset to extended header */
03100 } IMAGE_DOS_HEADER, *PIMAGE_DOS_HEADER;
03101
03102 #define IMAGE_DOS_SIGNATURE    0x5A4D    /* MZ */
03103 #define IMAGE_OS2_SIGNATURE    0x454E    /* NE */
03104 #define IMAGE_OS2_SIGNATURE_LE 0x454C    /* LE */
03105 #define IMAGE_OS2_SIGNATURE_LX 0x584C    /* LX */
03106 #define IMAGE_VXD_SIGNATURE    0x454C    /* LE */
03107 #define IMAGE_NT_SIGNATURE     0x00004550 /* PE00 */
03108
03109 /*
03110  * This is the Windows executable (NE) header.
03111  * the name IMAGE_OS2_HEADER is misleading, but in the SDK this way.
03112  */
03113 typedef struct
03114 {
03115     WORD    ne_magic;        /* 00 NE signature 'NE' */
03116     BYTE    ne_ver;          /* 02 Linker version number */
03117     BYTE    ne_rev;          /* 03 Linker revision number */
03118     WORD    ne_enttab;       /* 04 Offset to entry table relative to NE */
03119     WORD    ne_cbenttab;     /* 06 Length of entry table in bytes */
03120     LONG    ne_crc;          /* 08 Checksum */
03121     WORD    ne_flags;        /* 0c Flags about segments in this file */
03122     WORD    ne_autodata;     /* 0e Automatic data segment number */
03123     WORD    ne_heap;         /* 10 Initial size of local heap */
03124     WORD    ne_stack;        /* 12 Initial size of stack */
03125     DWORD    ne_csip;        /* 14 Initial CS:IP */
03126     DWORD    ne_sssp;        /* 18 Initial SS:SP */
03127     WORD    ne_cseg;         /* 1c # of entries in segment table */
03128     WORD    ne_cmod;         /* 1e # of entries in module reference tab. */
03129     WORD    ne_cbnrestab;    /* 20 Length of nonresident-name table */
03130     WORD    ne_segtab;       /* 22 Offset to segment table */
03131     WORD    ne_rsrctab;      /* 24 Offset to resource table */
03132     WORD    ne_restab;       /* 26 Offset to resident-name table */
03133     WORD    ne_modtab;       /* 28 Offset to module reference table */
03134     WORD    ne_imptab;       /* 2a Offset to imported name table */
03135     DWORD    ne_nrestab;     /* 2c Offset to nonresident-name table */
03136     WORD    ne_cmovent;      /* 30 # of movable entry points */
03137     WORD    ne_align;        /* 32 Logical sector alignment shift count */
03138     WORD    ne_cres;         /* 34 # of resource segments */
03139     BYTE    ne_exetyp;       /* 36 Flags indicating target OS */
03140     BYTE    ne_flagsothers;  /* 37 Additional information flags */
03141     WORD    fastload_offset; /* 38 Offset to fast load area (should be ne_pretthunks) */
03142     WORD    fastload_length; /* 3a Length of fast load area (should be ne_psegrebytes) */
03143     WORD    ne_swaparea;     /* 3c Reserved by Microsoft */
03144     WORD    ne_expver;       /* 3e Expected Windows version number */
03145 } IMAGE_OS2_HEADER, *PIMAGE_OS2_HEADER;
03146
03147 typedef struct _IMAGE_VXD_HEADER {
03148     WORD    e32_magic;
03149     BYTE    e32_border;
03150     BYTE    e32_worder;
03151     DWORD    e32_level;
03152     WORD    e32_cpu;
03153     WORD    e32_os;

```

```

03154     DWORD e32_ver;
03155     DWORD e32_mflags;
03156     DWORD e32_mpages;
03157     DWORD e32_startobj;
03158     DWORD e32_eip;
03159     DWORD e32_stackobj;
03160     DWORD e32_esp;
03161     DWORD e32_pagesize;
03162     DWORD e32_lastpagesize;
03163     DWORD e32_fixupsize;
03164     DWORD e32_fixupsum;
03165     DWORD e32_ldrsize;
03166     DWORD e32_ldrsum;
03167     DWORD e32_objtab;
03168     DWORD e32_objcnt;
03169     DWORD e32_objmap;
03170     DWORD e32_itermap;
03171     DWORD e32_rsrctab;
03172     DWORD e32_rsrcnt;
03173     DWORD e32_restab;
03174     DWORD e32_enttab;
03175     DWORD e32_dirtab;
03176     DWORD e32_dircnt;
03177     DWORD e32_fpagetab;
03178     DWORD e32_frextab;
03179     DWORD e32_impmod;
03180     DWORD e32_impmodcnt;
03181     DWORD e32_impproc;
03182     DWORD e32_pagesum;
03183     DWORD e32_datapage;
03184     DWORD e32_preload;
03185     DWORD e32_nrestab;
03186     DWORD e32_cbnrestab;
03187     DWORD e32_nressum;
03188     DWORD e32_autodata;
03189     DWORD e32_debuginfo;
03190     DWORD e32_debuglen;
03191     DWORD e32_instpreload;
03192     DWORD e32_instdemand;
03193     DWORD e32_heapsize;
03194     BYTE   e32_res3[12];
03195     DWORD e32_winresoff;
03196     DWORD e32_winreslen;
03197     WORD   e32_devid;
03198     WORD   e32_ddkver;
03199 } IMAGE_VXD_HEADER, *PIMAGE_VXD_HEADER;
03200
03201
03202 /* These defines describe the meanings of the bits in the Characteristics
03203    field */
03204
03205 #define IMAGE_FILE_RELOCS_STRIPPED    0x0001 /* No relocation info */
03206 #define IMAGE_FILE_EXECUTABLE_IMAGE  0x0002
03207 #define IMAGE_FILE_LINE_NUMS_STRIPPED 0x0004
03208 #define IMAGE_FILE_LOCAL_SYMS_STRIPPED 0x0008
03209 #define IMAGE_FILE_16BIT_MACHINE     0x0040
03210 #define IMAGE_FILE_BYTES_REVERSED_LO 0x0080
03211 #define IMAGE_FILE_32BIT_MACHINE     0x0100
03212 #define IMAGE_FILE_DEBUG_STRIPPED    0x0200
03213 #define IMAGE_FILE_SYSTEM             0x1000
03214 #define IMAGE_FILE_DLL               0x2000
03215 #define IMAGE_FILE_BYTES_REVERSED_HI 0x8000
03216
03217 /* These are the settings of the Machine field. */
03218 #define IMAGE_FILE_MACHINE_UNKNOWN    0
03219 #define IMAGE_FILE_MACHINE_I860       0x14d
03220 #define IMAGE_FILE_MACHINE_I386       0x14c
03221 #define IMAGE_FILE_MACHINE_R3000      0x162
03222 #define IMAGE_FILE_MACHINE_R4000      0x166
03223 #define IMAGE_FILE_MACHINE_R10000     0x168
03224 #define IMAGE_FILE_MACHINE_ALPHA      0x184
03225 #define IMAGE_FILE_MACHINE_POWERPC    0x1f0
03226
03227 #define IMAGE_SIZEOF_FILE_HEADER      20
03228
03229 /* Possible Magic values */
03230 #define IMAGE_NT_OPTIONAL_HDR_MAGIC    0x10b
03231 #define IMAGE_ROM_OPTIONAL_HDR_MAGIC  0x107
03232
03233 /* These are indexes into the DataDirectory array */
03234 #define IMAGE_FILE_EXPORT_DIRECTORY   0
03235 #define IMAGE_FILE_IMPORT_DIRECTORY   1
03236 #define IMAGE_FILE_RESOURCE_DIRECTORY 2
03237 #define IMAGE_FILE_EXCEPTION_DIRECTORY 3
03238 #define IMAGE_FILE_SECURITY_DIRECTORY 4
03239 #define IMAGE_FILE_BASE_RELOCATION_TABLE 5
03240 #define IMAGE_FILE_DEBUG_DIRECTORY   6

```

```

03241 #define IMAGE_FILE_DESCRIPTION_STRING      7
03242 #define IMAGE_FILE_MACHINE_VALUE            8 /* Mips */
03243 #define IMAGE_FILE_THREAD_LOCAL_STORAGE      9
03244 #define IMAGE_FILE_CALLBACK_DIRECTORY       10
03245
03246 /* Directory Entries, indices into the DataDirectory array */
03247
03248 #define IMAGE_DIRECTORY_ENTRY_EXPORT          0
03249 #define IMAGE_DIRECTORY_ENTRY_IMPORT          1
03250 #define IMAGE_DIRECTORY_ENTRY_RESOURCE        2
03251 #define IMAGE_DIRECTORY_ENTRY_EXCEPTION      3
03252 #define IMAGE_DIRECTORY_ENTRY_SECURITY        4
03253 #define IMAGE_DIRECTORY_ENTRY_BASERELOC       5
03254 #define IMAGE_DIRECTORY_ENTRY_DEBUG           6
03255 #define IMAGE_DIRECTORY_ENTRY_COPYRIGHT       7
03256 #define IMAGE_DIRECTORY_ENTRY_GLOBALPTR       8 /* (MIPS GP) */
03257 #define IMAGE_DIRECTORY_ENTRY_TLS             9
03258 #define IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG    10
03259 #define IMAGE_DIRECTORY_ENTRY_BOUND_IMPORT    11
03260 #define IMAGE_DIRECTORY_ENTRY_IAT            12 /* Import Address Table */
03261 #define IMAGE_DIRECTORY_ENTRY_DELAY_IMPORT    13
03262 #define IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR  14
03263
03264 /* Subsystem Values */
03265
03266 #define IMAGE_SUBSYSTEM_UNKNOWN              0
03267 #define IMAGE_SUBSYSTEM_NATIVE               1
03268 #define IMAGE_SUBSYSTEM_WINDOWS_GUI          2 /* Windows GUI subsystem */
03269 #define IMAGE_SUBSYSTEM_WINDOWS_CUI          3 /* Windows character subsystem*/
03270 #define IMAGE_SUBSYSTEM_OS2_CUI              5
03271 #define IMAGE_SUBSYSTEM_POSIX_CUI            7
03272
03273 typedef struct _IMAGE_FILE_HEADER {
03274     WORD Machine;
03275     WORD NumberOfSections;
03276     DWORD TimeDateStamp;
03277     DWORD PointerToSymbolTable;
03278     DWORD NumberOfSymbols;
03279     WORD SizeOfOptionalHeader;
03280     WORD Characteristics;
03281 } IMAGE_FILE_HEADER, *PIMAGE_FILE_HEADER;
03282
03283 typedef struct _IMAGE_DATA_DIRECTORY {
03284     DWORD VirtualAddress;
03285     DWORD Size;
03286 } IMAGE_DATA_DIRECTORY, *PIMAGE_DATA_DIRECTORY;
03287
03288 #define IMAGE_NUMBEROF_DIRECTORY_ENTRIES 16
03289
03290 typedef struct _IMAGE_OPTIONAL_HEADER {
03291
03292     /* Standard fields */
03293
03294     WORD Magic; /* 0x10b or 0x107 */ /* 0x00 */
03295     BYTE MajorLinkerVersion;
03296     BYTE MinorLinkerVersion;
03297     DWORD SizeOfCode;
03298     DWORD SizeOfInitializedData;
03299     DWORD SizeOfUninitializedData;
03300     DWORD AddressOfEntryPoint; /* 0x10 */
03301     DWORD BaseOfCode;
03302     DWORD BaseOfData;
03303
03304     /* NT additional fields */
03305
03306     DWORD ImageBase;
03307     DWORD SectionAlignment; /* 0x20 */
03308     DWORD FileAlignment;
03309     WORD MajorOperatingSystemVersion;
03310     WORD MinorOperatingSystemVersion;
03311     WORD MajorImageVersion;
03312     WORD MinorImageVersion;
03313     WORD MajorSubsystemVersion; /* 0x30 */
03314     WORD MinorSubsystemVersion;
03315     DWORD Win32VersionValue;
03316     DWORD SizeOfImage;
03317     DWORD SizeOfHeaders;
03318     DWORD CheckSum; /* 0x40 */
03319     WORD Subsystem;
03320     WORD DllCharacteristics;
03321     DWORD SizeOfStackReserve;
03322     DWORD SizeOfStackCommit;
03323     DWORD SizeOfHeapReserve; /* 0x50 */
03324     DWORD SizeOfHeapCommit;
03325     DWORD LoaderFlags;
03326     DWORD NumberOfRvaAndSizes;
03327     IMAGE_DATA_DIRECTORY DataDirectory[IMAGE_NUMBEROF_DIRECTORY_ENTRIES]; /* 0x60 */

```

```

03328 } IMAGE_OPTIONAL_HEADER, *PIMAGE_OPTIONAL_HEADER;
03329
03330 typedef struct _IMAGE_NT_HEADERS {
03331     DWORD Signature; /* "PE"\0\0 */
03332     IMAGE_FILE_HEADER FileHeader;
03333     IMAGE_OPTIONAL_HEADER OptionalHeader;
03334 } IMAGE_NT_HEADERS, *PIMAGE_NT_HEADERS;
03335
03336 #define IMAGE_SIZEOF_SHORT_NAME 8
03337
03338 typedef struct _IMAGE_SECTION_HEADER {
03339     BYTE Name[IMAGE_SIZEOF_SHORT_NAME];
03340     union {
03341         DWORD PhysicalAddress;
03342         DWORD VirtualSize;
03343     } Misc;
03344     DWORD VirtualAddress;
03345     DWORD SizeOfRawData;
03346     DWORD PointerToRawData;
03347     DWORD PointerToRelocations;
03348     DWORD PointerToLinenumbers;
03349     WORD NumberOfRelocations;
03350     WORD NumberOfLinenumbers;
03351     DWORD Characteristics;
03352 } IMAGE_SECTION_HEADER, *PIMAGE_SECTION_HEADER;
03353
03354 #define IMAGE_SIZEOF_SECTION_HEADER 40
03355
03356 #define IMAGE_FIRST_SECTION(nthead) \
03357     ((PIMAGE_SECTION_HEADER) ((LPBYTE)&((PIMAGE_NT_HEADERS)(nthead))->OptionalHeader + \
03358     ((PIMAGE_NT_HEADERS)(nthead))->FileHeader.SizeOfOptionalHeader))
03359
03360 /* These defines are for the Characteristics bitfield. */
03361 /* #define IMAGE_SCN_TYPE_REG 0x00000000 - Reserved */
03362 /* #define IMAGE_SCN_TYPE_DSECT 0x00000001 - Reserved */
03363 /* #define IMAGE_SCN_TYPE_NOLOAD 0x00000002 - Reserved */
03364 /* #define IMAGE_SCN_TYPE_GROUP 0x00000004 - Reserved */
03365 /* #define IMAGE_SCN_TYPE_NO_PAD 0x00000008 - Reserved */
03366 /* #define IMAGE_SCN_TYPE_COPY 0x00000010 - Reserved */
03367
03368 #define IMAGE_SCN_CNT_CODE 0x00000020
03369 #define IMAGE_SCN_CNT_INITIALIZED_DATA 0x00000040
03370 #define IMAGE_SCN_CNT_UNINITIALIZED_DATA 0x00000080
03371
03372 #define IMAGE_SCN_LNK_OTHER 0x00000100
03373 #define IMAGE_SCN_LNK_INFO 0x00000200
03374 /* #define IMAGE_SCN_TYPE_OVER 0x00000400 - Reserved */
03375 #define IMAGE_SCN_LNK_REMOVE 0x00000800
03376 #define IMAGE_SCN_LNK_COMDAT 0x00001000
03377
03378 /* 0x00002000 - Reserved */
03379 /* #define IMAGE_SCN_MEM_PROTECTED 0x00004000 - Obsolete */
03380 #define IMAGE_SCN_MEM_FARDATA 0x00008000
03381
03382 /* #define IMAGE_SCN_MEM_SYSHEAP 0x00010000 - Obsolete */
03383 #define IMAGE_SCN_MEM_PURGEABLE 0x00020000
03384 #define IMAGE_SCN_MEM_16BIT 0x00020000
03385 #define IMAGE_SCN_MEM_LOCKED 0x00040000
03386 #define IMAGE_SCN_MEM_PRELOAD 0x00080000
03387
03388 #define IMAGE_SCN_ALIGN_1BYTES 0x00100000
03389 #define IMAGE_SCN_ALIGN_2BYTES 0x00200000
03390 #define IMAGE_SCN_ALIGN_4BYTES 0x00300000
03391 #define IMAGE_SCN_ALIGN_8BYTES 0x00400000
03392 #define IMAGE_SCN_ALIGN_16BYTES 0x00500000 /* Default */
03393 #define IMAGE_SCN_ALIGN_32BYTES 0x00600000
03394 #define IMAGE_SCN_ALIGN_64BYTES 0x00700000
03395 /* 0x00800000 - Unused */
03396
03397 #define IMAGE_SCN_LNK_NRELOC_OVFL 0x01000000
03398
03399
03400 #define IMAGE_SCN_MEM_DISCARDABLE 0x02000000
03401 #define IMAGE_SCN_MEM_NOT_CACHED 0x04000000
03402 #define IMAGE_SCN_MEM_NOT_PAGED 0x08000000
03403 #define IMAGE_SCN_MEM_SHARED 0x10000000
03404 #define IMAGE_SCN_MEM_EXECUTE 0x20000000
03405 #define IMAGE_SCN_MEM_READ 0x40000000
03406 #define IMAGE_SCN_MEM_WRITE 0x80000000
03407
03408 #include "pshpack2.h"
03409
03410 typedef struct _IMAGE_SYMBOL {
03411     union {
03412         BYTE ShortName[8];
03413         struct {
03414             DWORD Short;

```

```

03415         DWORD    Long;
03416     } Name;
03417     DWORD    LongName[2];
03418 } N;
03419     DWORD    Value;
03420     SHORT    SectionNumber;
03421     WORD     Type;
03422     BYTE     StorageClass;
03423     BYTE     NumberOfAuxSymbols;
03424 } IMAGE_SYMBOL;
03425 typedef IMAGE_SYMBOL *PIMAGE_SYMBOL;
03426
03427 #define IMAGE_SIZEOF_SYMBOL 18
03428
03429 typedef struct _IMAGE_LINENUMBER {
03430     union {
03431         DWORD    SymbolTableIndex;
03432         DWORD    VirtualAddress;
03433     } Type;
03434     WORD     Linenumber;
03435 } IMAGE_LINENUMBER;
03436 typedef IMAGE_LINENUMBER *PIMAGE_LINENUMBER;
03437
03438 #define IMAGE_SIZEOF_LINENUMBER 6
03439
03440 typedef union _IMAGE_AUX_SYMBOL {
03441     struct {
03442         DWORD    TagIndex;
03443         union {
03444             struct {
03445                 WORD    Linenumber;
03446                 WORD    Size;
03447             } LnSz;
03448             DWORD    TotalSize;
03449         } Misc;
03450         union {
03451             struct {
03452                 DWORD    PointerToLinenumber;
03453                 DWORD    PointerToNextFunction;
03454             } Function;
03455             struct {
03456                 WORD    Dimension[4];
03457             } Array;
03458         } FcnAry;
03459         WORD    TvIndex;
03460     } Sym;
03461     struct {
03462         BYTE     Name[IMAGE_SIZEOF_SYMBOL];
03463     } File;
03464     struct {
03465         DWORD    Length;
03466         WORD     NumberOfRelocations;
03467         WORD     NumberOfLinenumbers;
03468         DWORD    CheckSum;
03469         SHORT    Number;
03470         BYTE     Selection;
03471     } Section;
03472 } IMAGE_AUX_SYMBOL;
03473 typedef IMAGE_AUX_SYMBOL *PIMAGE_AUX_SYMBOL;
03474
03475 #define IMAGE_SIZEOF_AUX_SYMBOL 18
03476
03477 #include "poppack.h"
03478
03479 #define IMAGE_SYM_UNDEFINED          (SHORT)0
03480 #define IMAGE_SYM_ABSOLUTE          (SHORT)-1
03481 #define IMAGE_SYM_DEBUG              (SHORT)-2
03482
03483 #define IMAGE_SYM_TYPE_NULL          0x0000
03484 #define IMAGE_SYM_TYPE_VOID          0x0001
03485 #define IMAGE_SYM_TYPE_CHAR          0x0002
03486 #define IMAGE_SYM_TYPE_SHORT         0x0003
03487 #define IMAGE_SYM_TYPE_INT           0x0004
03488 #define IMAGE_SYM_TYPE_LONG          0x0005
03489 #define IMAGE_SYM_TYPE_FLOAT         0x0006
03490 #define IMAGE_SYM_TYPE_DOUBLE        0x0007
03491 #define IMAGE_SYM_TYPE_STRUCT        0x0008
03492 #define IMAGE_SYM_TYPE_UNION         0x0009
03493 #define IMAGE_SYM_TYPE_ENUM          0x000A
03494 #define IMAGE_SYM_TYPE_MOE          0x000B
03495 #define IMAGE_SYM_TYPE_BYTE          0x000C
03496 #define IMAGE_SYM_TYPE_WORD          0x000D
03497 #define IMAGE_SYM_TYPE_UINT          0x000E
03498 #define IMAGE_SYM_TYPE_DWORD         0x000F
03499 #define IMAGE_SYM_TYPE_PCODE         0x8000
03500
03501 #define IMAGE_SYM_DTYPE_NULL          0

```

```

03502 #define IMAGE_SYM_DTYPE_POINTER 1
03503 #define IMAGE_SYM_DTYPE_FUNCTION 2
03504 #define IMAGE_SYM_DTYPE_ARRAY 3
03505
03506 #define IMAGE_SYM_CLASS_END_OF_FUNCTION (BYTE)-1
03507 #define IMAGE_SYM_CLASS_NULL 0x0000
03508 #define IMAGE_SYM_CLASS_AUTOMATIC 0x0001
03509 #define IMAGE_SYM_CLASS_EXTERNAL 0x0002
03510 #define IMAGE_SYM_CLASS_STATIC 0x0003
03511 #define IMAGE_SYM_CLASS_REGISTER 0x0004
03512 #define IMAGE_SYM_CLASS_EXTERNAL_DEF 0x0005
03513 #define IMAGE_SYM_CLASS_LABEL 0x0006
03514 #define IMAGE_SYM_CLASS_UNDEFINED_LABEL 0x0007
03515 #define IMAGE_SYM_CLASS_MEMBER_OF_STRUCT 0x0008
03516 #define IMAGE_SYM_CLASS_ARGUMENT 0x0009
03517 #define IMAGE_SYM_CLASS_STRUCT_TAG 0x000A
03518 #define IMAGE_SYM_CLASS_MEMBER_OF_UNION 0x000B
03519 #define IMAGE_SYM_CLASS_UNION_TAG 0x000C
03520 #define IMAGE_SYM_CLASS_TYPE_DEFINITION 0x000D
03521 #define IMAGE_SYM_CLASS_UNDEFINED_STATIC 0x000E
03522 #define IMAGE_SYM_CLASS_ENUM_TAG 0x000F
03523 #define IMAGE_SYM_CLASS_MEMBER_OF_ENUM 0x0010
03524 #define IMAGE_SYM_CLASS_REGISTER_PARAM 0x0011
03525 #define IMAGE_SYM_CLASS_BIT_FIELD 0x0012
03526
03527 #define IMAGE_SYM_CLASS_FAR_EXTERNAL 0x0044
03528 #define IMAGE_SYM_CLASS_BLOCK 0x0064
03529 #define IMAGE_SYM_CLASS_FUNCTION 0x0065
03530 #define IMAGE_SYM_CLASS_END_OF_STRUCT 0x0066
03531 #define IMAGE_SYM_CLASS_FILE 0x0067
03532 #define IMAGE_SYM_CLASS_SECTION 0x0068
03533 #define IMAGE_SYM_CLASS_WEAK_EXTERNAL 0x0069
03534
03535 #define N_BTMASK 0x000F
03536 #define N_TMASK 0x0030
03537 #define N_TMASK1 0x00C0
03538 #define N_TMASK2 0x00F0
03539 #define N_BTSHFT 4
03540 #define N_TSHFT 2
03541
03542 #define BYTYPE(x) ((x) & N_BTMASK)
03543
03544 #ifndef ISPTR
03545 #define ISPTR(x) (((x) & N_TMASK) == (IMAGE_SYM_DTYPE_POINTER << N_BTSHFT))
03546 #endif
03547
03548 #ifndef ISFCN
03549 #define ISFCN(x) (((x) & N_TMASK) == (IMAGE_SYM_DTYPE_FUNCTION << N_BTSHFT))
03550 #endif
03551
03552 #ifndef ISARY
03553 #define ISARY(x) (((x) & N_TMASK) == (IMAGE_SYM_DTYPE_ARRAY << N_BTSHFT))
03554 #endif
03555
03556 #ifndef ISTAG
03557 #define ISTAG(x) ((x) == IMAGE_SYM_CLASS_STRUCT_TAG || (x) == IMAGE_SYM_CLASS_UNION_TAG ||
(x) == IMAGE_SYM_CLASS_ENUM_TAG)
03558 #endif
03559
03560 #ifndef INCREF
03561 #define INCREF(x) (((x) & ~N_BTMASK) << N_TSHFT) | (IMAGE_SYM_DTYPE_POINTER << N_BTSHFT) | ((x) & N_BTMASK)
03562 #endif
03563 #ifndef DECREF
03564 #define DECREF(x) (((x) >> N_TSHFT) & ~N_BTMASK) | ((x) & N_BTMASK)
03565 #endif
03566
03567 #define IMAGE_COMDAT_SELECT_NODUPPLICATES 1
03568 #define IMAGE_COMDAT_SELECT_ANY 2
03569 #define IMAGE_COMDAT_SELECT_SAME_SIZE 3
03570 #define IMAGE_COMDAT_SELECT_EXACT_MATCH 4
03571 #define IMAGE_COMDAT_SELECT_ASSOCIATIVE 5
03572 #define IMAGE_COMDAT_SELECT_LARGEST 6
03573 #define IMAGE_COMDAT_SELECT_NEWEST 7
03574
03575 #define IMAGE_WEAK_EXTERN_SEARCH_NOLIBRARY 1
03576 #define IMAGE_WEAK_EXTERN_SEARCH_LIBRARY 2
03577 #define IMAGE_WEAK_EXTERN_SEARCH_ALIAS 3
03578
03579 /* Export module directory */
03580
03581 typedef struct _IMAGE_EXPORT_DIRECTORY {
03582     DWORD Characteristics;
03583     DWORD TimeDateStamp;
03584     WORD MajorVersion;
03585     WORD MinorVersion;
03586     DWORD Name;
03587     DWORD Base;

```

```

03588     DWORD    NumberOfFunctions;
03589     DWORD    NumberOfNames;
03590     DWORD    AddressOfFunctions;
03591     DWORD    AddressOfNames;
03592     DWORD    AddressOfNameOrdinals;
03593 } IMAGE_EXPORT_DIRECTORY, *PIMAGE_EXPORT_DIRECTORY;
03594
03595 /* Import name entry */
03596 typedef struct _IMAGE_IMPORT_BY_NAME {
03597     WORD    Hint;
03598     BYTE    Name[1];
03599 } IMAGE_IMPORT_BY_NAME, *PIMAGE_IMPORT_BY_NAME;
03600
03601 /* Import thunk */
03602 typedef struct _IMAGE_THUNK_DATA {
03603     union {
03604         LPBYTE    ForwarderString;
03605         PDWORD    Function;
03606         DWORD     Ordinal;
03607         PIMAGE_IMPORT_BY_NAME    AddressOfData;
03608     } u;
03609 } IMAGE_THUNK_DATA, *PIMAGE_THUNK_DATA;
03610
03611 /* Import module directory */
03612
03613 typedef struct _IMAGE_IMPORT_DESCRIPTOR {
03614     union {
03615         DWORD    Characteristics; /* 0 for terminating null import descriptor */
03616         PIMAGE_THUNK_DATA    OriginalFirstThunk; /* RVA to original unbound IAT */
03617     } u;
03618     DWORD    TimeDateStamp; /* 0 if not bound,
03619                             * -1 if bound, and real date\time stamp
03620                             * in IMAGE_DIRECTORY_ENTRY_BOUND_IMPORT
03621                             * (new BIND)
03622                             * otherwise date/time stamp of DLL bound to
03623                             * (Old BIND)
03624                             */
03625     DWORD    ForwarderChain; /* -1 if no forwarders */
03626     DWORD    Name;
03627     /* RVA to IAT (if bound this IAT has actual addresses) */
03628     PIMAGE_THUNK_DATA    FirstThunk;
03629 } IMAGE_IMPORT_DESCRIPTOR, *PIMAGE_IMPORT_DESCRIPTOR;
03630
03631 #define IMAGE_ORDINAL_FLAG        0x80000000
03632 #define IMAGE_SNAP_BY_ORDINAL(Ordinal)    ((Ordinal & IMAGE_ORDINAL_FLAG) != 0)
03633 #define IMAGE_ORDINAL(Ordinal)    (Ordinal & 0xffff)
03634
03635 typedef struct _IMAGE_BOUND_IMPORT_DESCRIPTOR {
03636 {
03637     DWORD    TimeDateStamp;
03638     WORD     OffsetModuleName;
03639     WORD     NumberOfModuleForwarderRefs;
03640 /* Array of zero or more IMAGE_BOUND_FORWARDER_REF follows */
03641 } IMAGE_BOUND_IMPORT_DESCRIPTOR, *PIMAGE_BOUND_IMPORT_DESCRIPTOR;
03642
03643 typedef struct _IMAGE_BOUND_FORWARDER_REF {
03644 {
03645     DWORD    TimeDateStamp;
03646     WORD     OffsetModuleName;
03647     WORD     Reserved;
03648 } IMAGE_BOUND_FORWARDER_REF, *PIMAGE_BOUND_FORWARDER_REF;
03649
03650 #include "pshpack2.h"
03651
03652 typedef struct _IMAGE_BASE_RELOCATION {
03653 {
03654     DWORD    VirtualAddress;
03655     DWORD    SizeOfBlock;
03656     /* WORD TypeOffset[1]; */
03657 } IMAGE_BASE_RELOCATION, *PIMAGE_BASE_RELOCATION;
03658
03659 typedef struct _IMAGE_RELOCATION {
03660 {
03661     union {
03662         DWORD    VirtualAddress;
03663         DWORD    RelocCount;
03664     } u;
03665     DWORD    SymbolTableIndex;
03666     WORD     Type;
03667 } IMAGE_RELOCATION;
03668 typedef IMAGE_RELOCATION *PIMAGE_RELOCATION;
03669
03670 #include "poppack.h"
03671
03672 #define IMAGE_SIZEOF_RELOCATION 10
03673
03674 /* generic relocation types */

```

```
03675 #define IMAGE_REL_BASED_ABSOLUTE 0
03676 #define IMAGE_REL_BASED_HIGH 1
03677 #define IMAGE_REL_BASED_LOW 2
03678 #define IMAGE_REL_BASED_HIGHLOW 3
03679 #define IMAGE_REL_BASED_HIGHADJ 4
03680 #define IMAGE_REL_BASED_MIPS_JMPADDR 5
03681 #define IMAGE_REL_BASED_SECTION 6
03682 #define IMAGE_REL_BASED_REL 7
03683 #define IMAGE_REL_BASED_MIPS_JMPADDR16 9
03684 #define IMAGE_REL_BASED_IA64_IMM64 9 /* yes, 9 too */
03685 #define IMAGE_REL_BASED_DIR64 10
03686 #define IMAGE_REL_BASED_HIGH3ADJ 11
03687
03688 /* I386 relocation types */
03689 #define IMAGE_REL_I386_ABSOLUTE 0
03690 #define IMAGE_REL_I386_DIR16 1
03691 #define IMAGE_REL_I386_REL16 2
03692 #define IMAGE_REL_I386_DIR32 6
03693 #define IMAGE_REL_I386_DIR32NB 7
03694 #define IMAGE_REL_I386_SEG12 9
03695 #define IMAGE_REL_I386_SECTION 10
03696 #define IMAGE_REL_I386_SECREL 11
03697 #define IMAGE_REL_I386_REL32 20
03698
03699 /* MIPS relocation types */
03700 #define IMAGE_REL_MIPS_ABSOLUTE 0x0000
03701 #define IMAGE_REL_MIPS_REFHALF 0x0001
03702 #define IMAGE_REL_MIPS_REFWORD 0x0002
03703 #define IMAGE_REL_MIPS_JMPADDR 0x0003
03704 #define IMAGE_REL_MIPS_REFHI 0x0004
03705 #define IMAGE_REL_MIPS_REFLO 0x0005
03706 #define IMAGE_REL_MIPS_GPREL 0x0006
03707 #define IMAGE_REL_MIPS_LITERAL 0x0007
03708 #define IMAGE_REL_MIPS_SECTION 0x000A
03709 #define IMAGE_REL_MIPS_SECREL 0x000B
03710 #define IMAGE_REL_MIPS_SECRELLO 0x000C
03711 #define IMAGE_REL_MIPS_SECRELHI 0x000D
03712 #define IMAGE_REL_MIPS_JMPADDR16 0x0010
03713 #define IMAGE_REL_MIPS_REFWORDNB 0x0022
03714 #define IMAGE_REL_MIPS_PAIR 0x0025
03715
03716 /* ALPHA relocation types */
03717 #define IMAGE_REL_ALPHA_ABSOLUTE 0x0000
03718 #define IMAGE_REL_ALPHA_REFLONG 0x0001
03719 #define IMAGE_REL_ALPHA_REFQUAD 0x0002
03720 #define IMAGE_REL_ALPHA_GPREL 0x0003
03721 #define IMAGE_REL_ALPHA_LITERAL 0x0004
03722 #define IMAGE_REL_ALPHA_LITUSE 0x0005
03723 #define IMAGE_REL_ALPHA_GPDISP 0x0006
03724 #define IMAGE_REL_ALPHA_BRADDR 0x0007
03725 #define IMAGE_REL_ALPHA_HINT 0x0008
03726 #define IMAGE_REL_ALPHA_INLINE_REFLONG 0x0009
03727 #define IMAGE_REL_ALPHA_REFHI 0x000A
03728 #define IMAGE_REL_ALPHA_REFLO 0x000B
03729 #define IMAGE_REL_ALPHA_PAIR 0x000C
03730 #define IMAGE_REL_ALPHA_MATCH 0x000D
03731 #define IMAGE_REL_ALPHA_SECTION 0x000E
03732 #define IMAGE_REL_ALPHA_SECREL 0x000F
03733 #define IMAGE_REL_ALPHA_REFLONGNB 0x0010
03734 #define IMAGE_REL_ALPHA_SECRELLO 0x0011
03735 #define IMAGE_REL_ALPHA_SECRELHI 0x0012
03736 #define IMAGE_REL_ALPHA_REFQ3 0x0013
03737 #define IMAGE_REL_ALPHA_REFQ2 0x0014
03738 #define IMAGE_REL_ALPHA_REFQ1 0x0015
03739 #define IMAGE_REL_ALPHA_GPRELLO 0x0016
03740 #define IMAGE_REL_ALPHA_GPRELHI 0x0017
03741
03742 /* PowerPC relocation types */
03743 #define IMAGE_REL_PPC_ABSOLUTE 0x0000
03744 #define IMAGE_REL_PPC_ADDR64 0x0001
03745 #define IMAGE_REL_PPC_ADDR 0x0002
03746 #define IMAGE_REL_PPC_ADDR24 0x0003
03747 #define IMAGE_REL_PPC_ADDR16 0x0004
03748 #define IMAGE_REL_PPC_ADDR14 0x0005
03749 #define IMAGE_REL_PPC_REL24 0x0006
03750 #define IMAGE_REL_PPC_REL14 0x0007
03751 #define IMAGE_REL_PPC_TOCREL16 0x0008
03752 #define IMAGE_REL_PPC_TOCREL14 0x0009
03753 #define IMAGE_REL_PPC_ADDR32NB 0x000A
03754 #define IMAGE_REL_PPC_SECREL 0x000B
03755 #define IMAGE_REL_PPC_SECTION 0x000C
03756 #define IMAGE_REL_PPC_IFGLUE 0x000D
03757 #define IMAGE_REL_PPC_IMGLUE 0x000E
03758 #define IMAGE_REL_PPC_SECREL16 0x000F
03759 #define IMAGE_REL_PPC_REFHI 0x0010
03760 #define IMAGE_REL_PPC_REFLO 0x0011
03761 #define IMAGE_REL_PPC_PAIR 0x0012
```

```

03762 #define IMAGE_REL_PPC_SECRELLO 0x0013
03763 #define IMAGE_REL_PPC_SECRELHI 0x0014
03764 #define IMAGE_REL_PPC_GPREL 0x0015
03765 #define IMAGE_REL_PPC_TYEMASK 0x00FF
03766 /* modifier bits */
03767 #define IMAGE_REL_PPC_NEG 0x0100
03768 #define IMAGE_REL_PPC_BRTAKEN 0x0200
03769 #define IMAGE_REL_PPC_BRNTAKEN 0x0400
03770 #define IMAGE_REL_PPC_TOCDEFN 0x0800
03771
03772 /* SH3 ? relocation type */
03773 #define IMAGE_REL_SH3_ABSOLUTE 0x0000
03774 #define IMAGE_REL_SH3_DIRECT16 0x0001
03775 #define IMAGE_REL_SH3_DIRECT 0x0002
03776 #define IMAGE_REL_SH3_DIRECT8 0x0003
03777 #define IMAGE_REL_SH3_DIRECT8_WORD 0x0004
03778 #define IMAGE_REL_SH3_DIRECT8_LONG 0x0005
03779 #define IMAGE_REL_SH3_DIRECT4 0x0006
03780 #define IMAGE_REL_SH3_DIRECT4_WORD 0x0007
03781 #define IMAGE_REL_SH3_DIRECT4_LONG 0x0008
03782 #define IMAGE_REL_SH3_PCREL8_WORD 0x0009
03783 #define IMAGE_REL_SH3_PCREL8_LONG 0x000A
03784 #define IMAGE_REL_SH3_PCREL12_WORD 0x000B
03785 #define IMAGE_REL_SH3_STARTOF_SECTION 0x000C
03786 #define IMAGE_REL_SH3_SIZEOF_SECTION 0x000D
03787 #define IMAGE_REL_SH3_SECTION 0x000E
03788 #define IMAGE_REL_SH3_SECREL 0x000F
03789 #define IMAGE_REL_SH3_DIRECT32_NB 0x0010
03790
03791 /* ARM (Archimedes?) relocation types */
03792 #define IMAGE_REL_ARM_ABSOLUTE 0x0000
03793 #define IMAGE_REL_ARM_ADDR 0x0001
03794 #define IMAGE_REL_ARM_ADDR32NB 0x0002
03795 #define IMAGE_REL_ARM_BRANCH24 0x0003
03796 #define IMAGE_REL_ARM_BRANCH11 0x0004
03797 #define IMAGE_REL_ARM_SECTION 0x000E
03798 #define IMAGE_REL_ARM_SECREL 0x000F
03799
03800 /* IA64 relocation types */
03801 #define IMAGE_REL_IA64_ABSOLUTE 0x0000
03802 #define IMAGE_REL_IA64_IMM14 0x0001
03803 #define IMAGE_REL_IA64_IMM22 0x0002
03804 #define IMAGE_REL_IA64_IMM64 0x0003
03805 #define IMAGE_REL_IA64_DIR 0x0004
03806 #define IMAGE_REL_IA64_DIR64 0x0005
03807 #define IMAGE_REL_IA64_PCREL21B 0x0006
03808 #define IMAGE_REL_IA64_PCREL21M 0x0007
03809 #define IMAGE_REL_IA64_PCREL21F 0x0008
03810 #define IMAGE_REL_IA64_GPREL22 0x0009
03811 #define IMAGE_REL_IA64_LTOFF22 0x000A
03812 #define IMAGE_REL_IA64_SECTION 0x000B
03813 #define IMAGE_REL_IA64_SECREL22 0x000C
03814 #define IMAGE_REL_IA64_SECREL64I 0x000D
03815 #define IMAGE_REL_IA64_SECREL 0x000E
03816 #define IMAGE_REL_IA64_LTOFF64 0x000F
03817 #define IMAGE_REL_IA64_DIR32NB 0x0010
03818 #define IMAGE_REL_IA64_RESERVED_11 0x0011
03819 #define IMAGE_REL_IA64_RESERVED_12 0x0012
03820 #define IMAGE_REL_IA64_RESERVED_13 0x0013
03821 #define IMAGE_REL_IA64_RESERVED_14 0x0014
03822 #define IMAGE_REL_IA64_RESERVED_15 0x0015
03823 #define IMAGE_REL_IA64_RESERVED_16 0x0016
03824 #define IMAGE_REL_IA64_ADDEND 0x001F
03825
03826 /* archive format */
03827
03828 #define IMAGE_ARCHIVE_START_SIZE 8
03829 #define IMAGE_ARCHIVE_START " !<arch>\n"
03830 #define IMAGE_ARCHIVE_END "`\n"
03831 #define IMAGE_ARCHIVE_PAD "\n"
03832 #define IMAGE_ARCHIVE_LINKER_MEMBER "/"
03833 #define IMAGE_ARCHIVE_LONGNAMES_MEMBER "//"
03834
03835 typedef struct _IMAGE_ARCHIVE_MEMBER_HEADER
03836 {
03837     BYTE Name[16];
03838     BYTE Date[12];
03839     BYTE UserID[6];
03840     BYTE GroupID[6];
03841     BYTE Mode[8];
03842     BYTE Size[10];
03843     BYTE EndHeader[2];
03844 } IMAGE_ARCHIVE_MEMBER_HEADER, *PIMAGE_ARCHIVE_MEMBER_HEADER;
03845
03846 #define IMAGE_SIZEOF_ARCHIVE_MEMBER_HDR 60
03847
03848 /*

```

```

03849  * Resource directory stuff
03850  */
03851  typedef struct _IMAGE_RESOURCE_DIRECTORY {
03852      DWORD      Characteristics;
03853      DWORD      TimeDateStamp;
03854      WORD       MajorVersion;
03855      WORD       MinorVersion;
03856      WORD       NumberOfNamedEntries;
03857      WORD       NumberOfIdEntries;
03858      /* IMAGE_RESOURCE_DIRECTORY_ENTRY DirectoryEntries[]; */
03859  } IMAGE_RESOURCE_DIRECTORY, *PIMAGE_RESOURCE_DIRECTORY;
03860
03861  #define IMAGE_RESOURCE_NAME_IS_STRING      0x80000000
03862  #define IMAGE_RESOURCE_DATA_IS_DIRECTORY  0x80000000
03863
03864  typedef struct _IMAGE_RESOURCE_DIRECTORY_ENTRY {
03865      union u1 {
03866          struct fleegle {
03867              #ifdef BITFIELDS_BIGENDIAN
03868                  unsigned NameIsString:1;
03869                  unsigned NameOffset:31;
03870              #else
03871                  unsigned NameOffset:31;
03872                  unsigned NameIsString:1;
03873              #endif
03874          } DUMMYSTRUCTNAME1;
03875          DWORD      Name;
03876          struct sneegle {
03877              #ifdef WORDS_BIGENDIAN
03878                  WORD      __pad;
03879                  WORD      Id;
03880              #else
03881                  WORD      Id;
03882                  WORD      __pad;
03883              #endif
03884          } DUMMYSTRUCTNAME2;
03885      } DUMMYUNIONNAME1;
03886      union u2 {
03887          DWORD      OffsetToData;
03888          struct drooper {
03889              #ifdef BITFIELDS_BIGENDIAN
03890                  unsigned DataIsDirectory:1;
03891                  unsigned OffsetToDirectory:31;
03892              #else
03893                  unsigned OffsetToDirectory:31;
03894                  unsigned DataIsDirectory:1;
03895              #endif
03896          } DUMMYSTRUCTNAME3;
03897      } DUMMYUNIONNAME2;
03898  } IMAGE_RESOURCE_DIRECTORY_ENTRY, *PIMAGE_RESOURCE_DIRECTORY_ENTRY;
03899
03900
03901  typedef struct _IMAGE_RESOURCE_DIRECTORY_STRING {
03902      WORD      Length;
03903      CHAR      NameString[ 1 ];
03904  } IMAGE_RESOURCE_DIRECTORY_STRING, *PIMAGE_RESOURCE_DIRECTORY_STRING;
03905
03906  typedef struct _IMAGE_RESOURCE_DIR_STRING_U {
03907      WORD      Length;
03908      WCHAR     NameString[ 1 ];
03909  } IMAGE_RESOURCE_DIR_STRING_U, *PIMAGE_RESOURCE_DIR_STRING_U;
03910
03911  typedef struct _IMAGE_RESOURCE_DATA_ENTRY {
03912      DWORD      OffsetToData;
03913      DWORD      Size;
03914      DWORD      CodePage;
03915      DWORD      ResourceHandle;
03916  } IMAGE_RESOURCE_DATA_ENTRY, *PIMAGE_RESOURCE_DATA_ENTRY;
03917
03918
03919  typedef VOID CALLBACK (*PIMAGE_TLS_CALLBACK) (
03920      LPVOID DllHandle, DWORD Reason, LPVOID Reserved
03921  );
03922
03923  typedef struct _IMAGE_TLS_DIRECTORY {
03924      DWORD      StartAddressOfRawData;
03925      DWORD      EndAddressOfRawData;
03926      LPDWORD    AddressOfIndex;
03927      PIMAGE_TLS_CALLBACK *AddressOfCallBacks;
03928      DWORD      SizeOfZeroFill;
03929      DWORD      Characteristics;
03930  } IMAGE_TLS_DIRECTORY, *PIMAGE_TLS_DIRECTORY;
03931
03932  typedef struct _IMAGE_DEBUG_DIRECTORY {
03933      DWORD      Characteristics;
03934      DWORD      TimeDateStamp;
03935      WORD       MajorVersion;

```

```

03936 WORD MinorVersion;
03937 DWORD Type;
03938 DWORD SizeOfData;
03939 DWORD AddressOfRawData;
03940 DWORD PointerToRawData;
03941 } IMAGE_DEBUG_DIRECTORY, *PIMAGE_DEBUG_DIRECTORY;
03942
03943 #define IMAGE_DEBUG_TYPE_UNKNOWN 0
03944 #define IMAGE_DEBUG_TYPE_COFF 1
03945 #define IMAGE_DEBUG_TYPE_CODEVIEW 2
03946 #define IMAGE_DEBUG_TYPE_FPO 3
03947 #define IMAGE_DEBUG_TYPE_MISC 4
03948 #define IMAGE_DEBUG_TYPE_EXCEPTION 5
03949 #define IMAGE_DEBUG_TYPE_FIXUP 6
03950 #define IMAGE_DEBUG_TYPE_OMAP_TO_SRC 7
03951 #define IMAGE_DEBUG_TYPE_OMAP_FROM_SRC 8
03952 #define IMAGE_DEBUG_TYPE_BORLAND 9
03953 #define IMAGE_DEBUG_TYPE_RESERVED10 10
03954
03955 typedef struct _IMAGE_COFF_SYMBOLS_HEADER {
03956     DWORD NumberOfSymbols;
03957     DWORD LvaToFirstSymbol;
03958     DWORD NumberOfLinenumbers;
03959     DWORD LvaToFirstLinenumber;
03960     DWORD RvaToFirstByteOfCode;
03961     DWORD RvaToLastByteOfCode;
03962     DWORD RvaToFirstByteOfData;
03963     DWORD RvaToLastByteOfData;
03964 } IMAGE_COFF_SYMBOLS_HEADER, *PIMAGE_COFF_SYMBOLS_HEADER;
03965
03966 #define FRAME_FPO 0
03967 #define FRAME_TRAP 1
03968 #define FRAME_TSS 2
03969 #define FRAME_NONFPO 3
03970
03971 typedef struct _FPO_DATA {
03972     DWORD ulOffStart;
03973     DWORD cbProcSize;
03974     DWORD cdwLocals;
03975     WORD cdwParams;
03976     unsigned cbProlog : 8;
03977     unsigned cbRegs : 3;
03978     unsigned fHasSEH : 1;
03979     unsigned fUseBP : 1;
03980     unsigned reserved : 1;
03981     unsigned cbFrame : 2;
03982 } FPO_DATA, *PFPO_DATA;
03983
03984 typedef struct _IMAGE_LOAD_CONFIG_DIRECTORY {
03985     DWORD Characteristics;
03986     DWORD TimeDateStamp;
03987     WORD MajorVersion;
03988     WORD MinorVersion;
03989     DWORD GlobalFlagsClear;
03990     DWORD GlobalFlagsSet;
03991     DWORD CriticalSectionDefaultTimeout;
03992     DWORD DeCommitFreeBlockThreshold;
03993     DWORD DeCommitTotalFreeThreshold;
03994     PVOID LockPrefixTable;
03995     DWORD MaximumAllocationSize;
03996     DWORD VirtualMemoryThreshold;
03997     DWORD ProcessHeapFlags;
03998     DWORD ProcessAffinityMask;
03999     WORD CSDVersion;
04000     WORD Reserved1;
04001     PVOID EditList;
04002     DWORD Reserved[1];
04003 } IMAGE_LOAD_CONFIG_DIRECTORY, *PIMAGE_LOAD_CONFIG_DIRECTORY;
04004
04005 typedef struct _IMAGE_FUNCTION_ENTRY {
04006     DWORD StartingAddress;
04007     DWORD EndingAddress;
04008     DWORD EndOfPrologue;
04009 } IMAGE_FUNCTION_ENTRY, *PIMAGE_FUNCTION_ENTRY;
04010
04011 #define IMAGE_DEBUG_MISC_EXENAME 1
04012
04013 typedef struct _IMAGE_DEBUG_MISC {
04014     DWORD DataType;
04015     DWORD Length;
04016     BYTE Unicode;
04017     BYTE Reserved[ 3 ];
04018     BYTE Data[ 1 ];
04019 } IMAGE_DEBUG_MISC, *PIMAGE_DEBUG_MISC;
04020
04021 /* This is the structure that appears at the very start of a .DBG file. */
04022

```

```

04023 typedef struct _IMAGE_SEPARATE_DEBUG_HEADER {
04024     WORD    Signature;
04025     WORD    Flags;
04026     WORD    Machine;
04027     WORD    Characteristics;
04028     DWORD   TimeDateStamp;
04029     DWORD   CheckSum;
04030     DWORD   ImageBase;
04031     DWORD   SizeOfImage;
04032     DWORD   NumberOfSections;
04033     DWORD   ExportedNamesSize;
04034     DWORD   DebugDirectorySize;
04035     DWORD   SectionAlignment;
04036     DWORD   Reserved[ 2 ];
04037 } IMAGE_SEPARATE_DEBUG_HEADER, *PIMAGE_SEPARATE_DEBUG_HEADER;
04038
04039 #define IMAGE_SEPARATE_DEBUG_SIGNATURE 0x4944
04040
04041
04042 typedef struct tagMESSAGE_RESOURCE_ENTRY {
04043     WORD    Length;
04044     WORD    Flags;
04045     BYTE    Text[1];
04046 } MESSAGE_RESOURCE_ENTRY, *PMESSAGE_RESOURCE_ENTRY;
04047 #define MESSAGE_RESOURCE_UNICODE 0x0001
04048
04049 typedef struct tagMESSAGE_RESOURCE_BLOCK {
04050     DWORD   LowId;
04051     DWORD   HighId;
04052     DWORD   OffsetToEntries;
04053 } MESSAGE_RESOURCE_BLOCK, *PMESSAGE_RESOURCE_BLOCK;
04054
04055 typedef struct tagMESSAGE_RESOURCE_DATA {
04056     DWORD   NumberOfBlocks;
04057     MESSAGE_RESOURCE_BLOCK Blocks[ 1 ];
04058 } MESSAGE_RESOURCE_DATA, *PMESSAGE_RESOURCE_DATA;
04059
04060 /*
04061  * Here follows typedefs for security and tokens.
04062  */
04063
04064 /*
04065  * First a constant for the following typedefs.
04066  */
04067
04068 #define ANYSIZE_ARRAY 1
04069
04070 /* FIXME: Orphan. What does it point to? */
04071 typedef PVOID PACCESS_TOKEN;
04072
04073 /*
04074  * TOKEN_INFORMATION_CLASS
04075  */
04076
04077 typedef enum _TOKEN_INFORMATION_CLASS {
04078     TokenUser = 1,
04079     TokenGroups,
04080     TokenPrivileges,
04081     TokenOwner,
04082     TokenPrimaryGroup,
04083     TokenDefaultDacl,
04084     TokenSource,
04085     TokenType,
04086     TokenImpersonationLevel,
04087     TokenStatistics
04088 } TOKEN_INFORMATION_CLASS;
04089
04090 #define TOKEN_TOKEN_ADJUST_DEFAULT 0x0080
04091 #define TOKEN_ADJUST_GROUPS 0x0040
04092 #define TOKEN_ADJUST_PRIVILEGES 0x0020
04093 #define TOKEN_ADJUST_SESSIONID 0x0100
04094 #define TOKEN_ASSIGN_PRIMARY 0x0001
04095 #define TOKEN_DUPLICATE 0x0002
04096 #define TOKEN_EXECUTE STANDARD_RIGHTS_EXECUTE
04097 #define TOKEN_IMPERSONATE 0x0004
04098 #define TOKEN_QUERY 0x0008
04099 #define TOKEN_QUERY_SOURCE 0x0010
04100 #define TOKEN_ADJUST_DEFAULT 0x0080
04101 #define TOKEN_READ (STANDARD_RIGHTS_READ|TOKEN_QUERY)
04102 #define TOKEN_WRITE (STANDARD_RIGHTS_WRITE | \
04103     TOKEN_ADJUST_PRIVILEGES | \
04104     TOKEN_ADJUST_GROUPS | \
04105     TOKEN_ADJUST_DEFAULT )
04106 #define TOKEN_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED | \
04107     TOKEN_ASSIGN_PRIMARY | \
04108     TOKEN_DUPLICATE | \
04109     TOKEN_IMPERSONATE | \

```

```

04110             TOKEN_QUERY | \
04111             TOKEN_QUERY_SOURCE | \
04112             TOKEN_ADJUST_PRIVILEGES | \
04113             TOKEN_ADJUST_GROUPS | \
04114             TOKEN_ADJUST_SESSIONID | \
04115             TOKEN_ADJUST_DEFAULT )
04116
04117 #ifndef _SECURITY_DEFINED
04118 #define _SECURITY_DEFINED
04119
04120
04121 typedef DWORD ACCESS_MASK, *PACCESS_MASK;
04122
04123 typedef struct _GENERIC_MAPPING {
04124     ACCESS_MASK GenericRead;
04125     ACCESS_MASK GenericWrite;
04126     ACCESS_MASK GenericExecute;
04127     ACCESS_MASK GenericAll;
04128 } GENERIC_MAPPING, *PGENERIC_MAPPING;
04129
04130 #ifndef SID_IDENTIFIER_AUTHORITY_DEFINED
04131 #define SID_IDENTIFIER_AUTHORITY_DEFINED
04132 typedef struct {
04133     BYTE Value[6];
04134 } SID_IDENTIFIER_AUTHORITY, *PSID_IDENTIFIER_AUTHORITY, *LPSID_IDENTIFIER_AUTHORITY;
04135 #endif /* !defined(SID_IDENTIFIER_AUTHORITY_DEFINED) */
04136
04137 #ifndef SID_DEFINED
04138 #define SID_DEFINED
04139 typedef struct _SID {
04140     BYTE Revision;
04141     BYTE SubAuthorityCount;
04142     SID_IDENTIFIER_AUTHORITY IdentifierAuthority;
04143     DWORD SubAuthority[1];
04144 } SID, *PSID;
04145 #endif /* !defined(SID_DEFINED) */
04146
04147 #define SID_REVISION (1) /* Current revision */
04148 #define SID_MAX_SUB_AUTHORITIES (15) /* current max subauths */
04149 #define SID_RECOMMENDED_SUB_AUTHORITIES (1) /* recommended subauths */
04150
04151
04152 /*
04153  * ACL
04154  */
04155
04156 #define ACL_REVISION1 1
04157 #define ACL_REVISION2 2
04158 #define ACL_REVISION3 3
04159 #define ACL_REVISION4 4
04160
04161 #define MIN_ACL_REVISION ACL_REVISION2
04162 #define MAX_ACL_REVISION ACL_REVISION4
04163
04164 typedef struct _ACL {
04165     BYTE AclRevision;
04166     BYTE Sbz1;
04167     WORD AclSize;
04168     WORD AceCount;
04169     WORD Sbz2;
04170 } ACL, *PACL;
04171
04172 /* SECURITY_DESCRIPTOR */
04173 #define SECURITY_DESCRIPTOR_REVISION 1
04174 #define SECURITY_DESCRIPTOR_REVISION1 1
04175
04176
04177 #define SE_OWNER_DEFAULTED 0x0001
04178 #define SE_GROUP_DEFAULTED 0x0002
04179 #define SE_DACL_PRESENT 0x0004
04180 #define SE_DACL_DEFAULTED 0x0008
04181 #define SE_SACL_PRESENT 0x0010
04182 #define SE_SACL_DEFAULTED 0x0020
04183 #define SE_SELF_RELATIVE 0x8000
04184
04185 typedef DWORD SECURITY_INFORMATION, *PSECURITY_INFORMATION;
04186 typedef WORD SECURITY_DESCRIPTOR_CONTROL, *PSECURITY_DESCRIPTOR_CONTROL;
04187
04188 /* The security descriptor structure */
04189 typedef struct {
04190     BYTE Revision;
04191     BYTE Sbz1;
04192     SECURITY_DESCRIPTOR_CONTROL Control;
04193     DWORD Owner;
04194     DWORD Group;
04195     DWORD SACL;
04196     DWORD Dacl;

```

```
04197 } SECURITY_DESCRIPTOR_RELATIVE, *PISECURITY_DESCRIPTOR_RELATIVE;
04198
04199 typedef struct {
04200     BYTE Revision;
04201     BYTE Sbz1;
04202     SECURITY_DESCRIPTOR_CONTROL Control;
04203     PSID Owner;
04204     PSID Group;
04205     PACL Sacl;
04206     PACL Dacl;
04207 } SECURITY_DESCRIPTOR, *PSECURITY_DESCRIPTOR;
04208
04209 #define SECURITY_DESCRIPTOR_MIN_LENGTH    (sizeof(SECURITY_DESCRIPTOR))
04210
04211 #endif /* _SECURITY_DEFINED */
04212
04213 /*
04214  * SID_AND_ATTRIBUTES
04215  */
04216
04217 typedef struct _SID_AND_ATTRIBUTES {
04218     PSID Sid;
04219     DWORD Attributes;
04220 } SID_AND_ATTRIBUTES ;
04221
04222 /* security entities */
04223 #define SECURITY_NULL_RID                (0x00000000L)
04224 #define SECURITY_WORLD_RID               (0x00000000L)
04225 #define SECURITY_LOCAL_RID               (0x00000000L)
04226
04227 #define SECURITY_NULL_SID_AUTHORITY      {0,0,0,0,0,0}
04228
04229 /* S-1-1 */
04230 #define SECURITY_WORLD_SID_AUTHORITY     {0,0,0,0,0,1}
04231
04232 /* S-1-2 */
04233 #define SECURITY_LOCAL_SID_AUTHORITY     {0,0,0,0,0,2}
04234
04235 /* S-1-3 */
04236 #define SECURITY_CREATOR_SID_AUTHORITY   {0,0,0,0,0,3}
04237 #define SECURITY_CREATOR_OWNER_RID       (0x00000000L)
04238 #define SECURITY_CREATOR_GROUP_RID       (0x00000001L)
04239 #define SECURITY_CREATOR_OWNER_SERVER_RID (0x00000002L)
04240 #define SECURITY_CREATOR_GROUP_SERVER_RID (0x00000003L)
04241
04242 /* S-1-4 */
04243 #define SECURITY_NON_UNIQUE_AUTHORITY    {0,0,0,0,0,4}
04244
04245 /* S-1-5 */
04246 #define SECURITY_NT_AUTHORITY             {0,0,0,0,0,5}
04247 #define SECURITY_DIALUP_RID               0x00000001L
04248 #define SECURITY_NETWORK_RID              0x00000002L
04249 #define SECURITY_BATCH_RID                0x00000003L
04250 #define SECURITY_INTERACTIVE_RID          0x00000004L
04251 #define SECURITY_LOGON_IDS_RID            0x00000005L
04252 #define SECURITY_SERVICE_RID              0x00000006L
04253 #define SECURITY_ANONYMOUS_LOGON_RID      0x00000007L
04254 #define SECURITY_PROXY_RID                0x00000008L
04255 #define SECURITY_ENTERPRISE_CONTROLLERS_RID 0x00000009L
04256 #define SECURITY_PRINCIPAL_SELF_RID       0x0000000AL
04257 #define SECURITY_AUTHENTICATED_USER_RID   0x0000000BL
04258 #define SECURITY_RESTRICTED_CODE_RID      0x0000000CL
04259 #define SECURITY_TERMINAL_SERVER_RID      0x0000000DL
04260 #define SECURITY_LOCAL_SYSTEM_RID         0x00000012L
04261 #define SECURITY_NT_NON_UNIQUE            0x00000015L
04262 #define SECURITY_BUILTIN_DOMAIN_RID       0x00000020L
04263
04264 #define DOMAIN_GROUP_RID_ADMINS           0x00000200L
04265 #define DOMAIN_GROUP_RID_USERS            0x00000201L
04266 #define DOMAIN_GROUP_RID_GUESTS          0x00000202L
04267
04268 #define DOMAIN_ALIAS_RID_ADMINS           0x00000220L
04269 #define DOMAIN_ALIAS_RID_USERS            0x00000221L
04270 #define DOMAIN_ALIAS_RID_GUESTS           0x00000222L
04271
04272 #define SECURITY_SERVER_LOGON_RID          SECURITY_ENTERPRISE_CONTROLLERS_RID
04273
04274 #define SECURITY_LOGON_IDS_RID_COUNT      (3L)
04275
04276 /*
04277  * TOKEN_USER
04278  */
04279
04280 typedef struct _TOKEN_USER {
04281     SID_AND_ATTRIBUTES User;
04282 } TOKEN_USER;
04283
```

```
04284 /*
04285  * TOKEN_GROUPS
04286 */
04287
04288 typedef struct _TOKEN_GROUPS {
04289     DWORD GroupCount;
04290     SID_AND_ATTRIBUTES Groups[ANYSIZE_ARRAY];
04291 } TOKEN_GROUPS;
04292
04293 /*
04294  * LUID_AND_ATTRIBUTES
04295 */
04296
04297 typedef union _LARGE_INTEGER {
04298     struct dorp {
04299         DWORD LowPart;
04300         LONG HighPart;
04301     } DUMMYSTRUCTNAME;
04302     LONGLONG QuadPart;
04303 } LARGE_INTEGER, *LPLARGE_INTEGER, *PLARGE_INTEGER;
04304
04305 typedef union _ULARGE_INTEGER {
04306     struct banana {
04307         DWORD LowPart;
04308         DWORD HighPart;
04309     } DUMMYSTRUCTNAME;
04310     ULONGLONG QuadPart;
04311 } ULARGE_INTEGER, *LPULARGE_INTEGER, *PULARGE_INTEGER;
04312
04313 /*
04314  * Locally Unique Identifier
04315 */
04316
04317 typedef struct _LUID {
04318     DWORD LowPart;
04319     LONG HighPart;
04320 } LUID, *PLUID;
04321
04322 #include "pshpack4.h"
04323 typedef struct _LUID_AND_ATTRIBUTES {
04324     LUID Luid;
04325     DWORD Attributes;
04326 } LUID_AND_ATTRIBUTES;
04327 #include "poppack.h"
04328
04329 /*
04330  * PRIVILEGE_SET
04331 */
04332
04333 typedef struct _PRIVILEGE_SET {
04334     DWORD PrivilegeCount;
04335     DWORD Control;
04336     LUID_AND_ATTRIBUTES Privilege[ANYSIZE_ARRAY];
04337 } PRIVILEGE_SET, *PPRIVILEGE_SET;
04338
04339 /*
04340  * TOKEN_PRIVILEGES
04341 */
04342
04343 typedef struct _TOKEN_PRIVILEGES {
04344     DWORD PrivilegeCount;
04345     LUID_AND_ATTRIBUTES Privileges[ANYSIZE_ARRAY];
04346 } TOKEN_PRIVILEGES, *PTOKEN_PRIVILEGES;
04347
04348 /*
04349  * TOKEN_OWNER
04350 */
04351
04352 typedef struct _TOKEN_OWNER {
04353     PSID Owner;
04354 } TOKEN_OWNER;
04355
04356 /*
04357  * TOKEN_PRIMARY_GROUP
04358 */
04359
04360 typedef struct _TOKEN_PRIMARY_GROUP {
04361     PSID PrimaryGroup;
04362 } TOKEN_PRIMARY_GROUP;
04363
04364 /*
04365  * TOKEN_DEFAULT_DACL
04366 */
04367
04368 typedef struct _TOKEN_DEFAULT_DACL {
04369     PACL DefaultDacl;
```

```
04371 } TOKEN_DEFAULT_DACL;
04372
04373 /*
04374  * TOKEN_SOURCE
04375  */
04376
04377 typedef struct _TOKEN_SOURCE {
04378     char Sourcename[8];
04379     LUID SourceIdentifier;
04380 } TOKEN_SOURCE;
04381
04382 /*
04383  * TOKEN_TYPE
04384  */
04385
04386 typedef enum tagTOKEN_TYPE {
04387     TokenPrimary = 1,
04388     TokenImpersonation
04389 } TOKEN_TYPE;
04390
04391 /*
04392  * SECURITY_IMPERSONATION_LEVEL
04393  */
04394
04395 typedef enum _SECURITY_IMPERSONATION_LEVEL {
04396     SecurityAnonymous,
04397     SecurityIdentification,
04398     SecurityImpersonation,
04399     SecurityDelegation
04400 } SECURITY_IMPERSONATION_LEVEL, *PSECURITY_IMPERSONATION_LEVEL;
04401
04402
04403 typedef BOOLEAN SECURITY_CONTEXT_TRACKING_MODE,
04404     * PSECURITY_CONTEXT_TRACKING_MODE;
04405
04406 /*
04407  * Quality of Service
04408  */
04409
04410 typedef struct _SECURITY_QUALITY_OF_SERVICE {
04411     DWORD Length;
04412     SECURITY_IMPERSONATION_LEVEL ImpersonationLevel;
04413     SECURITY_CONTEXT_TRACKING_MODE ContextTrackingMode;
04414     BOOLEAN EffectiveOnly;
04415 } SECURITY_QUALITY_OF_SERVICE, *PSECURITY_QUALITY_OF_SERVICE;
04416
04417 /*
04418  * TOKEN_STATISTICS
04419  */
04420
04421 typedef struct _TOKEN_STATISTICS {
04422     LUID TokenId;
04423     LUID AuthenticationId;
04424     LARGE_INTEGER ExpirationTime;
04425     TOKEN_TYPE TokenType;
04426     SECURITY_IMPERSONATION_LEVEL ImpersonationLevel;
04427     DWORD DynamicCharged;
04428     DWORD DynamicAvailable;
04429     DWORD GroupCount;
04430     DWORD PrivilegeCount;
04431     LUID ModifiedId;
04432 } TOKEN_STATISTICS;
04433
04434 /*
04435  * ACLs of NT
04436  */
04437 #define ACL_REVISION 2
04438
04439 #define ACL_REVISION1 1
04440 #define ACL_REVISION2 2
04441
04442 /* ACEs, directly starting after an ACL */
04443 typedef struct _ACE_HEADER {
04444     BYTE AceType;
04445     BYTE AceFlags;
04446     WORD AceSize;
04447 } ACE_HEADER, *PACE_HEADER;
04448
04449 /* AceType */
04450 #define ACCESS_ALLOWED_ACE_TYPE 0
04451 #define ACCESS_DENIED_ACE_TYPE 1
04452 #define SYSTEM_AUDIT_ACE_TYPE 2
04453 #define SYSTEM_ALARM_ACE_TYPE 3
04454
04455 /* inherit AceFlags */
04456 #define OBJECT_INHERIT_ACE 0x01
04457 #define CONTAINER_INHERIT_ACE 0x02
```

```

04458 #define NO_PROPAGATE_INHERIT_ACE    0x04
04459 #define INHERIT_ONLY_ACE             0x08
04460 #define VALID_INHERIT_FLAGS          0x0F
04461
04462 /* AceFlags mask for what events we (should) audit */
04463 #define SUCCESSFUL_ACCESS_ACE_FLAG    0x40
04464 #define FAILED_ACCESS_ACE_FLAG        0x80
04465
04466 /* different ACEs depending on AceType
04467  * SidStart marks the begin of a SID
04468  * so the thing finally looks like this:
04469  * 0: ACE_HEADER
04470  * 4: ACCESS_MASK
04471  * 8... : SID
04472  */
04473 typedef struct _ACCESS_ALLOWED_ACE {
04474     ACE_HEADER Header;
04475     DWORD      Mask;
04476     DWORD      SidStart;
04477 } ACCESS_ALLOWED_ACE, *PACCESS_ALLOWED_ACE;
04478
04479 typedef struct _ACCESS_DENIED_ACE {
04480     ACE_HEADER Header;
04481     DWORD      Mask;
04482     DWORD      SidStart;
04483 } ACCESS_DENIED_ACE, *PACCESS_DENIED_ACE;
04484
04485 typedef struct _SYSTEM_AUDIT_ACE {
04486     ACE_HEADER Header;
04487     DWORD      Mask;
04488     DWORD      SidStart;
04489 } SYSTEM_AUDIT_ACE, *PSYSTEM_AUDIT_ACE;
04490
04491 typedef struct _SYSTEM_ALARM_ACE {
04492     ACE_HEADER Header;
04493     DWORD      Mask;
04494     DWORD      SidStart;
04495 } SYSTEM_ALARM_ACE, *PSYSTEM_ALARM_ACE;
04496
04497 typedef enum tagSID_NAME_USE {
04498     SidTypeUser = 1,
04499     SidTypeGroup,
04500     SidTypeDomain,
04501     SidTypeAlias,
04502     SidTypeWellKnownGroup,
04503     SidTypeDeletedAccount,
04504     SidTypeInvalid,
04505     SidTypeUnknown
04506 } SID_NAME_USE, *PSID_NAME_USE;
04507
04508 /* Access rights */
04509
04510 /* DELETE may be already defined via /usr/include/arpa/nameser_compat.h */
04511 #undef DELETE
04512 #define DELETE                0x00010000
04513 #define READ_CONTROL           0x00020000
04514 #define WRITE_DAC              0x00040000
04515 #define WRITE_OWNER            0x00080000
04516 #define SYNCHRONIZE            0x00100000
04517 #define STANDARD_RIGHTS_REQUIRED 0x000f0000
04518
04519 #define STANDARD_RIGHTS_READ    READ_CONTROL
04520 #define STANDARD_RIGHTS_WRITE  READ_CONTROL
04521 #define STANDARD_RIGHTS_EXECUTE READ_CONTROL
04522
04523 #define STANDARD_RIGHTS_ALL     0x001f0000
04524
04525 #define SPECIFIC_RIGHTS_ALL     0x0000ffff
04526
04527 #define GENERIC_READ            0x80000000
04528 #define GENERIC_WRITE           0x40000000
04529 #define GENERIC_EXECUTE         0x20000000
04530 #define GENERIC_ALL             0x10000000
04531
04532 #define MAXIMUM_ALLOWED         0x02000000
04533 #define ACCESS_SYSTEM_SECURITY  0x01000000
04534
04535 #define EVENT_MODIFY_STATE      0x0002
04536 #define EVENT_ALL_ACCESS        (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x3)
04537
04538 #define SEMAPHORE_MODIFY_STATE  0x0002
04539 #define SEMAPHORE_ALL_ACCESS    (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x3)
04540
04541 #define MUTEX_MODIFY_STATE      0x0001
04542 #define MUTEX_ALL_ACCESS        (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x1)
04543
04544 #define TIMER_QUERY_STATE       0x0001

```

```

04545 #define TIMER_MODIFY_STATE          0x0002
04546 #define TIMER_ALL_ACCESS              (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x3)
04547
04548 #define PROCESS_TERMINATE             0x0001
04549 #define PROCESS_CREATE_THREAD         0x0002
04550 #define PROCESS_VM_OPERATION          0x0008
04551 #define PROCESS_VM_READ               0x0010
04552 #define PROCESS_VM_WRITE              0x0020
04553 #define PROCESS_DUP_HANDLE            0x0040
04554 #define PROCESS_CREATE_PROCESS        0x0080
04555 #define PROCESS_SET_QUOTA             0x0100
04556 #define PROCESS_SET_INFORMATION       0x0200
04557 #define PROCESS_QUERY_INFORMATION     0x0400
04558 #define PROCESS_ALL_ACCESS             (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0xfff)
04559
04560 #define THREAD_TERMINATE               0x0001
04561 #define THREAD_SUSPEND_RESUME          0x0002
04562 #define THREAD_GET_CONTEXT             0x0008
04563 #define THREAD_SET_CONTEXT             0x0010
04564 #define THREAD_SET_INFORMATION         0x0020
04565 #define THREAD_QUERY_INFORMATION       0x0040
04566 #define THREAD_SET_THREAD_TOKEN        0x0080
04567 #define THREAD_IMPERSONATE             0x0100
04568 #define THREAD_DIRECT_IMPERSONATION    0x0200
04569 #define THREAD_ALL_ACCESS              (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x3ff)
04570
04571 #define THREAD_BASE_PRIORITY_LOWRT     15
04572 #define THREAD_BASE_PRIORITY_MAX       2
04573 #define THREAD_BASE_PRIORITY_MIN       -2
04574 #define THREAD_BASE_PRIORITY_IDLE      -15
04575
04576 #define FILE_READ_DATA                 0x0001 /* file & pipe */
04577 #define FILE_LIST_DIRECTORY            0x0001 /* directory */
04578 #define FILE_WRITE_DATA                0x0002 /* file & pipe */
04579 #define FILE_ADD_FILE                  0x0002 /* directory */
04580 #define FILE_APPEND_DATA               0x0004 /* file */
04581 #define FILE_ADD_SUBDIRECTORY           0x0004 /* directory */
04582 #define FILE_CREATE_PIPE_INSTANCE      0x0004 /* named pipe */
04583 #define FILE_READ_EA                   0x0008 /* file & directory */
04584 #define FILE_READ_PROPERTIES            FILE_READ_EA
04585 #define FILE_WRITE_EA                   0x0010 /* file & directory */
04586 #define FILE_WRITE_PROPERTIES           FILE_WRITE_EA
04587 #define FILE_EXECUTE                   0x0020 /* file */
04588 #define FILE_TRAVERSE                   0x0020 /* directory */
04589 #define FILE_DELETE_CHILD              0x0040 /* directory */
04590 #define FILE_READ_ATTRIBUTES           0x0080 /* all */
04591 #define FILE_WRITE_ATTRIBUTES           0x0100 /* all */
04592 #define FILE_ALL_ACCESS                 (STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x1ff)
04593
04594 #define FILE_GENERIC_READ               (STANDARD_RIGHTS_READ | FILE_READ_DATA | \
04595 FILE_READ_ATTRIBUTES | FILE_READ_EA | \
04596 SYNCHRONIZE)
04597 #define FILE_GENERIC_WRITE              (STANDARD_RIGHTS_WRITE | FILE_WRITE_DATA | \
04598 FILE_WRITE_ATTRIBUTES | FILE_WRITE_EA | \
04599 FILE_APPEND_DATA | SYNCHRONIZE)
04600 #define FILE_GENERIC_EXECUTE            (STANDARD_RIGHTS_EXECUTE | FILE_EXECUTE | \
04601 FILE_READ_ATTRIBUTES | SYNCHRONIZE)
04602
04603
04604 /* File attribute flags */
04605 #define FILE_SHARE_READ                 0x00000001L
04606 #define FILE_SHARE_WRITE                0x00000002L
04607 #define FILE_SHARE_DELETE               0x00000004L
04608 #define FILE_ATTRIBUTE_READONLY         0x00000001L
04609 #define FILE_ATTRIBUTE_HIDDEN           0x00000002L
04610 #define FILE_ATTRIBUTE_SYSTEM           0x00000004L
04611 #define FILE_ATTRIBUTE_LABEL            0x00000008L /* Not in Windows API */
04612 #define FILE_ATTRIBUTE_DIRECTORY        0x00000010L
04613 #define FILE_ATTRIBUTE_ARCHIVE          0x00000020L
04614 #define FILE_ATTRIBUTE_NORMAL           0x00000080L
04615 #define FILE_ATTRIBUTE_TEMPORARY        0x00000100L
04616 #define FILE_ATTRIBUTE_ATOMIC_WRITE     0x00000200L
04617 #define FILE_ATTRIBUTE_XACTION_WRITE    0x00000400L
04618 #define FILE_ATTRIBUTE_COMPRESSED       0x00000800L
04619 #define FILE_ATTRIBUTE_OFFLINE          0x00001000L
04620 #define FILE_ATTRIBUTE_SYMLINK          0x80000000L /* Not in Windows API */
04621
04622 /* File notification flags */
04623 #define FILE_NOTIFY_CHANGE_FILE_NAME    0x00000001
04624 #define FILE_NOTIFY_CHANGE_DIR_NAME     0x00000002
04625 #define FILE_NOTIFY_CHANGE_ATTRIBUTES   0x00000004
04626 #define FILE_NOTIFY_CHANGE_SIZE         0x00000008
04627 #define FILE_NOTIFY_CHANGE_LAST_WRITE   0x00000010
04628 #define FILE_NOTIFY_CHANGE_LAST_ACCESS  0x00000020
04629 #define FILE_NOTIFY_CHANGE_CREATION     0x00000040
04630 #define FILE_NOTIFY_CHANGE_SECURITY     0x00000100
04631

```

```

04632 #define FILE_ACTION_ADDED 0x00000001
04633 #define FILE_ACTION_REMOVED 0x00000002
04634 #define FILE_ACTION_MODIFIED 0x00000003
04635 #define FILE_ACTION_RENAMED_OLD_NAME 0x00000004
04636 #define FILE_ACTION_RENAMED_NEW_NAME 0x00000005
04637
04638
04639 #define FILE_CASE_SENSITIVE_SEARCH 0x00000001
04640 #define FILE_CASE_PRESERVED_NAMES 0x00000002
04641 #define FILE_UNICODE_ON_DISK 0x00000004
04642 #define FILE_PERSISTENT_ACLS 0x00000008
04643 #define FILE_FILE_COMPRESSION 0x00000010
04644 #define FILE_VOLUME_IS_COMPRESSED 0x00008000
04645
04646 /* File alignments (NT) */
04647 #define FILE_BYTE_ALIGNMENT 0x00000000
04648 #define FILE_WORD_ALIGNMENT 0x00000001
04649 #define FILE_LONG_ALIGNMENT 0x00000003
04650 #define FILE_QUAD_ALIGNMENT 0x00000007
04651 #define FILE_OCTA_ALIGNMENT 0x0000000f
04652 #define FILE_32_BYTE_ALIGNMENT 0x0000001f
04653 #define FILE_64_BYTE_ALIGNMENT 0x0000003f
04654 #define FILE_128_BYTE_ALIGNMENT 0x0000007f
04655 #define FILE_256_BYTE_ALIGNMENT 0x000000ff
04656 #define FILE_512_BYTE_ALIGNMENT 0x000001ff
04657
04658 #define REG_NONE 0 /* no type */
04659 #define REG_SZ 1 /* string type (ASCII) */
04660 #define REG_EXPAND_SZ 2 /* string, includes %ENVVAR% (expanded by caller) (ASCII) */
04661 #define REG_BINARY 3 /* binary format, callerspecific */
04662 /* YES, REG_DWORD == REG_DWORD_LITTLE_ENDIAN */
04663 #define REG_DWORD 4 /* DWORD in little endian format */
04664 #define REG_DWORD_LITTLE_ENDIAN 4 /* DWORD in little endian format */
04665 #define REG_DWORD_BIG_ENDIAN 5 /* DWORD in big endian format */
04666 #define REG_LINK 6 /* symbolic link (UNICODE) */
04667 #define REG_MULTI_SZ 7 /* multiple strings, delimited by \0, terminated by \0\0 (ASCII) */
04668 #define REG_RESOURCE_LIST 8 /* resource list? huh? */
04669 #define REG_FULL_RESOURCE_DESCRIPTOR 9 /* full resource descriptor? huh? */
04670 #define REG_RESOURCE_REQUIREMENTS_LIST 10
04671
04672 /* ----- begin registry ----- */
04673
04674 /* Registry security values */
04675 #define OWNER_SECURITY_INFORMATION 0x00000001
04676 #define GROUP_SECURITY_INFORMATION 0x00000002
04677 #define DACL_SECURITY_INFORMATION 0x00000004
04678 #define SACL_SECURITY_INFORMATION 0x00000008
04679
04680 #define REG_OPTION_RESERVED 0x00000000
04681 #define REG_OPTION_NON_VOLATILE 0x00000000
04682 #define REG_OPTION_VOLATILE 0x00000001
04683 #define REG_OPTION_CREATE_LINK 0x00000002
04684 #define REG_OPTION_BACKUP_RESTORE 0x00000004 /* FIXME */
04685 #define REG_OPTION_OPEN_LINK 0x00000008
04686 #define REG_LEGAL_OPTION ((REG_OPTION_RESERVED| \
04687 REG_OPTION_NON_VOLATILE| \
04688 REG_OPTION_VOLATILE| \
04689 REG_OPTION_CREATE_LINK| \
04690 REG_OPTION_BACKUP_RESTORE| \
04691 REG_OPTION_OPEN_LINK)
04692
04693
04694 #define REG_CREATED_NEW_KEY 0x00000001
04695 #define REG_OPENED_EXISTING_KEY 0x00000002
04696
04697 /* For RegNotifyChangeKeyValue */
04698 #define REG_NOTIFY_CHANGE_NAME 0x1
04699
04700 #define KEY_QUERY_VALUE 0x00000001
04701 #define KEY_SET_VALUE 0x00000002
04702 #define KEY_CREATE_SUB_KEY 0x00000004
04703 #define KEY_ENUMERATE_SUB_KEYS 0x00000008
04704 #define KEY_NOTIFY 0x00000010
04705 #define KEY_CREATE_LINK 0x00000020
04706
04707 #define KEY_READ ((STANDARD_RIGHTS_READ| \
04708 KEY_QUERY_VALUE| \
04709 KEY_ENUMERATE_SUB_KEYS| \
04710 KEY_NOTIFY) \
04711 & (~SYNCHRONIZE) \
04712 )
04713 #define KEY_WRITE ((STANDARD_RIGHTS_WRITE| \
04714 KEY_SET_VALUE| \
04715 KEY_CREATE_SUB_KEY) \
04716 & (~SYNCHRONIZE) \
04717 )
04718 #define KEY_EXECUTE ((KEY_READ) \

```

```

04719         & (~SYNCHRONIZE)) \
04720     )
04721 #define KEY_ALL_ACCESS ((STANDARD_RIGHTS_ALL| \
04722     KEY_QUERY_VALUE| \
04723     KEY_SET_VALUE| \
04724     KEY_CREATE_SUB_KEY| \
04725     KEY_ENUMERATE_SUB_KEYS| \
04726     KEY_NOTIFY| \
04727     KEY_CREATE_LINK) \
04728     & (~SYNCHRONIZE) \
04729     )
04730 /* ----- end registry ----- */
04731
04732
04733 #define EVENTLOG_SUCCESS 0x0000
04734 #define EVENTLOG_ERROR_TYPE 0x0001
04735 #define EVENTLOG_WARNING_TYPE 0x0002
04736 #define EVENTLOG_INFORMATION_TYPE 0x0004
04737 #define EVENTLOG_AUDIT_SUCCESS 0x0008
04738 #define EVENTLOG_AUDIT_FAILURE 0x0010
04739
04740 #define SERVICE_BOOT_START 0x00000000
04741 #define SERVICE_SYSTEM_START 0x00000001
04742 #define SERVICE_AUTO_START 0x00000002
04743 #define SERVICE_DEMAND_START 0x00000003
04744 #define SERVICE_DISABLED 0x00000004
04745
04746 #define SERVICE_ERROR_IGNORE 0x00000000
04747 #define SERVICE_ERROR_NORMAL 0x00000001
04748 #define SERVICE_ERROR_SEVERE 0x00000002
04749 #define SERVICE_ERROR_CRITICAL 0x00000003
04750
04751 /* Service types */
04752 #define SERVICE_KERNEL_DRIVER 0x00000001
04753 #define SERVICE_FILE_SYSTEM_DRIVER 0x00000002
04754 #define SERVICE_ADAPTER 0x00000004
04755 #define SERVICE_RECOGNIZER_DRIVER 0x00000008
04756
04757 #define SERVICE_DRIVER ( SERVICE_KERNEL_DRIVER | SERVICE_FILE_SYSTEM_DRIVER | \
04758     SERVICE_RECOGNIZER_DRIVER )
04759
04760 #define SERVICE_WIN32_OWN_PROCESS 0x00000010
04761 #define SERVICE_WIN32_SHARE_PROCESS 0x00000020
04762 #define SERVICE_WIN32 (SERVICE_WIN32_OWN_PROCESS | SERVICE_WIN32_SHARE_PROCESS)
04763
04764 #define SERVICE_INTERACTIVE_PROCESS 0x00000100
04765
04766 #define SERVICE_TYPE_ALL ( SERVICE_WIN32 | SERVICE_ADAPTER | \
04767     SERVICE_DRIVER | SERVICE_INTERACTIVE_PROCESS )
04768
04769
04770 typedef enum _CM_SERVICE_NODE_TYPE
04771 {
04772     DriverType = SERVICE_KERNEL_DRIVER,
04773     FileSystemType = SERVICE_FILE_SYSTEM_DRIVER,
04774     Win32ServiceOwnProcess = SERVICE_WIN32_OWN_PROCESS,
04775     Win32ServiceShareProcess = SERVICE_WIN32_SHARE_PROCESS,
04776     AdapterType = SERVICE_ADAPTER,
04777     RecognizerType = SERVICE_RECOGNIZER_DRIVER
04778 } SERVICE_NODE_TYPE;
04779
04780 typedef enum _CM_SERVICE_LOAD_TYPE
04781 {
04782     BootLoad = SERVICE_BOOT_START,
04783     SystemLoad = SERVICE_SYSTEM_START,
04784     AutoLoad = SERVICE_AUTO_START,
04785     DemandLoad = SERVICE_DEMAND_START,
04786     DisableLoad = SERVICE_DISABLED
04787 } SERVICE_LOAD_TYPE;
04788
04789 typedef enum _CM_ERROR_CONTROL_TYPE
04790 {
04791     IgnoreError = SERVICE_ERROR_IGNORE,
04792     NormalError = SERVICE_ERROR_NORMAL,
04793     SevereError = SERVICE_ERROR_SEVERE,
04794     CriticalError = SERVICE_ERROR_CRITICAL
04795 } SERVICE_ERROR_TYPE;
04796
04797
04798
04799 #define RtlEqualMemory(Destination, Source, Length) (!memcmp((Destination), (Source), (Length)))
04800 #define RtlMoveMemory(Destination, Source, Length) memmove((Destination), (Source), (Length))
04801 #define RtlCopyMemory(Destination, Source, Length) memcpy((Destination), (Source), (Length))
04802 #define RtlFillMemory(Destination, Length, Fill) memset((Destination), (Fill), (Length))
04803 #define RtlZeroMemory(Destination, Length) memset((Destination), 0, (Length))
04804
04805 #include "guiddef.h"

```

```

04806
04807 typedef struct _RTL_CRITICAL_SECTION_DEBUG
04808 {
04809     WORD    Type;
04810     WORD    CreatorBackTraceIndex;
04811     struct _RTL_CRITICAL_SECTION *CriticalSection;
04812     LIST_ENTRY ProcessLocksList;
04813     DWORD    EntryCount;
04814     DWORD    ContentionCount;
04815     DWORD    Spare[ 2 ];
04816 } RTL_CRITICAL_SECTION_DEBUG, *PRTL_CRITICAL_SECTION_DEBUG, RTL_RESOURCE_DEBUG, *PRTL_RESOURCE_DEBUG;
04817
04818 typedef struct _RTL_CRITICAL_SECTION {
04819     PRTL_CRITICAL_SECTION_DEBUG DebugInfo;
04820     LONG    LockCount;
04821     LONG    RecursionCount;
04822     HANDLE    OwningThread;
04823     HANDLE    LockSemaphore;
04824     ULONG_PTR    SpinCount;
04825 } RTL_CRITICAL_SECTION, *PRTL_CRITICAL_SECTION;
04826
04827 #endif /* __WINE_WINNT_H */

```

## 5.13 winuser.h

```

00001 #ifndef _WINUSER_
00002 #define _WINUSER_
00003
00004 #ifndef RC_INVOKED
00005 #include <stdarg.h>
00006 #endif
00007
00008 #ifdef __cplusplus
00009 extern "C" {
00010 #endif
00011
00012 /* Define a bunch of callback types */
00013
00014 #if defined(STRICT) || defined(__WINE__)
00015 typedef BOOL    CALLBACK (*DLGPROC) (HWND, UINT, WPARAM, LPARAM);
00016 typedef BOOL    CALLBACK (*DRAWSTATEPROC) (HDC, LPARAM, WPARAM, int, int);
00017 typedef INT     CALLBACK (*EDITWORDBREAKPROCA) (LPSTR, INT, INT, INT);
00018 typedef INT     CALLBACK (*EDITWORDBREAKPROCW) (LPWSTR, INT, INT, INT);
00019 typedef BOOL    CALLBACK (*GRAYSTRINGPROC) (HDC, LPARAM, INT);
00020 typedef LRESULT CALLBACK (*HOOKPROC) (INT, WPARAM, LPARAM);
00021 typedef BOOL    CALLBACK (*NAMEENUMPROCA) (LPSTR, LPARAM);
00022 typedef BOOL    CALLBACK (*NAMEENUMPROCW) (LPWSTR, LPARAM);
00023 typedef BOOL    CALLBACK (*PROPENUMPROCA) (HWND, LPCSTR, HANDLE);
00024 typedef BOOL    CALLBACK (*PROPENUMPROCW) (HWND, LPCWSTR, HANDLE);
00025 typedef BOOL    CALLBACK (*PROPENUMPROCEXA) (HWND, LPCSTR, HANDLE, ULONG_PTR);
00026 typedef BOOL    CALLBACK (*PROPENUMPROCEXW) (HWND, LPCWSTR, HANDLE, ULONG_PTR);
00027 typedef VOID    CALLBACK (*SENDASYNCPROC) (HWND, UINT, ULONG_PTR, LRESULT);
00028 typedef VOID    CALLBACK (*TIMERPROC) (HWND, UINT, UINT, DWORD);
00029 typedef BOOL    CALLBACK (*WNDENUMPROC) (HWND, LPARAM);
00030 #else
00031 typedef FARPROC DLGPROC;
00032 typedef FARPROC DRAWSTATEPROC;
00033 typedef FARPROC EDITWORDBREAKPROCA;
00034 typedef FARPROC EDITWORDBREAKPROCW;
00035 typedef FARPROC GRAYSTRINGPROC;
00036 typedef FARPROC HOOKPROC;
00037 typedef FARPROC NAMEENUMPROCA;
00038 typedef FARPROC NAMEENUMPROCW;
00039 typedef FARPROC PROPENUMPROCA;
00040 typedef FARPROC PROPENUMPROCW;
00041 typedef FARPROC PROPENUMPROCEXA;
00042 typedef FARPROC PROPENUMPROCEXW;
00043 typedef FARPROC SENDASYNCPROC;
00044 typedef FARPROC TIMERPROC;
00045 typedef FARPROC WNDENUMPROC;
00046 #endif /* STRICT || __WINE__ */
00047
00048 typedef NAMEENUMPROCA WINSTAENUMPROCA;
00049 typedef NAMEENUMPROCA DESKTOPENUMPROCA;
00050 typedef NAMEENUMPROCW WINSTAENUMPROCW;
00051 typedef NAMEENUMPROCW DESKTOPENUMPROCW;
00052
00053 typedef LRESULT CALLBACK (*WNDPROC) (HWND, UINT, WPARAM, LPARAM);
00054
00055 DECL_WINELIB_TYPE_AW(DESKTOPENUMPROC)
00056 DECL_WINELIB_TYPE_AW(EDITWORDBREAKPROC)
00057 DECL_WINELIB_TYPE_AW(NAMEENUMPROC)
00058 DECL_WINELIB_TYPE_AW(PROPENUMPROC)
00059 DECL_WINELIB_TYPE_AW(PROPENUMPROCEX)

```

```

00060 DECL_WINELIB_TYPE_AW(WINSTAENUMPROC)
00061
00062
00063 typedef HANDLE HDWP;
00064
00065 /* flags for FILTERKEYS dwFlags field */
00066 #define FKF_AVAILABLE 0x00000002
00067 #define FKF_CLICKON 0x00000040
00068 #define FKF_FILTERKEYSON 0x00000001
00069 #define FKF_HOTKEYACTIVE 0x00000004
00070 #define FKF_HOTKEYSOUND 0x00000010
00071 #define FKF_CONFIRMHOTKEY 0x00000008
00072 #define FKF_INDICATOR 0x00000020
00073
00074 typedef struct tagFILTERKEYS
00075 {
00076     UINT cbSize;
00077     DWORD dwFlags;
00078     DWORD iWaitMSec;
00079     DWORD iDelayMSec;
00080     DWORD iRepeatMSec;
00081     DWORD iBounceMSec;
00082 } FILTERKEYS, *PFILTERKEYS, *LPFILTERKEYS;
00083
00084 /* flags for TOGGLEKEYS dwFlags field */
00085 #define TKF_AVAILABLE 0x00000002
00086 #define TKF_CONFIRMHOTKEY 0x00000008
00087 #define TKF_HOTKEYACTIVE 0x00000004
00088 #define TKF_HOTKEYSOUND 0x00000010
00089 #define TKF_TOGGLEKEYSON 0x00000001
00090
00091 typedef struct tagTOGGLEKEYS
00092 {
00093     DWORD cbSize;
00094     DWORD dwFlags;
00095 } TOGGLEKEYS, *PTOGGLEKEYS, *LPTOGGLEKEYS;
00096
00097 /* flags for MOUSEKEYS dwFlags field */
00098 #define MKF_AVAILABLE 0x00000002
00099 #define MKF_CONFIRMHOTKEY 0x00000008
00100 #define MKF_HOTKEYACTIVE 0x00000004
00101 #define MKF_HOTKEYSOUND 0x00000010
00102 #define MKF_INDICATOR 0x00000020
00103 #define MKF_MOUSEKEYSON 0x00000001
00104 #define MKF_MODIFIERS 0x00000040
00105 #define MKF_REPLACENUMBERS 0x00000080
00106
00107 typedef struct tagMOUSEKEYS
00108 {
00109     UINT cbSize;
00110     DWORD dwFlags;
00111     DWORD iMaxSpeed;
00112     DWORD iTimeToMaxSpeed;
00113     DWORD iCtrlSpeed;
00114     DWORD dwReserved1;
00115     DWORD dwReserved2;
00116 } MOUSEKEYS, *PMOUSEKEYS, *LPMOUSEKEYS;
00117
00118 /* flags for STICKYKEYS dwFlags field */
00119 #define SKF_AUDIBLEFEEDBACK 0x00000040
00120 #define SKF_AVAILABLE 0x00000002
00121 #define SKF_CONFIRMHOTKEY 0x00000008
00122 #define SKF_HOTKEYACTIVE 0x00000004
00123 #define SKF_HOTKEYSOUND 0x00000010
00124 #define SKF_INDICATOR 0x00000020
00125 #define SKF_STICKYKEYSON 0x00000001
00126 #define SKF_TRISTATE 0x00000080
00127 #define SKF_TWOKEYSOFF 0x00000100
00128
00129 typedef struct tagSTICKYKEYS
00130 {
00131     DWORD cbSize;
00132     DWORD dwFlags;
00133 } STICKYKEYS, *PSTICKYKEYS, *LPSTICKYKEYS;
00134
00135 /* flags for ACCESTIMEOUT dwFlags field */
00136 #define ATF_ONOFFFEEDBACK 0x00000002
00137 #define ATF_AVAILABLE 0x00000004
00138 #define ATF_TIMEOUTON 0x00000001
00139
00140 typedef struct tagACCESTIMEOUT
00141 {
00142     UINT cbSize;
00143     DWORD dwFlags;
00144     DWORD iTimeOutMSec;
00145 } ACCESTIMEOUT, *PACCESTIMEOUT, *LPACCESTIMEOUT;
00146

```

```

00147 /* flags for SERIALKEYS dwFlags field */
00148 #define SERKF_ACTIVE 0x00000008
00149 #define SERKF_AVAILABLE 0x00000002
00150 #define SERKF_INDICATOR 0x00000004
00151 #define SERKF_SERIALKEYSON 0x00000001
00152
00153 typedef struct tagSERIALKEYSA
00154 {
00155     UINT cbSize;
00156     DWORD dwFlags;
00157     LPSTR lpszActivePort;
00158     LPSTR lpszPort;
00159     UINT iBaudRate;
00160     UINT iPortState;
00161     UINT iActive;
00162 } SERIALKEYSA, *LP SERIALKEYSA;
00163
00164 typedef struct tagSERIALKEYSW {
00165     UINT cbSize;
00166     DWORD dwFlags;
00167     LPWSTR lpszActivePort;
00168     LPWSTR lpszPort;
00169     UINT iBaudRate;
00170     UINT iPortState;
00171     UINT iActive;
00172 } SERIALKEYSW, *LP SERIALKEYSW;
00173
00174 DECL_WINELIB_TYPE_AW(SERIALKEYS)
00175 DECL_WINELIB_TYPE_AW(LP SERIALKEYS)
00176
00177 /* flags for SOUNDSENTRY dwFlags field */
00178 #define SSF_AVAILABLE 0x00000002
00179 #define SSF_SOUNDSENTRYON 0x00000001
00180
00181 #define SSTF_BORDER 0x00000002
00182 #define SSTF_CHARS 0x00000001
00183 #define SSTF_DISPLAY 0x00000003
00184 #define SSTF_NONE 0x00000000
00185
00186 #define SSGF_DISPLAY 0x00000003
00187 #define SSGF_NONE 0x00000000
00188
00189 #define SSWF_DISPLAY 0x00000003
00190 #define SSWF_NONE 0x00000000
00191 #define SSWF_TITLE 0x00000001
00192 #define SSWF_WINDOW 0x00000002
00193
00194 typedef struct tagSOUNDSENTRYA
00195 {
00196     UINT cbSize;
00197     DWORD dwFlags;
00198     DWORD iFSTextEffect;
00199     DWORD iFSTextEffectMSec;
00200     DWORD iFSTextEffectColorBits;
00201     DWORD iFSGrafeEffect;
00202     DWORD iFSGrafeEffectMSec;
00203     DWORD iFSGrafeEffectColor;
00204     DWORD iWindowsEffect;
00205     DWORD iWindowsEffectMSec;
00206     LPSTR lpszWindowsEffectDLL;
00207     DWORD iWindowsEffectOrdinal;
00208 } SOUNDSENTRYA, *LP SOUNDSENTRYA;
00209
00210 typedef struct tagSOUNDSENTRYW
00211 {
00212     UINT cbSize;
00213     DWORD dwFlags;
00214     DWORD iFSTextEffect;
00215     DWORD iFSTextEffectMSec;
00216     DWORD iFSTextEffectColorBits;
00217     DWORD iFSGrafeEffect;
00218     DWORD iFSGrafeEffectMSec;
00219     DWORD iFSGrafeEffectColor;
00220     DWORD iWindowsEffect;
00221     DWORD iWindowsEffectMSec;
00222     LPWSTR lpszWindowsEffectDLL;
00223     DWORD iWindowsEffectOrdinal;
00224 } SOUNDSENTRYW, *LP SOUNDSENTRYW;
00225
00226 DECL_WINELIB_TYPE_AW(SOUNDSENTRY)
00227 DECL_WINELIB_TYPE_AW(LP SOUNDSENTRY)
00228
00229 /* flags for HIGHCONTRAST dwFlags field */
00230 #define HCF_HIGHCONTRASTON 0x00000001
00231 #define HCF_AVAILABLE 0x00000002
00232 #define HCF_HOTKEYACTIVE 0x00000004
00233 #define HCF_CONFIRMHOTKEY 0x00000008

```

```

00234 #define HCF_HOTKEYSOUND      0x00000010
00235 #define HCF_INDICATOR        0x00000020
00236 #define HCF_HOTKEYAVAILABLE  0x00000040
00237
00238 typedef struct tagHIGHCONTRASTA
00239 {
00240     UINT    cbSize;
00241     DWORD   dwFlags;
00242     LPSTR    lpszDefaultScheme;
00243 } HIGHCONTRASTA, *LPHIGHCONTRASTA;
00244
00245 typedef struct tagHIGHCONTRASTW
00246 {
00247     UINT    cbSize;
00248     DWORD   dwFlags;
00249     LPWSTR   lpszDefaultScheme;
00250 } HIGHCONTRASTW, *LPHIGHCONTRASTW;
00251
00252 DECL_WINELIB_TYPE_AW(HIGHCONTRAST)
00253 DECL_WINELIB_TYPE_AW(LPHIGHCONTRAST)
00254
00255 typedef struct
00256 {
00257     UINT    message;
00258     UINT    paramL;
00259     UINT    paramH;
00260     DWORD   time;
00261     HWND    hwnd;
00262 } EVENTMSG, *PEVENTMSG, *LPEVENTMSG;
00263
00264 /* WH_KEYBOARD_LL structure */
00265 typedef struct tagKBDLLHOOKSTRUCT
00266 {
00267     DWORD    vkCode;
00268     DWORD    scanCode;
00269     DWORD    flags;
00270     DWORD    time;
00271     ULONG_PTR dwExtraInfo;
00272 } KBDLLHOOKSTRUCT, *LPKBDLLHOOKSTRUCT, *PKBDLLHOOKSTRUCT;
00273
00274 #define LLKHF_EXTENDED      (KF_EXTENDED » 8)
00275 #define LLKHF_INJECTED      0x00000010
00276 #define LLKHF_ALTDOWN      (KF_ALTDOWN » 8)
00277 #define LLKHF_UP            (KF_UP » 8)
00278
00279 /* WH_MOUSE_LL structure */
00280 typedef struct tagMSLLHOOKSTRUCT
00281 {
00282     POINT    pt;
00283     DWORD    mouseData;
00284     DWORD    flags;
00285     DWORD    time;
00286     ULONG_PTR dwExtraInfo;
00287 } MSLLHOOKSTRUCT, *LPMSLLHOOKSTRUCT, *PMSLLHOOKSTRUCT;
00288
00289 #define LLMHF_INJECTED      0x00000001
00290
00291 /* Mouse hook structure */
00292
00293 typedef struct
00294 {
00295     POINT    pt;
00296     HWND    hwnd;
00297     UINT    wHitTestCode;
00298     DWORD    dwExtraInfo;
00299 } MOUSEHOOKSTRUCT, *PMOUSEHOOKSTRUCT, *LPMOUSEHOOKSTRUCT;
00300
00301
00302 /* Hardware hook structure */
00303
00304 typedef struct
00305 {
00306     HWND    hwnd;
00307     UINT    wMessage;
00308     WPARAM    wParam;
00309     LPARAM    lParam;
00310 } HARDWAREHOOKSTRUCT, *PHARDWAREHOOKSTRUCT, *LPHARDWAREHOOKSTRUCT;
00311
00312
00313 /* Debug hook structure */
00314
00315 typedef struct
00316 {
00317     DWORD    idThread;
00318     DWORD    idThreadInstaller;
00319     LPARAM    lParam;
00320     WPARAM    wParam;

```

```

00321     INT         code;
00322 } DEBUGHOOKINFO, *PDEBUGHOOKINFO, *LPDEBUGHOOKINFO;
00323
00324 #define HKL_PREV    0
00325 #define HKL_NEXT    1
00326
00327 #define KLF_ACTIVATE      0x00000001
00328 #define KLF_SUBSTITUTE_OK 0x00000002
00329 #define KLF_UNLOADPREVIOUS 0x00000004
00330 #define KLF_REORDER       0x00000008
00331 #define KLF_REPLACELANG   0x00000010
00332 #define KLF_NOTELLSHELL   0x00000080
00333
00334 #define KL_NAMELENGTH     9
00335
00336 typedef struct tagMOUSEINPUT
00337 {
00338     LONG    dx;
00339     LONG    dy;
00340     DWORD   mouseData;
00341     DWORD   dwFlags;
00342     DWORD   time;
00343     ULONG_PTR dwExtraInfo;
00344 } MOUSEINPUT, *PMOUSEINPUT, *LPMOUSEINPUT;
00345
00346 typedef struct tagKEYBDINPUT
00347 {
00348     WORD    wVk;
00349     WORD    wScan;
00350     DWORD   dwFlags;
00351     DWORD   time;
00352     ULONG_PTR dwExtraInfo;
00353 } KEYBDINPUT, *PKEYBDINPUT, *LPKEYBDINPUT;
00354
00355 typedef struct tagHARDWAREINPUT
00356 {
00357     DWORD   uMsg;
00358     WORD    wParamL;
00359     WORD    wParamH;
00360 } HARDWAREINPUT, *PHARDWAREINPUT, *LPHARDWAREINPUT;
00361
00362 #define INPUT_MOUSE      0
00363 #define INPUT_KEYBOARD   1
00364 #define INPUT_HARDWARE   2
00365
00366 typedef struct tagINPUT
00367 {
00368     DWORD type;
00369     union
00370     {
00371         MOUSEINPUT    mi;
00372         KEYBDINPUT     ki;
00373         HARDWAREINPUT  hi;
00374     } DUMMYUNIONNAME;
00375 } INPUT, *PINPUT, *LPINPUT;
00376
00377
00378 /***** Dialogs *****/
00379
00380 /* Gcc on Solaris has a version of this that we don't care about */
00381 #undef FSHIFT
00382
00383 #define FVIRTKEY    TRUE           /* Assumed to be == TRUE */
00384 #define FNOINVERT   0x02
00385 #define FSHIFT      0x04
00386 #define FCONTROL     0x08
00387 #define FALT          0x10
00388
00389
00390 typedef struct tagANIMATIONINFO
00391 {
00392     UINT    cbSize;
00393     INT     iMinAnimate;
00394 } ANIMATIONINFO, *LPANIMATIONINFO;
00395
00396 typedef struct tagNMHDR
00397 {
00398     HWND    hwndFrom;
00399     UINT    idFrom;
00400     UINT    code;
00401 } NMHDR, *LPNMHDR;
00402
00403 typedef struct
00404 {
00405     UINT    cbSize;
00406     INT     iTabLength;
00407     INT     iLeftMargin;

```

```

00408     INT iRightMargin;
00409     UINT uiLengthDrawn;
00410 } DRAWTEXTPARAMS, *LPDRAWTEXTPARAMS;
00411
00412 #define WM_USER          0x0400
00413
00414 #define DT_EDITCONTROL    0x00002000
00415 #define DT_PATH_ELLIPSIS 0x00004000
00416 #define DT_END_ELLIPSIS  0x00008000
00417 #define DT_MODIFYSTRING   0x00010000
00418 #define DT_RTLREADING     0x00020000
00419 #define DT_WORD_ELLIPSIS 0x00040000
00420
00421 typedef struct
00422 {
00423     LPARAM      lParam;
00424     WPARAM      wParam;
00425     UINT         message;
00426     HWND        hwnd;
00427 } CWPSTRUCT, *PCWPSTRUCT, *LPCWPSTRUCT;
00428
00429 typedef struct
00430 {
00431     LRESULT      lResult;
00432     LPARAM      lParam;
00433     WPARAM      wParam;
00434     DWORD       message;
00435     HWND        hwnd;
00436 } CWPRETSTRUCT, *PCWPRETSTRUCT, *LPCWPRETSTRUCT;
00437
00438 typedef struct
00439 {
00440     UINT         length;
00441     UINT         flags;
00442     UINT         showCmd;
00443     POINT        ptMinPosition WINE_PACKED;
00444     POINT        ptMaxPosition WINE_PACKED;
00445     RECT         rcNormalPosition WINE_PACKED;
00446 } WINDOWPLACEMENT, *PWINDOWPLACEMENT, *LPWINDOWPLACEMENT;
00447
00448 /* WINDOWPLACEMENT flags */
00449 #define WPF_SETMINPOSITION 0x0001
00450 #define WPF_RESTORETOMAXIMIZED 0x0002
00451
00452 /***** Dialogs *****/
00453
00454 #define MAKEINTRESOURCEA(i) (LPSTR)((DWORD)((WORD)(i)))
00455 #define MAKEINTRESOURCEW(i) (LPWSTR)((DWORD)((WORD)(i)))
00456 #define MAKEINTRESOURCE WINELIB_NAME_AW(MAKEINTRESOURCE)
00457
00458 /* Predefined resource types */
00459 #define RT_CURSORA      MAKEINTRESOURCEA(1)
00460 #define RT_CURSORW      MAKEINTRESOURCEW(1)
00461 #define RT_CURSOR       WINELIB_NAME_AW(RT_CURSOR)
00462 #define RT_BITMAPA      MAKEINTRESOURCEA(2)
00463 #define RT_BITMAPW      MAKEINTRESOURCEW(2)
00464 #define RT_BITMAP       WINELIB_NAME_AW(RT_BITMAP)
00465 #define RT_ICONA        MAKEINTRESOURCEA(3)
00466 #define RT_ICONW        MAKEINTRESOURCEW(3)
00467 #define RT_ICON         WINELIB_NAME_AW(RT_ICON)
00468 #define RT_MENUA        MAKEINTRESOURCEA(4)
00469 #define RT_MEN UW        MAKEINTRESOURCEW(4)
00470 #define RT_MENU         WINELIB_NAME_AW(RT_MENU)
00471 #define RT_DIALOGA      MAKEINTRESOURCEA(5)
00472 #define RT_DIALOGW      MAKEINTRESOURCEW(5)
00473 #define RT_DIALOG       WINELIB_NAME_AW(RT_DIALOG)
00474 #define RT_STRINGA      MAKEINTRESOURCEA(6)
00475 #define RT_STRINGW      MAKEINTRESOURCEW(6)
00476 #define RT_STRING       WINELIB_NAME_AW(RT_STRING)
00477 #define RT_FONTDIRA     MAKEINTRESOURCEA(7)
00478 #define RT_FONTDIRW     MAKEINTRESOURCEW(7)
00479 #define RT_FONTDIR      WINELIB_NAME_AW(RT_FONTDIR)
00480 #define RT_FONTA        MAKEINTRESOURCEA(8)
00481 #define RT_FONTW        MAKEINTRESOURCEW(8)
00482 #define RT_FONT         WINELIB_NAME_AW(RT_FONT)
00483 #define RT_ACCELERATORA  MAKEINTRESOURCEA(9)
00484 #define RT_ACCELERATORW  MAKEINTRESOURCEW(9)
00485 #define RT_ACCELERATOR   WINELIB_NAME_AW(RT_ACCELERATOR)
00486 #define RT_RCDATAA      MAKEINTRESOURCEA(10)
00487 #define RT_RCDATAW      MAKEINTRESOURCEW(10)
00488 #define RT_RCDATA       WINELIB_NAME_AW(RT_RCDATA)
00489 #define RT_MESSAGETABLEA MAKEINTRESOURCEA(11)
00490 #define RT_MESSAGETABLEW MAKEINTRESOURCEW(11)
00491 #define RT_MESSAGETABLE  WINELIB_NAME_AW(RT_MESSAGETABLE)
00492 #define RT_GROUP_CURSORA MAKEINTRESOURCEA(12)
00493 #define RT_GROUP_CURSORW MAKEINTRESOURCEW(12)
00494 #define RT_GROUP_CURSOR  WINELIB_NAME_AW(RT_GROUP_CURSOR)

```

```

00495 #define RT_GROUP_CURSOR      WINELIB_NAME_AW(RT_GROUP_CURSOR)
00496 #define RT_GROUP_ICONA        MAKEINTRESOURCEA(14)
00497 #define RT_GROUP_ICONW        MAKEINTRESOURCEW(14)
00498 #define RT_GROUP_ICON         WINELIB_NAME_AW(RT_GROUP_ICON)
00499 #define RT_VERSIONA           MAKEINTRESOURCEA(16)
00500 #define RT_VERSIONW           MAKEINTRESOURCEW(16)
00501 #define RT_VERSION             WINELIB_NAME_AW(RT_VERSION)
00502 #define RT_DLGINCLUDEA        MAKEINTRESOURCEA(17)
00503 #define RT_DLGINCLUDEW        MAKEINTRESOURCEW(17)
00504 #define RT_DLGINCLUDE         WINELIB_NAME_AW(RT_DLGINCLUDE)
00505 #define RT_PLUGPLAYA          MAKEINTRESOURCEA(19)
00506 #define RT_PLUGPLAYW          MAKEINTRESOURCEW(19)
00507 #define RT_PLUGPLAY           WINELIB_NAME_AW(RT_PLUGPLAY)
00508 #define RT_VXDA               MAKEINTRESOURCEA(20)
00509 #define RT_VXDW               MAKEINTRESOURCEW(20)
00510 #define RT_VXD                WINELIB_NAME_AW(RT_VXD)
00511 #define RT_ANICURSORA         MAKEINTRESOURCEA(21)
00512 #define RT_ANICURSOW          MAKEINTRESOURCEW(21)
00513 #define RT_ANICURSOR          WINELIB_NAME_AW(RT_ANICURSOR)
00514 #define RT_ANIICONA           MAKEINTRESOURCEA(22)
00515 #define RT_ANIICONW           MAKEINTRESOURCEW(22)
00516 #define RT_ANIICON            WINELIB_NAME_AW(RT_ANIICON)
00517 #define RT_HTMLA              MAKEINTRESOURCEA(23)
00518 #define RT_HTMLW              MAKEINTRESOURCEW(23)
00519 #define RT_HTML               WINELIB_NAME_AW(RT_HTML)
00520
00521
00522 /* cbWndExtra bytes for dialog class */
00523 #define DLGWINDOWEXTRA        30
00524
00525 /* Button control styles */
00526 #define BS_PUSHBUTTON          0x00000000L
00527 #define BS_DEFPUSHBUTTON      0x00000001L
00528 #define BS_CHECKBOX            0x00000002L
00529 #define BS_AUTOCHECKBOX       0x00000003L
00530 #define BS_RADIOBUTTON        0x00000004L
00531 #define BS_3STATE              0x00000005L
00532 #define BS_AUTO3STATE         0x00000006L
00533 #define BS_GROUPBOX           0x00000007L
00534 #define BS_USERBUTTON         0x00000008L
00535 #define BS_AUTORADIOBUTTON    0x00000009L
00536 #define BS_OWNERDRAW          0x0000000BL
00537 #define BS_LEFTTEXT           0x00000020L
00538 #define BS_RIGHTBUTTON        BS_LEFTTEXT
00539
00540 #define BS_TEXT                0x00000000L
00541 #define BS_ICON               0x00000040L
00542 #define BS_BITMAP             0x00000080L
00543 #define BS_LEFT               0x00000100L
00544 #define BS_RIGHT              0x00000200L
00545 #define BS_CENTER             0x00000300L
00546 #define BS_TOP                0x00000400L
00547 #define BS_BOTTOM             0x00000800L
00548 #define BS_VCENTER            0x00000C00L
00549 #define BS_PUSHLIKE           0x00001000L
00550 #define BS_MULTILINE          0x00002000L
00551 #define BS_NOTIFY             0x00004000L
00552 #define BS_FLAT               0x00008000L
00553
00554 /* Dialog styles */
00555 #define DS_ABSALIGN           0x0001
00556 #define DS_SYSMODAL           0x0002
00557 #define DS_3DLOOK             0x0004 /* win95 */
00558 #define DS_FIXEDSYS           0x0008 /* win95 */
00559 #define DS_NOFAILCREATE       0x0010 /* win95 */
00560 #define DS_LOCALEEDIT         0x0020
00561 #define DS_SETFONT            0x0040
00562 #define DS_MODALFRAME         0x0080
00563 #define DS_NOIDLEMSG          0x0100
00564 #define DS_SETFOREGROUND      0x0200 /* win95 */
00565 #define DS_CONTROL            0x0400 /* win95 */
00566 #define DS_CENTER             0x0800 /* win95 */
00567 #define DS_CENTERMOUSE        0x1000 /* win95 */
00568 #define DS_CONTEXTHELP        0x2000 /* win95 */
00569
00570
00571 /* Dialog messages */
00572 #define DM_GETDEFID           (WM_USER+0)
00573 #define DM_SETDEFID           (WM_USER+1)
00574 #define DM_REPOSITION         (WM_USER+2)
00575
00576 #define DC_HASDEFID           0x534b
00577
00578 /* Owner draw control types */
00579 #define ODT_MENU              1
00580 #define ODT_LISTBOX           2
00581 #define ODT_COMBOBOX          3

```

```
00582 #define ODT_BUTTON      4
00583 #define ODT_STATIC        5
00584
00585 /* Owner draw actions */
00586 #define ODA_DRAWENTIRE    0x0001
00587 #define ODA_SELECT        0x0002
00588 #define ODA_FOCUS         0x0004
00589
00590 /* Owner draw state */
00591 #define ODS_SELECTED      0x0001
00592 #define ODS_GRAYED        0x0002
00593 #define ODS_DISABLED      0x0004
00594 #define ODS_CHECKED        0x0008
00595 #define ODS_FOCUS         0x0010
00596 #define ODS_COMBOBOXEDIT 0x1000
00597 #define ODS_HOTLIGHT      0x0040
00598 #define ODS_INACTIVE      0x0080
00599
00600 /* Edit control styles */
00601 #define ES_LEFT            0x00000000
00602 #define ES_CENTER          0x00000001
00603 #define ES_RIGHT           0x00000002
00604 #define ES_MULTILINE       0x00000004
00605 #define ES_UPPERCASE       0x00000008
00606 #define ES_LOWERCASE       0x00000010
00607 #define ES_PASSWORD        0x00000020
00608 #define ES_AUTOVSCROLL     0x00000040
00609 #define ES_AUTOHSCROLL     0x00000080
00610 #define ES_NOHIDESEL       0x00000100
00611 #define ES_COMBO           0x00000200 /* Undocumented. Parent is a combobox */
00612 #define ES_OEMCONVERT      0x00000400
00613 #define ES_READONLY        0x00000800
00614 #define ES_WANTRETURN      0x00001000
00615 #define ES_NUMBER          0x00002000
00616
00617 /* OEM Resource Ordinal Numbers */
00618 #define OBM_CLOSED         32731
00619 #define OBM_TRTYPE         32732
00620 #define OBM_LFARROWI       32734
00621 #define OBM_RGARROWI       32735
00622 #define OBM_DNARROWI       32736
00623 #define OBM_UPARROWI       32737
00624 #define OBM_COMBO          32738
00625 #define OBM_MNARROW        32739
00626 #define OBM_LFARROWD       32740
00627 #define OBM_RGARROWD       32741
00628 #define OBM_DNARROWD       32742
00629 #define OBM_UPARROWD       32743
00630 #define OBM_RESTORED        32744
00631 #define OBM_ZOOMD          32745
00632 #define OBM_REDUCED         32746
00633 #define OBM_RESTORE        32747
00634 #define OBM_ZOOM           32748
00635 #define OBM_REDUCE          32749
00636 #define OBM_LFARROW        32750
00637 #define OBM_RGARROW        32751
00638 #define OBM_DNARROW        32752
00639 #define OBM_UPARROW        32753
00640 #define OBM_CLOSE          32754
00641 #define OBM_OLD_RESTORE     32755
00642 #define OBM_OLD_ZOOM        32756
00643 #define OBM_OLD_REDUCE      32757
00644 #define OBM_BTNCORNERS     32758
00645 #define OBM_CHECKBOXES     32759
00646 #define OBM_CHECK           32760
00647 #define OBM_BTSIZE         32761
00648 #define OBM_OLD_LFARROW     32762
00649 #define OBM_OLD_RGARROW     32763
00650 #define OBM_OLD_DNARROW     32764
00651 #define OBM_OLD_UPARROW     32765
00652 #define OBM_SIZE            32766
00653 #define OBM_OLD_CLOSE       32767
00654
00655 #define OCR_NORMAL          32512
00656 #define OCR_IBEAM           32513
00657 #define OCR_WAIT            32514
00658 #define OCR_CROSS           32515
00659 #define OCR_UP              32516
00660 #define OCR_SIZE            32640
00661 #define OCR_ICON            32641
00662 #define OCR_SIZENWSE        32642
00663 #define OCR_SIZENESW        32643
00664 #define OCR_SIZEWE          32644
00665 #define OCR_SIZENS          32645
00666 #define OCR_SIZEALL         32646
00667 #define OCR_ICOCUR          32647
00668 #define OCR_NO              32648
```

```

00669 #define OCR_HAND 32649
00670 #define OCR_APPSTARTING 32650
00671 #define OCR_HELP 32651
00672
00673 /* only defined in wine (FIXME) */
00674 #define OCR_DRAGOBJECT 32653
00675
00676 #define OIC_SAMPLE 32512
00677 #define OIC_HAND 32513
00678 #define OIC_QUES 32514
00679 #define OIC_BANG 32515
00680 #define OIC_NOTE 32516
00681 #define OIC_WINLOGO 32517
00682 #define OIC_WARNING OIC_BANG
00683 #define OIC_ERROR OIC_HAND
00684 #define OIC_INFORMATION OIC_NOTE
00685
00686 #ifndef NOCOLOR
00687
00688 #define COLOR_SCROLLBAR 0
00689 #define COLOR_BACKGROUND 1
00690 #define COLOR_ACTIVECAPTION 2
00691 #define COLOR_INACTIVECAPTION 3
00692 #define COLOR_MENU 4
00693 #define COLOR_WINDOW 5
00694 #define COLOR_WINDOWFRAME 6
00695 #define COLOR_MENUTEXT 7
00696 #define COLOR_WINDOWTEXT 8
00697 #define COLOR_CAPTIONTEXT 9
00698 #define COLOR_ACTIVEBORDER 10
00699 #define COLOR_INACTIVEBORDER 11
00700 #define COLOR_APPWORKSPACE 12
00701 #define COLOR_HIGHLIGHT 13
00702 #define COLOR_HIGHLIGHTTEXT 14
00703 #define COLOR_BTNFACE 15
00704 #define COLOR_BTNSHADOW 16
00705 #define COLOR_GRAYTEXT 17
00706 #define COLOR_BTNTEXT 18
00707 #define COLOR_INACTIVECAPTIONTEXT 19
00708 #define COLOR_BTNHIGHLIGHT 20
00709 /* win95 colors */
00710 #define COLOR_3DDKSHADOW 21
00711 #define COLOR_3DLIGHT 22
00712 #define COLOR_INFOTEXT 23
00713 #define COLOR_INFOBK 24
00714 #define COLOR_DESKTOP COLOR_BACKGROUND
00715 #define COLOR_3DFACE COLOR_BTNFACE
00716 #define COLOR_3DSHADOW COLOR_BTNSHADOW
00717 #define COLOR_3DHIGHLIGHT COLOR_BTNHIGHLIGHT
00718 #define COLOR_3DHILIGHT COLOR_BTNHIGHLIGHT
00719 #define COLOR_BTNHILIGHT COLOR_BTNHIGHLIGHT
00720 /* win98 colors */
00721 #define COLOR_ALTERNATEBTNFACE 25 /* undocumented, constant's name unknown */
00722 #define COLOR_HOTLIGHT 26
00723 #define COLOR_GRADIENTACTIVECAPTION 27
00724 #define COLOR_GRADIENTINACTIVECAPTION 28
00725
00726 /* WM_CTLCOLOR values */
00727 #define CTLCOLOR_MSGBOX 0
00728 #define CTLCOLOR_EDIT 1
00729 #define CTLCOLOR_LISTBOX 2
00730 #define CTLCOLOR_BTN 3
00731 #define CTLCOLOR_DLG 4
00732 #define CTLCOLOR_SCROLLBAR 5
00733 #define CTLCOLOR_STATIC 6
00734
00735 COLORREF WINAPI GetSysColor(INT);
00736 BOOL WINAPI SetSysColors(INT, const INT*, const COLORREF*);
00737
00738 #endif /* NOCOLOR */
00739
00740 /* Edit control messages */
00741 #define EM_GETSEL 0x00b0
00742 #define EM_SETSEL 0x00b1
00743 #define EM_GETRECT 0x00b2
00744 #define EM_SETRECT 0x00b3
00745 #define EM_SETRECTNP 0x00b4
00746 #define EM_SCROLL 0x00b5
00747 #define EM_LINESCROLL 0x00b6
00748 #define EM_SCROLLCARET 0x00b7
00749 #define EM_GETMODIFY 0x00b8
00750 #define EM_SETMODIFY 0x00b9
00751 #define EM_GETLINECOUNT 0x00ba
00752 #define EM_LINEINDEX 0x00bb
00753 #define EM_SETHANDLE 0x00bc
00754 #define EM_GETHANDLE 0x00bd
00755 #define EM_GETTHUMB 0x00be

```

```
00756 /* FIXME : missing from specs 0x00bf and 0x00c0 */
00757 #define EM_LINELENGTH 0x00c1
00758 #define EM_REPLACESEL 0x00c2
00759 /* FIXME : missing from specs 0x00c3 */
00760 #define EM_GETLINE 0x00c4
00761 #define EM_LIMITTEXT 0x00c5
00762 #define EM_CANUNDO 0x00c6
00763 #define EM_UNDO 0x00c7
00764 #define EM_FMTLINES 0x00c8
00765 #define EM_LINEFROMCHAR 0x00c9
00766 /* FIXME : missing from specs 0x00ca */
00767 #define EM_SETTABSTOPS 0x00cb
00768 #define EM_SETPASSWORDCHAR 0x00cc
00769 #define EM_EMPTYUNDOBUFFER 0x00cd
00770 #define EM_GETFIRSTVISIBLELINE 0x00ce
00771 #define EM_SETREADONLY 0x00cf
00772 #define EM_SETWORDBREAKPROC 0x00d0
00773 #define EM_GETWORDBREAKPROC 0x00d1
00774 #define EM_GETPASSWORDCHAR 0x00d2
00775 #define EM_SETMARGINS 0x00d3
00776 #define EM_GETMARGINS 0x00d4
00777 #define EM_GETLIMITTEXT 0x00d5
00778 #define EM_POSFROMCHAR 0x00d6
00779 #define EM_CHARFROMPOS 0x00d7
00780 /* a name change since win95 */
00781 #define EM_SETLIMITTEXT EM_LIMITTEXT
00782
00783 /* EDITWORDBREAKPROC code values */
00784 #define WB_LEFT 0
00785 #define WB_RIGHT 1
00786 #define WB_ISDELIMITER 2
00787
00788 /* Edit control notification codes */
00789 #define EN_SETFOCUS 0x0100
00790 #define EN_KILLFOCUS 0x0200
00791 #define EN_CHANGE 0x0300
00792 #define EN_UPDATE 0x0400
00793 #define EN_ERRSPACE 0x0500
00794 #define EN_MAXTEXT 0x0501
00795 #define EN_HSCROLL 0x0601
00796 #define EN_VSCROLL 0x0602
00797
00798 /* New since win95 : EM_SETMARGIN parameters */
00799 #define EC_LEFTMARGIN 0x0001
00800 #define EC_RIGHTMARGIN 0x0002
00801 #define EC_USEFONTINFO 0xffff
00802
00803
00804 /* GetSystemMetrics() codes */
00805 #define SM_CXSCREEN 0
00806 #define SM_CYSCREEN 1
00807 #define SM_CXVSCROLL 2
00808 #define SM_CXHSCROLL 3
00809 #define SM_CYCAPTION 4
00810 #define SM_CXBORDER 5
00811 #define SM_CYBORDER 6
00812 #define SM_CXDLGFRAME 7
00813 #define SM_CYDLGFRAME 8
00814 #define SM_CYVTHUMB 9
00815 #define SM_CXHTHUMB 10
00816 #define SM_CXICON 11
00817 #define SM_CYICON 12
00818 #define SM_CXCURSOR 13
00819 #define SM_CYCURSOR 14
00820 #define SM_CYMENU 15
00821 #define SM_CXFULLSCREEN 16
00822 #define SM_CYFULLSCREEN 17
00823 #define SM_CYKANJIWINDOW 18
00824 #define SM_MOUSEPRESENT 19
00825 #define SM_CYVSCROLL 20
00826 #define SM_CXHSCROLL 21
00827 #define SM_DEBUG 22
00828 #define SM_SWAPBUTTON 23
00829 #define SM_RESERVED1 24
00830 #define SM_RESERVED2 25
00831 #define SM_RESERVED3 26
00832 #define SM_RESERVED4 27
00833 #define SM_CXMIN 28
00834 #define SM_CYMIN 29
00835 #define SM_CXSIZE 30
00836 #define SM_CYSIZE 31
00837 #define SM_CXFRAME 32
00838 #define SM_CYFRAME 33
00839 #define SM_CXMINTRACK 34
00840 #define SM_CYMINTRACK 35
00841 #define SM_CXDOUBLECLK 36
00842 #define SM_CYDOUBLECLK 37
```

```

00843 #define SM_CXICONSPACING      38
00844 #define SM_CYICONSPACING      39
00845 #define SM_MENUDROPALIGNMENT  40
00846 #define SM_PENWINDOWS          41
00847 #define SM_DBCSENABLED        42
00848 #define SM_CMOUSEBUTTONS      43
00849 #define SM_CXFIXEDFRAME        SM_CXDLGFRAME
00850 #define SM_CYFIXEDFRAME        SM_CYDLGFRAME
00851 #define SM_CXSIZEFRAME         SM_CXFRAME
00852 #define SM_CYSIZEFRAME         SM_CYFRAME
00853 #define SM_SECURE              44
00854 #define SM_CXEDGE              45
00855 #define SM_CYEDGE              46
00856 #define SM_CXMINSPPACING      47
00857 #define SM_CYMINSPPACING      48
00858 #define SM_CXSMICON           49
00859 #define SM_CYSMICON           50
00860 #define SM_CYSMCAPTION        51
00861 #define SM_CXSMSIZE           52
00862 #define SM_CYSMSIZE           53
00863 #define SM_CXMENUSIZE         54
00864 #define SM_CYMENUSIZE         55
00865 #define SM_ARRANGE             56
00866 #define SM_CXMINIMIZED        57
00867 #define SM_CYMINIMIZED        58
00868 #define SM_CXMAXTRACK          59
00869 #define SM_CYMAXTRACK          60
00870 #define SM_CXMAXIMIZED        61
00871 #define SM_CYMAXIMIZED        62
00872 #define SM_NETWORK            63
00873 #define SM_CLEANBOOT           67
00874 #define SM_CXDRAG              68
00875 #define SM_CYDRAG              69
00876 #define SM_SHOWSOUNDS         70
00877 #define SM_CXMENUCHECK        71
00878 #define SM_CYMENUCHECK        72
00879 #define SM_SLOWMACHINE        73
00880 #define SM_MIDEASTENABLED      74
00881 #define SM_MOUSEWHEELPRESENT   75
00882 #define SM_XVIRTUALSCREEN      76
00883 #define SM_YVIRTUALSCREEN      77
00884 #define SM_CXVIRTUALSCREEN     78
00885 #define SM_CYVIRTUALSCREEN     79
00886 #define SM_CMONITORS           80
00887 #define SM_SAMEDISPLAYFORMAT   81
00888 #define SM_CMETRICS            83
00889
00890
00891 /* Messages */
00892
00893 /* WM_GETDLGCODE values */
00894
00895
00896 #define WM_NULL                 0x0000
00897 #define WM_CREATE               0x0001
00898 #define WM_DESTROY              0x0002
00899 #define WM_MOVE                 0x0003
00900 #define WM_SIZEWAIT            0x0004
00901 #define WM_SIZE                 0x0005
00902 #define WM_ACTIVATE             0x0006
00903 #define WM_SETFOCUS             0x0007
00904 #define WM_KILLFOCUS            0x0008
00905 #define WM_SETVISIBLE           0x0009
00906 #define WM_ENABLE               0x000a
00907 #define WM_SETREDRAW            0x000b
00908 #define WM_SETTEXT              0x000c
00909 #define WM_GETTEXT              0x000d
00910 #define WM_GETTEXTLENGTH       0x000e
00911 #define WM_PAINT                0x000f
00912 #define WM_CLOSE                0x0010
00913 #define WM_QUERYENDSESSION      0x0011
00914 #define WM_QUIT                 0x0012
00915 #define WM_QUERYOPEN            0x0013
00916 #define WM_ERASEBKGD           0x0014
00917 #define WM_SYSCOLORCHANGE       0x0015
00918 #define WM_ENDSESSION           0x0016
00919 #define WM_SYSTEMERROR          0x0017
00920 #define WM_SHOWWINDOW           0x0018
00921 #define WM_CTLCOLOR             0x0019
00922 #define WM_WININICHANGE         0x001a
00923 #define WM_SETTINGCHANGE        WM_WININICHANGE
00924 #define WM_DEVMODECHANGE        0x001b
00925 #define WM_ACTIVATEAPP          0x001c
00926 #define WM_FONTCHANGE           0x001d
00927 #define WM_TIMECHANGE           0x001e
00928 #define WM_CANCELMODE           0x001f
00929 #define WM_SETCURSOR            0x0020

```

```

00930 #define WM_MOUSEACTIVATE      0x0021
00931 #define WM_CHILDACTIVATE        0x0022
00932 #define WM_QUEUESYNC            0x0023
00933 #define WM_GETMINMAXINFO        0x0024
00934
00935 #define WM_PAINTICON             0x0026
00936 #define WM_ICONERASEBKGD        0x0027
00937 #define WM_NEXTDLGCTL           0x0028
00938 #define WM_ALTTABACTIVE         0x0029
00939 #define WM_SPOOLERSTATUS        0x002a
00940 #define WM_DRAWITEM             0x002b
00941 #define WM_MEASUREITEM          0x002c
00942 #define WM_DELETEITEM           0x002d
00943 #define WM_KEYTOITEM            0x002e
00944 #define WM_CHAROITEM           0x002f
00945 #define WM_SETFONT              0x0030
00946 #define WM_GETFONT              0x0031
00947 #define WM_SETHOTKEY            0x0032
00948 #define WM_GETHOTKEY            0x0033
00949 #define WM_FILESYSCCHANGE       0x0034
00950 #define WM_ISACTIVEICON         0x0035
00951 #define WM_QUERYPARKICON        0x0036
00952 #define WM_QUERYDRAGON         0x0037
00953 #define WM_QUERYSAVESTATE       0x0038
00954 #define WM_COMPAREITEM          0x0039
00955 #define WM_TESTING              0x003a
00956
00957 #define WM_OTHERWINDOWCREATED    0x003c
00958 #define WM_OTHERWINDOWDESTROYED 0x003d
00959 #define WM_ACTIVATESHELLWINDOW  0x003e
00960
00961 #define WM_COMPACTING            0x0041
00962
00963 #define WM_COMMNOTIFY           0x0044
00964 #define WM_WINDOWPOSCHANGING    0x0046
00965 #define WM_WINDOWPOSCHANGED     0x0047
00966 #define WM_POWER                0x0048
00967
00968 /* Win32 4.0 messages */
00969 #define WM_COPYDATA             0x004a
00970 #define WM_CANCELJOURNAL        0x004b
00971 #define WM_NOTIFY               0x004e
00972 #define WM_INPUTLANGCHANGEREQUEST 0x0050
00973 #define WM_INPUTLANGCHANGE      0x0051
00974 #define WM_TCARD                0x0052
00975 #define WM_HELP                 0x0053
00976 #define WM_USERCHANGED          0x0054
00977 #define WM_NOTIFYFORMAT         0x0055
00978
00979 #define WM_CONTEXTMENU          0x007b
00980 #define WM_STYLECHANGING        0x007c
00981 #define WM_STYLECHANGED         0x007d
00982 #define WM_DISPLAYCHANGE        0x007e
00983 #define WM_GETICON              0x007f
00984 #define WM_SETICON              0x0080
00985
00986 /* Non-client system messages */
00987 #define WM_NCCREATE              0x0081
00988 #define WM_NCDESTROY             0x0082
00989 #define WM_NCCALCSIZE           0x0083
00990 #define WM_NCHITTEST            0x0084
00991 #define WM_NCPAINT              0x0085
00992 #define WM_NCACTIVATE           0x0086
00993
00994 #define WM_GETDLGCODE            0x0087
00995 #define WM_SYNCPAINT            0x0088
00996 #define WM_SYNCTASK             0x0089
00997
00998 /* Non-client mouse messages */
00999 #define WM_NCMOUSEMOVE          0x00a0
01000 #define WM_NCLBUTTONDOWN        0x00a1
01001 #define WM_NCLBUTTONUP          0x00a2
01002 #define WM_NCLBUTTONDBLCLK      0x00a3
01003 #define WM_NCRBUTTONDOWN        0x00a4
01004 #define WM_NCRBUTTONUP          0x00a5
01005 #define WM_NCRBUTTONDBLCLK      0x00a6
01006 #define WM_NCMBUTTONDOWN       0x00a7
01007 #define WM_NCMBUTTONUP          0x00a8
01008 #define WM_NCMBUTTONDBLCLK      0x00a9
01009
01010 #define WM_NCXBUTTONDOWN        0x00ab
01011 #define WM_NCXBUTTONUP          0x00ac
01012 #define WM_NCXBUTTONDBLCLK      0x00ad
01013
01014 /* Keyboard messages */
01015 #define WM_KEYDOWN              0x0100
01016 #define WM_KEYUP                0x0101

```

```
01017 #define WM_CHAR                0x0102
01018 #define WM_DEADCHAR             0x0103
01019 #define WM_SYSKEYDOWN           0x0104
01020 #define WM_SYSKEYUP              0x0105
01021 #define WM_SYSCHAR               0x0106
01022 #define WM_SYSDEADCHAR          0x0107
01023 #define WM_KEYFIRST              WM_KEYDOWN
01024 #define WM_KEYLAST              0x0108
01025
01026 /* Win32 4.0 messages for IME */
01027 #define WM_IME_STARTCOMPOSITION  0x010d
01028 #define WM_IME_ENDCOMPOSITION    0x010e
01029 #define WM_IME_COMPOSITION      0x010f
01030 #define WM_IME_KEYLAST          0x010f
01031
01032 #define WM_INITDIALOG            0x0110
01033 #define WM_COMMAND               0x0111
01034 #define WM_SYSCOMMAND           0x0112
01035 #define WM_TIMER                0x0113
01036 #define WM_SYSTIMER             0x0118
01037
01038 /* scroll messages */
01039 #define WM_HSCROLL              0x0114
01040 #define WM_VSCROLL              0x0115
01041
01042 /* Menu messages */
01043 #define WM_INITMENU             0x0116
01044 #define WM_INITMENUPOPUP       0x0117
01045
01046 #define WM_MENUSELECT          0x011F
01047 #define WM_MENUCHAR            0x0120
01048 #define WM_ENTERIDLE           0x0121
01049
01050 #define WM_LBTRACKPOINT        0x0131
01051
01052 /* Win32 CTLCOLOR messages */
01053 #define WM_CTLCOLOMSGBOX       0x0132
01054 #define WM_CTLCOLOREDIT        0x0133
01055 #define WM_CTLCOLORLISTBOX     0x0134
01056 #define WM_CTLCOLORBTN         0x0135
01057 #define WM_CTLCOLORDLG         0x0136
01058 #define WM_CTLCOLORSCROLLBAR   0x0137
01059 #define WM_CTLCOLORSTATIC      0x0138
01060
01061 /* Mouse messages */
01062 #define WM_MOUSEMOVE           0x0200
01063 #define WM_LBUTTONDOWN         0x0201
01064 #define WM_LBUTTONUP           0x0202
01065 #define WM_LBUTTONDBLCLK      0x0203
01066 #define WM_RBUTTONDOWN         0x0204
01067 #define WM_RBUTTONUP           0x0205
01068 #define WM_RBUTTONDBLCLK      0x0206
01069 #define WM_MBUTTONDOWN         0x0207
01070 #define WM_MBUTTONUP           0x0208
01071 #define WM_MBUTTONDBLCLK      0x0209
01072 #define WM_MOUSEWHEEL          0x020A
01073 #define WM_XBUTTONDOWN         0x020B
01074 #define WM_XBUTTONUP           0x020C
01075 #define WM_XBUTTONDBLCLK      0x020D
01076
01077 #define WM_MOUSEFIRST          0x0200
01078 #define WM_MOUSELAST          0x020D
01079
01080 #define WHEEL_DELTA            120
01081 #define WHEEL_PAGESCROLL      (UINT_MAX)
01082 #define WM_PARENTNOTIFY        0x0210
01083 #define WM_ENTERMENULOOP       0x0211
01084 #define WM_EXITMENULOOP        0x0212
01085 #define WM_NEXTMENU            0x0213
01086
01087 /* Win32 4.0 messages */
01088 #define WM_SIZING               0x0214
01089 #define WM_CAPTURECHANGED      0x0215
01090 #define WM_MOVING               0x0216
01091 #define WM_POWERBROADCAST       0x0218
01092 #define WM_DEVICECHANGE         0x0219
01093
01094 /* wParam for WM_SIZING message */
01095 #define WMSZ_LEFT               1
01096 #define WMSZ_RIGHT              2
01097 #define WMSZ_TOP                3
01098 #define WMSZ_Topleft            4
01099 #define WMSZ_TOPRIGHT           5
01100 #define WMSZ_BOTTOM             6
01101 #define WMSZ_BOTTOMLEFT         7
01102 #define WMSZ_BOTTOMRIGHT        8
01103
```

```
01104  /* MDI messages */
01105  #define WM_MDICREATE      0x0220
01106  #define WM_MDIDESTROY    0x0221
01107  #define WM_MDIACTIVATE   0x0222
01108  #define WM_MDIRESTORE    0x0223
01109  #define WM_MDINEXT       0x0224
01110  #define WM_MDIMAXIMIZE   0x0225
01111  #define WM_MDITILE       0x0226
01112  #define WM_MDICASCADE    0x0227
01113  #define WM_MDIICONARRANGE 0x0228
01114  #define WM_MDIGETACTIVE  0x0229
01115  #define WM_MDIREFRESHMENU 0x0234
01116
01117  /* D&D messages */
01118  #define WM_DROPBJECT      0x022A
01119  #define WM_QUERYDROPOBJECT 0x022B
01120  #define WM_BEGINDRAG     0x022C
01121  #define WM_DRAGLOOP      0x022D
01122  #define WM_DRAGSELECT    0x022E
01123  #define WM_DRAGMOVE      0x022F
01124  #define WM_MDISETMENU    0x0230
01125
01126  #define WM_ENTERSIZEMOVE  0x0231
01127  #define WM_EXITSIZEMOVE  0x0232
01128  #define WM_DROPFILES     0x0233
01129
01130
01131  /* Win32 4.0 messages for IME */
01132  #define WM_IME_SETCONTEXT 0x0281
01133  #define WM_IME_NOTIFY     0x0282
01134  #define WM_IME_CONTROL    0x0283
01135  #define WM_IME_COMPOSITIONFULL 0x0284
01136  #define WM_IME_SELECT     0x0285
01137  #define WM_IME_CHAR       0x0286
01138  /* Win32 5.0 messages for IME */
01139  #define WM_IME_REQUEST    0x0288
01140
01141  /* Win32 4.0 messages for IME */
01142  #define WM_IME_KEYDOWN    0x0290
01143  #define WM_IME_KEYUP      0x0291
01144
01145  /* Clipboard command messages */
01146  #define WM_CUT            0x0300
01147  #define WM_COPY           0x0301
01148  #define WM_PASTE          0x0302
01149  #define WM_CLEAR          0x0303
01150  #define WM_UNDO           0x0304
01151
01152  /* Clipboard owner messages */
01153  #define WM_RENDERFORMAT   0x0305
01154  #define WM_RENDERALLFORMATS 0x0306
01155  #define WM_DESTROYCLIPBOARD 0x0307
01156
01157  /* Clipboard viewer messages */
01158  #define WM_DRAWCLIPBOARD  0x0308
01159  #define WM_PAINTCLIPBOARD 0x0309
01160  #define WM_VSCROLLCLIPBOARD 0x030A
01161  #define WM_SIZECLIPBOARD  0x030B
01162  #define WM_ASKCBFORMATNAME 0x030C
01163  #define WM_CHANGECHAIN     0x030D
01164  #define WM_HSCROLLCLIPBOARD 0x030E
01165
01166  #define WM_QUERYNEWPALETTE 0x030F
01167  #define WM_PALETTEISCHANGING 0x0310
01168  #define WM_PALETTECHANGED  0x0311
01169  #define WM_HOTKEY          0x0312
01170
01171  #define WM_PRINT           0x0317
01172  #define WM_PRINTCLIENT    0x0318
01173
01174  #define WM_PENWINFIRST     0x0380
01175  #define WM_PENWINLAST     0x038F
01176
01177
01178  #define WM_APP             0x8000
01179
01180  /* MsgWaitForMultipleObjectsEx flags */
01181  #define MWMO_WAITALL      0x0001
01182  #define MWMO_ALERTTABLE   0x0002
01183  #define MWMO_INPUTAVAILABLE 0x0004
01184
01185  #define DLGC_WANTARROWS    0x0001
01186  #define DLGC_WANTTAB       0x0002
01187  #define DLGC_WANTALLKEYS   0x0004
01188  #define DLGC_WANTMESSAGE   0x0004
01189  #define DLGC_HASSETSEL     0x0008
01190  #define DLGC_DEFPUSHBUTTON 0x0010
```

```

01191 #define DLGC_UNDEFPUSHBUTTON 0x0020
01192 #define DLGC_RADIOBUTTON      0x0040
01193 #define DLGC_WANTCHARS        0x0080
01194 #define DLGC_STATIC            0x0100
01195 #define DLGC_BUTTON            0x2000
01196
01197 /* Standard dialog button IDs */
01198 #define IDOK                    1
01199 #define IDCANCEL                2
01200 #define IDABORT                3
01201 #define IDRETRY                4
01202 #define IDIGNORE               5
01203 #define IDYES                  6
01204 #define IDNO                   7
01205 #define IDCLOSE                8
01206 #define IDHELP                 9
01207
01208 /***** Window classes *****/
01209
01210 typedef struct tagCREATESTRUCTA
01211 {
01212     LPVOID      lpCreateParams;
01213     HINSTANCE   hInstance;
01214     HMENU       hMenu;
01215     HWND        hwndParent;
01216     INT         cy;
01217     INT         cx;
01218     INT         y;
01219     INT         x;
01220     LONG        style;
01221     LPCSTR      lpszName;
01222     LPCSTR      lpszClass;
01223     DWORD       dwExStyle;
01224 } CREATESTRUCTA, *LPCREATESTRUCTA;
01225
01226 typedef struct
01227 {
01228     LPVOID      lpCreateParams;
01229     HINSTANCE   hInstance;
01230     HMENU       hMenu;
01231     HWND        hwndParent;
01232     INT         cy;
01233     INT         cx;
01234     INT         y;
01235     INT         x;
01236     LONG        style;
01237     LPCWSTR     lpszName;
01238     LPCWSTR     lpszClass;
01239     DWORD       dwExStyle;
01240 } CREATESTRUCTW, *LPCREATESTRUCTW;
01241
01242 DECL_WINELIB_TYPE_AW(CREATESTRUCT)
01243 DECL_WINELIB_TYPE_AW(LPCREATESTRUCT)
01244
01245 typedef struct
01246 {
01247     HDC      hdc;
01248     BOOL     fErase;
01249     RECT     rcPaint;
01250     BOOL     fRestore;
01251     BOOL     fIncUpdate;
01252     BYTE     rgbReserved[32];
01253 } PAINTSTRUCT, *PPAINTSTRUCT, *LPPAINTSTRUCT;
01254
01255 typedef struct
01256 {
01257     HMENU     hWindowMenu;
01258     UINT      idFirstChild;
01259 } CLIENTCREATESTRUCT, *LPCLIENTCREATESTRUCT;
01260
01261
01262 typedef struct
01263 {
01264     LPCSTR     szClass;
01265     LPCSTR     szTitle;
01266     HINSTANCE  hOwner;
01267     INT        x;
01268     INT        y;
01269     INT        cx;
01270     INT        cy;
01271     DWORD      style;
01272     LPARAM     lParam;
01273 } MDICREATESTRUCTA, *LPMDCREATESTRUCTA;
01274
01275 typedef struct
01276 {
01277     LPCWSTR     szClass;

```

```

01278     LPCWSTR      szTitle;
01279     HINSTANCE     hOwner;
01280     INT           x;
01281     INT           y;
01282     INT           cx;
01283     INT           cy;
01284     DWORD         style;
01285     LPARAM        lParam;
01286 } MDICREATESTRUCTW, *LPMDCREATESTRUCTW;
01287
01288 DECL_WINELIB_TYPE_AW(MDICREATESTRUCT)
01289 DECL_WINELIB_TYPE_AW(LPMDCREATESTRUCT)
01290
01291 #define MDITILE_VERTICAL      0x0000
01292 #define MDITILE_HORIZONTAL   0x0001
01293 #define MDITILE_SKIPDISABLED 0x0002
01294
01295 #define MDIS_ALLCHILDSTYLES  0x0001
01296
01297 typedef struct {
01298     DWORD     styleOld;
01299     DWORD     styleNew;
01300 } STYLESTRUCT, *LPSTYLESTRUCT;
01301
01302 #define WC_DIALOGA MAKEINTATOMA(0x8002)
01303 #define WC_DIALOGW MAKEINTATOMW(0x8002)
01304 #define WC_DIALOG  WINELIB_NAME_AW(WC_DIALOG)
01305
01306 /* Offsets for GetWindowLong() and GetWindowWord() */
01307 #define GWL_USERDATA      (-21)
01308 #define GWL_EXSTYLE       (-20)
01309 #define GWL_STYLE         (-16)
01310 #define GWL_ID            (-12)
01311 #define GWL_HWNDPARENT    (-8)
01312 #define GWL_HINSTANCE     (-6)
01313 #define GWL_WNDPROC       (-4)
01314 #define DWL_MSGRESULT     0
01315 #define DWL_DLGPROC       4
01316 #define DWL_USER          8
01317
01318 /* GetWindow() constants */
01319 #define GW_HWNDFIRST      0
01320 #define GW_HWNDLAST      1
01321 #define GW_HWNDNEXT      2
01322 #define GW_HWNDPREV      3
01323 #define GW_OWNER         4
01324 #define GW_CHILD         5
01325
01326 /* GetAncestor() constants */
01327 #define GA_PARENT         1
01328 #define GA_ROOT           2
01329 #define GA_ROOTOWNER      3
01330
01331 /* WM_GETMINMAXINFO struct */
01332 typedef struct
01333 {
01334     POINT      ptReserved;
01335     POINT      ptMaxSize;
01336     POINT      ptMaxPosition;
01337     POINT      ptMinTrackSize;
01338     POINT      ptMaxTrackSize;
01339 } MINMAXINFO, *PMINMAXINFO, *LPMINMAXINFO;
01340
01341
01342 /* RedrawWindow() flags */
01343 #define RDW_INVALIDATE     0x0001
01344 #define RDW_INTERNALPAINT 0x0002
01345 #define RDW_ERASE         0x0004
01346 #define RDW_VALIDATE     0x0008
01347 #define RDW_NOINTERNALPAINT 0x0010
01348 #define RDW_NOERASE       0x0020
01349 #define RDW_NOCHILDREN    0x0040
01350 #define RDW_ALLCHILDREN   0x0080
01351 #define RDW_UPDATENOW     0x0100
01352 #define RDW_ERASENOW      0x0200
01353 #define RDW_FRAME         0x0400
01354 #define RDW_NOFRAME       0x0800
01355
01356 /* debug flags */
01357 #define DBGFILL_ALLOC     0xfd
01358 #define DBGFILL_FREE      0xfb
01359 #define DBGFILL_BUFFER    0xf9
01360 #define DBGFILL_STACK     0xf7
01361
01362 /* WM_WINDOWPOSCHANGING/CHANGED struct */
01363 typedef struct tagWINDOWPOS
01364 {

```

```

01365     HWND  hwnd;
01366     HWND  hwndInsertAfter;
01367     INT    x;
01368     INT    y;
01369     INT    cx;
01370     INT    cy;
01371     UINT   flags;
01372 } WINDOWPOS, *PWINDOWPOS, *LPWINDOWPOS;
01373
01374
01375 /* WM_MOUSEACTIVATE return values */
01376 #define MA_ACTIVATE      1
01377 #define MA_ACTIVATEANDEAT 2
01378 #define MA_NOACTIVATE    3
01379 #define MA_NOACTIVATEANDEAT 4
01380
01381 /* WM_ACTIVATE wParam values */
01382 #define WA_INACTIVE      0
01383 #define WA_ACTIVE        1
01384 #define WA_CLICKACTIVE   2
01385
01386 /* WM_GETICON/WM_SETICON params values */
01387 #define ICON_SMALL       0
01388 #define ICON_BIG         1
01389
01390 /* WM_NCCALCSIZE parameter structure */
01391 typedef struct
01392 {
01393     RECT      rgrc[3];
01394     WINDOWPOS *lppos;
01395 } NCCALCSIZE_PARAMS, *LPNCCALCSIZE_PARAMS;
01396
01397
01398 /* WM_NCCALCSIZE return flags */
01399 #define WVR_ALIGNTOP      0x0010
01400 #define WVR_ALIGNLEFT    0x0020
01401 #define WVR_ALIGNBOTTOM  0x0040
01402 #define WVR_ALIGNRIGHT   0x0080
01403 #define WVR_HREDRAW       0x0100
01404 #define WVR_VREDRAW       0x0200
01405 #define WVR_REDRAW        (WVR_HREDRAW | WVR_VREDRAW)
01406 #define WVR_VALIDRECTS    0x0400
01407
01408 /* WM_NCHITTEST return codes */
01409 #define HTERROR            (-2)
01410 #define HTTRANSPARENT      (-1)
01411 #define HTNOWHERE          0
01412 #define HTCLIENT           1
01413 #define HTCAPTION          2
01414 #define HTSYSMENU          3
01415 #define HTSIZE              4
01416 #define HTMENU             5
01417 #define HTHSCROLL          6
01418 #define HTVSCROLL          7
01419 #define HTMINBUTTON        8
01420 #define HTMAXBUTTON        9
01421 #define HTLEFT             10
01422 #define HTRIGHT            11
01423 #define HTTOP              12
01424 #define HTTOPLEFT          13
01425 #define HTTOPRIGHT         14
01426 #define HTBOTTOM           15
01427 #define HTBOTTOMLEFT       16
01428 #define HTBOTTOMRIGHT      17
01429 #define HTBORDER           18
01430 #define HTGROWBOX           HTSIZE
01431 #define HTREDUCE            HTMINBUTTON
01432 #define HTZOOM              HTMAXBUTTON
01433 #define HTOBJECT            19
01434 #define HTCLOSE             20
01435 #define HTHelp              21
01436 #define HTSIZEFIRST         HTLEFT
01437 #define HTSIZELAST          HTBOTTOMRIGHT
01438
01439 /* SendMessageTimeout flags */
01440 #define SMT_NORMAL         0x0000
01441 #define SMT_BLOCK          0x0001
01442 #define SMT_ABORTIFHUNG    0x0002
01443 #define SMT_NOTIMEOUTIFNOTHUNG 0x0008
01444
01445 /* WM_SYSCOMMAND parameters */
01446 #ifndef SC_SIZE /* at least HP-UX: already defined in /usr/include/sys/signal.h */
01447 #undef SC_SIZE
01448 #endif
01449 #define SC_SIZE            0xf000
01450 #define SC_MOVE            0xf010
01451 #define SC_MINIMIZE        0xf020

```

```

01452 #define SC_MAXIMIZE      0xf030
01453 #define SC_NEXTWINDOW     0xf040
01454 #define SC_PREVWINDOW     0xf050
01455 #define SC_CLOSE          0xf060
01456 #define SC_VSCROLL        0xf070
01457 #define SC_HSCROLL        0xf080
01458 #define SC_MOUSEMENU      0xf090
01459 #define SC_KEYMENU        0xf100
01460 #define SC_ARRANGE        0xf110
01461 #define SC_RESTORE        0xf120
01462 #define SC_TASKLIST       0xf130
01463 #define SC_SCREENSAVE     0xf140
01464 #define SC_HOTKEY         0xf150
01465 /* Win32 4.0 */
01466 #define SC_DEFAULT        0xf160
01467 #define SC_MONITORPOWER   0xf170
01468 #define SC_CONTEXTHELP    0xf180
01469 #define SC_SEPARATOR      0xf00f
01470
01471 /* obsolete names(SC_ICON and SC_ZOOM) */
01472 #define SC_ICON           SC_MINIMIZE
01473 #define SC_ZOOM           SC_MAXIMIZE
01474
01475
01476 #define CS_VREDRAW         0x0001
01477 #define CS_HREDRAW         0x0002
01478 #define CS_KEYCVTWINDOW   0x0004
01479 #define CS_DBLCLKS        0x0008
01480 #define CS_OWNDC          0x0020
01481 #define CS_CLASSDC        0x0040
01482 #define CS_PARENTDC       0x0080
01483 #define CS_NOKEYCVT       0x0100
01484 #define CS_NOCLOSE        0x0200
01485 #define CS_SAVEBITS       0x0800
01486 #define CS_BYTEALIGNCLIENT 0x1000
01487 #define CS_BYTEALIGNWINDOW 0x2000
01488 #define CS_GLOBALCLASS    0x4000
01489 #define CS_IME             0x00010000
01490
01491 #define PRF_CHECKVISIBLE   0x00000001L
01492 #define PRF_NONCLIENT     0x00000002L
01493 #define PRF_CLIENT         0x00000004L
01494 #define PRF_ERASEBKGDND   0x00000008L
01495 #define PRF_CHILDREN      0x00000010L
01496 #define PRF_OWNED         0x00000020L
01497
01498 /* Offsets for GetClassLong() and GetClassWord() */
01499 #define GCL_MENUNAME       (-8)
01500 #define GCL_HBRBACKGROUND (-10)
01501 #define GCL_HCURSOR        (-12)
01502 #define GCL_HICON          (-14)
01503 #define GCL_HMODULE        (-16)
01504 #define GCL_CBWNDEXTRA     (-18)
01505 #define GCL_CBCLSEXTRA     (-20)
01506 #define GCL_WNDPROC        (-24)
01507 #define GCL_STYLE          (-26)
01508 #define GCW_ATOM           (-32)
01509 #define GCL_HICONSM        (-34)
01510
01511
01512 /***** Window hooks *****/
01513
01514 /* Hook values */
01515 #define WH_MIN              (-1)
01516 #define WH_MSGFILTER        (-1)
01517 #define WH_JOURNALRECORD    0
01518 #define WH_JOURNALPLAYBACK 1
01519 #define WH_KEYBOARD         2
01520 #define WH_GETMESSAGE       3
01521 #define WH_CALLWNDPROC      4
01522 #define WH_CBT              5
01523 #define WH_SYSMSGFILTER     6
01524 #define WH_MOUSE            7
01525 #define WH_HARDWARE         8
01526 #define WH_DEBUG            9
01527 #define WH_SHELL            10
01528 #define WH_FOREGROUNDIDLE   11
01529 #define WH_CALLWNDPROCRET   12
01530 #define WH_KEYBOARD_LL      13
01531 #define WH_MOUSE_LL         14
01532 #define WH_MAX              14
01533
01534 #define WH_MINHOOK          WH_MIN
01535 #define WH_MAXHOOK          WH_MAX
01536
01537 /* Hook action codes */
01538 #define HC_ACTION            0

```

```

01539 #define HC_GETNEXT          1
01540 #define HC_SKIP              2
01541 #define HC_NOREMOVE          3
01542 #define HC_NOREM             HC_NOREMOVE
01543 #define HC_SYSMODALON        4
01544 #define HC_SYSMODALOFF       5
01545
01546 /* CallMsgFilter() values */
01547 #define MSGF_DIALOGBOX        0
01548 #define MSGF_MESSAGEBOX       1
01549 #define MSGF_MENU             2
01550 #define MSGF_MOVE             3
01551 #define MSGF_SIZE             4
01552 #define MSGF_SCROLLBAR        5
01553 #define MSGF_NEXTWINDOW       6
01554 #define MSGF_MAX              8
01555 #define MSGF_USER             0x1000
01556 #define MSGF_DDEMGR           0x8001
01557
01558 typedef struct
01559 {
01560     UINT        style;
01561     WNDPROC     lpfnWndProc;
01562     INT         cbClsExtra;
01563     INT         cbWndExtra;
01564     HINSTANCE    hInstance;
01565     HICON        hIcon;
01566     HCURSOR      hCursor;
01567     HBRUSH       hbrBackground;
01568     LPCSTR       lpszMenuName;
01569     LPCSTR       lpszClassName;
01570 } WNDCLASSA, *PWNDCLASSA, *LPWNDCLASSA;
01571
01572 typedef struct
01573 {
01574     UINT        style;
01575     WNDPROC     lpfnWndProc;
01576     INT         cbClsExtra;
01577     INT         cbWndExtra;
01578     HINSTANCE    hInstance;
01579     HICON        hIcon;
01580     HCURSOR      hCursor;
01581     HBRUSH       hbrBackground;
01582     LPCWSTR      lpszMenuName;
01583     LPCWSTR      lpszClassName;
01584 } WNDCLASSW, *PWNDCLASSW, *LPWNDCLASSW;
01585
01586 DECL_WINELIB_TYPE_AW(WNDCLASS)
01587 DECL_WINELIB_TYPE_AW(PWNDCLASS)
01588 DECL_WINELIB_TYPE_AW(LPWNDCLASS)
01589
01590 typedef struct {
01591     DWORD dwData;
01592     DWORD cbData;
01593     LPVOID lpData;
01594 } COPYDATASTRUCT, *PCOPYDATASTRUCT;
01595
01596 typedef struct {
01597     HMENU hmenuIn;
01598     HMENU hmenuNext;
01599     HWND hwndNext;
01600 } MDINEXTMENU, *PMDINEXTMENU, *LPMDINEXTMENU;
01601
01602 typedef struct
01603 {
01604     DWORD    mkSize;
01605     CHAR     mkKeyList;
01606     CHAR     szKeyphrase[1];
01607 } MULTIKEYHELPA, *PMULTIKEYHELPA, *LPMULTIKEYHELPA;
01608
01609 typedef struct
01610 {
01611     DWORD    mkSize;
01612     WCHAR    mkKeyList;
01613     WCHAR    szKeyphrase[1];
01614 } MULTIKEYHELPAW, *PMULTIKEYHELPAW, *LPMULTIKEYHELPAW;
01615
01616 DECL_WINELIB_TYPE_AW(MULTIKEYHELP)
01617 DECL_WINELIB_TYPE_AW(PMULTIKEYHELP)
01618 DECL_WINELIB_TYPE_AW(LPMULTIKEYHELP)
01619
01620 typedef struct {
01621     int wStructSize;
01622     int x;
01623     int y;
01624     int dx;
01625     int dy;

```

```

01626     int wMax;
01627     CHAR rgchMember[2];
01628 } HELPPWININFOA, *PHELPPWININFOA, *LPHELPPWININFOA;
01629
01630 typedef struct {
01631     int wStructSize;
01632     int x;
01633     int y;
01634     int dx;
01635     int dy;
01636     int wMax;
01637     WCHAR rgchMember[2];
01638 } HELPPWININFOW, *PHELPPWININFOW, *LPHELPPWININFOW;
01639
01640 DECL_WINELIB_TYPE_AW(HELPPWININFO)
01641 DECL_WINELIB_TYPE_AW(PHELPPWININFO)
01642 DECL_WINELIB_TYPE_AW(LPHELPPWININFO)
01643
01644 #define HELP_CONTEXT      0x0001
01645 #define HELP_QUIT         0x0002
01646 #define HELP_INDEX       0x0003
01647 #define HELP_CONTENTS    0x0003
01648 #define HELP_HELPONHELP  0x0004
01649 #define HELP_SETINDEX    0x0005
01650 #define HELP_SETCONTENTS 0x0005
01651 #define HELP_CONTEXTPOPUP 0x0008
01652 #define HELP_FORCEFILE   0x0009
01653 #define HELP_KEY         0x0101
01654 #define HELP_COMMAND     0x0102
01655 #define HELP_PARTIALKEY  0x0105
01656 #define HELP_MULTIKY     0x0201
01657 #define HELP_SETWINPOS    0x0203
01658 #define HELP_CONTEXTMENU 0x000a
01659 #define HELP_FINDER      0x000b
01660 #define HELP_WM_HELP     0x000c
01661 #define HELP_SETPOPUP_POS 0x000d
01662
01663 #define HELP_TCARD      0x8000
01664 #define HELP_TCARD_DATA 0x0010
01665 #define HELP_TCARD_OTHER_CALLER 0x0011
01666
01667
01668     /* ChangeDisplaySettings return codes */
01669
01670 #define DISP_CHANGE_SUCCESSFUL 0
01671 #define DISP_CHANGE_RESTART   1
01672 #define DISP_CHANGE_FAILED    (-1)
01673 #define DISP_CHANGE_BADMODE   (-2)
01674 #define DISP_CHANGE_NOTUPDATED (-3)
01675 #define DISP_CHANGE_BADFLAGS  (-4)
01676 #define DISP_CHANGE_BADPARAM  (-5)
01677
01678 /* ChangeDisplaySettings.dwFlags */
01679 #define CDS_UPDATEREGISTRY 0x00000001
01680 #define CDS_TEST           0x00000002
01681 #define CDS_FULLSCREEN     0x00000004
01682 #define CDS_GLOBAL         0x00000008
01683 #define CDS_SET_PRIMARY    0x00000010
01684 #define CDS_RESET          0x40000000
01685 #define CDS_SETRECT        0x20000000
01686 #define CDS_NORESET        0x10000000
01687
01688 typedef struct
01689 {
01690     UINT        cbSize;
01691     UINT        style;
01692     WNDPROC      lpfnWndProc;
01693     INT         cbClsExtra;
01694     INT         cbWndExtra;
01695     HINSTANCE    hInstance;
01696     HICON        hIcon;
01697     HCURSOR      hCursor;
01698     HBRUSH       hbrBackground;
01699     LPCSTR       lpszMenuName;
01700     LPCSTR       lpszClassName;
01701     HICON        hIconSm;
01702 } WNDCLASSEX, *PWNDCLASSEX, *LPWNDCLASSEX;
01703
01704 typedef struct
01705 {
01706     UINT        cbSize;
01707     UINT        style;
01708     WNDPROC      lpfnWndProc;
01709     INT         cbClsExtra;
01710     INT         cbWndExtra;
01711     HINSTANCE    hInstance;
01712     HICON        hIcon;

```

```

01713     HCURSOR    hCursor;
01714     HBRUSH     hbrBackground;
01715     LPCWSTR    lpszMenuName;
01716     LPCWSTR    lpszClassName;
01717     HICON      hIconSm;
01718 } WNDCLASSEXW, *PWNDCLASSEXW, *LPWNDCLASSEXW;
01719
01720 DECL_WINELIB_TYPE_AW(WNDCLASSEX)
01721 DECL_WINELIB_TYPE_AW(PWNDCLASSEX)
01722 DECL_WINELIB_TYPE_AW(LPWNDCLASSEX)
01723
01724 typedef struct tagMSG
01725 {
01726     HWND        hwnd;
01727     UINT        message;
01728     WPARAM      wParam;
01729     LPARAM      lParam;
01730     DWORD       time;
01731     POINT       pt;
01732 } MSG, *PMSG, *LPMSG;
01733
01734 #define POINTSTOPOINT(pt, pts) \
01735     { (pt).x = (LONG) (SHORT) LOWORD(* (LONG*) &pts); \
01736       (pt).y = (LONG) (SHORT) HIWORD(* (LONG*) &pts); }
01737
01738 #define POINTTOPOINTS(pt)      (MAKELONG((short)((pt).x), (short)((pt).y)))
01739
01740
01741 /* Cursors / Icons */
01742
01743 typedef struct {
01744     BOOL        fIcon;
01745     DWORD       xHotspot;
01746     DWORD       yHotspot;
01747     HBITMAP     hbmMask;
01748     HBITMAP     hbmColor;
01749 } ICONINFO, *PICONINFO;
01750
01751
01752 /* this is the 6 byte accel struct used in Win32 when presented to the user */
01753 typedef struct
01754 {
01755     BYTE        fVirt;
01756     BYTE        pad0;
01757     WORD        key;
01758     WORD        cmd;
01759 } ACCEL, *LPACCEL;
01760
01761
01762 /* Flags for TrackPopupMenu */
01763 #define TPM_LEFTBUTTON    0x0000
01764 #define TPM_RIGHTBUTTON  0x0002
01765 #define TPM_LEFTALIGN    0x0000
01766 #define TPM_CENTERALIGN  0x0004
01767 #define TPM_RIGHTALIGN   0x0008
01768 #define TPM_TOPALIGN     0x0000
01769 #define TPM_VCENTERALIGN 0x0010
01770 #define TPM_BOTTOMALIGN  0x0020
01771 #define TPM_HORIZONTAL   0x0000
01772 #define TPM_VERTICAL     0x0040
01773 #define TPM_NONOTIFY     0x0080
01774 #define TPM_RETURNCMD    0x0100
01775
01776 typedef struct
01777 {
01778     UINT        cbSize;
01779     RECT        rcExclude;
01780 } TPMPARAMS, *LPTMPPARAMS;
01781
01782 /* FIXME: not sure this one is correct */
01783 typedef struct {
01784     UINT        cbSize;
01785     UINT        fMask;
01786     UINT        fType;
01787     UINT        fState;
01788     UINT        wID;
01789     HMENU       hSubMenu;
01790     HBITMAP     hbmpChecked;
01791     HBITMAP     hbmpUnchecked;
01792     DWORD       dwItemData;
01793     LPSTR       dwTypeData;
01794     UINT        cch;
01795     HBITMAP     hbmpItem;
01796 } MENUITEMINFOA, *LPMENUITEMINFOA;
01797
01798 typedef struct {
01799     UINT        cbSize;

```

```

01800     UINT     fMask;
01801     UINT     fType;
01802     UINT     fState;
01803     UINT     wID;
01804     HMENU     hSubMenu;
01805     HBITMAP    hbmpChecked;
01806     HBITMAP    hbmpUnchecked;
01807     DWORD     dwItemData;
01808     LPWSTR     dwTypeData;
01809     UINT     cch;
01810     HBITMAP    hbmpItem;
01811 } MENUITEMINFOW, *LPMENUITEMINFOW;
01812
01813 DECL_WINELIB_TYPE_AW(MENUITEMINFO)
01814 DECL_WINELIB_TYPE_AW(LPMENUITEMINFO)
01815 typedef const MENUITEMINFOA *LPCMENUITEMINFOA;
01816 typedef const MENUITEMINFOW *LPCMENUITEMINFOW;
01817 DECL_WINELIB_TYPE_AW(LPCMENUITEMINFO)
01818
01819 typedef struct {
01820     DWORD     cbSize;
01821     DWORD     fMask;
01822     DWORD     dwStyle;
01823     UINT     cyMax;
01824     HBRUSH     hbrBack;
01825     DWORD     dwContextHelpID;
01826     DWORD     dwMenuData;
01827 } MENUINFO, *LPMENUINFO;
01828
01829 typedef const MENUINFO *LPCMENUINFO;
01830
01831 #define MIM_MAXHEIGHT      0x00000001
01832 #define MIM_BACKGROUND    0x00000002
01833 #define MIM_HELPID        0x00000004
01834 #define MIM_MENUDATA      0x00000008
01835 #define MIM_STYLE          0x00000010
01836 #define MIM_APPLYTOSUBMENUS 0x80000000
01837
01838 typedef struct {
01839     WORD versionNumber;
01840     WORD offset;
01841 } MENUITEMTEMPLATEHEADER, *PMENUITEMTEMPLATEHEADER;
01842
01843
01844 typedef struct {
01845     WORD mtOption;
01846     WORD mtID;
01847     WCHAR mtString[1];
01848 } MENUITEMTEMPLATE, *PMENUITEMTEMPLATE;
01849
01850
01851 typedef VOID     MENUTEMPLATE;
01852 typedef PVOID    *LPMENUTEMPLATE;
01853
01854 /* Field specifiers for MENUITEMINFO[AW] type. */
01855 #define MIIM_STATE      0x00000001
01856 #define MIIM_ID         0x00000002
01857 #define MIIM_SUBMENU    0x00000004
01858 #define MIIM_CHECKMARKS 0x00000008
01859 #define MIIM_TYPE       0x00000010
01860 #define MIIM_DATA       0x00000020
01861 #define MIIM_STRING     0x00000040
01862 #define MIIM_BITMAP     0x00000080
01863 #define MIIM_FTYPE      0x00000100
01864
01865 #define HBMMENU_CALLBACK ((HBITMAP) -1)
01866 #define HBMMENU_SYSTEM   ((HBITMAP) 1)
01867 #define HBMMENU_MBAR_RESTORE ((HBITMAP) 2)
01868 #define HBMMENU_MBAR_MINIMIZE ((HBITMAP) 3)
01869 #define HBMMENU_MBAR_CLOSE ((HBITMAP) 5)
01870 #define HBMMENU_MBAR_CLOSE_D ((HBITMAP) 6)
01871 #define HBMMENU_MBAR_MINIMIZE_D ((HBITMAP) 7)
01872 #define HBMMENU_POPUP_CLOSE ((HBITMAP) 8)
01873 #define HBMMENU_POPUP_RESTORE ((HBITMAP) 9)
01874 #define HBMMENU_POPUP_MAXIMIZE ((HBITMAP) 10)
01875 #define HBMMENU_POPUP_MINIMIZE ((HBITMAP) 11)
01876
01877 /* WM_H/VSCROLL commands */
01878 #define SB_LINEUP        0
01879 #define SB_LINELEFT      0
01880 #define SB_LINEDOWN      1
01881 #define SB_LINERIGHT     1
01882 #define SB_PAGEUP        2
01883 #define SB_PAGELLEFT     2
01884 #define SB_PAGEDOWN      3
01885 #define SB_PAGERIGHT     3
01886 #define SB_THUMBPOSITION  4

```

```
01887 #define SB_THUMBTRACK      5
01888 #define SB_TOP                6
01889 #define SB_LEFT               6
01890 #define SB_BOTTOM             7
01891 #define SB_RIGHT              7
01892 #define SB_ENDSCROLL          8
01893
01894 /* Scroll bar selection constants */
01895 #define SB_HORZ                0
01896 #define SB_VERT               1
01897 #define SB_CTL                2
01898 #define SB_BOTH               3
01899
01900 /* Scrollbar styles */
01901 #define SBS_HORZ               0x0000L
01902 #define SBS_VERT              0x0001L
01903 #define SBS_TOPALIGN          0x0002L
01904 #define SBS_LEFTALIGN         0x0002L
01905 #define SBS_BOTTOMALIGN       0x0004L
01906 #define SBS_RIGHTALIGN        0x0004L
01907 #define SBS_SIZEBOXTOPLEFTALIGN 0x0002L
01908 #define SBS_SIZEBOXBOTTOMRIGHTALIGN 0x0004L
01909 #define SBS_SIZEBOX           0x0008L
01910 #define SBS_SIZEGRIP          0x0010L
01911
01912 /* EnableScrollBar() flags */
01913 #define ESB_ENABLE_BOTH       0x0000
01914 #define ESB_DISABLE_BOTH      0x0003
01915
01916 #define ESB_DISABLE_LEFT      0x0001
01917 #define ESB_DISABLE_RIGHT     0x0002
01918
01919 #define ESB_DISABLE_UP        0x0001
01920 #define ESB_DISABLE_DOWN      0x0002
01921
01922 #define ESB_DISABLE_LTUP      ESB_DISABLE_LEFT
01923 #define ESB_DISABLE_RTDN      ESB_DISABLE_RIGHT
01924
01925 /* Win32 button control messages */
01926 #define BM_GETCHECK           0x00f0
01927 #define BM_SETCHECK           0x00f1
01928 #define BM_GETSTATE           0x00f2
01929 #define BM_SETSTATE           0x00f3
01930 #define BM_SETSTYLE           0x00f4
01931 #define BM_CLICK              0x00f5
01932 #define BM_GETIMAGE           0x00f6
01933 #define BM_SETIMAGE           0x00f7
01934 /* Winelib button control messages */
01935
01936 /* Button notification codes */
01937 #define BN_CLICKED            0
01938 #define BN_PAINT              1
01939 #define BN_HILITE             2
01940 #define BN_UNHILITE           3
01941 #define BN_DISABLE            4
01942 #define BN_DOUBLECLICKED      5
01943 #define BN_DBLCLK             BN_DOUBLECLICKED
01944
01945 /* Button states */
01946 #define BST_UNCHECKED          0x0000
01947 #define BST_CHECKED            0x0001
01948 #define BST_INDETERMINATE      0x0002
01949 #define BST_PUSHD              0x0004
01950 #define BST_FOCUS              0x0008
01951
01952 /* Static Control Styles */
01953 #define SS_LEFT                0x00000000L
01954 #define SS_CENTER              0x00000001L
01955 #define SS_RIGHT               0x00000002L
01956 #define SS_ICON                0x00000003L
01957 #define SS_BLACKRECT           0x00000004L
01958 #define SS_GRAYRECT            0x00000005L
01959 #define SS_WHITERECT           0x00000006L
01960 #define SS_BLACKFRAME          0x00000007L
01961 #define SS_GRAYFRAME           0x00000008L
01962 #define SS_WHITEFRAME          0x00000009L
01963
01964 #define SS_SIMPLE              0x0000000BL
01965 #define SS_LEFTNOWORDWRAP      0x0000000CL
01966
01967 #define SS_OWNERDRAW           0x0000000DL
01968 #define SS_BITMAP              0x0000000EL
01969 #define SS_ENHMETAFILE         0x0000000FL
01970
01971 #define SS_ETCHEDHORZ          0x00000010L
01972 #define SS_ETCHEDVERT          0x00000011L
01973 #define SS_ETCHEDFRAME         0x00000012L
```

```
01974 #define SS_TYPMASK            0x0000001FL
01975
01976 #define SS_NOPREFIX              0x000000080L
01977 #define SS_NOTIFY                0x000000100L
01978 #define SS_CENTERIMAGE          0x000000200L
01979 #define SS_RIGHTJUST            0x000000400L
01980 #define SS_REALSIZEIMAGE        0x000000800L
01981 #define SS_SUNKEN               0x000001000L
01982
01983 /* Static Control Messages */
01984 #define STM_SETICON              0x0170
01985 #define STM_GETICON              0x0171
01986 #define STM_SETIMAGE             0x0172
01987 #define STM_GETIMAGE             0x0173
01988 #define STM_MSGMAX              0x0174
01989
01990 #define STN_CLICKED              0
01991 #define STN_DBLCLK              1
01992 #define STN_ENABLE              2
01993 #define STN_DISABLE             3
01994
01995 /* Scrollbar messages */
01996 #define SBM_SETPOS               0x00e0
01997 #define SBM_GETPOS               0x00e1
01998 #define SBM_SETRANGE            0x00e2
01999 #define SBM_GETRANGE            0x00e3
02000 #define SBM_ENABLE_ARROWS       0x00e4
02001 #define SBM_SETRANGEREDRAW      0x00e6
02002 #define SBM_SETSCROLLINFO       0x00e9
02003 #define SBM_GETSCROLLINFO       0x00ea
02004
02005 /* Scrollbar info */
02006 typedef struct
02007 {
02008     UINT    cbSize;
02009     UINT    fMask;
02010     INT     nMin;
02011     INT     nMax;
02012     UINT    nPage;
02013     INT     nPos;
02014     INT     nTrackPos;
02015 } SCROLLINFO, *LPSCROLLINFO;
02016
02017 typedef const SCROLLINFO *LPCSCROLLINFO;
02018
02019 /* GetScrollInfo() flags */
02020 #define SIF_RANGE                0x0001
02021 #define SIF_PAGE                 0x0002
02022 #define SIF_POS                  0x0004
02023 #define SIF_DISABLENOSCROLL     0x0008
02024 #define SIF_TRACKPOS             0x0010
02025 #define SIF_ALL                  (SIF_RANGE | SIF_PAGE | SIF_POS | SIF_TRACKPOS)
02026
02027 /* Listbox styles */
02028 #define LBS_NOTIFY                0x0001
02029 #define LBS_SORT                  0x0002
02030 #define LBS_NOREDRAW             0x0004
02031 #define LBS_MULTIPLESEL          0x0008
02032 #define LBS_OWNERDRAWFIXED       0x0010
02033 #define LBS_OWNERDRAWVARIABLE    0x0020
02034 #define LBS_HASSTRINGS           0x0040
02035 #define LBS_USETABSTOPS           0x0080
02036 #define LBS_NOINTEGRALHEIGHT     0x0100
02037 #define LBS_MULTICOLUMN          0x0200
02038 #define LBS_WANTKEYBOARDINPUT    0x0400
02039 #define LBS_EXTENDEDSEL          0x0800
02040 #define LBS_DISABLENOSCROLL     0x1000
02041 #define LBS_NODATA               0x2000
02042 #define LBS_NOSEL                0x4000
02043 #define LBS_STANDARD             (LBS_NOTIFY | LBS_SORT | WS_VSCROLL | WS_BORDER)
02044
02045 /* Listbox messages */
02046 #define LB_ADDSTRING              0x0180
02047 #define LB_INSERTSTRING           0x0181
02048 #define LB_DELETESTRING          0x0182
02049 #define LB_SELITEMRANGEEX        0x0183
02050 #define LB_RESETCONTENT          0x0184
02051 #define LB_SETSEL                0x0185
02052 #define LB_SETCURSEL             0x0186
02053 #define LB_GETSEL                0x0187
02054 #define LB_GETCURSEL             0x0188
02055 #define LB_GETTEXT               0x0189
02056 #define LB_GETTEXTLEN            0x018a
02057 #define LB_GETCOUNT             0x018b
02058 #define LB_SELECTSTRING          0x018c
02059 #define LB_DIR                   0x018d
02060 #define LB_GETTOPINDEX           0x018e
```

```

02061 #define LB_FINDSTRING          0x018f
02062 #define LB_GETSELCOUNT          0x0190
02063 #define LB_GETSELITEMS           0x0191
02064 #define LB_SETTABSTOPS           0x0192
02065 #define LB_GETHORIZONTALEXTENT   0x0193
02066 #define LB_SETHORIZONTALEXTENT   0x0194
02067 #define LB_SETCOLUMNWIDTH        0x0195
02068 #define LB_ADDFILE                0x0196
02069 #define LB_SETTOPINDEX            0x0197
02070 #define LB_GETITEMRECT            0x0198
02071 #define LB_GETITEMDATA            0x0199
02072 #define LB_SETITEMDATA            0x019a
02073 #define LB_SELITEMRANGE           0x019b
02074 #define LB_SETANCHORINDEX         0x019c
02075 #define LB_GETANCHORINDEX         0x019d
02076 #define LB_SETCARETINDEX          0x019e
02077 #define LB_GETCARETINDEX          0x019f
02078 #define LB_SETITEMHEIGHT          0x01a0
02079 #define LB_GETITEMHEIGHT          0x01a1
02080 #define LB_FINDSTRINGEXACT        0x01a2
02081 #define LB_CARETON                0x01a3
02082 #define LB_CARETOFF              0x01a4
02083 #define LB_SETLOCALE              0x01a5
02084 #define LB_GETLOCALE              0x01a6
02085 #define LB_SETCOUNT               0x01a7
02086 #define LB_INITSTORAGE            0x01a8
02087 #define LB_ITEMFROMPOINT          0x01a9
02088
02089 /* Listbox notification codes */
02090 #define LBN_ERRSPACE              (-2)
02091 #define LBN_SELCHANGE             1
02092 #define LBN_DBLCLK                2
02093 #define LBN_SELCANCEL             3
02094 #define LBN_SETFOCUS              4
02095 #define LBN_KILLFOCUS             5
02096
02097 /* Listbox message return values */
02098 #define LB_OKAY                   0
02099 #define LB_ERR                    (-1)
02100 #define LB_ERRSPACE               (-2)
02101
02102 #define LB_CTLCODE                0L
02103
02104 /* Combo box styles */
02105 #define CBS_SIMPLE                 0x0001L
02106 #define CBS_DROPDOWN              0x0002L
02107 #define CBS_DROPDOWNLIST          0x0003L
02108 #define CBS_OWNERDRAWFIXED        0x0010L
02109 #define CBS_OWNERDRAWVARIABLE     0x0020L
02110 #define CBS_AUTOHSCROLL           0x0040L
02111 #define CBS_OEMCONVERT            0x0080L
02112 #define CBS_SORT                  0x0100L
02113 #define CBS_HASSTRINGS            0x0200L
02114 #define CBS_NOINTEGRALHEIGHT      0x0400L
02115 #define CBS_DISABLENOSCROLL       0x0800L
02116
02117 #define CBS_UPPERCASE             0x2000L
02118 #define CBS_LOWERCASE             0x4000L
02119
02120
02121 /* Combo box messages */
02122 #define CB_GETEDITSEL             0x0140
02123 #define CB_LIMITTEXT              0x0141
02124 #define CB_SETEXTEDITSEL          0x0142
02125 #define CB_ADDSTRING              0x0143
02126 #define CB_DELETESTRING           0x0144
02127 #define CB_DIR                    0x0145
02128 #define CB_GETCOUNT             0x0146
02129 #define CB_GETCURSEL              0x0147
02130 #define CB_GETLBTEXT              0x0148
02131 #define CB_GETLBTEXTLEN           0x0149
02132 #define CB_INSERTSTRING           0x014a
02133 #define CB_RESETCONTENT           0x014b
02134 #define CB_FINDSTRING             0x014c
02135 #define CB_SELECTSTRING           0x014d
02136 #define CB_SETCURSEL              0x014e
02137 #define CB_SHOWDROPDOWN           0x014f
02138 #define CB_GETITEMDATA            0x0150
02139 #define CB_SETITEMDATA            0x0151
02140 #define CB_GETDROPPEDCONTROLRECT  0x0152
02141 #define CB_SETITEMHEIGHT          0x0153
02142 #define CB_GETITEMHEIGHT          0x0154
02143 #define CB_SETEXTENDEDUI          0x0155
02144 #define CB_GETEXTENDEDUI          0x0156
02145 #define CB_GETDROPPEDSTATE        0x0157
02146 #define CB_FINDSTRINGEXACT        0x0158
02147 #define CB_SETLOCALE              0x0159

```

```

02148 #define CB_GETLOCALE          0x015a
02149 #define CB_GETTOPINDEX         0x015b
02150 #define CB_SETTOPINDEX         0x015c
02151 #define CB_GETHORIZONTALEXTENT 0x015d
02152 #define CB_SETHORIZONTALEXTENT 0x015e
02153 #define CB_GETDROPPEDWIDTH     0x015f
02154 #define CB_SETDROPPEDWIDTH     0x0160
02155 #define CB_INITSTORAGE         0x0161
02156
02157 /* Combo box notification codes */
02158 #define CBN_ERRSPACE           (-1)
02159 #define CBN_SELCHANGE          1
02160 #define CBN_DBLCLK             2
02161 #define CBN_SETFOCUS           3
02162 #define CBN_KILLFOCUS          4
02163 #define CBN_EDITCHANGE         5
02164 #define CBN_EDITUPDATE         6
02165 #define CBN_DROPDOWN           7
02166 #define CBN_CLOSEUP            8
02167 #define CBN_SELENDOK           9
02168 #define CBN_SELENDCANCEL       10
02169
02170 /* Combo box message return values */
02171 #define CB_OKAY                 0
02172 #define CB_ERR                  (-1)
02173 #define CB_ERRSPACE             (-2)
02174
02175 #define MB_OK                   0x00000000
02176 #define MB_OKCANCEL             0x00000001
02177 #define MB_ABORTRETRYIGNORE     0x00000002
02178 #define MB_YESNOCANCEL          0x00000003
02179 #define MB_YESNO                0x00000004
02180 #define MB_RETRYCANCEL          0x00000005
02181 #define MB_TYPEMASK             0x0000000F
02182
02183 #define MB_ICONHAND             0x00000010
02184 #define MB_ICONQUESTION         0x00000020
02185 #define MB_ICONEXCLAMATION     0x00000030
02186 #define MB_ICONASTERISK         0x00000040
02187 #define MB_USERICON             0x00000080
02188 #define MB_ICONMASK            0x000000F0
02189
02190 #define MB_ICONINFORMATION      MB_ICONASTERISK
02191 #define MB_ICONSTOP             MB_ICONHAND
02192 #define MB_ICONWARNING          MB_ICONEXCLAMATION
02193 #define MB_ICONERROR            MB_ICONHAND
02194
02195 #define MB_DEFBUTTON1           0x00000000
02196 #define MB_DEFBUTTON2           0x00000100
02197 #define MB_DEFBUTTON3           0x00000200
02198 #define MB_DEFBUTTON4           0x00000300
02199 #define MB_DEFMASK              0x00000F00
02200
02201 #define MB_APPLMODAL            0x00000000
02202 #define MB_SYSTEMMODAL          0x00001000
02203 #define MB_TASKMODAL            0x00002000
02204 #define MB_MODEMASK             0x00003000
02205
02206 #define MB_HELP                 0x00004000
02207 #define MB_NOFOCUS              0x00008000
02208 #define MB_MISCMASK             0x0000C000
02209
02210 #define MB_SETFOREGROUND        0x00010000
02211 #define MB_DEFAULT_DESKTOP_ONLY 0x00020000
02212 #define MB_SERVICE_NOTIFICATION 0x00040000
02213 #define MB_TOPMOST              0x00040000
02214 #define MB_RIGHT                0x00080000
02215 #define MB_RTLREADING           0x00100000
02216
02217 #define HELPINFO_WINDOW         0x0001
02218 #define HELPINFO_MENUITEM      0x0002
02219
02220 /* Structure pointed to by lParam of WM_HELP */
02221 typedef struct
02222 {
02223     UINT    cbSize;          /* Size in bytes of this struct */
02224     INT     iContextType;    /* Either HELPINFO_WINDOW or HELPINFO_MENUITEM */
02225     INT     iCtrlId;         /* Control Id or a Menu item Id. */
02226     HANDLE  hItemHandle;     /* hWnd of control or hMenu. */
02227     DWORD   dwContextId;     /* Context Id associated with this item */
02228     POINT   MousePos;        /* Mouse Position in screen co-ordinates */
02229 } HELPINFO, *LPHELPINFO;
02230
02231 typedef void CALLBACK (*MSGBOXCALLBACK) (LPHELPINFO lpHelpInfo);
02232
02233 typedef struct
02234 {

```

```

02235     UINT      cbSize;
02236     HWND      hwndOwner;
02237     HINSTANCE  hInstance;
02238     LPCSTR     lpstrText;
02239     LPCSTR     lpstrCaption;
02240     DWORD      dwStyle;
02241     LPCSTR     lpstrIcon;
02242     DWORD      dwContextHelpId;
02243     MSGBOXCALLBACK lpfnMsgBoxCallback;
02244     DWORD      dwLanguageId;
02245 } MSGBOXPARAMSA, *PMSGBOXPARAMSA, *LPMSGBOXPARAMSA;
02246
02247 typedef struct
02248 {
02249     UINT      cbSize;
02250     HWND      hwndOwner;
02251     HINSTANCE  hInstance;
02252     LPCWSTR     lpstrText;
02253     LPCWSTR     lpstrCaption;
02254     DWORD      dwStyle;
02255     LPCWSTR     lpstrIcon;
02256     DWORD      dwContextHelpId;
02257     MSGBOXCALLBACK lpfnMsgBoxCallback;
02258     DWORD      dwLanguageId;
02259 } MSGBOXPARAMSW, *PMSGBOXPARAMSW, *LPMSGBOXPARAMSW;
02260
02261 DECL_WINELIB_TYPE_AW(MSGBOXPARAMS)
02262 DECL_WINELIB_TYPE_AW(PMSGBOXPARAMS)
02263 DECL_WINELIB_TYPE_AW(LPMSGBOXPARAMS)
02264
02265 #define MONITOR_DEFAULTTONULL      0x00000000
02266 #define MONITOR_DEFAULTTOPRIMARY  0x00000001
02267 #define MONITOR_DEFAULTTONEAREST  0x00000002
02268
02269 #define MONITORINFOF_PRIMARY      0x00000001
02270
02271 #ifndef CCHDEVICENAME
02272 #define CCHDEVICENAME 32
02273 #endif
02274
02275 typedef struct tagMONITORINFO
02276 {
02277     DWORD      cbSize;
02278     RECT        rcMonitor;
02279     RECT        rcWork;
02280     DWORD      dwFlags;
02281 } MONITORINFO, *LPMONITORINFO;
02282
02283 typedef struct tagMONITORINFOEXA
02284 {
02285     MONITORINFO dummy;
02286     CHAR        szDevice[CCHDEVICENAME];
02287 } MONITORINFOEXA, *LPMONITORINFOEXA;
02288
02289 typedef struct tagMONITORINFOEXW
02290 {
02291     MONITORINFO dummy;
02292     WCHAR        szDevice[CCHDEVICENAME];
02293 } MONITORINFOEXW, *LPMONITORINFOEXW;
02294
02295 DECL_WINELIB_TYPE_AW(MONITORINFOEX)
02296 DECL_WINELIB_TYPE_AW(LPMONITORINFOEX)
02297
02298 typedef BOOL CALLBACK (*MONITORENUMPROC) (HMONITOR, HDC, LPRECT, LPARAM);
02299
02300 #include "pshpack2.h"
02301
02302 /* FIXME: use this instead of LPCVOID for CreateDialogIndirectParam
02303    and DialogBoxIndirectParam */
02304 typedef struct tagDLGTEMPLATE
02305 {
02306     DWORD style;
02307     DWORD dwExtendedStyle;
02308     WORD  cdit;
02309     short x;
02310     short y;
02311     short cx;
02312     short cy;
02313 } DLGTEMPLATE;
02314
02315 typedef DLGTEMPLATE *LPDLGTEMPLATEA;
02316 typedef DLGTEMPLATE *LPDLGTEMPLATEW;
02317 DECL_WINELIB_TYPE_AW(LPDLGTEMPLATE)
02318 typedef const DLGTEMPLATE *LPCDLGTEMPLATEA;
02319 typedef const DLGTEMPLATE *LPCDLGTEMPLATEW;
02320 DECL_WINELIB_TYPE_AW(LPCDLGTEMPLATE)
02321

```

```

02322 typedef struct tagDLGITEMTEMPLATE
02323 {
02324     DWORD style;
02325     DWORD dwExtendedStyle;
02326     short x;
02327     short y;
02328     short cx;
02329     short cy;
02330     WORD id;
02331 } DLGITEMTEMPLATE;
02332
02333 typedef DLGITEMTEMPLATE *PDLGITEMTEMPLATEA;
02334 typedef DLGITEMTEMPLATE *PDLGITEMTEMPLATEW;
02335 DECL_WINELIB_TYPE_AW(PDLGITEMTEMPLATE)
02336 typedef DLGITEMTEMPLATE *LPDLGITEMTEMPLATEA;
02337 typedef DLGITEMTEMPLATE *LPDLGITEMTEMPLATEW;
02338 DECL_WINELIB_TYPE_AW(LPDLGITEMTEMPLATE)
02339
02340 #include "poppack.h"
02341
02342 /* CBT hook values */
02343 #define HCBT_MOVESIZE 0
02344 #define HCBT_MINMAX 1
02345 #define HCBT_QS 2
02346 #define HCBT_CREATEWND 3
02347 #define HCBT_DESTROYWND 4
02348 #define HCBT_ACTIVATE 5
02349 #define HCBT_CLICKSKIPPED 6
02350 #define HCBT_KEYSKIPPED 7
02351 #define HCBT_SYSCOMMAND 8
02352 #define HCBT_SETFOCUS 9
02353
02354 /* CBT hook structures */
02355
02356 typedef struct
02357 {
02358     CREATESTRUCTA *lpcs;
02359     HWND hwndInsertAfter;
02360 } CBT_CREATEWND, *PCBT_CREATEWND;
02361
02362 typedef struct
02363 {
02364     CREATESTRUCTW *lpcs;
02365     HWND hwndInsertAfter;
02366 } CBT_CREATEWNDW, *PCBT_CREATEWNDW;
02367
02368 DECL_WINELIB_TYPE_AW(CBT_CREATEWND)
02369 DECL_WINELIB_TYPE_AW(PCBT_CREATEWND)
02370
02371 typedef struct
02372 {
02373     BOOL fMouse;
02374     HWND hwndActive;
02375 } CBTACTIVATESTRUCT, *PCBTACTIVATESTRUCT;
02376
02377
02378 /* modifiers for RegisterHotKey */
02379 #define MOD_ALT 0x0001
02380 #define MOD_CONTROL 0x0002
02381 #define MOD_SHIFT 0x0004
02382 #define MOD_WIN 0x0008
02383
02384 /* ids for RegisterHotKey */
02385 #define IDHOT_SNAPWINDOW (-1) /* SHIFT-PRINTSCRN */
02386 #define IDHOT_SNAPDESKTOP (-2) /* PRINTSCRN */
02387
02388 /* keyboard event flags */
02389 #define KEYEVENTF_EXTENDEDKEY 0x0001
02390 #define KEYEVENTF_KEYUP 0x0002
02391
02392 /* mouse event flags */
02393 #define MOUSEEVENTF_MOVE 0x0001
02394 #define MOUSEEVENTF_LEFTDOWN 0x0002
02395 #define MOUSEEVENTF_LEFTUP 0x0004
02396 #define MOUSEEVENTF_RIGHTDOWN 0x0008
02397 #define MOUSEEVENTF_RIGHTUP 0x0010
02398 #define MOUSEEVENTF_MIDDLEDOWN 0x0020
02399 #define MOUSEEVENTF_MIDDLEUP 0x0040
02400 #define MOUSEEVENTF_WHEEL 0x0800
02401 #define MOUSEEVENTF_ABSOLUTE 0x8000
02402
02403 /* ExitWindows() flags */
02404 #define EW_RESTARTWINDOWS 0x0042
02405 #define EW_REBOOTSYSTEM 0x0043
02406 #define EW_EXITANDEXECAPP 0x0044
02407
02408 /* ExitWindowsEx() flags */

```

```
02409 #define EWX_LOGOFF          0
02410 #define EWX_SHUTDOWN         1
02411 #define EWX_REBOOT           2
02412 #define EWX_FORCE             4
02413 #define EWX_POWEROFF          8
02414
02415 /* SetLastErrorEx types */
02416 #define SLE_ERROR              0x00000001
02417 #define SLE_MINORERROR        0x00000002
02418 #define SLE_WARNING           0x00000003
02419
02420 /* Predefined resources */
02421 #define IDI_APPLICATIONA MAKEINTRESOURCEA(32512)
02422 #define IDI_APPLICATIONW MAKEINTRESOURCEW(32512)
02423 #define IDI_APPLICATION    WINELIB_NAME_AW(IDI_APPLICATION)
02424 #define IDI_HANDA          MAKEINTRESOURCEA(32513)
02425 #define IDI_HANDW          MAKEINTRESOURCEW(32513)
02426 #define IDI_HAND           WINELIB_NAME_AW(IDI_HAND)
02427 #define IDI_QUESTIONA      MAKEINTRESOURCEA(32514)
02428 #define IDI_QUESTIONW      MAKEINTRESOURCEW(32514)
02429 #define IDI_QUESTION       WINELIB_NAME_AW(IDI_QUESTION)
02430 #define IDI_EXCLAMATIONA   MAKEINTRESOURCEA(32515)
02431 #define IDI_EXCLAMATIONW   MAKEINTRESOURCEW(32515)
02432 #define IDI_EXCLAMATION    WINELIB_NAME_AW(IDI_EXCLAMATION)
02433 #define IDI_ASTERISKA      MAKEINTRESOURCEA(32516)
02434 #define IDI_ASTERISKW      MAKEINTRESOURCEW(32516)
02435 #define IDI_ASTERISK       WINELIB_NAME_AW(IDI_ASTERISK)
02436 #define IDI_WINLOGOA       MAKEINTRESOURCEA(32517)
02437 #define IDI_WINLOGOW       MAKEINTRESOURCEW(32517)
02438 #define IDI_WINLOGO        WINELIB_NAME_AW(IDI_WINLOGO)
02439
02440 #define IDI_WARNING        IDI_EXCLAMATION
02441 #define IDI_ERROR          IDI_HAND
02442 #define IDI_INFORMATION    IDI_ASTERISK
02443
02444 #define IDC_ARROWA         MAKEINTRESOURCEA(32512)
02445 #define IDC_ARROWW         MAKEINTRESOURCEW(32512)
02446 #define IDC_ARROW          WINELIB_NAME_AW(IDC_ARROW)
02447 #define IDC_IBEAMA         MAKEINTRESOURCEA(32513)
02448 #define IDC_IBEAMW         MAKEINTRESOURCEW(32513)
02449 #define IDC_IBEAM          WINELIB_NAME_AW(IDC_IBEAM)
02450 #define IDC_WAITA          MAKEINTRESOURCEA(32514)
02451 #define IDC_WAITW          MAKEINTRESOURCEW(32514)
02452 #define IDC_WAIT           WINELIB_NAME_AW(IDC_WAIT)
02453 #define IDC_CROSSA         MAKEINTRESOURCEA(32515)
02454 #define IDC_CROSSW         MAKEINTRESOURCEW(32515)
02455 #define IDC_CROSS          WINELIB_NAME_AW(IDC_CROSS)
02456 #define IDC_UPARROWA       MAKEINTRESOURCEA(32516)
02457 #define IDC_UPARROWW       MAKEINTRESOURCEW(32516)
02458 #define IDC_UPARROW        WINELIB_NAME_AW(IDC_UPARROW)
02459 #define IDC_SIZEA          MAKEINTRESOURCEA(32640)
02460 #define IDC_SIZEW          MAKEINTRESOURCEW(32640)
02461 #define IDC_SIZE           WINELIB_NAME_AW(IDC_SIZE)
02462 #define IDC_ICONA          MAKEINTRESOURCEA(32641)
02463 #define IDC_ICONW          MAKEINTRESOURCEW(32641)
02464 #define IDC_ICON           WINELIB_NAME_AW(IDC_ICON)
02465 #define IDC_SIZENWSEA      MAKEINTRESOURCEA(32642)
02466 #define IDC_SIZENWSEW      MAKEINTRESOURCEW(32642)
02467 #define IDC_SIZENWSE       WINELIB_NAME_AW(IDC_SIZENWSE)
02468 #define IDC_SIZENESWA      MAKEINTRESOURCEA(32643)
02469 #define IDC_SIZENESWW      MAKEINTRESOURCEW(32643)
02470 #define IDC_SIZENESW       WINELIB_NAME_AW(IDC_SIZENESW)
02471 #define IDC_SIZEWEA        MAKEINTRESOURCEA(32644)
02472 #define IDC_SIZEWEW        MAKEINTRESOURCEW(32644)
02473 #define IDC_SIZEWE         WINELIB_NAME_AW(IDC_SIZEWE)
02474 #define IDC_SIZENSA        MAKEINTRESOURCEA(32645)
02475 #define IDC_SIZENSW        MAKEINTRESOURCEW(32645)
02476 #define IDC_SIZENS         WINELIB_NAME_AW(IDC_SIZENS)
02477 #define IDC_SIZEALLA       MAKEINTRESOURCEA(32646)
02478 #define IDC_SIZEALLW       MAKEINTRESOURCEW(32646)
02479 #define IDC_SIZEALL        WINELIB_NAME_AW(IDC_SIZEALL)
02480 #define IDC_NOA            MAKEINTRESOURCEA(32648)
02481 #define IDC_NOW            MAKEINTRESOURCEW(32648)
02482 #define IDC_NO             WINELIB_NAME_AW(IDC_NO)
02483 #define IDC_HANDA          MAKEINTRESOURCEA(32649)
02484 #define IDC_HANDW          MAKEINTRESOURCEW(32649)
02485 #define IDC_HAND           WINELIB_NAME_AW(IDC_HAND)
02486 #define IDC_APPSTARTINGA   MAKEINTRESOURCEA(32650)
02487 #define IDC_APPSTARTINGW   MAKEINTRESOURCEW(32650)
02488 #define IDC_APPSTARTING    WINELIB_NAME_AW(IDC_APPSTARTING)
02489 #define IDC_HELPA          MAKEINTRESOURCEA(32651)
02490 #define IDC_HELPW          MAKEINTRESOURCEW(32651)
02491 #define IDC_HELP           WINELIB_NAME_AW(IDC_HELP)
02492
02493 #define MNC_IGNORE 0
02494 #define MNC_CLOSE 1
02495 #define MNC_EXECUTE 2
```

```
02496 #define MNC_SELECT 3
02497
02498 /* SystemParametersInfo */
02499 /* defines below are for all win versions */
02500 #define SPI_GETBEEP 1
02501 #define SPI_SETBEEP 2
02502 #define SPI_GETMOUSE 3
02503 #define SPI_SETMOUSE 4
02504 #define SPI_GETBORDER 5
02505 #define SPI_SETBORDER 6
02506 #define SPI_GETKEYBOARDSPEED 10
02507 #define SPI_SETKEYBOARDSPEED 11
02508 #define SPI_LANGDRIVER 12
02509 #define SPI_ICONHORIZONTALSPACING 13
02510 #define SPI_GETSCREENSAVETIMEOUT 14
02511 #define SPI_SETSCREENSAVETIMEOUT 15
02512 #define SPI_GETSCREENSAVEACTIVE 16
02513 #define SPI_SETSCREENSAVEACTIVE 17
02514 #define SPI_GETGRIDGRANULARITY 18
02515 #define SPI_SETGRIDGRANULARITY 19
02516 #define SPI_SETDESKWALLPAPER 20
02517 #define SPI_SETDESKPATTERN 21
02518 #define SPI_GETKEYBOARDDELAY 22
02519 #define SPI_SETKEYBOARDDELAY 23
02520 #define SPI_ICONVERTICALSPACING 24
02521 #define SPI_GETICONTITLEWRAP 25
02522 #define SPI_SETICONTITLEWRAP 26
02523 #define SPI_GETMENUDROPALIGNMENT 27
02524 #define SPI_SETMENUDROPALIGNMENT 28
02525 #define SPI_SETDOUBLECLKWIDTH 29
02526 #define SPI_SETDOUBLECLKHEIGHT 30
02527 #define SPI_GETICONTITLELOGFONT 31
02528 #define SPI_SETDOUBLECLICKTIME 32
02529 #define SPI_SETMOUSEBUTTONSWAP 33
02530 #define SPI_SETICONTITLELOGFONT 34
02531 #define SPI_GETFASTTASKSWITCH 35
02532 #define SPI_SETFASTTASKSWITCH 36
02533 #define SPI_SETDRAGFULLWINDOWS 37
02534 #define SPI_GETDRAGFULLWINDOWS 38
02535
02536 #define SPI_GETFILTERKEYS 50
02537 #define SPI_SETFILTERKEYS 51
02538 #define SPI_GETTOGGLEKEYS 52
02539 #define SPI_SETTOGGLEKEYS 53
02540 #define SPI_GETMOUSEKEYS 54
02541 #define SPI_SETMOUSEKEYS 55
02542 #define SPI_GETSHOWSOUNDS 56
02543 #define SPI_SETSHOWSOUNDS 57
02544 #define SPI_GETSTICKYKEYS 58
02545 #define SPI_SETSTICKYKEYS 59
02546 #define SPI_GETACCESSTIMEOUT 60
02547 #define SPI_SETACCESSTIMEOUT 61
02548
02549 #define SPI_GETSOUNDSENTRY 64
02550 #define SPI_SETSOUNDSENTRY 65
02551
02552 /* defines below are for all win versions WINVER >= 0x0400 */
02553 #define SPI_SETDRAGFULLWINDOWS 37
02554 #define SPI_GETDRAGFULLWINDOWS 38
02555 #define SPI_GETNONCLIENTMETRICS 41
02556 #define SPI_SETNONCLIENTMETRICS 42
02557 #define SPI_GETMINIMIZEDMETRICS 43
02558 #define SPI_SETMINIMIZEDMETRICS 44
02559 #define SPI_GETICONMETRICS 45
02560 #define SPI_SETICONMETRICS 46
02561 #define SPI_SETWORKAREA 47
02562 #define SPI_GETWORKAREA 48
02563 #define SPI_SETPENWINDOWS 49
02564
02565 #define SPI_GETSERIALKEYS 62
02566 #define SPI_SETSERIALKEYS 63
02567 #define SPI_GETHIGHCONTRAST 66
02568 #define SPI_SETHIGHCONTRAST 67
02569 #define SPI_GETKEYBOARDPREF 68
02570 #define SPI_SETKEYBOARDPREF 69
02571 #define SPI_GETSCREENREADER 70
02572 #define SPI_SETSCREENREADER 71
02573 #define SPI_GETANIMATION 72
02574 #define SPI_SETANIMATION 73
02575 #define SPI_GETFONTSMOOTHING 74
02576 #define SPI_SETFONTSMOOTHING 75
02577 #define SPI_SETDRAGWIDTH 76
02578 #define SPI_SETDRAGHEIGHT 77
02579 #define SPI_SETHANDHELD 78
02580 #define SPI_GETLOWPOWERTIMEOUT 79
02581 #define SPI_GETPOWEROFFTIMEOUT 80
02582 #define SPI_SETLOWPOWERTIMEOUT 81
```

```

02583 #define SPI_SETPOWEROFFTIMEOUT      82
02584 #define SPI_GETLOWPOWERACTIVE         83
02585 #define SPI_GETPOWEROFFACTIVE         84
02586 #define SPI_SETLOWPOWERACTIVE         85
02587 #define SPI_SETPOWEROFFACTIVE         86
02588 #define SPI_SETCURSORS                 87
02589 #define SPI_SETICONS                   88
02590 #define SPI_GETDEFAULTINPUTLANG       89
02591 #define SPI_SETDEFAULTINPUTLANG       90
02592 #define SPI_SETLANGTOGGLE             91
02593 #define SPI_GETWINDOWSEXTENSION       92
02594 #define SPI_SETMOUSETRAILS            93
02595 #define SPI_GETMOUSETRAILS            94
02596 #define SPI_SETSCREENSAVERRUNNING     97
02597 #define SPI_SCREENSAVERRUNNING        SPI_SETSCREENSAVERRUNNING
02598
02599 /* defines below are for all win versions (_WIN32_WINNT >= 0x0400) ||
02600  * (_WIN32_WINDOWS > 0x0400) */
02601 #define SPI_GETMOUSEHOVERWIDTH         98
02602 #define SPI_SETMOUSEHOVERWIDTH         99
02603 #define SPI_GETMOUSEHOVERHEIGHT       100
02604 #define SPI_SETMOUSEHOVERHEIGHT       101
02605 #define SPI_GETMOUSEHOVERTIME         102
02606 #define SPI_SETMOUSEHOVERTIME         103
02607 #define SPI_GETWHEELSCROLLLINES       104
02608 #define SPI_SETWHEELSCROLLLINES       105
02609 #define SPI_GETMENUSHOWDELAY           106
02610 #define SPI_SETMENUSHOWDELAY           107
02611
02612 #define SPI_GETSHOWIMEUI               110
02613 #define SPI_SETSHOWIMEUI               111
02614
02615 /* defines below are for all win versions WINVER >= 0x0500 */
02616 #define SPI_GETMOUSESPEED              112
02617 #define SPI_SETMOUSESPEED              113
02618 #define SPI_GETSCREENSAVERRUNNING      114
02619 #define SPI_GETDESKWALLPAPER           115
02620
02621 #define SPI_GETACTIVEWINDOWTRACKING    0x1000
02622 #define SPI_SETACTIVEWINDOWTRACKING    0x1001
02623 #define SPI_GETMENUANIMATION           0x1002
02624 #define SPI_SETMENUANIMATION           0x1003
02625 #define SPI_GETCOMBOBOXANIMATION       0x1004
02626 #define SPI_SETCOMBOBOXANIMATION       0x1005
02627 #define SPI_GETLISTBOXSMOOTHSCROLLING  0x1006
02628 #define SPI_SETLISTBOXSMOOTHSCROLLING  0x1007
02629 #define SPI_GETGRADIENTCAPTIONS        0x1008
02630 #define SPI_SETGRADIENTCAPTIONS        0x1009
02631 #define SPI_GETMENUUNDERLINES          0x100A
02632 #define SPI_SETMENUUNDERLINES          0x100B
02633 #define SPI_GETACTIVEWNDTRKZORDER      0x100C
02634 #define SPI_SETACTIVEWNDTRKZORDER      0x100D
02635 #define SPI_GETHOTTRACKING              0x100E
02636 #define SPI_SETHOTTRACKING              0x100F
02637 #define SPI_GETFOREGROUNDLOCKTIMEOUT   0x2000
02638 #define SPI_SETFOREGROUNDLOCKTIMEOUT   0x2001
02639 #define SPI_GETACTIVEWNDTRKTIMEOUT     0x2002
02640 #define SPI_SETACTIVEWNDTRKTIMEOUT     0x2003
02641 #define SPI_GETFOREGROUNDFLASHCOUNT   0x2004
02642 #define SPI_SETFOREGROUNDFLASHCOUNT   0x2005
02643
02644 /* SystemParametersInfo flags */
02645
02646 #define SPIF_UPDATEINIFILE              1
02647 #define SPIF_SENDWININICHANGE           2
02648 #define SPIF_SENDCHANGE                  SPIF_SENDWININICHANGE
02649
02650 #if defined(_WINGDI_) && !defined(NOCDI)
02651 typedef struct {
02652     UINT        cbSize;
02653     INT          iBorderWidth;
02654     INT          iScrollWidth;
02655     INT          iScrollHeight;
02656     INT          iCaptionWidth;
02657     INT          iCaptionHeight;
02658     LOGFONTA     lfCaptionFont;
02659     INT          iSmCaptionWidth;
02660     INT          iSmCaptionHeight;
02661     LOGFONTA     lfSmCaptionFont;
02662     INT          iMenuWidth;
02663     INT          iMenuHeight;
02664     LOGFONTA     lfMenuFont;
02665     LOGFONTA     lfStatusFont;
02666     LOGFONTA     lfMessageFont;
02667 } NONCLIENTMETRICS, *PNONCLIENTMETRICS, *LPNONCLIENTMETRICS;
02668
02669 typedef struct {

```

```

02670     UINT        cbSize;
02671     INT          iBorderWidth;
02672     INT          iScrollWidth;
02673     INT          iScrollHeight;
02674     INT          iCaptionWidth;
02675     INT          iCaptionHeight;
02676     LOGFONTW     lfCaptionFont;
02677     INT          iSmCaptionWidth;
02678     INT          iSmCaptionHeight;
02679     LOGFONTW     lfSmCaptionFont;
02680     INT          iMenuWidth;
02681     INT          iMenuHeight;
02682     LOGFONTW     lfMenuFont;
02683     LOGFONTW     lfStatusFont;
02684     LOGFONTW     lfMessageFont;
02685 } NONCLIENTMETRICSW, *PNONCLIENTMETRICSW, *LPNONCLIENTMETRICSW;
02686
02687 DECL_WINELIB_TYPE_AW(NONCLIENTMETRICS)
02688 DECL_WINELIB_TYPE_AW(PNONCLIENTMETRICS)
02689 DECL_WINELIB_TYPE_AW(LPNONCLIENTMETRICS)
02690
02691 typedef struct tagICONMETRICS {
02692     UINT cbSize;
02693     int iHorzSpacing;
02694     int iVertSpacing;
02695     int iTitleWrap;
02696     LOGFONTA lfFont;
02697 } ICONMETRICS, *PICONMETRICS, *LPICONMETRICS;
02698
02699 typedef struct tagICONMETRICSW {
02700     UINT cbSize;
02701     int iHorzSpacing;
02702     int iVertSpacing;
02703     int iTitleWrap;
02704     LOGFONTW lfFont;
02705 } ICONMETRICSW, *PICONMETRICSW, *LPICONMETRICSW;
02706
02707 DECL_WINELIB_TYPE_AW(ICONMETRICS)
02708 DECL_WINELIB_TYPE_AW(PICONMETRICS)
02709 DECL_WINELIB_TYPE_AW(LPICONMETRICS)
02710 #endif /* defined(_WINGDI_) && !defined(NOCDI) */
02711
02712 #define ARW_BOTTOMLEFT      0x0000L
02713 #define ARW_BOTTOMRIGHT    0x0001L
02714 #define ARW_TOPLEFT        0x0002L
02715 #define ARW_TOPRIGHT       0x0003L
02716 #define ARW_STARTMASK      0x0003L
02717 #define ARW_STARTRIGHT     0x0001L
02718 #define ARW_STARTTOP       0x0002L
02719
02720 #define ARW_LEFT            0x0000L
02721 #define ARW_RIGHT           0x0000L
02722 #define ARW_UP              0x0004L
02723 #define ARW_DOWN            0x0004L
02724 #define ARW_HIDE            0x0008L
02725
02726 typedef struct tagMINIMIZEDMETRICS {
02727     UINT cbSize;
02728     int iWidth;
02729     int iHorzGap;
02730     int iVertGap;
02731     int iArrange;
02732 } MINIMIZEDMETRICS, *PMINIMIZEDMETRICS, *LPMINIMIZEDMETRICS;
02733
02734 /* Window Styles */
02735 #define WS_OVERLAPPED      0x00000000L
02736 #define WS_POPUP           0x80000000L
02737 #define WS_CHILD           0x40000000L
02738 #define WS_MINIMIZE        0x20000000L
02739 #define WS_VISIBLE        0x10000000L
02740 #define WS_DISABLED        0x08000000L
02741 #define WS_CLIPSIBLINGS    0x04000000L
02742 #define WS_CLIPCHILDREN    0x02000000L
02743 #define WS_MAXIMIZE        0x01000000L
02744 #define WS_CAPTION         0x00C00000L
02745 #define WS_BORDER          0x00800000L
02746 #define WS_DLGFRAME        0x00400000L
02747 #define WS_VSCROLL         0x00200000L
02748 #define WS_HSCROLL         0x00100000L
02749 #define WS_SYSMENU         0x00080000L
02750 #define WS_THICKFRAME      0x00040000L
02751 #define WS_GROUP           0x00020000L
02752 #define WS_TABSTOP         0x00010000L
02753 #define WS_MINIMIZEBOX     0x00020000L
02754 #define WS_MAXIMIZEBOX     0x00010000L
02755 #define WS_TILED           WS_OVERLAPPED
02756 #define WS_ICONIC         WS_MINIMIZE

```

```

02757 #define WS_SIZEBOX          WS_THICKFRAME
02758 #define WS_OVERLAPPEDWINDOW (WS_OVERLAPPED | WS_CAPTION | WS_SYSMENU | WS_THICKFRAME| WS_MINIMIZEBOX |
    WS_MAXIMIZEBOX)
02759 #define WS_POPUPWINDOW (WS_POPUP | WS_BORDER | WS_SYSMENU)
02760 #define WS_CHILDWINDOW (WS_CHILD)
02761 #define WS_TILEDWINDOW (WS_OVERLAPPEDWINDOW)
02762
02763 /* Window extended styles */
02764 #define WS_EX_DLGMODALFRAME    0x00000001L
02765 #define WS_EX_DRAGDETECT      0x00000002L
02766 #define WS_EX_NOPARENTNOTIFY  0x00000004L
02767 #define WS_EX_TOPMOST         0x00000008L
02768 #define WS_EX_ACCEPTFILES     0x00000010L
02769 #define WS_EX_TRANSPARENT     0x00000020L
02770
02771 /* New Win95/WinNT4 styles */
02772 #define WS_EX_MDICHILD        0x00000040L
02773 #define WS_EX_TOOLWINDOW      0x00000080L
02774 #define WS_EX_WINDOWEDGE      0x00000100L
02775 #define WS_EX_CLIENTEDGE      0x00000200L
02776 #define WS_EX_CONTEXTHELP     0x00000400L
02777 #define WS_EX_RIGHT            0x00001000L
02778 #define WS_EX_LEFT            0x00000000L
02779 #define WS_EX_RTLREADING      0x00002000L
02780 #define WS_EX_LTRREADING      0x00000000L
02781 #define WS_EX_LEFTSCROLLBAR   0x00004000L
02782 #define WS_EX_RIGHTSCROLLBAR  0x00000000L
02783 #define WS_EX_CONTROLPARENT   0x00010000L
02784 #define WS_EX_STATICEDGE      0x00020000L
02785 #define WS_EX_APPWINDOW       0x00040000L
02786
02787 #define WS_EX_OVERLAPPEDWINDOW (WS_EX_WINDOWEDGE|WS_EX_CLIENTEDGE)
02788 #define WS_EX_PALETTEWINDOW   (WS_EX_WINDOWEDGE|WS_EX_TOOLWINDOW|WS_EX_TOPMOST)
02789
02790 /* New Win2000 styles */
02791 #define WS_EX_LAYERED          0x00080000L
02792
02793 /* WINE internal... */
02794 #define WS_EX_TRAYWINDOW       0x80000000L
02795 #define WS_EX_MANAGED          0x40000000L /* Window managed by the window system */
02796
02797 /* Window scrolling */
02798 #define SW_SCROLLCHILDREN      0x0001
02799 #define SW_INVALIDATE          0x0002
02800 #define SW_ERASE                0x0004
02801
02802 /* CreateWindow() coordinates */
02803 #define CW_USEDEFAULT ((INT)0x80000000)
02804
02805 /* ChildWindowFromPointEx Flags */
02806 #define CWP_ALL                0x0000
02807 #define CWP_SKIPINVISIBLE      0x0001
02808 #define CWP_SKIPDISABLED       0x0002
02809 #define CWP_SKIPTRANSPARENT    0x0004
02810
02811 /* PeekMessage() options */
02812 #define PM_NOREMOVE            0x0000
02813 #define PM_REMOVE              0x0001
02814 #define PM_NOYIELD             0x0002
02815
02816 /* AnimateWindow() flags */
02817 #define AW_SLIDE                0x00040000
02818 #define AW_ACTIVATE            0x00020000
02819 #define AW_BLEND                0x00080000
02820 #define AW_HIDE                0x00010000
02821 #define AW_CENTER              0x00000010
02822 #define AW_HOR_POSITIVE        0x00000001
02823 #define AW_HOR_NEGATIVE        0x00000002
02824 #define AW_VER_POSITIVE        0x00000004
02825 #define AW_VER_NEGATIVE        0x00000008
02826
02827 /* WM_SHOWWINDOW wParam codes */
02828 #define SW_PARENTCLOSING        1
02829 #define SW_OTHERMAXIMIZED       2
02830 #define SW_PARENTOPENING        3
02831 #define SW_OTHERRESTORED        4
02832
02833 /* ShowWindow() codes */
02834 #define SW_HIDE                 0
02835 #define SW_SHOWNORMAL           1
02836 #define SW_NORMAL               1
02837 #define SW_SHOWMINIMIZED        2
02838 #define SW_SHOWMAXIMIZED        3
02839 #define SW_MAXIMIZE             3
02840 #define SW_SHOWNOACTIVATE       4
02841 #define SW_SHOW                 5
02842 #define SW_MINIMIZE             6

```

```
02843 #define SW_SHOWMINNOACTIVE 7
02844 #define SW_SHOWNA 8
02845 #define SW_RESTORE 9
02846 #define SW_SHOWDEFAULT 10
02847 #define SW_MAX 10
02848 #define SW_NORMALNA 0xCC /* undoc. flag in MinMaximize */
02849
02850 /* WM_SIZE message wParam values */
02851 #define SIZE_RESTORED 0
02852 #define SIZE_MINIMIZED 1
02853 #define SIZE_MAXIMIZED 2
02854 #define SIZE_MAXSHOW 3
02855 #define SIZE_MAXHIDE 4
02856 #define SIZENORMAL SIZE_RESTORED
02857 #define SIZEICONIC SIZE_MINIMIZED
02858 #define SIZEFULLSCREEN SIZE_MAXIMIZED
02859 #define SIZEZOOMSHOW SIZE_MAXSHOW
02860 #define SIZEZOOMHIDE SIZE_MAXHIDE
02861
02862 /* SetWindowPos() and WINDOWPOS flags */
02863 #define SWP_NOSIZE 0x0001
02864 #define SWP_NOMOVE 0x0002
02865 #define SWP_NOZORDER 0x0004
02866 #define SWP_NOREDRAW 0x0008
02867 #define SWP_NOACTIVATE 0x0010
02868 #define SWP_FRAMECHANGED 0x0020 /* The frame changed: send WM_NCCALCSIZE */
02869 #define SWP_SHOWWINDOW 0x0040
02870 #define SWP_HIDEWINDOW 0x0080
02871 #define SWP_NOCOPYBITS 0x0100
02872 #define SWP_NOOWNERZORDER 0x0200 /* Don't do owner Z ordering */
02873
02874 #define SWP_DRAWFRAME SWP_FRAMECHANGED
02875 #define SWP_NOREPOSITION SWP_NOOWNERZORDER
02876
02877 #define SWP_NOSENDCHANGING 0x0400
02878 #define SWP_DEFERERASE 0x2000
02879 #define SWP_ASYNCWINDOWPOS 0x4000
02880
02881 #define HWND_DESKTOP ((HWND)0)
02882 #define HWND_BROADCAST ((HWND)0xffff)
02883
02884 /* SetWindowPos() hwndInsertAfter field values */
02885 #define HWND_TOP ((HWND)0)
02886 #define HWND_BOTTOM ((HWND)1)
02887 #define HWND_TOPMOST ((HWND)-1)
02888 #define HWND_NOTOPMOST ((HWND)-2)
02889 #define HWND_MESSAGE ((HWND)-3)
02890
02891 /* GetDCEX flags */
02892 #define DCX_WINDOW 0x00000001
02893 #define DCX_CACHE 0x00000002
02894 #define DCX_NORESETATTRS 0x00000004
02895 #define DCX_CLIPCHILDREN 0x00000008
02896 #define DCX_CLIPSIBLINGS 0x00000010
02897 #define DCX_PARENTCLIP 0x00000020
02898 #define DCX_EXCLUDEGRN 0x00000040
02899 #define DCX_INTERSECTGRN 0x00000080
02900 #define DCX_EXCLUDEUPDATE 0x00000100
02901 #define DCX_INTERSECTUPDATE 0x00000200
02902 #define DCX_LOCKWINDOWUPDATE 0x00000400
02903 #define DCX_USESTYLE 0x00010000
02904 #define DCX_NORECOMPUTE 0x00100000
02905 #define DCX_VALIDATE 0x00200000
02906
02907 #define MF_INSERT 0x0000
02908 #define MF_CHANGE 0x0080
02909 #define MF_APPEND 0x0100
02910 #define MF_DELETE 0x0200
02911 #define MF_REMOVE 0x1000
02912 #define MF_END 0x0080
02913
02914 #define MF_ENABLED 0x0000
02915 #define MF_GRAYED 0x0001
02916 #define MF_DISABLED 0x0002
02917 #define MF_STRING 0x0000
02918 #define MF_BITMAP 0x0004
02919 #define MF_UNCHECKED 0x0000
02920 #define MF_CHECKED 0x0008
02921 #define MF_POPUP 0x0010
02922 #define MF_MENUBARBREAK 0x0020
02923 #define MF_MENUBREAK 0x0040
02924 #define MF_UNHILITE 0x0000
02925 #define MF_HILITE 0x0080
02926 #define MF_OWNERDRAW 0x0100
02927 #define MF_USECHECKBITMAPS 0x0200
02928 #define MF_BYCOMMAND 0x0000
02929 #define MF_BYPOSITION 0x0400
```

```
02930 #define MF_SEPARATOR      0x0800
02931 #define MF_DEFAULT          0x1000
02932 #define MF_SYSMENU          0x2000
02933 #define MF_HELP              0x4000
02934 #define MF_RIGHTJUSTIFY     0x4000
02935 #define MF_MOUSESELECT      0x8000
02936
02937 /* Flags for extended menu item types. */
02938 #define MFT_STRING            MF_STRING
02939 #define MFT_BITMAP            MF_BITMAP
02940 #define MFT_MENUBARBREAK     MF_MENUBARBREAK
02941 #define MFT_MENUBREAK        MF_MENUBREAK
02942 #define MFT_OWNERDRAW         MF_OWNERDRAW
02943 #define MFT_RADIOCHECK        0x00000200L
02944 #define MFT_SEPARATOR         MF_SEPARATOR
02945 #define MFT_RIGHTORDER        0x00002000L
02946 #define MFT_RIGHTJUSTIFY     MF_RIGHTJUSTIFY
02947
02948 /* Flags for extended menu item states. */
02949 #define MFS_GRAYED            0x00000003L
02950 #define MFS_DISABLED          MFS_GRAYED
02951 #define MFS_CHECKED           MF_CHECKED
02952 #define MFS_HILITE            MF_HILITE
02953 #define MFS_ENABLED           MF_ENABLED
02954 #define MFS_UNCHECKED         MF_UNCHECKED
02955 #define MFS_UNHILITE          MF_UNHILITE
02956 #define MFS_DEFAULT           MF_DEFAULT
02957 #define MFS_MASK               0x0000108BL
02958 #define MFS_HOTTRACKDRAWN     0x10000000L
02959 #define MFS_CACHEDBMP         0x20000000L
02960 #define MFS_BOTTOMGAPDROP     0x40000000L
02961 #define MFS_TOPGAPDROP        0x80000000L
02962 #define MFS_GAPDROP           0xC0000000L
02963
02964 /* for GetMenuDefaultItem */
02965 #define GMDI_USEDISABLED      0x00001L
02966 #define GMDI_GOINTOPOPUPS     0x00002L
02967
02968 #define DT_TOP 0
02969 #define DT_LEFT 0
02970 #define DT_CENTER 1
02971 #define DT_RIGHT 2
02972 #define DT_VCENTER 4
02973 #define DT_BOTTOM 8
02974 #define DT_WORDBREAK 16
02975 #define DT_SINGLELINE 32
02976 #define DT_EXPANDTABS 64
02977 #define DT_TABSTOP 128
02978 #define DT_NOCLIP 256
02979 #define DT_EXTERNALLEADING 512
02980 #define DT_CALCRECT 1024
02981 #define DT_NOPREFIX 2048
02982 #define DT_INTERNAL 4096
02983
02984 /* DrawCaption()/DrawCaptionTemp() flags */
02985 #define DC_ACTIVE              0x0001
02986 #define DC_SMALLCAP            0x0002
02987 #define DC_ICON                 0x0004
02988 #define DC_TEXT                 0x0008
02989 #define DC_INBUTTON            0x0010
02990
02991 /* DrawEdge() flags */
02992 #define BDR_RAISEDOUTER        0x0001
02993 #define BDR_SUNKENOUTER        0x0002
02994 #define BDR_RAISEDINNER        0x0004
02995 #define BDR_SUNKENINNER        0x0008
02996
02997 #define BDR_OUTER               0x0003
02998 #define BDR_INNER               0x000c
02999 #define BDR_RAISED              0x0005
03000 #define BDR_SUNKEN              0x000a
03001
03002 #define EDGE_RAISED             (BDR_RAISEDOUTER | BDR_RAISEDINNER)
03003 #define EDGE_SUNKEN             (BDR_SUNKENOUTER | BDR_SUNKENINNER)
03004 #define EDGE_ETCHED             (BDR_SUNKENOUTER | BDR_RAISEDINNER)
03005 #define EDGE_BUMP               (BDR_RAISEDOUTER | BDR_SUNKENINNER)
03006
03007 /* border flags */
03008 #define BF_LEFT                 0x0001
03009 #define BF_TOP                  0x0002
03010 #define BF_RIGHT                0x0004
03011 #define BF_BOTTOM               0x0008
03012 #define BF_DIAGONAL             0x0010
03013 #define BF_MIDDLE               0x0800 /* Fill in the middle */
03014 #define BF_SOFT                 0x1000 /* For softer buttons */
03015 #define BF_ADJUST               0x2000 /* Calculate the space left over */
03016 #define BF_FLAT                 0x4000 /* For flat rather than 3D borders */
```

```

03017 #define BF_MONO                0x8000 /* For monochrome borders */
03018 #define BF_TOPLEFT               (BF_TOP | BF_LEFT)
03019 #define BF_TOPRIGHT              (BF_TOP | BF_RIGHT)
03020 #define BF_BOTTOMLEFT            (BF_BOTTOM | BF_LEFT)
03021 #define BF_BOTTOMRIGHT           (BF_BOTTOM | BF_RIGHT)
03022 #define BF_RECT                  (BF_LEFT | BF_TOP | BF_RIGHT | BF_BOTTOM)
03023 #define BF_DIAGONAL_ENDTOPRIGHT (BF_DIAGONAL | BF_TOP | BF_RIGHT)
03024 #define BF_DIAGONAL_ENDTOPLEFT  (BF_DIAGONAL | BF_TOP | BF_LEFT)
03025 #define BF_DIAGONAL_ENDBOTTOMLEFT (BF_DIAGONAL | BF_BOTTOM | BF_LEFT)
03026 #define BF_DIAGONAL_ENDBOTTOMRIGHT (BF_DIAGONAL | BF_BOTTOM | BF_RIGHT)
03027
03028 /* DrawFrameControl() uType's */
03029
03030 #define DFC_CAPTION               1
03031 #define DFC_MENU                  2
03032 #define DFC_SCROLL                3
03033 #define DFC_BUTTON                4
03034
03035 /* uState's */
03036
03037 #define DFCS_CAPTIONCLOSE         0x0000
03038 #define DFCS_CAPTIONMIN          0x0001
03039 #define DFCS_CAPTIONMAX          0x0002
03040 #define DFCS_CAPTIONRESTORE      0x0003
03041 #define DFCS_CAPTIONHELP         0x0004 /* Windows 95 only */
03042
03043 #define DFCS_MENUARROW           0x0000
03044 #define DFCS_MENUCHECK           0x0001
03045 #define DFCS_MENUBULLET          0x0002
03046 #define DFCS_MENUARROWRIGHT      0x0004
03047
03048 #define DFCS_SCROLLUP            0x0000
03049 #define DFCS_SCROLLDOWN          0x0001
03050 #define DFCS_SCROLLLEFT         0x0002
03051 #define DFCS_SCROLLRIGHT        0x0003
03052 #define DFCS_SCROLLCOMBOBOX      0x0005
03053 #define DFCS_SCROLLSIZEGRIP      0x0008
03054 #define DFCS_SCROLLSIZEGRIPRIGHT 0x0010
03055
03056 #define DFCS_BUTTONCHECK         0x0000
03057 #define DFCS_BUTTONRADIOIMAGE    0x0001
03058 #define DFCS_BUTTONRADIOMASK     0x0002 /* to draw nonsquare button */
03059 #define DFCS_BUTTONRADIO         0x0004
03060 #define DFCS_BUTTON3STATE        0x0008
03061 #define DFCS_BUTTONPUSH          0x0010
03062
03063 /* additional state of the control */
03064
03065 #define DFCS_INACTIVE            0x0100
03066 #define DFCS_PUSHED              0x0200
03067 #define DFCS_CHECKED             0x0400
03068 #define DFCS_ADJUSTRECT          0x2000 /* exclude surrounding edge */
03069 #define DFCS_FLAT                0x4000
03070 #define DFCS_MONO                0x8000
03071
03072 /* Image type */
03073 #define DST_COMPLEX               0x0000
03074 #define DST_TEXT                  0x0001
03075 #define DST_PREFIXTEXT            0x0002
03076 #define DST_ICON                  0x0003
03077 #define DST_BITMAP                0x0004
03078
03079 /* State type */
03080 #define DSS_NORMAL                0x0000
03081 #define DSS_UNION                 0x0010 /* Gray string appearance */
03082 #define DSS_DISABLED              0x0020
03083 #define DSS_DEFAULT                0x0040 /* Make it bold */
03084 #define DSS_MONO                  0x0080
03085 #define DSS_RIGHT                 0x8000
03086
03087 typedef struct
03088 {
03089     UINT        CtlType;
03090     UINT        CtlID;
03091     UINT        itemID;
03092     UINT        itemAction;
03093     UINT        itemState;
03094     HWND        hwndItem;
03095     HDC         hDC;
03096     RECT        rcItem WINE_PACKED;
03097     DWORD       itemData WINE_PACKED;
03098 } DRAWITEMSTRUCT, *PDRAWITEMSTRUCT, *LPDRAWITEMSTRUCT;
03099
03100
03101 typedef struct
03102 {
03103     UINT        CtlType;

```

```

03104     UINT        CtlID;
03105     UINT        itemID;
03106     UINT        itemWidth;
03107     UINT        itemHeight;
03108     DWORD       itemData;
03109 } MEASUREITEMSTRUCT, *PMEASUREITEMSTRUCT, *LPMEASUREITEMSTRUCT;
03110
03111
03112 typedef struct
03113 {
03114     UINT        CtlType;
03115     UINT        CtlID;
03116     UINT        itemID;
03117     HWND        hwndItem;
03118     DWORD       itemData;
03119 } DELETEITEMSTRUCT, *PDELETEITEMSTRUCT, *LPDELETEITEMSTRUCT;
03120
03121
03122 typedef struct
03123 {
03124     UINT        CtlType;
03125     UINT        CtlID;
03126     HWND        hwndItem;
03127     UINT        itemID1;
03128     DWORD       itemData1;
03129     UINT        itemID2;
03130     DWORD       itemData2;
03131     DWORD       dwLocaleId;
03132 } COMPAREITEMSTRUCT, *PCOMPAREITEMSTRUCT, *LPCOMPAREITEMSTRUCT;
03133
03134
03135 /* WM_KEYUP/DOWN/CHAR HIWORD(lParam) flags */
03136 #define KF_EXTENDED      0x0100
03137 #define KF_DLGMODE       0x0800
03138 #define KF_MENUMODE      0x1000
03139 #define KF_ALTDOWN       0x2000
03140 #define KF_REPEAT        0x4000
03141 #define KF_UP            0x8000
03142
03143 /* Virtual key codes */
03144 #define VK_LBUTTON       0x01
03145 #define VK_RBUTTON       0x02
03146 #define VK_CANCEL        0x03
03147 #define VK_MBUTTON       0x04
03148 #define VK_XBUTTON1      0x05
03149 #define VK_XBUTTON2      0x06
03150 /*                                0x07  Undefined */
03151 #define VK_BACK          0x08
03152 #define VK_TAB           0x09
03153 /*                                0x0A-0x0B  Undefined */
03154 #define VK_CLEAR         0x0C
03155 #define VK_RETURN        0x0D
03156 /*                                0x0E-0x0F  Undefined */
03157 #define VK_SHIFT         0x10
03158 #define VK_CONTROL       0x11
03159 #define VK_MENU          0x12
03160 #define VK_PAUSE         0x13
03161 #define VK_CAPITAL       0x14
03162 /*                                0x15-0x19  Reserved for Kanji systems */
03163 /*                                0x1A      Undefined */
03164 #define VK_ESCAPE        0x1B
03165 /*                                0x1C-0x1F  Reserved for Kanji systems */
03166 #define VK_SPACE         0x20
03167 #define VK_PRIOR         0x21
03168 #define VK_NEXT          0x22
03169 #define VK_END           0x23
03170 #define VK_HOME          0x24
03171 #define VK_LEFT          0x25
03172 #define VK_UP            0x26
03173 #define VK_RIGHT         0x27
03174 #define VK_DOWN          0x28
03175 #define VK_SELECT        0x29
03176 #define VK_PRINT         0x2A /* OEM specific in Windows 3.1 SDK */
03177 #define VK_EXECUTE       0x2B
03178 #define VK_SNAPSHOT      0x2C
03179 #define VK_INSERT        0x2D
03180 #define VK_DELETE        0x2E
03181 #define VK_HELP          0x2F
03182 #define VK_0             0x30
03183 #define VK_1             0x31
03184 #define VK_2             0x32
03185 #define VK_3             0x33
03186 #define VK_4             0x34
03187 #define VK_5             0x35
03188 #define VK_6             0x36
03189 #define VK_7             0x37
03190 #define VK_8             0x38

```

```
03191 #define VK_9                0x39
03192 /*                          0x3A-0x40  Undefined */
03193 #define VK_A                0x41
03194 #define VK_B                0x42
03195 #define VK_C                0x43
03196 #define VK_D                0x44
03197 #define VK_E                0x45
03198 #define VK_F                0x46
03199 #define VK_G                0x47
03200 #define VK_H                0x48
03201 #define VK_I                0x49
03202 #define VK_J                0x4A
03203 #define VK_K                0x4B
03204 #define VK_L                0x4C
03205 #define VK_M                0x4D
03206 #define VK_N                0x4E
03207 #define VK_O                0x4F
03208 #define VK_P                0x50
03209 #define VK_Q                0x51
03210 #define VK_R                0x52
03211 #define VK_S                0x53
03212 #define VK_T                0x54
03213 #define VK_U                0x55
03214 #define VK_V                0x56
03215 #define VK_W                0x57
03216 #define VK_X                0x58
03217 #define VK_Y                0x59
03218 #define VK_Z                0x5A
03219
03220 #define VK_LWIN              0x5B
03221 #define VK_RWIN              0x5C
03222 #define VK_APPS              0x5D
03223 /*                          0x5E-0x5F  Unassigned */
03224 #define VK_NUMPAD0           0x60
03225 #define VK_NUMPAD1           0x61
03226 #define VK_NUMPAD2           0x62
03227 #define VK_NUMPAD3           0x63
03228 #define VK_NUMPAD4           0x64
03229 #define VK_NUMPAD5           0x65
03230 #define VK_NUMPAD6           0x66
03231 #define VK_NUMPAD7           0x67
03232 #define VK_NUMPAD8           0x68
03233 #define VK_NUMPAD9           0x69
03234 #define VK_MULTIPLY          0x6A
03235 #define VK_ADD               0x6B
03236 #define VK_SEPARATOR         0x6C
03237 #define VK_SUBTRACT          0x6D
03238 #define VK_DECIMAL           0x6E
03239 #define VK_DIVIDE            0x6F
03240 #define VK_F1                0x70
03241 #define VK_F2                0x71
03242 #define VK_F3                0x72
03243 #define VK_F4                0x73
03244 #define VK_F5                0x74
03245 #define VK_F6                0x75
03246 #define VK_F7                0x76
03247 #define VK_F8                0x77
03248 #define VK_F9                0x78
03249 #define VK_F10               0x79
03250 #define VK_F11               0x7A
03251 #define VK_F12               0x7B
03252 #define VK_F13               0x7C
03253 #define VK_F14               0x7D
03254 #define VK_F15               0x7E
03255 #define VK_F16               0x7F
03256 #define VK_F17               0x80
03257 #define VK_F18               0x81
03258 #define VK_F19               0x82
03259 #define VK_F20               0x83
03260 #define VK_F21               0x84
03261 #define VK_F22               0x85
03262 #define VK_F23               0x86
03263 #define VK_F24               0x87
03264 /*                          0x88-0x8F  Unassigned */
03265 #define VK_NUMLOCK            0x90
03266 #define VK_SCROLL            0x91
03267 /*                          0x92-0x9F  Unassigned */
03268 /*
03269  * differencing between right and left shift/control/alt key.
03270  * Used only by GetAsyncKeyState() and GetKeyState().
03271  */
03272 #define VK_LSHIFT             0xA0
03273 #define VK_RSHIFT             0xA1
03274 #define VK_LCONTROL           0xA2
03275 #define VK_RCONTROL           0xA3
03276 #define VK_LMENU              0xA4
03277 #define VK_RMENU              0xA5
```

```

03278 /*                                0xA6-0xB9  Unassigned */
03279 #define VK_OEM_1                    0xBA
03280 #define VK_OEM_PLUS                  0xBB
03281 #define VK_OEM_COMMA                0xBC
03282 #define VK_OEM_MINUS                0xBD
03283 #define VK_OEM_PERIOD              0xBE
03284 #define VK_OEM_2                    0xBF
03285 #define VK_OEM_3                    0xC0
03286 /*                                0xC1-0xDA  Unassigned */
03287 #define VK_OEM_4                    0xDB
03288 #define VK_OEM_5                    0xDC
03289 #define VK_OEM_6                    0xDD
03290 #define VK_OEM_7                    0xDE
03291 #define VK_OEM_8                    0xDF
03292 /*                                0xE0      OEM specific */
03293 #define VK_OEM_AX                   0xE1  /* "AX" key on Japanese AX keyboard */
03294 #define VK_OEM_102                 0xE2  /* "<>" or "|" on RT 102-key keyboard */
03295 #define VK_ICO_HELP                0xE3  /* Help key on ICO */
03296 #define VK_ICO_00                  0xE4  /* 00 key on ICO */
03297 #define VK_PROCESSKEY               0xE5
03298
03299 /*                                0xE6      OEM specific */
03300 /*                                0xE7-0xE8  Unassigned */
03301 /*                                0xE9-0xF5  OEM specific */
03302
03303 #define VK_ATTN                     0xF6
03304 #define VK_CRSEL                    0xF7
03305 #define VK_EXSEL                    0xF8
03306 #define VK_EREOF                    0xF9
03307 #define VK_PLAY                     0xFA
03308 #define VK_ZOOM                     0xFB
03309 #define VK_NONAME                   0xFC
03310 #define VK_PA1                      0xFD
03311 #define VK_OEM_CLEAR                0xFE
03312
03313 /* Key status flags for mouse events */
03314 #define MK_LBUTTON                  0x0001
03315 #define MK_RBUTTON                  0x0002
03316 #define MK_SHIFT                    0x0004
03317 #define MK_CONTROL                  0x0008
03318 #define MK_MBUTTON                  0x0010
03319 #define MK_XBUTTON1                 0x0020
03320 #define MK_XBUTTON2                 0x0040
03321
03322 /* Queue status flags */
03323 #define QS_KEY                      0x0001
03324 #define QS_MOUSEMOVE                0x0002
03325 #define QS_MOUSEBUTTON              0x0004
03326 #define QS_MOUSE                    (QS_MOUSEMOVE | QS_MOUSEBUTTON)
03327 #define QS_POSTMESSAGE              0x0008
03328 #define QS_TIMER                    0x0010
03329 #define QS_PAINT                    0x0020
03330 #define QS_SENDMESSAGE              0x0040
03331 #define QS_HOTKEY                   0x0080
03332 #define QS_INPUT                    (QS_MOUSE | QS_KEY)
03333 #define QS_ALLEVENTS                (QS_INPUT | QS_POSTMESSAGE | QS_TIMER | QS_PAINT | QS_HOTKEY)
03334 #define QS_ALLINPUT                 (QS_ALLEVENTS | QS_SENDMESSAGE)
03335
03336 /* Extra (undocumented) queue wake bits - see "Undoc. Windows" */
03337 #define QS_SMRRESULT                0x8000
03338
03339 /* InSendMessageEx flags */
03340 #define ISMEX_NOSEND                0x00000000
03341 #define ISMEX_SEND                   0x00000001
03342 #define ISMEX_NOTIFY                 0x00000002
03343 #define ISMEX_CALLBACK               0x00000004
03344 #define ISMEX_REPLIED               0x00000008
03345
03346 #define DDL_READWRITE               0x0000
03347 #define DDL_READONLY                0x0001
03348 #define DDL_HIDDEN                   0x0002
03349 #define DDL_SYSTEM                   0x0004
03350 #define DDL_DIRECTORY               0x0010
03351 #define DDL_ARCHIVE                 0x0020
03352
03353 #define DDL_POSTMSGS                0x2000
03354 #define DDL_DRIVES                   0x4000
03355 #define DDL_EXCLUSIVE               0x8000
03356
03357 /* Shell hook values */
03358 #define HSELL_WINDOWCREATED          1
03359 #define HSELL_WINDOWDESTROYED        2
03360 #define HSELL_ACTIVATESHELLWINDOW    3
03361
03362 /* Predefined Clipboard Formats */
03363 #define CF_TEXT                      1
03364 #define CF_BITMAP                    2

```

```

03365 #define CF_METAFILEPICT      3
03366 #define CF_SYLK                4
03367 #define CF_DIF                 5
03368 #define CF_TIFF                6
03369 #define CF_OEMTEXT             7
03370 #define CF_DIB                 8
03371 #define CF_PALETTE             9
03372 #define CF_PENDATA             10
03373 #define CF_RIFF                11
03374 #define CF_WAVE                12
03375 #define CF_UNICODETEXT         13
03376 #define CF_ENHMETAFILE         14
03377 #define CF_HDROP               15
03378 #define CF_LOCALE              16
03379 #define CF_DIBV5               17
03380 #define CF_MAX                 18
03381
03382 #define CF_OWNERDISPLAY         0x0080
03383 #define CF_DSPTEXT             0x0081
03384 #define CF_DSPBITMAP           0x0082
03385 #define CF_DSPMETAFILEPICT     0x0083
03386 #define CF_DSPENHMETAFILE      0x008E
03387
03388 /* "Private" formats don't get GlobalFree()'d */
03389 #define CF_PRIVATEFIRST         0x0200
03390 #define CF_PRIVATELAST         0x02FF
03391
03392 /* "GDI OBJ" formats do get DeleteObject()'d */
03393 #define CF_GDIOBJFIRST         0x0300
03394 #define CF_GDIOBJLAST         0x03FF
03395
03396
03397 /* types of LoadImage */
03398 #define IMAGE_BITMAP            0
03399 #define IMAGE_ICON              1
03400 #define IMAGE_CURSOR            2
03401 #define IMAGE_ENHMETAFILE       3
03402
03403 /* loadflags to LoadImage */
03404 #define LR_DEFAULTCOLOR         0x0000
03405 #define LR_MONOCHROME           0x0001
03406 #define LR_COLOR                0x0002
03407 #define LR_COPYRETURNORG        0x0004
03408 #define LR_COPYDELETEORG        0x0008
03409 #define LR_LOADFROMFILE         0x0010
03410 #define LR_LOADTRANSPARENT      0x0020
03411 #define LR_DEFAULTSIZE          0x0040
03412 #define LR_VGA_COLOR            0x0080
03413 #define LR_LOADMAP3DCOLORS      0x1000
03414 #define LR_CREATEDIBSECTION     0x2000
03415 #define LR_COPYFROMRESOURCE     0x4000
03416 #define LR_SHARED               0x8000
03417
03418 /* Flags for DrawIconEx. */
03419 #define DI_MASK                  1
03420 #define DI_IMAGE                 2
03421 #define DI_NORMAL                (DI_MASK | DI_IMAGE)
03422 #define DI_COMPAT                4
03423 #define DI_DEFAULTSIZE           8
03424
03425 /* WM_NOTIFYFORMAT commands and return values */
03426 #define NFR_ANSI                 1
03427 #define NFR_UNICODE              2
03428 #define NF_QUERY                 3
03429 #define NF_QUERY                 4
03430
03431 /* RegisterDeviceNotification stuff */
03432 typedef PVOID HDEVNOTIFY;
03433 typedef HDEVNOTIFY *PHDEVNOTIFY;
03434
03435 #define DEVICE_NOTIFY_WINDOW_HANDLE 0x00000000
03436
03437 #define EnumTaskWindows(handle,proc,lparam) \
03438 EnumThreadWindows(handle,proc,lparam)
03439 #define OemToAnsiA OemToCharA
03440 #define OemToAnsiW OemToCharW
03441 #define OemToAnsi WINELIB_NAME_AW(OemToAnsi)
03442 #define OemToAnsiBuffA OemToCharBuffA
03443 #define OemToAnsiBuffW OemToCharBuffW
03444 #define OemToAnsiBuff WINELIB_NAME_AW(OemToAnsiBuff)
03445 #define AnsiToOemA CharToOemA
03446 #define AnsiToOemW CharToOemW
03447 #define AnsiToOem WINELIB_NAME_AW(AnsiToOem)
03448 #define AnsiToOemBuffA CharToOemBuffA
03449 #define AnsiToOemBuffW CharToOemBuffW
03450 #define AnsiToOemBuff WINELIB_NAME_AW(AnsiToOemBuff)
03451

```

```

03452 #if defined(_WINGDI_) && !defined(NOCDI)
03453 LONG WINAPI ChangeDisplaySettingsA(LPDEVMODEA,DWORD);
03454 LONG WINAPI ChangeDisplaySettingsW(LPDEVMODEW,DWORD);
03455 #define ChangeDisplaySettings WINELIB_NAME_AW(ChangeDisplaySettings)
03456 LONG WINAPI ChangeDisplaySettingsExA(LPCSTR,LPDEVMODEA,HWND,DWORD,LPARAM);
03457 LONG WINAPI ChangeDisplaySettingsExW(LPCWSTR,LPDEVMODEW,HWND,DWORD,LPARAM);
03458 #define ChangeDisplaySettingsEx WINELIB_NAME_AW(ChangeDisplaySettingsEx)
03459 BOOL WINAPI EnumDisplayDevicesA(LPVOID,DWORD,LPDISPLAY_DEVICEA,DWORD);
03460 BOOL WINAPI EnumDisplayDevicesW(LPVOID,DWORD,LPDISPLAY_DEVICEW,DWORD);
03461 #define EnumDisplayDevices WINELIB_NAME_AW(EnumDisplayDevices)
03462 BOOL WINAPI EnumDisplaySettingsA(LPCSTR,DWORD,LPDEVMODEA);
03463 BOOL WINAPI EnumDisplaySettingsW(LPCWSTR,DWORD,LPDEVMODEW);
03464 #define EnumDisplaySettings WINELIB_NAME_AW(EnumDisplaySettings)
03465 #endif /* defined(_WINGDI_) && !defined(NOCDI) */
03466
03467 HKL WINAPI ActivateKeyboardLayout(HKL,UINT);
03468 LONG WINAPI BroadcastSystemMessage(DWORD,LPDWORD,UINT,WPARAM,LPARAM);
03469 WORD WINAPI CascadeWindows(HWND,UINT, const LPRECT,UINT, const HWND *);
03470 INT WINAPI CopyAcceleratorTableA(HACCEL,LPACCEL,INT);
03471 INT WINAPI CopyAcceleratorTableW(HACCEL,LPACCEL,INT);
03472 #define CopyAcceleratorTable WINELIB_NAME_AW(CopyAcceleratorTable)
03473 HACCEL WINAPI CreateAcceleratorTableA(LPACCEL,INT);
03474 HACCEL WINAPI CreateAcceleratorTableW(LPACCEL,INT);
03475 #define CreateAcceleratorTable WINELIB_NAME_AW(CreateAcceleratorTable)
03476 HICON WINAPI CreateIconIndirect(PICONINFO);
03477 BOOL WINAPI DestroyAcceleratorTable(HACCEL);
03478 BOOL WINAPI EnumDesktopsA(HWINSTA,DESKTOPENUMPROCA,LPARAM);
03479 BOOL WINAPI EnumDesktopsW(HWINSTA,DESKTOPENUMPROCW,LPARAM);
03480 #define EnumDesktops WINELIB_NAME_AW(EnumDesktops)
03481 BOOL WINAPI EnumDisplayMonitors(HDC,LPRECT,MONITORENUMPROC,LPARAM);
03482 INT WINAPI EnumPropsExA(HWND,PROPENUMPROCEXA,LPARAM);
03483 INT WINAPI EnumPropsExW(HWND,PROPENUMPROCEXW,LPARAM);
03484 #define EnumPropsEx WINELIB_NAME_AW(EnumPropsEx)
03485 BOOL WINAPI EnumThreadWindows(DWORD,WNDENUMPROC,LPARAM);
03486 BOOL WINAPI ExitWindowsEx(UINT,DWORD);
03487 BOOL WINAPI GetIconInfo(HICON,PICONINFO);
03488 HKL WINAPI GetKeyboardLayout(DWORD);
03489 INT WINAPI GetKeyboardLayoutList(INT,HKL *);
03490 DWORD WINAPI GetMenuContextHelpId(HMENU);
03491 UINT WINAPI GetMenuDefaultItem(HMENU,UINT,UINT);
03492 BOOL WINAPI GetMenuInfo(HMENU,LPMENUINFO);
03493 BOOL WINAPI GetMenuItemInfoA(HMENU,UINT,BOOL,MENUTITEMINFOA*);
03494 BOOL WINAPI GetMenuItemInfoW(HMENU,UINT,BOOL,MENUTITEMINFOW*);
03495 #define GetMenuItemInfo WINELIB_NAME_AW(GetMenuItemInfo)
03496 BOOL WINAPI GetMonitorInfoA(HMONITOR,LPMONITORINFO);
03497 BOOL WINAPI GetMonitorInfoW(HMONITOR,LPMONITORINFO);
03498 #define GetMonitorInfo WINELIB_NAME_AW(GetMonitorInfo)
03499 DWORD WINAPI GetWindowContextHelpId(HWND);
03500 DWORD WINAPI GetWindowThreadProcessId(HWND,LPDWORD);
03501 BOOL WINAPI IsWindowUnicode(HWND);
03502 HKL WINAPI LoadKeyboardLayoutA(LPCSTR,UINT);
03503 HKL WINAPI LoadKeyboardLayoutW(LPCWSTR,UINT);
03504 #define LoadKeyboardLayout WINELIB_NAME_AW(LoadKeyboardLayout)
03505 INT WINAPI MessageBoxExA(HWND,LPCSTR,LPCSTR,UINT,WORD);
03506 INT WINAPI MessageBoxExW(HWND,LPCWSTR,LPCWSTR,UINT,WORD);
03507 #define MessageBoxEx WINELIB_NAME_AW(MessageBoxEx)
03508 HMONITOR WINAPI MonitorFromPoint(POINT,DWORD);
03509 HMONITOR WINAPI MonitorFromRect(LPRECT,DWORD);
03510 HMONITOR WINAPI MonitorFromWindow(HWND,DWORD);
03511 DWORD WINAPI MsgWaitForMultipleObjects(DWORD,CONST HANDLE*,BOOL,DWORD,DWORD);
03512 DWORD WINAPI MsgWaitForMultipleObjectsEx(DWORD,CONST HANDLE*,DWORD,DWORD,DWORD);
03513 BOOL WINAPI PaintDesktop(HDC);
03514 BOOL WINAPI PostThreadMessageA(DWORD,UINT,WPARAM,LPARAM);
03515 BOOL WINAPI PostThreadMessageW(DWORD,UINT,WPARAM,LPARAM);
03516 #define PostThreadMessage WINELIB_NAME_AW(PostThreadMessage)
03517 BOOL WINAPI RegisterHotKey(HWND,INT,UINT,UINT);
03518 HDEVNOTIFY WINAPI RegisterDeviceNotificationA(HANDLE,LPVOID,DWORD);
03519 HDEVNOTIFY WINAPI RegisterDeviceNotificationW(HANDLE,LPVOID,DWORD);
03520 #define RegisterDeviceNotification WINELIB_NAME_AW(RegisterDeviceNotification)
03521 BOOL WINAPI SendMessageCallbackA(HWND,UINT,WPARAM,LPARAM,SENDASYNCPROC,ULONG_PTR);
03522 BOOL WINAPI SendMessageCallbackW(HWND,UINT,WPARAM,LPARAM,SENDASYNCPROC,ULONG_PTR);
03523 #define SendMessageCallback WINELIB_NAME_AW(SendMessageCallback)
03524 BOOL WINAPI SendNotifyMessageA(HWND,UINT,WPARAM,LPARAM);
03525 BOOL WINAPI SendNotifyMessageW(HWND,UINT,WPARAM,LPARAM);
03526 #define SendNotifyMessage WINELIB_NAME_AW(SendNotifyMessage)
03527 VOID WINAPI SetDebugErrorLevel(DWORD);
03528 VOID WINAPI SetLastErrorEx(DWORD,DWORD);
03529 BOOL WINAPI SetMenuDefaultItem(HMENU,UINT,UINT);
03530 BOOL WINAPI SetMenuInfo(HMENU,LPMENUINFO);
03531 BOOL WINAPI SetMenuItemInfoA(HMENU,UINT,BOOL, const MENUTITEMINFOA*);
03532 BOOL WINAPI SetMenuItemInfoW(HMENU,UINT,BOOL, const MENUTITEMINFOW*);
03533 #define SetMenuItemInfo WINELIB_NAME_AW(SetMenuItemInfo)
03534 BOOL WINAPI SetWindowContextHelpId(HWND,DWORD);
03535 WORD WINAPI TileWindows(HWND,UINT, const LPRECT,UINT, const HWND *);
03536
03537 INT WINAPI ToUnicode(UINT,UINT,PBYTE,LPWSTR,int,UINT);
03538 BOOL WINAPI TrackPopupMenuEx(HMENU,UINT,INT,INT,HWND,

```

```

03539                                     LPTPM_PARAMS);
03540 BOOL      WINAPI UnregisterDeviceNotification(HDEVNOTIFY);
03541 BOOL      WINAPI UnregisterHotKey(HWND, INT);
03542 DWORD     WINAPI WaitForInputIdle(HANDLE, DWORD);
03543 VOID      WINAPI keybd_event(BYTE, BYTE, DWORD, DWORD);
03544 VOID      WINAPI mouse_event(DWORD, DWORD, DWORD, DWORD, DWORD);
03545
03546 /* Declarations for functions that are the same in Win16 and Win32 */
03547 VOID      WINAPI EndMenu(void);
03548 DWORD     WINAPI GetDialogBaseUnits(void);
03549 BOOL      WINAPI GetKeyboardState(LPBYTE);
03550 DWORD     WINAPI GetMenuCheckMarkDimensions(void);
03551 LONG      WINAPI GetMessageExtraInfo(void);
03552 DWORD     WINAPI GetMessagePos(void);
03553 LONG      WINAPI GetMessageTime(void);
03554 DWORD     WINAPI OemKeyScan(WORD);
03555 BOOL      WINAPI ReleaseCapture(void);
03556 BOOL      WINAPI SetKeyboardState(LPBYTE);
03557
03558 /* Declarations for functions that change between Win16 and Win32 */
03559
03560 BOOL      WINAPI AdjustWindowRect(LPRECT, DWORD, BOOL);
03561 BOOL      WINAPI AdjustWindowRectEx(LPRECT, DWORD, BOOL, DWORD);
03562 BOOL      WINAPI AnimateWindow(HWND, DWORD, DWORD);
03563 #define     AnsiLowerA CharLowerA
03564 #define     AnsiLowerW CharLowerW
03565 #define     AnsiLower WINELIB_NAME_AW(AnsiLower)
03566 #define     AnsiLowerBuffA CharLowerBuffA
03567 #define     AnsiLowerBuffW CharLowerBuffW
03568 #define     AnsiLowerBuff WINELIB_NAME_AW(AnsiLowerBuff)
03569 #define     AnsiNextA CharNextA
03570 #define     AnsiNextW CharNextW
03571 #define     AnsiNext WINELIB_NAME_AW(AnsiNext)
03572 #define     AnsiPrevA CharPrevA
03573 #define     AnsiPrevW CharPrevW
03574 #define     AnsiPrev WINELIB_NAME_AW(AnsiPrev)
03575 #define     AnsiUpperA CharUpperA
03576 #define     AnsiUpperW CharUpperW
03577 #define     AnsiUpper WINELIB_NAME_AW(AnsiUpper)
03578 #define     AnsiUpperBuffA CharUpperBuffA
03579 #define     AnsiUpperBuffW CharUpperBuffW
03580 #define     AnsiUpperBuff WINELIB_NAME_AW(AnsiUpperBuff)
03581 BOOL      WINAPI AnyPopup(void);
03582 BOOL      WINAPI AppendMenuA(HMENU, UINT, UINT, LPCSTR);
03583 BOOL      WINAPI AppendMenuW(HMENU, UINT, UINT, LPCWSTR);
03584 #define     AppendMenu WINELIB_NAME_AW(AppendMenu)
03585 UINT      WINAPI ArrangeIconicWindows(HWND);
03586 HDWP      WINAPI BeginDeferWindowPos(INT);
03587 HDC        WINAPI BeginPaint(HWND, LPPAINTSTRUCT);
03588 BOOL      WINAPI BringWindowToTop(HWND);
03589 void       WINAPI CalcChildScroll(HWND, INT);
03590 BOOL      WINAPI CallMsgFilterA(LPMSG, INT);
03591 BOOL      WINAPI CallMsgFilterW(LPMSG, INT);
03592 #define     CallMsgFilter WINELIB_NAME_AW(CallMsgFilter)
03593 LRESULT    WINAPI CallNextHookEx(HHOOK, INT, WPARAM, LPARAM);
03594 LRESULT    WINAPI CallWindowProcA(WNDPROC, HWND, UINT, WPARAM, LPARAM);
03595 LRESULT    WINAPI CallWindowProcW(WNDPROC, HWND, UINT, WPARAM, LPARAM);
03596 #define     CallWindowProc WINELIB_NAME_AW(CallWindowProc)
03597 BOOL      WINAPI ChangeClipboardChain(HWND, HWND);
03598 BOOL      WINAPI ChangeMenuA(HMENU, UINT, LPCSTR, UINT, UINT);
03599 BOOL      WINAPI ChangeMenuW(HMENU, UINT, LPCWSTR, UINT, UINT);
03600 #define     ChangeMenu WINELIB_NAME_AW(ChangeMenu)
03601 LPSTR      WINAPI CharLowerA(LPSTR);
03602 LPWSTR     WINAPI CharLowerW(LPWSTR);
03603 #define     CharLower WINELIB_NAME_AW(CharLower)
03604 DWORD     WINAPI CharLowerBuffA(LPSTR, DWORD);
03605 DWORD     WINAPI CharLowerBuffW(LPWSTR, DWORD);
03606 #define     CharLowerBuff WINELIB_NAME_AW(CharLowerBuff)
03607 LPSTR      WINAPI CharNextA(LPCSTR);
03608 LPWSTR     WINAPI CharNextW(LPCWSTR);
03609 #define     CharNext WINELIB_NAME_AW(CharNext)
03610 LPSTR      WINAPI CharNextExA(WORD, LPCSTR, DWORD);
03611 /* no CharNextExW (doesn't make sense) */
03612 LPSTR      WINAPI CharPrevA(LPCSTR, LPCSTR);
03613 LPWSTR     WINAPI CharPrevW(LPCWSTR, LPCWSTR);
03614 #define     CharPrev WINELIB_NAME_AW(CharPrev)
03615 LPSTR      WINAPI CharPrevExA(WORD, LPCSTR, LPCSTR, DWORD);
03616 /* no CharPrevExW (doesn't make sense) */
03617 LPSTR      WINAPI CharUpperA(LPSTR);
03618 LPWSTR     WINAPI CharUpperW(LPWSTR);
03619 #define     CharUpper WINELIB_NAME_AW(CharUpper)
03620 DWORD     WINAPI CharUpperBuffA(LPSTR, DWORD);
03621 DWORD     WINAPI CharUpperBuffW(LPWSTR, DWORD);
03622 #define     CharUpperBuff WINELIB_NAME_AW(CharUpperBuff)
03623 BOOL      WINAPI CharToOemA(LPCSTR, LPSTR);
03624 BOOL      WINAPI CharToOemW(LPCWSTR, LPSTR);
03625 #define     CharToOem WINELIB_NAME_AW(CharToOem)

```

```

03626 BOOL        WINAPI CharToOemBuffA(LPCSTR,LPSTR,DWORD);
03627 BOOL        WINAPI CharToOemBuffW(LPCWSTR,LPSTR,DWORD);
03628 #define        CharToOemBuff WINELIB_NAME_AW(CharToOemBuff)
03629 BOOL        WINAPI CheckDlgButton(HWND,INT,UINT);
03630 DWORD        WINAPI CheckMenuItem(HMENU,UINT,UINT);
03631 BOOL        WINAPI CheckMenuRadioItem(HMENU,UINT,UINT,UINT,UINT);
03632 BOOL        WINAPI CheckRadioButton(HWND,UINT,UINT,UINT);
03633 HWND        WINAPI ChildWindowFromPoint(HWND,POINT);
03634 HWND        WINAPI ChildWindowFromPointEx(HWND,POINT,UINT);
03635 BOOL        WINAPI ClientToScreen(HWND,LPPOINT);
03636 BOOL        WINAPI ClipCursor(const RECT*);
03637 BOOL        WINAPI CloseClipboard(void);
03638 BOOL        WINAPI CloseDesktop(HDESK);
03639 BOOL        WINAPI CloseWindow(HWND);
03640 BOOL        WINAPI CloseWindowStation(HWINSTA);
03641 #define        CopyCursor(cur) ((HCURSOR)CopyIcon((HICON)(cur)))
03642 HICON        WINAPI CopyIcon(HICON);
03643 HICON        WINAPI CopyImage(HANDLE,UINT,INT,INT,UINT);
03644 BOOL        WINAPI CopyRect(RECT*,const RECT*);
03645 INT        WINAPI CountClipboardFormats(void);
03646 BOOL        WINAPI CreateCaret(HWND,HBITMAP,INT,INT);
03647 HCURSOR        WINAPI CreateCursor(HINSTANCE,INT,INT,INT,INT,LPCVOID,LPCVOID);
03648 #define        CreateDialogA(inst,ptr,hwnd,dlg) \
03649             CreateDialogParamA(inst,ptr,hwnd,dlg,0)
03650 #define        CreateDialogW(inst,ptr,hwnd,dlg) \
03651             CreateDialogParamW(inst,ptr,hwnd,dlg,0)
03652 #define        CreateDialog WINELIB_NAME_AW(CreateDialog)
03653 #define        CreateDialogIndirectA(inst,ptr,hwnd,dlg) \
03654             CreateDialogIndirectParamA(inst,ptr,hwnd,dlg,0)
03655 #define        CreateDialogIndirectW(inst,ptr,hwnd,dlg) \
03656             CreateDialogIndirectParamW(inst,ptr,hwnd,dlg,0)
03657 #define        CreateDialogIndirect WINELIB_NAME_AW(CreateDialogIndirect)
03658 HWND        WINAPI CreateDialogIndirectParamA(HINSTANCE,LPCVOID,HWND,
03659             DLGPROC,LPARAM);
03660 HWND        WINAPI CreateDialogIndirectParamW(HINSTANCE,LPCVOID,HWND,
03661             DLGPROC,LPARAM);
03662 #define        CreateDialogIndirectParam WINELIB_NAME_AW(CreateDialogIndirectParam)
03663 HWND        WINAPI CreateDialogParamA(HINSTANCE,LPCSTR,HWND,DLGPROC,LPARAM);
03664 HWND        WINAPI CreateDialogParamW(HINSTANCE,LPCWSTR,HWND,DLGPROC,LPARAM);
03665 #define        CreateDialogParam WINELIB_NAME_AW(CreateDialogParam)
03666 HICON        WINAPI CreateIcon(HINSTANCE,INT,INT,BYTE,BYTE,LPCVOID,LPCVOID);
03667 HICON        WINAPI CreateIconFromResource(LPBYTE,UINT,BOOL,DWORD);
03668 HICON        WINAPI CreateIconFromResourceEx(LPBYTE,UINT,BOOL,DWORD,INT,INT,UINT);
03669 HMENU        WINAPI CreateMenu(void);
03670 HMENU        WINAPI CreatePopupMenu(void);
03671 #define        CreateWindowA(className,titleName,style,x,y,width,height,\
03672             parent,menu,instance,param) \
03673             CreateWindowExA(0,className,titleName,style,x,y,width,height,\
03674             parent,menu,instance,param)
03675 #define        CreateWindowW(className,titleName,style,x,y,width,height,\
03676             parent,menu,instance,param) \
03677             CreateWindowExW(0,className,titleName,style,x,y,width,height,\
03678             parent,menu,instance,param)
03679 #define        CreateWindow WINELIB_NAME_AW(CreateWindow)
03680 HWND        WINAPI CreateWindowExA(DWORD,LPCSTR,LPCSTR,DWORD,INT,INT,
03681             INT,INT,HWND,HMENU,HINSTANCE,LPVOID);
03682 HWND        WINAPI CreateWindowExW(DWORD,LPCWSTR,LPCWSTR,DWORD,INT,INT,
03683             INT,INT,HWND,HMENU,HINSTANCE,LPVOID);
03684 #define        CreateWindowEx WINELIB_NAME_AW(CreateWindowEx)
03685 HWINSTA        WINAPI CreateWindowStationA(LPSTR,DWORD,DWORD,LPSECURITY_ATTRIBUTES);
03686 HWINSTA        WINAPI CreateWindowStationW(LPWSTR,DWORD,DWORD,LPSECURITY_ATTRIBUTES);
03687 #define        CreateWindowStation WINELIB_NAME_AW(CreateWindowStation)
03688 HWND        WINAPI CreateMDIWindowA(LPCSTR,LPCSTR,DWORD,INT,INT,
03689             INT,INT,HWND,HINSTANCE,LPARAM);
03690 HWND        WINAPI CreateMDIWindowW(LPCWSTR,LPCWSTR,DWORD,INT,INT,
03691             INT,INT,HWND,HINSTANCE,LPARAM);
03692 #define        CreateMDIWindow WINELIB_NAME_AW(CreateMDIWindow)
03693 LRESULT        WINAPI DefDlgProcA(HWND,UINT,WPARAM,LPARAM);
03694 LRESULT        WINAPI DefDlgProcW(HWND,UINT,WPARAM,LPARAM);
03695 #define        DefDlgProc WINELIB_NAME_AW(DefDlgProc)
03696 HDWP        WINAPI DeferWindowPos(HDWP,HWND,HWND,INT,INT,INT,UINT);
03697 LRESULT        WINAPI DefFrameProcA(HWND,HWND,UINT,WPARAM,LPARAM);
03698 LRESULT        WINAPI DefFrameProcW(HWND,HWND,UINT,WPARAM,LPARAM);
03699 #define        DefFrameProc WINELIB_NAME_AW(DefFrameProc)
03700 #define        DefHookProc(code,wparam,lparam,phook) \
03701             CallNextHookEx(*(phook),code,wparam,lparam)
03702 LRESULT        WINAPI DefMDIChildProcA(HWND,UINT,WPARAM,LPARAM);
03703 LRESULT        WINAPI DefMDIChildProcW(HWND,UINT,WPARAM,LPARAM);
03704 #define        DefMDIChildProc WINELIB_NAME_AW(DefMDIChildProc)
03705 LRESULT        WINAPI DefWindowProcA(HWND,UINT,WPARAM,LPARAM);
03706 LRESULT        WINAPI DefWindowProcW(HWND,UINT,WPARAM,LPARAM);
03707 #define        DefWindowProc WINELIB_NAME_AW(DefWindowProc)
03708 BOOL        WINAPI DeleteMenu(HMENU,UINT,UINT);
03709 BOOL        WINAPI DestroyCaret(void);
03710 BOOL        WINAPI DestroyCursor(HCURSOR);
03711 BOOL        WINAPI DestroyIcon(HICON);
03712 BOOL        WINAPI DestroyMenu(HMENU);

```

```

03713 BOOL        WINAPI DestroyWindow (HWND);
03714 #define        DialogBoxA (inst, template, owner, func) \
03715               DialogBoxParamA (inst, template, owner, func, 0)
03716 #define        DialogBoxW (inst, template, owner, func) \
03717               DialogBoxParamW (inst, template, owner, func, 0)
03718 #define        DialogBox WINELIB_NAME_AW (DialogBox)
03719 #define        DialogBoxIndirectA (inst, template, owner, func) \
03720               DialogBoxIndirectParamA (inst, template, owner, func, 0)
03721 #define        DialogBoxIndirectW (inst, template, owner, func) \
03722               DialogBoxIndirectParamW (inst, template, owner, func, 0)
03723 #define        DialogBoxIndirect WINELIB_NAME_AW (DialogBoxIndirect)
03724 INT WINAPI DialogBoxIndirectParamA (HINSTANCE, LPCVOID, HWND, DLGPROC, LPARAM);
03725 INT WINAPI DialogBoxIndirectParamW (HINSTANCE, LPCVOID, HWND, DLGPROC, LPARAM);
03726 #define        DialogBoxIndirectParam WINELIB_NAME_AW (DialogBoxIndirectParam)
03727 INT WINAPI DialogBoxParamA (HINSTANCE, LPCSTR, HWND, DLGPROC, LPARAM);
03728 INT WINAPI DialogBoxParamW (HINSTANCE, LPCWSTR, HWND, DLGPROC, LPARAM);
03729 #define        DialogBoxParam WINELIB_NAME_AW (DialogBoxParam)
03730 LONG WINAPI DispatchMessageA (const MSG*);
03731 LONG WINAPI DispatchMessageW (const MSG*);
03732 #define        DispatchMessage WINELIB_NAME_AW (DispatchMessage)
03733 INT WINAPI DlgDirListA (HWND, LPSTR, INT, INT, UINT);
03734 INT WINAPI DlgDirListW (HWND, LPWSTR, INT, INT, UINT);
03735 #define        DlgDirList WINELIB_NAME_AW (DlgDirList)
03736 INT WINAPI DlgDirListComboBoxA (HWND, LPSTR, INT, INT, UINT);
03737 INT WINAPI DlgDirListComboBoxW (HWND, LPWSTR, INT, INT, UINT);
03738 #define        DlgDirListComboBox WINELIB_NAME_AW (DlgDirListComboBox)
03739 BOOL WINAPI DlgDirSelectComboBoxExA (HWND, LPSTR, INT, INT);
03740 BOOL WINAPI DlgDirSelectComboBoxExW (HWND, LPWSTR, INT, INT);
03741 #define        DlgDirSelectComboBoxEx WINELIB_NAME_AW (DlgDirSelectComboBoxEx)
03742 BOOL WINAPI DlgDirSelectExA (HWND, LPSTR, INT, INT);
03743 BOOL WINAPI DlgDirSelectExW (HWND, LPWSTR, INT, INT);
03744 #define        DlgDirSelectEx WINELIB_NAME_AW (DlgDirSelectEx)
03745 BOOL WINAPI DragDetect (HWND, POINT);
03746 DWORD WINAPI DragObject (HWND, HWND, UINT, DWORD, HCURSOR);
03747 BOOL WINAPI DrawAnimatedRects (HWND, int, const RECT*, const RECT*);
03748 BOOL WINAPI DrawCaption (HWND, HDC, const RECT*, UINT);
03749 BOOL WINAPI DrawCaptionTempA (HWND, HDC, const RECT*, HFONT, HICON, LPCSTR, UINT);
03750 BOOL WINAPI DrawCaptionTempW (HWND, HDC, const RECT*, HFONT, HICON, LPCWSTR, UINT);
03751 #define        DrawCaptionTemp WINELIB_NAME_AW (DrawCaptionTemp)
03752 BOOL WINAPI DrawEdge (HDC, LPRECT, UINT, UINT);
03753 BOOL WINAPI DrawFocusRect (HDC, const RECT*);
03754 BOOL WINAPI DrawFrameControl (HDC, LPRECT, UINT, UINT);
03755 BOOL WINAPI DrawIcon (HDC, INT, INT, HICON);
03756 BOOL WINAPI DrawIconEx (HDC, INT, INT, HICON, INT, INT, UINT, HBRUSH, UINT);
03757 BOOL WINAPI DrawMenuBar (HWND);
03758 BOOL WINAPI DrawStateA (HDC, HBRUSH, DRAWSTATEPROC, LPARAM, WPARAM, INT, INT, INT, INT, INT);
03759 BOOL WINAPI DrawStateW (HDC, HBRUSH, DRAWSTATEPROC, LPARAM, WPARAM, INT, INT, INT, INT, INT);
03760 #define        DrawState WINELIB_NAME_AW (DrawState)
03761 INT WINAPI DrawTextA (HDC, LPCSTR, INT, LPRECT, UINT);
03762 INT WINAPI DrawTextW (HDC, LPCWSTR, INT, LPRECT, UINT);
03763 #define        DrawText WINELIB_NAME_AW (DrawText)
03764 INT WINAPI DrawTextExA (HDC, LPSTR, INT, LPRECT, UINT, LPDRAWTEXTPARAMS);
03765 INT WINAPI DrawTextExW (HDC, LPWSTR, INT, LPRECT, UINT, LPDRAWTEXTPARAMS);
03766 #define        DrawTextEx WINELIB_NAME_AW (DrawTextEx)
03767 BOOL WINAPI EmptyClipboard (void);
03768 UINT WINAPI EnableMenuItem (HMENU, UINT, UINT);
03769 BOOL WINAPI EnableScrollBar (HWND, INT, UINT);
03770 BOOL WINAPI EnableWindow (HWND, BOOL);
03771 BOOL WINAPI EndDeferWindowPos (HDWP);
03772 BOOL WINAPI EndDialog (HWND, INT);
03773 BOOL WINAPI EndPaint (HWND, const PAINTSTRUCT*);
03774 BOOL WINAPI EnumChildWindows (HWND, WNDENUMPROC, LPARAM);
03775 UINT WINAPI EnumClipboardFormats (UINT);
03776 INT WINAPI EnumPropsA (HWND, PROPENUMPROCA);
03777 INT WINAPI EnumPropsW (HWND, PROPENUMPROCW);
03778 #define        EnumProps WINELIB_NAME_AW (EnumProps)
03779 BOOL WINAPI EnumWindows (WNDENUMPROC, LPARAM);
03780 BOOL WINAPI EnumWindowStationsA (WINSTAENUMPROCA, LPARAM);
03781 BOOL WINAPI EnumWindowStationsW (WINSTAENUMPROCW, LPARAM);
03782 #define        EnumWindowStations WINELIB_NAME_AW (EnumWindowStations)
03783 BOOL WINAPI EqualRect (const RECT*, const RECT*);
03784 INT WINAPI ExcludeUpdateRgn (HDC, HWND);
03785 #define        ExitWindows (a,b) ExitWindowsEx (EWX_LOGOFF, 0xffffffff)
03786 INT WINAPI FillRect (HDC, const RECT*, HBRUSH);
03787 HWND WINAPI FindWindowA (LPCSTR, LPCSTR);
03788 HWND WINAPI FindWindowW (LPCWSTR, LPCWSTR);
03789 #define        FindWindow WINELIB_NAME_AW (FindWindow)
03790 HWND WINAPI FindWindowExA (HWND, HWND, LPCSTR, LPCSTR);
03791 HWND WINAPI FindWindowExW (HWND, HWND, LPCWSTR, LPCWSTR);
03792 #define        FindWindowEx WINELIB_NAME_AW (FindWindowEx)
03793 BOOL WINAPI FlashWindow (HWND, BOOL);
03794 INT WINAPI FrameRect (HDC, const RECT*, HBRUSH);
03795 HWND WINAPI GetActiveWindow (void);
03796 HWND WINAPI GetAncestor (HWND, UINT);
03797 DWORD WINAPI GetAppCompatFlags (HTASK);
03798 WORD WINAPI GetAsyncKeyState (INT);
03799 HWND WINAPI GetCapture (void);

```

```

03800 UINT          WINAPI GetCaretBlinkTime(void);
03801 BOOL           WINAPI GetCaretPos(LPPOINT);
03802 BOOL           WINAPI GetClassInfoA(HINSTANCE, LPCSTR, WNDCLASSA *);
03803 BOOL           WINAPI GetClassInfoW(HINSTANCE, LPCWSTR, WNDCLASSW *);
03804 #define         GetClassInfo WINELIB_NAME_AW(GetClassInfo)
03805 BOOL           WINAPI GetClassInfoExA(HINSTANCE, LPCSTR, WNDCLASSEXA *);
03806 BOOL           WINAPI GetClassInfoExW(HINSTANCE, LPCWSTR, WNDCLASSEXW *);
03807 #define         GetClassInfoEx WINELIB_NAME_AW(GetClassInfoEx)
03808 LONG            WINAPI GetClassLongA(HWND, INT);
03809 LONG            WINAPI GetClassLongW(HWND, INT);
03810 #define         GetClassLong WINELIB_NAME_AW(GetClassLong)
03811 INT             WINAPI GetClassNameA(HWND, LPSTR, INT);
03812 INT             WINAPI GetClassNameW(HWND, LPWSTR, INT);
03813 #define         GetClassName WINELIB_NAME_AW(GetClassName)
03814 WORD            WINAPI GetClassWord(HWND, INT);
03815 BOOL           WINAPI GetClientRect(HWND, LPRECT);
03816 HANDLE         WINAPI GetClipboardData(UINT);
03817 INT            WINAPI GetClipboardFormatNameA(UINT, LPSTR, INT);
03818 INT            WINAPI GetClipboardFormatNameW(UINT, LPWSTR, INT);
03819 #define         GetClipboardFormatName WINELIB_NAME_AW(GetClipboardFormatName)
03820 HWND           WINAPI GetClipboardOwner(void);
03821 HWND           WINAPI GetClipboardViewer(void);
03822 BOOL           WINAPI GetClipCursor(LPRECT);
03823 HCURSOR        WINAPI GetCursor(void);
03824 BOOL           WINAPI GetCursorPos(LPPOINT);
03825 HDC            WINAPI GetDC(HWND);
03826 HDC            WINAPI GetDCEX(HWND, HRGN, DWORD);
03827 HWND           WINAPI GetDesktopWindow(void);
03828 INT            WINAPI GetDlgCtrlID(HWND);
03829 HWND           WINAPI GetDlgItem(HWND, INT);
03830 UINT           WINAPI GetDlgItemInt(HWND, INT, BOOL*, BOOL);
03831 INT            WINAPI GetDlgItemTextA(HWND, INT, LPSTR, UINT);
03832 INT            WINAPI GetDlgItemTextW(HWND, INT, LPWSTR, UINT);
03833 #define         GetDlgItemText WINELIB_NAME_AW(GetDlgItemText)
03834 UINT           WINAPI GetDoubleClickTime(void);
03835 HWND           WINAPI GetFocus(void);
03836 HWND           WINAPI GetForegroundWindow(void);
03837 BOOL           WINAPI GetInputState(void);
03838 UINT           WINAPI GetInternalWindowPos(HWND, LPRECT, LPPOINT);
03839 UINT           WINAPI GetKBCodePage(void);
03840 INT            WINAPI GetKeyboardType(INT);
03841 INT            WINAPI GetKeyNameTextA(LONG, LPSTR, INT);
03842 INT            WINAPI GetKeyNameTextW(LONG, LPWSTR, INT);
03843 #define         GetKeyNameText WINELIB_NAME_AW(GetKeyNameText)
03844 INT            WINAPI GetKeyboardLayoutNameA(LPSTR);
03845 INT            WINAPI GetKeyboardLayoutNameW(LPWSTR);
03846 #define         GetKeyboardLayoutName WINELIB_NAME_AW(GetKeyboardLayoutName)
03847 SHORT          WINAPI GetKeyState(INT);
03848 HWND           WINAPI GetLastActivePopup(HWND);
03849 HMENU           WINAPI GetMenu(HWND);
03850 INT            WINAPI GetMenuItemCount(HMENU);
03851 UINT           WINAPI GetMenuItemID(HMENU, INT);
03852 BOOL           WINAPI GetMenuItemRect(HWND, HMENU, UINT, LPRECT);
03853 UINT           WINAPI GetMenuState(HMENU, UINT, UINT);
03854 INT            WINAPI GetMenuStringA(HMENU, UINT, LPSTR, INT, UINT);
03855 INT            WINAPI GetMenuStringW(HMENU, UINT, LPWSTR, INT, UINT);
03856 #define         GetMenuString WINELIB_NAME_AW(GetMenuString)
03857 BOOL           WINAPI GetMessageA(LPMSG, HWND, UINT, UINT);
03858 BOOL           WINAPI GetMessageW(LPMSG, HWND, UINT, UINT);
03859 #define         GetMessage WINELIB_NAME_AW(GetMessage)
03860 HWND           WINAPI GetNextDlgGroupItem(HWND, HWND, BOOL);
03861 HWND           WINAPI GetNextDlgTabItem(HWND, HWND, BOOL);
03862 #define         GetNextWindow GetWindow
03863 HWND           WINAPI GetOpenClipboardWindow(void);
03864 HWND           WINAPI GetParent(HWND);
03865 INT            WINAPI GetPriorityClipboardFormat(UINT*, INT);
03866 BOOL           WINAPI GetProcessDefaultLayout(DWORD*);
03867 HANDLE         WINAPI GetPropA(HWND, LPCSTR);
03868 HANDLE         WINAPI GetPropW(HWND, LPCWSTR);
03869 #define         GetProp WINELIB_NAME_AW(GetProp)
03870 DWORD          WINAPI GetQueueStatus(UINT);
03871 BOOL           WINAPI GetScrollInfo(HWND, INT, LPSCROLLINFO);
03872 INT            WINAPI GetScrollPos(HWND, INT);
03873 BOOL           WINAPI GetScrollRange(HWND, INT, LPINT, LPINT);
03874 HWND           WINAPI GetShellWindow(void);
03875 HMENU           WINAPI GetSubMenu(HMENU, INT);
03876 HBRUSH          WINAPI GetSysColorBrush(INT);
03877 #define         GetSysModalWindow() ((HWND)0)
03878 HMENU           WINAPI GetSystemMenu(HWND, BOOL);
03879 INT            WINAPI GetSystemMetrics(INT);
03880 DWORD          WINAPI GetTabbedTextExtentA(HDC, LPCSTR, INT, INT, const INT*);
03881 DWORD          WINAPI GetTabbedTextExtentW(HDC, LPCWSTR, INT, INT, const INT*);
03882 #define         GetTabbedTextExtent WINELIB_NAME_AW(GetTabbedTextExtent)
03883 HWND           WINAPI GetTopWindow(HWND);
03884 BOOL           WINAPI GetUpdateRect(HWND, LPRECT, BOOL);
03885 INT            WINAPI GetUpdateRgn(HWND, HRGN, BOOL);
03886 BOOL           WINAPI GetUserObjectInformationA(HANDLE, INT, LPVOID, DWORD, LPDWORD);

```

```
03887 BOOL        WINAPI GetUserObjectInformationW (HANDLE, INT, LPVOID, DWORD, LPDWORD);
03888 #define        GetUserObjectInformation WINELIB_NAME_AW (GetUserObjectInformation)
03889 HWND        WINAPI GetWindow (HWND, UINT);
03890 HDC        WINAPI GetWindowDC (HWND);
03891 LONG        WINAPI GetWindowLongA (HWND, INT);
03892 LONG        WINAPI GetWindowLongW (HWND, INT);
03893 #define        GetWindowLong WINELIB_NAME_AW (GetWindowLong)
03894 BOOL        WINAPI GetWindowPlacement (HWND, LPWINDOWPLACEMENT);
03895 BOOL        WINAPI GetWindowRect (HWND, LPRECT);
03896 INT        WINAPI GetWindowRgn (HWND, HRGN);
03897 HWINSTA     WINAPI GetProcessWindowStation (void);
03898 #define        GetWindowTask (hwnd) ((HTASK)GetWindowThreadProcessId (hwnd, NULL))
03899 INT        WINAPI GetWindowTextA (HWND, LPSTR, INT);
03900 INT        WINAPI GetWindowTextW (HWND, LPWSTR, INT);
03901 #define        GetWindowText WINELIB_NAME_AW (GetWindowText)
03902 INT        WINAPI GetWindowTextLengthA (HWND);
03903 INT        WINAPI GetWindowTextLengthW (HWND);
03904 #define        GetWindowTextLength WINELIB_NAME_AW (GetWindowTextLength)
03905 WORD        WINAPI GetWindowWord (HWND, INT);
03906 BOOL        WINAPI GrayStringA (HDC, HBRUSH, GRAYSTRINGPROC, LPARAM,
03907                                INT, INT, INT, INT, INT);
03908 BOOL        WINAPI GrayStringW (HDC, HBRUSH, GRAYSTRINGPROC, LPARAM,
03909                                INT, INT, INT, INT, INT);
03910 #define        GrayString WINELIB_NAME_AW (GrayString)
03911 BOOL        WINAPI HideCaret (HWND);
03912 BOOL        WINAPI HiliteMenuItem (HWND, HMENU, UINT, UINT);
03913 BOOL        WINAPI InflateRect (LPRECT, INT, INT);
03914 BOOL        WINAPI InSendMessage (void);
03915 DWORD       WINAPI InSendMessageEx (LPVOID);
03916 BOOL        WINAPI InsertMenuA (HMENU, UINT, UINT, LPCSTR);
03917 BOOL        WINAPI InsertMenuW (HMENU, UINT, UINT, LPCWSTR);
03918 #define        InsertMenu WINELIB_NAME_AW (InsertMenu)
03919 BOOL        WINAPI InsertMenuItemA (HMENU, UINT, BOOL, const MENUITEMINFO*);
03920 BOOL        WINAPI InsertMenuItemW (HMENU, UINT, BOOL, const MENUITEMINFOW*);
03921 #define        InsertMenuItem WINELIB_NAME_AW (InsertMenuItem)
03922 INT        WINAPI InternalGetWindowText (HWND, LPWSTR, INT);
03923 BOOL        WINAPI IntersectRect (LPRECT, const RECT*, const RECT*);
03924 BOOL        WINAPI InvalidateRect (HWND, const RECT*, BOOL);
03925 BOOL        WINAPI InvalidateRgn (HWND, HRGN, BOOL);
03926 BOOL        WINAPI InvertRect (HDC, const RECT*);
03927 BOOL        WINAPI IsCharAlphaA (CHAR);
03928 BOOL        WINAPI IsCharAlphaW (WCHAR);
03929 #define        IsCharAlpha WINELIB_NAME_AW (IsCharAlpha)
03930 BOOL        WINAPI IsCharAlphaNumericA (CHAR);
03931 BOOL        WINAPI IsCharAlphaNumericW (WCHAR);
03932 #define        IsCharAlphaNumeric WINELIB_NAME_AW (IsCharAlphaNumeric)
03933 BOOL        WINAPI IsCharLowerA (CHAR);
03934 BOOL        WINAPI IsCharLowerW (WCHAR);
03935 #define        IsCharLower WINELIB_NAME_AW (IsCharLower)
03936 BOOL        WINAPI IsCharUpperA (CHAR);
03937 BOOL        WINAPI IsCharUpperW (WCHAR);
03938 #define        IsCharUpper WINELIB_NAME_AW (IsCharUpper)
03939 BOOL        WINAPI IsChild (HWND, HWND);
03940 BOOL        WINAPI IsClipboardFormatAvailable (UINT);
03941 BOOL        WINAPI IsDialogMessageA (HWND, LPMSG);
03942 BOOL        WINAPI IsDialogMessageW (HWND, LPMSG);
03943 #define        IsDialogMessage WINELIB_NAME_AW (IsDialogMessage)
03944 UINT        WINAPI IsDlgButtonChecked (HWND, UINT);
03945 BOOL        WINAPI IsIconic (HWND);
03946 BOOL        WINAPI IsMenu (HMENU);
03947 BOOL        WINAPI IsRectEmpty (const RECT*);
03948 BOOL        WINAPI IsWindow (HWND);
03949 BOOL        WINAPI IsWindowEnabled (HWND);
03950 BOOL        WINAPI IsWindowVisible (HWND);
03951 BOOL        WINAPI IsZoomed (HWND);
03952 BOOL        WINAPI KillSystemTimer (HWND, UINT);
03953 BOOL        WINAPI KillTimer (HWND, UINT);
03954 HACCEL      WINAPI LoadAcceleratorsA (HINSTANCE, LPCSTR);
03955 HACCEL      WINAPI LoadAcceleratorsW (HINSTANCE, LPCWSTR);
03956 #define        LoadAccelerators WINELIB_NAME_AW (LoadAccelerators)
03957 HBITMAP     WINAPI LoadBitmapA (HINSTANCE, LPCSTR);
03958 HBITMAP     WINAPI LoadBitmapW (HINSTANCE, LPCWSTR);
03959 #define        LoadBitmap WINELIB_NAME_AW (LoadBitmap)
03960 HCURSOR     WINAPI LoadCursorA (HINSTANCE, LPCSTR);
03961 HCURSOR     WINAPI LoadCursorW (HINSTANCE, LPCWSTR);
03962 #define        LoadCursor WINELIB_NAME_AW (LoadCursor)
03963 HCURSOR     WINAPI LoadCursorFromFileA (LPCSTR);
03964 HCURSOR     WINAPI LoadCursorFromFileW (LPCWSTR);
03965 #define        LoadCursorFromFile WINELIB_NAME_AW (LoadCursorFromFile)
03966 HICON       WINAPI LoadIconA (HINSTANCE, LPCSTR);
03967 HICON       WINAPI LoadIconW (HINSTANCE, LPCWSTR);
03968 #define        LoadIcon WINELIB_NAME_AW (LoadIcon)
03969 HANDLE      WINAPI LoadImageA (HINSTANCE, LPCSTR, UINT, INT, INT, UINT);
03970 HANDLE      WINAPI LoadImageW (HINSTANCE, LPCWSTR, UINT, INT, INT, UINT);
03971 #define        LoadImage WINELIB_NAME_AW (LoadImage)
03972 HMENU       WINAPI LoadMenuA (HINSTANCE, LPCSTR);
03973 HMENU       WINAPI LoadMenuW (HINSTANCE, LPCWSTR);
```

```

03974 #define      LoadMenu WINELIB_NAME_AW(LoadMenu)
03975 HMENU      WINAPI LoadMenuIndirectA(LPCVOID);
03976 HMENU      WINAPI LoadMenuIndirectW(LPCVOID);
03977 #define      LoadMenuIndirect WINELIB_NAME_AW(LoadMenuIndirect)
03978 INT         WINAPI LoadStringA(HINSTANCE,UINT,LPSTR,INT);
03979 INT         WINAPI LoadStringW(HINSTANCE,UINT,LPWSTR,INT);
03980 #define      LoadString WINELIB_NAME_AW(LoadString)
03981 BOOL        WINAPI LockWindowUpdate(HWND);
03982 INT         WINAPI LookupIconIdFromDirectory(LPBYTE,BOOL);
03983 INT         WINAPI LookupIconIdFromDirectoryEx(LPBYTE,BOOL,INT,INT,UINT);
03984 UINT        WINAPI MapVirtualKeyA(UINT,UINT);
03985 UINT        WINAPI MapVirtualKeyW(UINT,UINT);
03986 #define      MapVirtualKey WINELIB_NAME_AW(MapVirtualKey)
03987 UINT        WINAPI MapVirtualKeyExA(UINT,UINT,HKL);
03988 UINT        WINAPI MapVirtualKeyExW(UINT,UINT,HKL);
03989 #define      MapVirtualKeyEx WINELIB_NAME_AW(MapVirtualKeyEx)
03990 BOOL        WINAPI MapDialogRect(HWND,LPRECT);
03991 INT         WINAPI MapWindowPoints(HWND,HWND,LPOINT,UINT);
03992 UINT        WINAPI MenuItemFromPoint(HWND,HMENU,POINT);
03993 BOOL        WINAPI MessageBeep(UINT);
03994 INT         WINAPI MessageBoxA(HWND,LPCSTR,LPCSTR,UINT);
03995 INT         WINAPI MessageBoxW(HWND,LPCWSTR,LPCWSTR,UINT);
03996 #define      MessageBox WINELIB_NAME_AW(MessageBox)
03997 INT         WINAPI MessageBoxIndirectA(LPMSGBOXPARAMSA);
03998 INT         WINAPI MessageBoxIndirectW(LPMSGBOXPARAMSW);
03999 #define      MessageBoxIndirect WINELIB_NAME_AW(MessageBoxIndirect)
04000 BOOL        WINAPI ModifyMenuA(HMENU,UINT,UINT,UINT,LPCSTR);
04001 BOOL        WINAPI ModifyMenuW(HMENU,UINT,UINT,UINT,LPCWSTR);
04002 #define      ModifyMenu WINELIB_NAME_AW(ModifyMenu)
04003 BOOL        WINAPI MoveWindow(HWND,INT,INT,INT,BOOL);
04004 BOOL        WINAPI OemToCharA(LPCSTR,LPSTR);
04005 BOOL        WINAPI OemToCharW(LPCSTR,LPWSTR);
04006 #define      OemToChar WINELIB_NAME_AW(OemToChar)
04007 BOOL        WINAPI OemToCharBuffA(LPCSTR,LPSTR,DWORD);
04008 BOOL        WINAPI OemToCharBuffW(LPCSTR,LPWSTR,DWORD);
04009 #define      OemToCharBuff WINELIB_NAME_AW(OemToCharBuff)
04010 BOOL        WINAPI OffsetRect(LPRECT,INT,INT);
04011 BOOL        WINAPI OpenClipboard(HWND);
04012 BOOL        WINAPI OpenIcon(HWND);
04013 HWINSTA     WINAPI OpenWindowStationA(LPSTR,BOOL,ACCESS_MASK);
04014 HWINSTA     WINAPI OpenWindowStationW(LPWSTR,BOOL,ACCESS_MASK);
04015 #define      OpenWindowStation WINELIB_NAME_AW(OpenWindowStation)
04016 BOOL        WINAPI PeekMessageA(LPMSG,HWND,UINT,UINT,UINT);
04017 BOOL        WINAPI PeekMessageW(LPMSG,HWND,UINT,UINT,UINT);
04018 #define      PeekMessage WINELIB_NAME_AW(PeekMessage)
04019 #define      PostAppMessageA(thread,msg,wparam,lparam) \
04020             PostThreadMessageA((DWORD)(thread),msg,wparam,lparam)
04021 #define      PostAppMessageW(thread,msg,wparam,lparam) \
04022             PostThreadMessageW((DWORD)(thread),msg,wparam,lparam)
04023 #define      PostAppMessage WINELIB_NAME_AW(PostAppMessage)
04024 BOOL        WINAPI PostMessageA(HWND,UINT,WPARAM,LPARAM);
04025 BOOL        WINAPI PostMessageW(HWND,UINT,WPARAM,LPARAM);
04026 #define      PostMessage WINELIB_NAME_AW(PostMessage)
04027 void        WINAPI PostQuitMessage(INT);
04028 BOOL        WINAPI PtInRect(const RECT*,POINT);
04029 BOOL        WINAPI RedrawWindow(HWND,const RECT*,HRGN,UINT);
04030 ATOM        WINAPI RegisterClassA(const WNDCLASSA *);
04031 ATOM        WINAPI RegisterClassW(const WNDCLASSW *);
04032 #define      RegisterClass WINELIB_NAME_AW(RegisterClass)
04033 ATOM        WINAPI RegisterClassExA(const WNDCLASSEXA *);
04034 ATOM        WINAPI RegisterClassExW(const WNDCLASSEXW *);
04035 #define      RegisterClassEx WINELIB_NAME_AW(RegisterClassEx)
04036 UINT        WINAPI RegisterClipboardFormatA(LPCSTR);
04037 UINT        WINAPI RegisterClipboardFormatW(LPCWSTR);
04038 #define      RegisterClipboardFormat WINELIB_NAME_AW(RegisterClipboardFormat)
04039 WORD        WINAPI RegisterWindowMessageA(LPCSTR);
04040 WORD        WINAPI RegisterWindowMessageW(LPCWSTR);
04041 #define      RegisterWindowMessage WINELIB_NAME_AW(RegisterWindowMessage)
04042 INT         WINAPI ReleasedDC(HWND,HDC);
04043 BOOL        WINAPI RemoveMenu(HMENU,UINT,UINT);
04044 HANDLE      WINAPI RemovePropA(HWND,LPCSTR);
04045 HANDLE      WINAPI RemovePropW(HWND,LPCWSTR);
04046 #define      RemoveProp WINELIB_NAME_AW(RemoveProp)
04047 BOOL        WINAPI ReplyMessage(LRESULT);
04048 BOOL        WINAPI ScreenToClient(HWND,LPOINT);
04049 VOID        WINAPI ScrollChildren(HWND,UINT,WPARAM,LPARAM);
04050 BOOL        WINAPI ScrollDC(HDC,INT,INT,const RECT*,const RECT*,HRGN,LPRECT);
04051 BOOL        WINAPI ScrollWindow(HWND,INT,INT,const RECT*,const RECT*);
04052 INT         WINAPI ScrollWindowEx(HWND,INT,INT,const RECT*,const RECT*,HRGN,LPRECT,UINT);
04053 LRESULT      WINAPI SendDlgItemMessageA(HWND,INT,UINT,WPARAM,LPARAM);
04054 LRESULT      WINAPI SendDlgItemMessageW(HWND,INT,UINT,WPARAM,LPARAM);
04055 #define      SendDlgItemMessage WINELIB_NAME_AW(SendDlgItemMessage)
04056 UINT        WINAPI SendInput(UINT,LPINPUT,int);
04057 LRESULT      WINAPI SendMessageA(HWND,UINT,WPARAM,LPARAM);
04058 LRESULT      WINAPI SendMessageW(HWND,UINT,WPARAM,LPARAM);
04059 #define      SendMessage WINELIB_NAME_AW(SendMessage)
04060 LRESULT      WINAPI SendMessageTimeoutA(HWND,UINT,WPARAM,LPARAM,UINT,UINT,LPDWORD);

```

```

04061 LRESULT      WINAPI SendMessageTimeoutW(HWND,UINT,WPARAM,LPARAM,UINT,UINT,LPDWORD);
04062 #define        SendMessageTimeout WINELIB_NAME_AW(SendMessageTimeout)
04063 HWND          WINAPI SetActiveWindow(HWND);
04064 HWND          WINAPI SetCapture(HWND);
04065 BOOL          WINAPI SetCaretBlinkTime(UINT);
04066 BOOL          WINAPI SetCaretPos(INT,INT);
04067 LONG          WINAPI SetClassLongA(HWND,INT,LONG);
04068 LONG          WINAPI SetClassLongW(HWND,INT,LONG);
04069 #define        SetClassLong WINELIB_NAME_AW(SetClassLong)
04070 WORD          WINAPI SetClassWord(HWND,INT,WORD);
04071 HANDLE        WINAPI SetClipboardData(UINT,HANDLE);
04072 HWND          WINAPI SetClipboardViewer(HWND);
04073 HCURSOR       WINAPI SetCursor(HCURSOR);
04074 BOOL          WINAPI SetCursorPos(INT,INT);
04075 BOOL          WINAPI SetDeskWallPaper(LPCSTR);
04076 BOOL          WINAPI SetDlgItemInt(HWND,INT,UINT,BOOL);
04077 BOOL          WINAPI SetDlgItemTextA(HWND,INT,LPCSTR);
04078 BOOL          WINAPI SetDlgItemTextW(HWND,INT,LPCWSTR);
04079 #define        SetDlgItemText WINELIB_NAME_AW(SetDlgItemText)
04080 BOOL          WINAPI SetDoubleClickTime(UINT);
04081 HWND          WINAPI SetFocus(HWND);
04082 BOOL          WINAPI SetForegroundWindow(HWND);
04083 void          WINAPI SetInternalWindowPos(HWND,UINT,LPRECT,LPPOINT);
04084 BOOL          WINAPI SetMenu(HWND,HMENU);
04085 BOOL          WINAPI SetMenuContextHelpId(HMENU,DWORD);
04086 BOOL          WINAPI SetMenuItemBitmaps(HMENU,UINT,UINT,HBITMAP,HBITMAP);
04087 BOOL          WINAPI SetMessageQueue(INT);
04088 BOOL          WINAPI SetProcessDefaultLayout(DWORD);
04089 BOOL          WINAPI SetProcessWindowStation(HWINSTA);
04090 HWND          WINAPI SetParent(HWND,HWND);
04091 BOOL          WINAPI SetPropA(HWND,LPCSTR,HANDLE);
04092 BOOL          WINAPI SetPropW(HWND,LPCWSTR,HANDLE);
04093 #define        SetProp WINELIB_NAME_AW(SetProp)
04094 BOOL          WINAPI SetRect(LPRECT,INT,INT,INT,INT);
04095 BOOL          WINAPI SetRectEmpty(LPRECT);
04096 INT           WINAPI SetScrollInfo(HWND,INT,const SCROLLINFO*,BOOL);
04097 INT           WINAPI SetScrollPos(HWND,INT,INT,BOOL);
04098 BOOL          WINAPI SetScrollRange(HWND,INT,INT,INT,BOOL);
04099 #define        SetSysModalWindow(hwnd) ((HWND)0)
04100 BOOL          WINAPI SetSystemCursor(HCURSOR,DWORD);
04101 BOOL          WINAPI SetSystemMenu(HWND,HMENU);
04102 UINT          WINAPI SetSystemTimer(HWND,UINT,UINT,TIMERPROC);
04103 UINT          WINAPI SetTimer(HWND,UINT,UINT,TIMERPROC);
04104 BOOL          WINAPI SetUserObjectSecurity(HANDLE,PSECURITY_INFORMATION,PSECURITY_DESCRIPTOR);
04105 LONG          WINAPI SetWindowLongA(HWND,INT,LONG);
04106 LONG          WINAPI SetWindowLongW(HWND,INT,LONG);
04107 #define        SetWindowLong WINELIB_NAME_AW(SetWindowLong)
04108 BOOL          WINAPI SetWindowPlacement(HWND,const WINDOWPLACEMENT*);
04109 HHOOK         WINAPI SetWindowsHookA(INT,HOOKPROC);
04110 HHOOK         WINAPI SetWindowsHookW(INT,HOOKPROC);
04111 #define        SetWindowsHook WINELIB_NAME_AW(SetWindowsHook)
04112 HHOOK         WINAPI SetWindowsHookExA(INT,HOOKPROC,HINSTANCE,DWORD);
04113 HHOOK         WINAPI SetWindowsHookExW(INT,HOOKPROC,HINSTANCE,DWORD);
04114 #define        SetWindowsHookEx WINELIB_NAME_AW(SetWindowsHookEx)
04115 BOOL          WINAPI SetWindowPos(HWND,HWND,INT,INT,INT,INT,UINT);
04116 INT           WINAPI SetWindowRgn(HWND,HRGN,BOOL);
04117 BOOL          WINAPI SetWindowTextA(HWND,LPCSTR);
04118 BOOL          WINAPI SetWindowTextW(HWND,LPCWSTR);
04119 #define        SetWindowText WINELIB_NAME_AW(SetWindowText)
04120 WORD          WINAPI SetWindowWord(HWND,INT,WORD);
04121 BOOL          WINAPI ShowCaret(HWND);
04122 INT           WINAPI ShowCursor(BOOL);
04123 BOOL          WINAPI ShowScrollBar(HWND,INT,BOOL);
04124 BOOL          WINAPI ShowOwnedPopups(HWND,BOOL);
04125 BOOL          WINAPI ShowWindow(HWND,INT);
04126 BOOL          WINAPI SubtractRect(LPRECT,const RECT*,const RECT*);
04127 BOOL          WINAPI SwapMouseButton(BOOL);
04128 VOID          WINAPI SwitchToThisWindow(HWND,BOOL);
04129 BOOL          WINAPI SystemParametersInfoA(UINT,UINT,LPVOID,UINT);
04130 BOOL          WINAPI SystemParametersInfoW(UINT,UINT,LPVOID,UINT);
04131 #define        SystemParametersInfo WINELIB_NAME_AW(SystemParametersInfo)
04132 LONG          WINAPI TabbedTextOutA(HDC,INT,INT,LPCSTR,INT,INT,const INT*,INT);
04133 LONG          WINAPI TabbedTextOutW(HDC,INT,INT,LPCWSTR,INT,INT,const INT*,INT);
04134 #define        TabbedTextOut WINELIB_NAME_AW(TabbedTextOut)
04135 INT           WINAPI ToAscii(UINT,UINT,LPBYTE,LPWORD,UINT);
04136 INT           WINAPI ToAsciiEx(UINT,UINT,LPBYTE,LPWORD,UINT,HKL);
04137 BOOL          WINAPI TrackPopupMenu(HMENU,UINT,INT,INT,INT,HWND,const RECT*);
04138 INT           WINAPI TranslateAccelerator(HWND,HACCEL,LPMMSG);
04139 BOOL          WINAPI TranslateMDISysAccel(HWND,LPMMSG);
04140 BOOL          WINAPI TranslateMessage(const MSG*);
04141 BOOL          WINAPI UnhookWindowsHook(INT,HOOKPROC);
04142 BOOL          WINAPI UnhookWindowsHookEx(HHOOK);
04143 BOOL          WINAPI UnionRect(LPRECT,const RECT*,const RECT*);
04144 BOOL          WINAPI UnregisterClassA(LPCSTR,HINSTANCE);
04145 BOOL          WINAPI UnregisterClassW(LPCWSTR,HINSTANCE);
04146 #define        UnregisterClass WINELIB_NAME_AW(UnregisterClass)
04147 VOID          WINAPI UpdateWindow(HWND);

```

```

04148 UINT      WINAPI UserRealizePalette(HDC);
04149 VOID      WINAPI ValidateRect(HWND, const RECT*);
04150 VOID      WINAPI ValidateRgn(HWND, HRGN);
04151 WORD      WINAPI VkKeyScanA(CHAR);
04152 WORD      WINAPI VkKeyScanW(WCHAR);
04153 #define      VkKeyScan WINELIB_NAME_AW(VkKeyScan)
04154 WORD      WINAPI VkKeyScanExA(CHAR, HKL);
04155 WORD      WINAPI VkKeyScanExW(WCHAR, HKL);
04156 #define      VkKeyScanEx WINELIB_NAME_AW(VkKeyScanEx)
04157 BOOL      WINAPI WaitMessage(void);
04158 HWND      WINAPI WindowFromDC(HDC);
04159 HWND      WINAPI WindowFromPoint(POINT);
04160 BOOL      WINAPI WinHelpA(HWND, LPCSTR, UINT, DWORD);
04161 BOOL      WINAPI WinHelpW(HWND, LPCWSTR, UINT, DWORD);
04162 #define      WinHelp WINELIB_NAME_AW(WinHelp)
04163 INT      WINAPIV wsprintfA(LPSTR, LPCSTR, ...);
04164 INT      WINAPIV wsprintfW(LPWSTR, LPCWSTR, ...);
04165 #define      wsprintf WINELIB_NAME_AW(wsprintf)
04166 INT      WINAPI wvsprintfA(LPSTR, LPCSTR, va_list);
04167 INT      WINAPI wvsprintfW(LPWSTR, LPCWSTR, va_list);
04168 #define      wvsprintf WINELIB_NAME_AW(wvsprintf)
04169
04170 /* Undocumented functions */
04171
04172 /* NOTE: This is SYSTEM.3, not USER.182, which is also named KillSystemTimer */
04173 WORD      WINAPI SYSTEM_KillSystemTimer(WORD);
04174
04175 HRESULT    WINAPI PrivateExtractIconsA(LPCSTR, INT, DWORD, DWORD, HICON*, DWORD, UINT, DWORD);
04176 HRESULT    WINAPI PrivateExtractIconsW(LPCWSTR, INT, DWORD, DWORD, HICON*, DWORD, UINT, DWORD);
04177
04178 /* Extra functions that don't exist in the Windows API */
04179
04180 HPEN      WINAPI GetSysColorPen(INT);
04181 INT      WINAPI wvsprintfA(LPSTR, UINT, LPCSTR, va_list);
04182 INT      WINAPI wvsprintfW(LPWSTR, UINT, LPCWSTR, va_list);
04183 #define      wvsprintf WINELIB_NAME_AW(wvsprintf)
04184
04185 #ifdef __cplusplus
04186 }
04187 #endif
04188
04189 #endif /* _WINUSER_ */

```

## 5.14 libemf.h

```

00001 /* -*- c++ -*-
00002  * EMF: A library for generating ECMA-234 Enhanced Metafiles
00003  * Copyright (C) 2002, 2003 lignum Computing, Inc. <dallenbarnett@users.sourceforge.net>
00004  * $Id: libemf.h 94 2020-04-25 18:46:06Z dallenbarnett $
00005  *
00006  * This library is free software; you can redistribute it and/or
00007  * modify it under the terms of the GNU Lesser General Public
00008  * License as published by the Free Software Foundation; either
00009  * version 2.1 of the License, or (at your option) any later version.
00010  *
00011  * This library is distributed in the hope that it will be useful,
00012  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00013  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
00014  * Lesser General Public License for more details.
00015  *
00016  * You should have received a copy of the GNU Lesser General Public
00017  * License along with this library; if not, write to the Free Software
00018  * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
00019  *
00020  */
00021 #ifndef _LIBEMF_H
00022 #define _LIBEMF_H 1
00023
00024 #include <cmath>
00025 #include <vector>
00026 #include <map>
00027 #include <functional>
00028 #include <algorithm>
00029 #include <stdexcept>
00030 #include <memory>
00031
00032 #include <config.h>
00033 #include <libEMF/emf.h>
00034
00035 #include <libEMF/wine/w16.h>
00036
00037 #ifdef ENABLE_EDITING
00038 #include <iconv.h>
00039 #include <errno.h>

```

```

00040 #endif
00041
00042 #define EMF_UNUSED(x) (void)x;
00043
00044 namespace EMF {
00045 #if 1
00050     const int XMAX_PIXELS = 1024; /*(INT_MAX)*/
00051 #else
00052     const int XMAX_PIXELS = 1280; /*(INT_MAX)*/
00053 #endif
00054 #if 1
00059     const int YMAX_PIXELS = 768; /*(INT_MAX)*/
00060 #else
00061     const int YMAX_PIXELS = 1024; /*(INT_MAX)*/
00062 #endif
00068     const int XMAX_MM = 320;
00074     const int YMAX_MM = 240;
00078     const int RESOLUTION = 96;
00082     static inline DWORD ROUND_TO_LONG ( DWORD n ) { return ((n+3)/4)*4; }
00083
00084     static bool bigEndian ( void );
00085
00086
00092     struct WCHARSTR {
00093         WCHAR *const string_;
00094         const int length_;
00100         WCHARSTR ( WCHAR *const string, const int length )
00101             : string_( string ), length_( length ) {}
00102     };
00103
00104
00110     struct CHARSTR {
00111         CHAR *const string_;
00112         const int length_;
00118         CHARSTR ( CHAR *const string, const int length )
00119             : string_( string ), length_( length ) {}
00120     };
00121
00122
00127     struct BYTEARRAY {
00128         BYTE *const array_;
00129         const int n_;
00135         BYTEARRAY ( BYTE *const array, const int n )
00136             : array_( array ), n_( n ) {}
00137     };
00138
00139
00143     struct POINTLARRAY {
00144         POINTL *const points_;
00145         const DWORD n_;
00151         POINTLARRAY ( POINTL *const points, const DWORD n )
00152             : points_( points ), n_( n ) {}
00153     };
00154
00155
00159     struct POINT16ARRAY {
00160         POINT16 *const points_;
00161         const DWORD n_;
00167         POINT16ARRAY ( POINT16 *const points, const DWORD n )
00168             : points_( points ), n_( n ) {}
00169     };
00170
00171
00175     struct INTARRAY {
00176         INT *const ints_;
00177         const DWORD n_;
00183         INTARRAY ( INT *const ints, const DWORD n )
00184             : ints_( ints ), n_( n ) {}
00185     };
00186
00187
00191     struct DWORDARRAY {
00192         DWORD *const dwords_;
00193         const DWORD n_;
00199         DWORDARRAY ( DWORD *const dwords, const DWORD n )
00200             : dwords_( dwords ), n_( n ) {}
00201     };
00202
00203
00207     struct PADDING {
00208         static const char padding_[4];
00209         const int size_;
00214         PADDING ( const int size ) : size_( size ) {}
00215     };
00216
00217
00225     class DATASTREAM {

```

```

00226     bool swap_;
00227     ::FILE* fp_;
00228 public:
00234     DATASTREAM ( ::FILE* fp = 0 ) : swap_( bigEndian() ), fp_( fp ) {}
00239     void setStream ( ::FILE* fp ) { fp_ = fp; }
00244     DATASTREAM& operator« ( const BYTE& byte )
00245     {
00246         fwrite( &byte, sizeof(BYTE), 1, fp_ );
00247         return *this;
00248     }
00253     DATASTREAM& operator» ( BYTE& byte )
00254     {
00255         fread( &byte, sizeof(BYTE), 1, fp_ );
00256         return *this;
00257     }
00262     DATASTREAM& operator« ( const WORD& word )
00263     {
00264         if ( swap_ ) {
00265             unsigned char const* p = (unsigned char const*)&word;
00266             fwrite( &p[1], sizeof(unsigned char), 1, fp_ );
00267             fwrite( &p[0], sizeof(unsigned char), 1, fp_ );
00268         }
00269         else
00270             fwrite( &word, sizeof(WORD), 1, fp_ );
00271         return *this;
00272     }
00277     DATASTREAM& operator» ( WORD& word )
00278     {
00279         if ( swap_ ) {
00280             unsigned char* p = (unsigned char*)&word;
00281             fread( &p[1], sizeof(unsigned char), 1, fp_ );
00282             fread( &p[0], sizeof(unsigned char), 1, fp_ );
00283         }
00284         else
00285             fread( &word, sizeof(WORD), 1, fp_ );
00286         return *this;
00287     }
00292     DATASTREAM& operator« ( const INT16& word )
00293     {
00294         if ( swap_ ) {
00295             unsigned char const* p = (unsigned char const*)&word;
00296             fwrite( &p[1], sizeof(unsigned char), 1, fp_ );
00297             fwrite( &p[0], sizeof(unsigned char), 1, fp_ );
00298         }
00299         else
00300             fwrite( &word, sizeof(INT16), 1, fp_ );
00301         return *this;
00302     }
00307     DATASTREAM& operator» ( INT16& word )
00308     {
00309         if ( swap_ ) {
00310             unsigned char* p = (unsigned char*)&word;
00311             fread( &p[1], sizeof(unsigned char), 1, fp_ );
00312             fread( &p[0], sizeof(unsigned char), 1, fp_ );
00313         }
00314         else
00315             fread( &word, sizeof(INT16), 1, fp_ );
00316         return *this;
00317     }
00322     DATASTREAM& operator« ( const DWORD& dword )
00323     {
00324         if ( swap_ ) {
00325             unsigned char const* p = (unsigned char const*)&dword;
00326             fwrite( &p[3], sizeof(unsigned char), 1, fp_ );
00327             fwrite( &p[2], sizeof(unsigned char), 1, fp_ );
00328             fwrite( &p[1], sizeof(unsigned char), 1, fp_ );
00329             fwrite( &p[0], sizeof(unsigned char), 1, fp_ );
00330         }
00331         else
00332             fwrite( &dword, sizeof(DWORD), 1, fp_ );
00333         return *this;
00334     }
00339     DATASTREAM& operator» ( DWORD& dword )
00340     {
00341         if ( swap_ ) {
00342             unsigned char* p = (unsigned char*)&dword;
00343             fread( &p[3], sizeof(unsigned char), 1, fp_ );
00344             fread( &p[2], sizeof(unsigned char), 1, fp_ );
00345             fread( &p[1], sizeof(unsigned char), 1, fp_ );
00346             fread( &p[0], sizeof(unsigned char), 1, fp_ );
00347         }
00348         else
00349             fread( &dword, sizeof(DWORD), 1, fp_ );
00350         return *this;
00351     }
00352     #if !defined( __LP64__ )
00357     DATASTREAM& operator« ( const LONG& long_ )

```

```

00358     {
00359         if ( swap_ ) {
00360             unsigned char const* p = (unsigned char const*)&long_;
00361             fwrite( &p[3], sizeof(unsigned char), 1, fp_ );
00362             fwrite( &p[2], sizeof(unsigned char), 1, fp_ );
00363             fwrite( &p[1], sizeof(unsigned char), 1, fp_ );
00364             fwrite( &p[0], sizeof(unsigned char), 1, fp_ );
00365         }
00366         else
00367             fwrite( &long_, sizeof(LONG), 1, fp_ );
00368         return *this;
00369     }
00374     DATASTREAM& operator» ( LONG& long_ )
00375     {
00376         if ( swap_ ) {
00377             unsigned char* p = (unsigned char*)&long_;
00378             fread( &p[3], sizeof(unsigned char), 1, fp_ );
00379             fread( &p[2], sizeof(unsigned char), 1, fp_ );
00380             fread( &p[1], sizeof(unsigned char), 1, fp_ );
00381             fread( &p[0], sizeof(unsigned char), 1, fp_ );
00382         }
00383         else
00384             fread( &long_, sizeof(LONG), 1, fp_ );
00385         return *this;
00386     }
00387 #endif /* __x86_64__ */
00392     DATASTREAM& operator« ( const INT& int_ )
00393     {
00394         if ( swap_ ) {
00395             unsigned char const* p = (unsigned char const*)&int_;
00396             fwrite( &p[3], sizeof(unsigned char), 1, fp_ );
00397             fwrite( &p[2], sizeof(unsigned char), 1, fp_ );
00398             fwrite( &p[1], sizeof(unsigned char), 1, fp_ );
00399             fwrite( &p[0], sizeof(unsigned char), 1, fp_ );
00400         }
00401         else
00402             fwrite( &int_, sizeof(INT), 1, fp_ );
00403         return *this;
00404     }
00409     DATASTREAM& operator» ( INT& int_ )
00410     {
00411         if ( swap_ ) {
00412             unsigned char* p = (unsigned char*)&int_;
00413             fread( &p[3], sizeof(unsigned char), 1, fp_ );
00414             fread( &p[2], sizeof(unsigned char), 1, fp_ );
00415             fread( &p[1], sizeof(unsigned char), 1, fp_ );
00416             fread( &p[0], sizeof(unsigned char), 1, fp_ );
00417         }
00418         else
00419             fread( &int_, sizeof(INT), 1, fp_ );
00420         return *this;
00421     }
00422 #if !defined(__LP64__)
00427     DATASTREAM& operator« ( const UINT& uint )
00428     {
00429         if ( swap_ ) {
00430             unsigned char const* p = (unsigned char const*)&uint;
00431             fwrite( &p[3], sizeof(unsigned char), 1, fp_ );
00432             fwrite( &p[2], sizeof(unsigned char), 1, fp_ );
00433             fwrite( &p[1], sizeof(unsigned char), 1, fp_ );
00434             fwrite( &p[0], sizeof(unsigned char), 1, fp_ );
00435         }
00436         else
00437             fwrite( &uint, sizeof(UINT), 1, fp_ );
00438         return *this;
00439     }
00444     DATASTREAM& operator» ( UINT& uint )
00445     {
00446         if ( swap_ ) {
00447             unsigned char* p = (unsigned char*)&uint;
00448             fread( &p[3], sizeof(unsigned char), 1, fp_ );
00449             fread( &p[2], sizeof(unsigned char), 1, fp_ );
00450             fread( &p[1], sizeof(unsigned char), 1, fp_ );
00451             fread( &p[0], sizeof(unsigned char), 1, fp_ );
00452         }
00453         else
00454             fread( &uint, sizeof(UINT), 1, fp_ );
00455         return *this;
00456     }
00457 #endif /* !__x86_64__ */
00462     DATASTREAM& operator« ( const FLOAT& float_ )
00463     {
00464         if ( swap_ ) {
00465             unsigned char const* p = (unsigned char const*)&float_;
00466             fwrite( &p[3], sizeof(unsigned char), 1, fp_ );
00467             fwrite( &p[2], sizeof(unsigned char), 1, fp_ );
00468             fwrite( &p[1], sizeof(unsigned char), 1, fp_ );

```

```

00469     fwrite( &p[0], sizeof(unsigned char), 1, fp_ );
00470     }
00471     else
00472     fwrite( &float_, sizeof(FLOAT), 1, fp_ );
00473     return *this;
00474     }
00475     DATASTREAM& operator» ( FLOAT& float_ )
00476     {
00477         if ( swap_ ) {
00478             unsigned char* p = (unsigned char*)&float_;
00479             fread( &p[3], sizeof(unsigned char), 1, fp_ );
00480             fread( &p[2], sizeof(unsigned char), 1, fp_ );
00481             fread( &p[1], sizeof(unsigned char), 1, fp_ );
00482             fread( &p[0], sizeof(unsigned char), 1, fp_ );
00483         }
00484         else
00485             fread( &float_, sizeof(FLOAT), 1, fp_ );
00486         return *this;
00487     }
00488     DATASTREAM& operator« ( const PADDING& padding )
00489     {
00490         if ( padding.size_ != 0 )
00491             fwrite( &padding.padding_, sizeof(CHAR), padding.size_, fp_ );
00492         return *this;
00493     }
00494     DATASTREAM& operator« ( const RECTL& rectl )
00495     {
00496         *this « rectl.left « rectl.top « rectl.right « rectl.bottom;
00497         return *this;
00498     }
00499     DATASTREAM& operator» ( RECTL& rectl )
00500     {
00501         *this » rectl.left » rectl.top » rectl.right » rectl.bottom;
00502         return *this;
00503     }
00504     DATASTREAM& operator« ( const SIZEL& sizel )
00505     {
00506         *this « sizel.cx « sizel.cy;
00507         return *this;
00508     }
00509     DATASTREAM& operator» ( SIZEL& sizel )
00510     {
00511         *this » sizel.cx » sizel.cy;
00512         return *this;
00513     }
00514     DATASTREAM& operator« ( const WCHARSTR& wcharstr )
00515     {
00516         for ( int i = 0; i < wcharstr.length_; i++ )
00517             *this « wcharstr.string_[i];
00518         return *this;
00519     }
00520     DATASTREAM& operator» ( WCHARSTR& wcharstr )
00521     {
00522         for ( int i = 0; i < wcharstr.length_; i++ )
00523             *this » wcharstr.string_[i];
00524         return *this;
00525     }
00526     DATASTREAM& operator« ( const CHARSTR& charstr )
00527     {
00528         fwrite( charstr.string_, sizeof(CHAR), charstr.length_, fp_ );
00529         return *this;
00530     }
00531     DATASTREAM& operator» ( CHARSTR& charstr )
00532     {
00533         fread( charstr.string_, sizeof(CHAR), charstr.length_, fp_ );
00534         return *this;
00535     }
00536     DATASTREAM& operator« ( const ::EMR& emr )
00537     {
00538         *this « emr.iType « emr.nSize;
00539         return *this;
00540     }
00541     DATASTREAM& operator» ( ::EMR& emr )
00542     {
00543         *this » emr.iType » emr.nSize;
00544         return *this;
00545     }
00546     DATASTREAM& operator« ( const POINT& point )
00547     {
00548         *this « point.x « point.y;
00549         return *this;
00550     }
00551     DATASTREAM& operator» ( POINT& point )
00552     {
00553         *this » point.x » point.y;
00554         return *this;
00555     }
00556     }

```

```

00616     DATASTREAM& operator« ( const POINTL& pointl )
00617     {
00618         *this « pointl.x « pointl.y;
00619         return *this;
00620     }
00621     DATASTREAM& operator» ( POINTL& pointl )
00622     {
00623         *this » pointl.x » pointl.y;
00624         return *this;
00625     }
00626     DATASTREAM& operator« ( const POINT16& point )
00627     {
00628         *this « point.x « point.y;
00629         return *this;
00630     }
00631     DATASTREAM& operator» ( POINT16& point )
00632     {
00633         *this » point.x » point.y;
00634         return *this;
00635     }
00636     DATASTREAM& operator« ( const XFORM& xform )
00637     {
00638         *this « xform.eM11 « xform.eM12 « xform.eM21 « xform.eM22
00639             « xform.eDx « xform.eDy;
00640         return *this;
00641     }
00642     DATASTREAM& operator» ( XFORM& xform )
00643     {
00644         *this » xform.eM11 » xform.eM12 » xform.eM21 » xform.eM22
00645             » xform.eDx » xform.eDy;
00646         return *this;
00647     }
00648     DATASTREAM& operator« ( const BYTEARRAY& array )
00649     {
00650         fwrite( array.array_, sizeof(BYTE), array.n_, fp_ );
00651         return *this;
00652     }
00653     DATASTREAM& operator» ( BYTEARRAY& array )
00654     {
00655         fread( array.array_, sizeof(BYTE), array.n_, fp_ );
00656         return *this;
00657     }
00658     DATASTREAM& operator« ( const POINTLARRAY& array )
00659     {
00660         for ( unsigned int i = 0; i < array.n_; i++ )
00661             *this « array.points_[i];
00662         return *this;
00663     }
00664     DATASTREAM& operator» ( POINTLARRAY& array )
00665     {
00666         for ( unsigned int i = 0; i < array.n_; i++ )
00667             *this » array.points_[i];
00668         return *this;
00669     }
00670     DATASTREAM& operator« ( const POINT16ARRAY& array )
00671     {
00672         for ( unsigned int i = 0; i < array.n_; i++ )
00673             *this « array.points_[i];
00674         return *this;
00675     }
00676     DATASTREAM& operator» ( POINT16ARRAY& array )
00677     {
00678         for ( unsigned int i = 0; i < array.n_; i++ )
00679             *this » array.points_[i];
00680         return *this;
00681     }
00682     DATASTREAM& operator« ( const INTARRAY& array )
00683     {
00684         for ( unsigned int i = 0; i < array.n_; i++ )
00685             *this « array.ints_[i];
00686         return *this;
00687     }
00688     DATASTREAM& operator» ( INTARRAY& array )
00689     {
00690         for ( unsigned int i = 0; i < array.n_; i++ )
00691             *this » array.ints_[i];
00692         return *this;
00693     }
00694     DATASTREAM& operator« ( const DWORDARRAY& array )
00695     {
00696         for ( unsigned int i = 0; i < array.n_; i++ )
00697             *this « array.dwords_[i];
00698         return *this;
00699     }
00700     DATASTREAM& operator» ( DWORDARRAY& array )
00701     {
00702         for ( unsigned int i = 0; i < array.n_; i++ )
00703             *this » array.dwords_[i];
00704     }

```

```

00763     *this » array.dwords_[i];
00764     return *this;
00765 }
00770 DATASTREAM& operator« ( const ::EMRTEXT& text )
00771 {
00772     *this « text.ptlReference « text.nChars « text.offString « text.fOptions
00773     « text.rcl « text.offDx;
00774     return *this;
00775 }
00780 DATASTREAM& operator« ( ::EMRTEXT& text )
00781 {
00782     *this » text.ptlReference » text.nChars » text.offString » text.fOptions
00783     » text.rcl » text.offDx;
00784     return *this;
00785 }
00790 DATASTREAM& operator« ( const LOGPEN& pen )
00791 {
00792     *this « pen.lopnStyle « pen.lopnWidth « pen.lopnColor;
00793     return *this;
00794 }
00799 DATASTREAM& operator« ( LOGPEN& pen )
00800 {
00801     *this » pen.lopnStyle » pen.lopnWidth » pen.lopnColor;
00802     return *this;
00803 }
00808 DATASTREAM& operator« ( const EXTLOGPEN& pen )
00809 {
00810     // *** How big is this structure if there are no style entries? ***
00811     *this « pen.elpPenStyle « pen.elpWidth « pen.elpBrushStyle « pen.elpColor
00812     « pen.elpHatch « pen.elpNumEntries;
00813     return *this;
00814 }
00819 DATASTREAM& operator« ( EXTLOGPEN& pen )
00820 {
00821     // *** How big is this structure if there are no style entries? ***
00822     *this » pen.elpPenStyle » pen.elpWidth » pen.elpBrushStyle » pen.elpColor
00823     » pen.elpHatch » pen.elpNumEntries;
00824     return *this;
00825 }
00830 DATASTREAM& operator« ( const LOGBRUSH& brush )
00831 {
00832     *this « brush.lbStyle « brush.lbColor « brush.lbHatch;
00833     return *this;
00834 }
00839 DATASTREAM& operator« ( LOGBRUSH& brush )
00840 {
00841     *this » brush.lbStyle » brush.lbColor » brush.lbHatch;
00842     return *this;
00843 }
00848 DATASTREAM& operator« ( const LOGFONTW& font )
00849 {
00850     *this « font.lfHeight « font.lfWidth « font.lfEscapement
00851     « font.lfOrientation « font.lfWeight « font.lfItalic
00852     « font.lfUnderline « font.lfStrikeOut « font.lfCharSet
00853     « font.lfOutPrecision « font.lfClipPrecision « font.lfQuality
00854     « font.lfPitchAndFamily
00855     « WCHARSTR( const_cast<WCHAR*>(font.lfFaceName), LF_FACESIZE );
00856     return *this;
00857 }
00862 DATASTREAM& operator« ( LOGFONT& font )
00863 {
00864     WCHARSTR wFaceName( font.lfFaceName, LF_FACESIZE );
00865
00866     *this » font.lfHeight » font.lfWidth » font.lfEscapement
00867     » font.lfOrientation » font.lfWeight » font.lfItalic
00868     » font.lfUnderline » font.lfStrikeOut » font.lfCharSet
00869     » font.lfOutPrecision » font.lfClipPrecision » font.lfQuality
00870     » font.lfPitchAndFamily
00871     » wFaceName;
00872     return *this;
00873 }
00878 DATASTREAM& operator« ( const PANOSE& panose )
00879 {
00880     fwrite( &panose, sizeof(PANOSE), 1, fp_ );
00881     return *this;
00882 }
00887 DATASTREAM& operator« ( PANOSE& panose )
00888 {
00889     fread( &panose, sizeof(PANOSE), 1, fp_ );
00890     return *this;
00891 }
00896 DATASTREAM& operator« ( const EXTLOGFONTW& font )
00897 {
00898     *this « font.elfLogFont
00899     « WCHARSTR( const_cast<WCHAR*>(font.elfFullName),
00900         LF_FULLFACESIZE )
00901     « WCHARSTR( const_cast<WCHAR*>(font.elfStyle), LF_FACESIZE )

```

```

00902     « font.elfVersion « font.elfStyleSize « font.elfMatch
00903     « font.elfReserved
00904     « BYTEARRAY( const_cast<BYTE*>(font.elfVendorId),
00905         ELF_VENDOR_SIZE )
00906     « font.elfCulture « font.elfPanose;
00907     return *this;
00908 }
00913 DATASTREAM& operator« ( EXTLOGFONTW& font )
00914 {
00915     WCHARSTR wFullName( font.elfFullName, LF_FULLFACESIZE );
00916     WCHARSTR wStyle( font.elfStyle, LF_FACESIZE );
00917     BYTEARRAY bVendorId( font.elfVendorId, ELF_VENDOR_SIZE );
00918     *this « font.elfLogFont
00919         « wFullName « wStyle
00920         « font.elfVersion « font.elfStyleSize « font.elfMatch
00921         « font.elfReserved « bVendorId
00922         « font.elfCulture « font.elfPanose;
00923     return *this;
00924 }
00929 DATASTREAM& operator« ( const LOGPALETTE& palette )
00930 {
00931     // *** How big is this structure if the palette is empty? ***
00932     *this « palette.palVersion « palette.palNumEntries;
00933     return *this;
00934 }
00939 DATASTREAM& operator« ( LOGPALETTE& palette )
00940 {
00941     // *** How big is this structure if the palette is empty? ***
00942     *this « palette.palVersion « palette.palNumEntries;
00943     return *this;
00944 }
00945 private:
00955 void fread ( void* ptr, size_t size, size_t nmemb, FILE* stream )
00956 {
00957     size_t res = ::fread( ptr, size, nmemb, stream );
00958     if ( res < nmemb ) {
00959         throw std::runtime_error( "Premature EOF on EMF stream" );
00960     }
00961 }
00971 void fwrite ( const void* ptr, size_t size, size_t nmemb, FILE* stream )
00972 {
00973     size_t res = ::fwrite( ptr, size, nmemb, stream );
00974     if ( res < nmemb ) {
00975         throw std::runtime_error( "error writing EMF stream" );
00976     }
00977 }
00978 };
00979
00980 class METAFILEDEVICECONTEXT;
00981
00982 class METARECORD {
00983 public:
00997     virtual void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const = 0;
01004     virtual bool serialize ( DATASTREAM ds ) = 0;
01010     virtual int size ( void ) const = 0;
01016     virtual ~METARECORD() { }
01017 #ifndef ENABLE_EDITING
01022     virtual void edit ( void ) const {}
01023 #endif
01024 };
01025
01026 #ifndef ENABLE_EDITING
01027     /* Miscellaneous editing routines */
01028     inline void edit_rectl ( const char* tag, const RECTL& rectl )
01029     {
01030         #if defined(__LP64__)
01031             const char* FMT = "\t%s\t: (%d, %d) - (%d, %d)\n";
01032         #else
01033             const char* FMT = "\t%s\t: (%ld, %ld) - (%ld, %ld)\n";
01034         #endif /* __x86_64__ */
01035         printf( FMT, tag, rectl.left, rectl.top, rectl.right, rectl.bottom );
01036     }
01037
01038     inline void edit_xform ( const char* tag, const XFORM& xform )
01039     {
01040         printf( "\t%s.eM11\t: %f\n", tag, xform.eM11 );
01041         printf( "\t%s.eM12\t: %f\n", tag, xform.eM12 );
01042         printf( "\t%s.eM21\t: %f\n", tag, xform.eM21 );
01043         printf( "\t%s.eM22\t: %f\n", tag, xform.eM22 );
01044         printf( "\t%s.eDx\t: %f\n", tag, xform.eDx );
01045         printf( "\t%s.eDy\t: %f\n", tag, xform.eDy );
01046     }
01047
01048     inline void edit_color ( const char* tag, const COLORREF& color )
01049     {
01050         #if defined(__LP64__)

```

```

01051     const char* FMT = "\t%s\t: R(0x%02x) G(0x%02x) B(0x%02x)\n";
01052 #else
01053     const char* FMT = "\t%s\t: R(0x%021x) G(0x%021x) B(0x%021x)\n";
01054 #endif /* __x86_64__ */
01055     printf( FMT, tag,
01056         GetRValue( color ), GetGValue( color ), GetBValue( color ) );
01057 }
01058
01059 inline void edit_size( const char* tag, const SIZE& size )
01060 {
01061 #if defined(__LP64__)
01062     const char* FMT = "\t%s\t: (%d, %d)\n";
01063 #else
01064     const char* FMT = "\t%s\t: (%ld, %ld)\n";
01065 #endif /* __x86_64__ */
01066     printf( FMT, tag, size.cx, size.cy );
01067 }
01068
01069 inline void edit_pointl( const char* tag, const POINTL& point )
01070 {
01071 #if defined(__LP64__)
01072     const char* FMT = "\t%s\t: (%d, %d)\n";
01073 #else
01074     const char* FMT = "\t%s\t: (%ld, %ld)\n";
01075 #endif /* __x86_64__ */
01076     printf( FMT, tag, point.x, point.y );
01077 }
01078
01079 inline void edit_pointlarray( const char* tag, const DWORD cptl,
01080                             const POINTL* points )
01081 {
01082 #if defined(__LP64__)
01083     const char* FMT0 = "\tcptl%s\t: %d\n";
01084     const char* FMT1 = "%d, %d\n";
01085     const char* FMT2 = "\t\t%s %d, %d\n";
01086 #else
01087     const char* FMT0 = "\tcptl%s\t: %ld\n";
01088     const char* FMT1 = "%ld, %ld\n";
01089     const char* FMT2 = "\t\t%s %ld, %ld\n";
01090 #endif /* __x86_64__ */
01091     printf( FMT0, tag, cptl );
01092     printf( "\taptl%s\t: ", tag );
01093     if ( cptl > 0 )
01094         printf( FMT1, points[0].x, points[0].y );
01095     else
01096         puts( "" );
01097     for ( DWORD i = 1; i < cptl; i++ )
01098         printf( FMT2, tag, points[i].x, points[i].y );
01099 }
01100
01101 inline void edit_pointl6array( const char* tag, const unsigned int cpts,
01102                              const POINTL6* points )
01103 {
01104     printf( "\tcpts%s\t: %d\n", tag, cpts );
01105     printf( "\taptl6%s\t: ", tag );
01106     if ( cpts > 0 )
01107         printf( "%d, %d\n", points[0].x, points[0].y );
01108     else
01109         puts( "" );
01110     for ( unsigned int i = 1; i < cpts; i++ )
01111         printf( "\t\t%s %d, %d\n", tag, points[i].x, points[i].y );
01112 }
01113
01114 inline void edit_pen_style( const char* tag, DWORD style )
01115 {
01116     printf( "\t%s\t: ", tag );
01117     switch ( style & PS_STYLE_MASK ) {
01118     case PS_SOLID: printf( "PS_SOLID" ); break;
01119     case PS_DASH: printf( "PS_DASH" ); break;
01120     case PS_DOT: printf( "PS_DOT" ); break;
01121     case PS_DASHDOT: printf( "PS_DASHDOT" ); break;
01122     case PS_DASHDOTDOT: printf( "PS_DASHDOTDOT" ); break;
01123     case PS_NULL: printf( "PS_NULL" ); break;
01124     case PS_INSIDEFRAME: printf( "PS_INSIDEFRAME" ); break;
01125     case PS_USERSTYLE: printf( "PS_USERSTYLE" ); break;
01126     case PS_ALTERNATE: printf( "PS_ALTERNATE" ); break;
01127     }
01128     switch ( style & PS_ENDCAP_MASK ) {
01129     case PS_ENDCAP_ROUND: printf( " | PS_ENDCAP_ROUND" ); break;
01130     case PS_ENDCAP_SQUARE: printf( " | PS_ENDCAP_SQUARE" ); break;
01131     case PS_ENDCAP_FLAT: printf( " | PS_ENDCAP_FLAT" ); break;
01132     }
01133     switch ( style & PS_JOIN_MASK ) {
01134     case PS_JOIN_ROUND: printf( " | PS_JOIN_ROUND" ); break;
01135     case PS_JOIN_BEVEL: printf( " | PS_JOIN_BEVEL" ); break;
01136     case PS_JOIN_MITER: printf( " | PS_JOIN_MITER" ); break;
01137     }

```

```

01138     switch ( style & PS_TYPE_MASK ) {
01139     case PS_COSMETIC: printf( " | PS_COSMETIC" ); break;
01140     case PS_GEOMETRIC: printf( " | PS_GEOMETRIC" ); break;
01141     }
01142     printf( "\n" );
01143 }
01144
01145 inline void edit_brush_style ( const char* tag, DWORD style )
01146 {
01147 #if defined(__LP64__)
01148     const char* FMT = "unknown(%d)";
01149 #else
01150     const char* FMT = "unknown(%ld)";
01151 #endif /* __x86_64__ */
01152     printf( "\t%s\t: ", tag );
01153     switch ( style ) {
01154     case BS_SOLID: printf( "BS_SOLID" ); break;
01155     case BS_NULL: printf( "BS_NULL" ); break;
01156     case BS_HATCHED: printf( "BS_HATCHED" ); break;
01157     case BS_PATTERN: printf( "BS_PATTERN" ); break;
01158     case BS_INDEXED: printf( "BS_INDEXED" ); break;
01159     case BS_DIBPATTERN: printf( "BS_DIBPATTERN" ); break;
01160     case BS_DIBPATTERNPT: printf( "BS_DIBPATTERNPT" ); break;
01161     case BS_PATTERN8X8: printf( "BS_PATTERN8X8" ); break;
01162     case BS_DIBPATTERN8X8: printf( "BS_DIBPATTERN8X8" ); break;
01163     case BS_MONOPATTERN: printf( "BS_DIBPATTERN8X8" ); break;
01164     default: printf( FMT, style );
01165     }
01166     printf( "\n" );
01167 }
01168
01169 inline void edit_brush_hatch ( const char* tag, DWORD hatch )
01170 {
01171 #if defined(__LP64__)
01172     const char* FMT = "unknown(%d)";
01173 #else
01174     const char* FMT = "unknown(%ld)";
01175 #endif /* __x86_64__ */
01176     printf( "\t%s\t: ", tag );
01177     switch ( hatch ) {
01178     case HS_HORIZONTAL: printf( "HS_HORIZONTAL" ); break;
01179     case HS_VERTICAL: printf( "HS_VERTICAL" ); break;
01180     case HS_FDIAGONAL: printf( "HS_FDIAGONAL" ); break;
01181     case HS_BDIAGONAL: printf( "HS_BDIAGONAL" ); break;
01182     case HS_CROSS: printf( "HS_CROSS" ); break;
01183     case HS_DIAGCROSS: printf( "HS_DIAGCROSS" ); break;
01184     default: printf( FMT, hatch );
01185     }
01186     printf( "\n" );
01187 }
01188 #endif
01189 enum OBJECTTYPE { O_METAFILEDEVICECONTEXT = OBJ_METADC,
01190                  O_FONT = OBJ_FONT,
01191                  O_PEN = OBJ_PEN,
01192                  O_EXTPEN = OBJ_EXTPEN,
01193                  O_BRUSH = OBJ_BRUSH,
01194                  O_PALETTE = OBJ_PAL };
01195
01196 #if 0
01197 static char* typStr ( OBJECTTYPE type )
01198 {
01199     switch (type) {
01200     case O_METAFILEDEVICECONTEXT:
01201         return "metafile device context";
01202     case O_FONT:
01203         return "font";
01204     case O_PEN:
01205         return "pen";
01206     case O_EXTPEN:
01207         return "extended pen";
01208     case O_BRUSH:
01209         return "brush";
01210     case O_PALETTE:
01211         return "palette";
01212     }
01213     return "unknown object";
01214 }
01215 #endif
01216
01217 class OBJECT {
01218 public:
01219     HGDIOBJ handle;
01220     virtual ~OBJECT () {}
01221     OBJECT ( void ) : handle( 0 ) {}
01222     virtual OBJECTTYPE getType ( void ) const = 0;
01223 };
01224
01225
01226

```

```

01253 class GRAPHICSOBJECT : public OBJECT {
01254 public:
01255     virtual ~GRAPHICSOBJECT () {}
01261     std::map< HDC, HGDIOBJ > contexts;
01268     virtual METARECORD* newEMR ( HDC dc, HGDIOBJ handle ) = 0;
01269 };
01270
01271 typedef METARECORD* (*METARECORDCTOR) (DATASTREAM&);
01272
01276 class GLOBALOBJECTS {
01280     std::vector<OBJECT*> objects;
01281
01288     std::map< DWORD, METARECORDCTOR > new_records;
01289
01290 public:
01291     GLOBALOBJECTS ( void );
01292     ~GLOBALOBJECTS ( void );
01293     HGDIOBJ add ( OBJECT* object );
01294     OBJECT* find ( const HGDIOBJ handle );
01295     void remove ( const OBJECT* object );
01296
01300     auto begin ( void ) const { return objects.begin(); }
01301
01305     auto end ( void ) const { return objects.end(); }
01306
01307     METARECORDCTOR newRecord ( DWORD iType ) const;
01308
01310     static EMF::METARECORD* new_eof ( DATASTREAM& ds );
01312     static EMF::METARECORD* new_setviewportorgex ( DATASTREAM& ds );
01314     static EMF::METARECORD* new_setwindoworgex ( DATASTREAM& ds );
01316     static EMF::METARECORD* new_setviewportextex ( DATASTREAM& ds );
01318     static EMF::METARECORD* new_setwindowextex ( DATASTREAM& ds );
01320     static EMF::METARECORD* new_scaleviewportextex ( DATASTREAM& ds );
01322     static EMF::METARECORD* new_scalewindowextex ( DATASTREAM& ds );
01324     static EMF::METARECORD* new_modifyworldtransform ( DATASTREAM& ds );
01326     static EMF::METARECORD* new_setworldtransform ( DATASTREAM& ds );
01328     static EMF::METARECORD* new_settextalign ( DATASTREAM& ds );
01330     static EMF::METARECORD* new_settextcolor ( DATASTREAM& ds );
01332     static EMF::METARECORD* new_setbkcolor ( DATASTREAM& ds );
01334     static EMF::METARECORD* new_setbkmode ( DATASTREAM& ds );
01336     static EMF::METARECORD* new_setpolyfillmode ( DATASTREAM& ds );
01338     static EMF::METARECORD* new_setmapmode ( DATASTREAM& ds );
01340     static EMF::METARECORD* new_selectobject ( DATASTREAM& ds );
01342     static EMF::METARECORD* new_deleteobject ( DATASTREAM& ds );
01344     static EMF::METARECORD* new_movetoex ( DATASTREAM& ds );
01346     static EMF::METARECORD* new_lineto ( DATASTREAM& ds );
01348     static EMF::METARECORD* new_arc ( DATASTREAM& ds );
01350     static EMF::METARECORD* new_arcto ( DATASTREAM& ds );
01352     static EMF::METARECORD* new_rectangle ( DATASTREAM& ds );
01354     static EMF::METARECORD* new_ellipse ( DATASTREAM& ds );
01356     static EMF::METARECORD* new_polyline ( DATASTREAM& ds );
01358     static EMF::METARECORD* new_polyline16 ( DATASTREAM& ds );
01360     static EMF::METARECORD* new_polygon ( DATASTREAM& ds );
01362     static EMF::METARECORD* new_polygon16 ( DATASTREAM& ds );
01364     static EMF::METARECORD* new_polypolygon ( DATASTREAM& ds );
01366     static EMF::METARECORD* new_polypolygon16 ( DATASTREAM& ds );
01368     static EMF::METARECORD* new_polybezier ( DATASTREAM& ds );
01370     static EMF::METARECORD* new_polybezier16 ( DATASTREAM& ds );
01372     static EMF::METARECORD* new_polybezierto ( DATASTREAM& ds );
01374     static EMF::METARECORD* new_polybezierto16 ( DATASTREAM& ds );
01376     static EMF::METARECORD* new_polylineto ( DATASTREAM& ds );
01378     static EMF::METARECORD* new_polylineto16 ( DATASTREAM& ds );
01380     static EMF::METARECORD* new_exttextouta ( DATASTREAM& ds );
01382     static EMF::METARECORD* new_exttextoutw ( DATASTREAM& ds );
01384     static EMF::METARECORD* new_setpixelv ( DATASTREAM& ds );
01386     static EMF::METARECORD* new_createpen ( DATASTREAM& ds );
01388     static EMF::METARECORD* new_extcreatepen ( DATASTREAM& ds );
01390     static EMF::METARECORD* new_createbrushindirect ( DATASTREAM& ds );
01392     static EMF::METARECORD* new_extcreatefontindirectw ( DATASTREAM& ds );
01394     static EMF::METARECORD* new_fillpath ( DATASTREAM& ds );
01396     static EMF::METARECORD* new_strokepath ( DATASTREAM& ds );
01398     static EMF::METARECORD* new_strokeandfillpath ( DATASTREAM& ds );
01400     static EMF::METARECORD* new_beginpath ( DATASTREAM& ds );
01402     static EMF::METARECORD* new_endpath ( DATASTREAM& ds );
01404     static EMF::METARECORD* new_closefigure ( DATASTREAM& ds );
01406     static EMF::METARECORD* new_savedc ( DATASTREAM& ds );
01408     static EMF::METARECORD* new_restoredc ( DATASTREAM& ds );
01410     static EMF::METARECORD* new_setmetargn ( DATASTREAM& ds );
01412     static EMF::METARECORD* new_setmiterlimit ( DATASTREAM& ds );
01413 };
01414
01415 extern GLOBALOBJECTS globalObjects;
01416
01418
01424 class ENHMETAHEADER : public METARECORD, public ::ENHMETAHEADER {
01425
01426     LPWSTR description_w{ nullptr };

```

```

01427     int description_size{ 0 };
01428
01429 public:
01436     ENHMETAHEADER ( LPCWSTR description = 0 )
01437     : description_w( 0 ), description_size( 0 )
01438     {
01439         iType = EMR_HEADER;
01440         nSize = sizeof( :ENHMETAHEADER );
01441
01442         // Compute the bounds
01443         RECTL default_bounds = { 0, 0, 0, 0 };
01444         rclBounds = default_bounds;
01445         RECTL default_frame = { 0, 0, 0, 0 };
01446         rclFrame = default_frame;
01447         dSignature = ENHMETA_SIGNATURE;
01448         nVersion = 0x10000;
01449         nBytes = nSize;
01450         nRecords = 1;
01451         nHandles = 0;
01452         sReserved = 0;
01453         nDescription = 0;
01454         offDescription = 0;
01455         nPalEntries = 0;
01456         szlDevice.cx = XMAX_PIXELS;
01457         szlDevice.cy = YMAX_PIXELS;
01458         szlMillimeters.cx = XMAX_MM;
01459         szlMillimeters.cy = YMAX_MM;
01460         //
01461         cbPixelFormat = 0;
01462         offPixelFormat = 0;
01463         bOpenGL = FALSE;
01464         //
01465 #if 1
01466         szlMicrometers.cx = 1000 * szlMillimeters.cx;
01467         szlMicrometers.cy = 1000 * szlMillimeters.cy;
01468 #endif
01469         if ( description ) {
01470             // Count the number of characters in the description
01471             int description_count = 0, nulls = 0;
01472             LPCWSTR description_p = description;
01473             while ( nulls < 3 ) {
01474                 description_count++;
01475                 if ( (*description_p++) == 0 ) nulls++;
01476             }
01477
01478             // Make sure that the TOTAL record length will be a multiple of 4
01479
01480             int record_size = ROUND_TO_LONG( sizeof( :ENHMETAHEADER ) +
01481                 sizeof( WCHAR ) * description_count );
01482             description_size =
01483                 (record_size - sizeof( :ENHMETAHEADER )) / sizeof( WCHAR );
01484
01485             std::unique_ptr<WCHAR[]>
01486                 description_tmp( new WCHAR[ description_size ] );
01487
01488             description_w = description_tmp.release();
01489
01490             memset( description_w, 0, sizeof(WCHAR) * description_size );
01491
01492             for ( int i=0; i<description_count; i++ )
01493                 description_w[i] = *description++;
01494
01495             nSize = nBytes = record_size;
01496             nDescription = description_count;
01497             offDescription = sizeof( :ENHMETAHEADER );
01498         }
01499     }
01500
01504     ~ENHMETAHEADER ( )
01505     {
01506         if ( description_w ) delete[] description_w;
01507     }
01512     bool serialize ( DATASTREAM ds )
01513     {
01514         ds « iType « nSize
01515         « rclBounds « rclFrame
01516         « dSignature « nVersion « nBytes « nRecords « nHandles « sReserved
01517         « nDescription « offDescription « nPalEntries
01518         « szlDevice « szlMillimeters
01519         « cbPixelFormat « offPixelFormat « bOpenGL
01520         « szlMicrometers
01521         « WCHARSTR( description_w, description_size );
01522         return true;
01523     }
01527     bool unserialize ( DATASTREAM ds )
01528     {
01529         ds » iType » nSize

```

```

01530     » rclBounds » rclFrame
01531     » dSignature » nVersion » nBytes » nRecords » nHandles » sReserved
01532     » nDescription » offDescription » nPalEntries
01533     » szlDevice » szlMillimeters;
01534
01535     // Some elements of the metafile header were added at later dates
01536
01537 #define OffsetOf( a, b ) ((unsigned int)(((char*)&((::ENHMETAHEADER*)a)->b)) - \
01538 (char*)((::ENHMETAHEADER*)a)))
01539 if ( OffsetOf( this, szlMicrometers ) <= offDescription )
01540     ds » cbPixelFormat » offPixelFormat » bOpenGL;
01541 #undef OffsetOf
01542 if ( sizeof(::ENHMETAHEADER) <= offDescription )
01543     ds » szlMicrometers;
01544
01545     // Should now probably check that the offset is correct...
01546
01547     int description_size_to_read = ( nSize - offDescription ) / sizeof(WCHAR);
01548
01549     if ( description_size_to_read < (int)nDescription ) {
01550         throw std::runtime_error( "record size inconsistent with description size" );
01551     }
01552
01553     description_size = max( 2, description_size_to_read );
01554
01555     std::unique_ptr<WCHAR[]> buffer( new WCHAR[description_size] );
01556
01557     WCHARSTR description( buffer.get(), description_size_to_read );
01558
01559     ds » description;
01560
01561     description_w = buffer.release();
01562
01563     // Make sure it's terminated properly.
01564     description_w[description_size-1] = 0;
01565     description_w[description_size-2] = 0;
01566
01567     return true;
01568 }
01569
01570 int size ( void ) const { return nSize; }
01571 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
01572 {
01573     // Actually handled by the destination device context.
01574     EMF_UNUSED(source);
01575     EMF_UNUSED(dc);
01576 }
01577
01578 #ifdef ENABLE_EDITING
01579 void edit ( void ) const
01580 {
01581     #if defined(__LP64__)
01582         const char* FMT0 = "\tiType\t\t\t: %d\n";
01583         const char* FMT1 = "\tnSize\t\t\t: %d\n";
01584         const char* FMT2 = "\tnBytes\t\t\t: %d\n";
01585         const char* FMT3 = "\tnRecords\t\t\t: %d\n";
01586         const char* FMT4 = "\tnDescription\t\t: %d\n";
01587         const char* FMT5 = "\toffDescription\t\t: %d\n";
01588         const char* FMT6 = "\tnPalEntries\t\t: %d\n";
01589         const char* FMT7 = "\tcbPixelFormat\t\t: %d\n";
01590         const char* FMT8 = "\toffPixelFormat\t\t: %d\n";
01591         const char* FMT9 = "\tbOpenGL\t\t\t: %d\n";
01592     #else
01593         const char* FMT0 = "\tiType\t\t\t: %ld\n";
01594         const char* FMT1 = "\tnSize\t\t\t: %ld\n";
01595         const char* FMT2 = "\tnBytes\t\t\t: %ld\n";
01596         const char* FMT3 = "\tnRecords\t\t\t: %ld\n";
01597         const char* FMT4 = "\tnDescription\t\t: %ld\n";
01598         const char* FMT5 = "\toffDescription\t\t: %ld\n";
01599         const char* FMT6 = "\tnPalEntries\t\t: %ld\n";
01600         const char* FMT7 = "\tcbPixelFormat\t\t: %ld\n";
01601         const char* FMT8 = "\toffPixelFormat\t\t: %ld\n";
01602         const char* FMT9 = "\tbOpenGL\t\t\t: %ld\n";
01603     #endif
01604
01605     printf( "*HEADER*\n" );
01606     printf( FMT0, iType );
01607     printf( FMT1, nSize );
01608     edit_rectl( "rclBounds\t", rclBounds );
01609     edit_rectl( "rclFrame\t", rclFrame );
01610     printf( "\tdSignature\t\t: %.4s\n", (const char*)&dSignature );
01611     printf( "\tnVersion\t\t: 0x%x\n", (unsigned int)nVersion );
01612     printf( FMT2, nBytes );
01613     printf( FMT3, nRecords );
01614     printf( "\tnHandles\t\t: %d\n", nHandles );
01615     printf( FMT4, nDescription );
01616     printf( FMT5, offDescription );
01617     printf( FMT6, nPalEntries );
01618     edit_sizel( "szlDevice\t", szlDevice );
01619     edit_sizel( "szlMillimeters\t", szlMillimeters );

```

```

01628
01629      /* Make a crude guess as to the age of this file */
01630 #define OffsetOf( a, b ) ((unsigned int)(((const char*)&((const ::ENHMETAHEADER*)a)->b)) - \
01631 (const char*)((const ::ENHMETAHEADER*)a)))
01632
01633      if ( OffsetOf( this, cbPixelFormat ) <= offDescription ) {
01634          printf( FMT7, cbPixelFormat );
01635          printf( FMT8, offPixelFormat );
01636          printf( FMT9, bOpenGL );
01637 #if 1
01638      if ( sizeof(::ENHMETAHEADER) <= offDescription ) {
01639          edit_size( "szlMicrometers\t", szlMicrometers );
01640      }
01641 #endif
01642      }
01643
01644 #undef OffsetOf
01645
01646      if ( nDescription != 0 ) {
01647          wchar_t last_w = 0;
01648          WCHAR* description = description_w;
01649
01650          printf( "\tDescription:" );
01651
01652          for ( DWORD i = 0; i < nDescription; i++ ) {
01653
01654              wchar_t w = *description++; /* This is not true, really. UNICODE is not
01655                                           * glibc's wide character representation */
01656
01657              if ( w != 0 ) {
01658                  if ( last_w == 0 ) printf( "\n\t\t" );
01659                  putchar( w );
01660              }
01661
01662              last_w = w;
01663          }
01664          printf( "\n" );
01665      }
01666 #endif /* ENABLE_EDITING */
01667 };
01668
01669 class EMREOF : public METARECORD, ::EMREOF {
01670 public:
01671     EMREOF ( void )
01672     {
01673         emr.iType = EMR_EOF;
01674         emr.nSize = sizeof( ::EMREOF );
01675         nPalEntries = 0;
01676         offPalEntries = 0;
01677         nSizeLast = 0;
01678     }
01679
01680     EMREOF ( DATASTREAM& ds )
01681     {
01682         ds » emr » nPalEntries » offPalEntries » nSizeLast;
01683     }
01684
01685     bool serialize ( DATASTREAM ds )
01686     {
01687         ds « emr « nPalEntries « offPalEntries « nSizeLast;
01688         return true;
01689     }
01690
01691     int size ( void ) const { return emr.nSize; }
01692     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
01693     {
01694         // Actually handled by the destination device context.
01695         EMF_UNUSED(source);
01696         EMF_UNUSED(dc);
01697     }
01698 #ifdef ENABLE_EDITING
01699     void edit ( void ) const
01700     {
01701         printf( "EOF*\n" );
01702     }
01703 #endif /* ENABLE_EDITING */
01704 };
01705
01706 class EMRSETVIEWPORTORGE : public METARECORD, ::EMRSETVIEWPORTORGE {
01707 public:
01708     EMRSETVIEWPORTORGE ( INT x, INT y )
01709     {
01710         emr.iType = EMR_SETVIEWPORTORGE;
01711         emr.nSize = sizeof( ::EMRSETVIEWPORTORGE );
01712     }

```

```

01750     ptlOrigin.x = x;
01751     ptlOrigin.y = y;
01752 }
01753 EMRSETVIEWPORTORGE ( DATASTREAM& ds )
01754 {
01755     ds » emr » ptlOrigin;
01756 }
01757 bool serialize ( DATASTREAM ds )
01758 {
01759     ds « emr « ptlOrigin;
01760     return true;
01761 }
01762 int size ( void ) const { return emr.nSize; }
01763 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
01764 {
01765     EMF_UNUSED(source);
01766     SetViewportOrgEx( dc, ptlOrigin.x, ptlOrigin.y, 0 );
01767 }
01768 #ifdef ENABLE_EDITING
01769 void edit ( void ) const
01770 {
01771     printf( "SETVIEWPORTORGE*\n" );
01772     edit_pointl( "ptlOrigin", ptlOrigin );
01773 }
01774 #endif /* ENABLE_EDITING */
01775 };
01776
01777 class EMRSETWINDOWORGE : public METARECORD, ::EMRSETWINDOWORGE {
01778 public:
01779     EMRSETWINDOWORGE ( INT x, INT y )
01780     {
01781         emr.iType = EMR_SETWINDOWORGE;
01782         emr.nSize = sizeof( ::EMRSETWINDOWORGE );
01783         ptlOrigin.x = x;
01784         ptlOrigin.y = y;
01785     }
01786     EMRSETWINDOWORGE ( DATASTREAM& ds )
01787     {
01788         ds » emr » ptlOrigin;
01789     }
01790     bool serialize ( DATASTREAM ds )
01791     {
01792         ds « emr « ptlOrigin;
01793         return true;
01794     }
01795     int size ( void ) const { return emr.nSize; }
01796     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
01797     {
01798         EMF_UNUSED(source);
01799         SetWindowOrgEx( dc, ptlOrigin.x, ptlOrigin.y, 0 );
01800     }
01801     #ifdef ENABLE_EDITING
01802     void edit ( void ) const
01803     {
01804         printf( "SETWINDOWORGE*\n" );
01805         edit_pointl( "ptlOrigin", ptlOrigin );
01806     }
01807     #endif /* ENABLE_EDITING */
01808 };
01809
01810 class EMRSETVIEWPORTEXT : public METARECORD, ::EMRSETVIEWPORTEXT {
01811 public:
01812     EMRSETVIEWPORTEXT ( INT cx, INT cy )
01813     {
01814         emr.iType = EMR_SETVIEWPORTEXT;
01815         emr.nSize = sizeof( ::EMRSETVIEWPORTEXT );
01816         szlExtent.cx = cx;
01817         szlExtent.cy = cy;
01818     }
01819     EMRSETVIEWPORTEXT ( DATASTREAM& ds )
01820     {
01821         ds » emr » szlExtent;
01822     }
01823     bool serialize ( DATASTREAM ds )
01824     {
01825         ds « emr « szlExtent;
01826         return true;
01827     }
01828     int size ( void ) const { return emr.nSize; }
01829     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
01830     {
01831         EMF_UNUSED(source);
01832         SetViewportExtEx( dc, szlExtent.cx, szlExtent.cy, 0 );
01833     }
01834     #ifdef ENABLE_EDITING
01835 
```

```

01911     void edit ( void ) const
01912     {
01913         printf( "*SETVIEWPORTETEX*\n" );
01914         edit_sizel( "szlExtent", szlExtent );
01915     }
01916 #endif /* ENABLE_EDITING */
01917 };
01918
01920
01925 class EMRSCALEVIEWPORTETEX : public METARECORD, ::EMRSCALEVIEWPORTETEX {
01926 public:
01933     EMRSCALEVIEWPORTETEX ( LONG x_num, LONG x_den, LONG y_num, LONG y_den )
01934     {
01935         emr.iType = EMR_SCALEVIEWPORTETEX;
01936         emr.nSize = sizeof( ::EMRSCALEVIEWPORTETEX );
01937         xNum = x_num;
01938         xDenom = x_den;
01939         yNum = y_num;
01940         yDenom = y_den;
01941     }
01946     EMRSCALEVIEWPORTETEX ( DATASTREAM& ds )
01947     {
01948         ds » emr » xNum » xDenom » yNum » yDenom;
01949     }
01953     bool serialize ( DATASTREAM ds )
01954     {
01955         ds « emr « xNum « xDenom « yNum « yDenom;
01956         return true;
01957     }
01961     int size ( void ) const { return emr.nSize; }
01967     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
01968     {
01969         EMF_UNUSED(source);
01970         ScaleViewportExtEx( dc, xNum, xDenom, yNum, yDenom, 0 );
01971     }
01972 #ifndef ENABLE_EDITING
01976     void edit ( void ) const
01977     {
01978         #if defined(__LP64__)
01979             const char* FMT0 = "\txNum\t: %d\n";
01980             const char* FMT1 = "\txDenom\t: %d\n";
01981             const char* FMT2 = "\tyNum\t: %d\n";
01982             const char* FMT3 = "\tyDenom\t: %d\n";
01983         #else
01984             const char* FMT0 = "\txNum\t: %ld\n";
01985             const char* FMT1 = "\txDenom\t: %ld\n";
01986             const char* FMT2 = "\tyNum\t: %ld\n";
01987             const char* FMT3 = "\tyDenom\t: %ld\n";
01988         #endif
01989         printf( "*SCALEVIEWPORTETEX*\n" );
01990         printf( FMT0, xNum );
01991         printf( FMT1, xDenom );
01992         printf( FMT2, yNum );
01993         printf( FMT3, yDenom );
01994     }
01995 #endif /* ENABLE_EDITING */
01996 };
01997
01999
02004 class EMRSETWINDOWEXTTEX : public METARECORD, ::EMRSETWINDOWEXTTEX {
02005 public:
02010     EMRSETWINDOWEXTTEX ( INT cx, INT cy )
02011     {
02012         emr.iType = EMR_SETWINDOWEXTTEX;
02013         emr.nSize = sizeof( ::EMRSETWINDOWEXTTEX );
02014         szlExtent.cx = cx;
02015         szlExtent.cy = cy;
02016     }
02021     EMRSETWINDOWEXTTEX ( DATASTREAM& ds )
02022     {
02023         ds » emr » szlExtent;
02024     }
02028     bool serialize ( DATASTREAM ds )
02029     {
02030         ds « emr « szlExtent;
02031         return true;
02032     }
02036     int size ( void ) const { return emr.nSize; }
02042     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02043     {
02044         EMF_UNUSED(source);
02045         SetWindowExtEx( dc, szlExtent.cx, szlExtent.cy, 0 );
02046     }
02047 #ifndef ENABLE_EDITING
02051     void edit ( void ) const
02052     {
02053         printf( "*SETWINDOWEXTTEX*\n" );

```

```

02054     edit_size( "szlExtent", szlExtent );
02055 }
02056 #endif /* ENABLE_EDITING */
02057 };
02058
02060
02065 class EMRSCALEWINDOWEXTEx : public METARECORD, ::EMRSCALEWINDOWEXTEx {
02066 public:
02073     EMRSCALEWINDOWEXTEx ( LONG x_num, LONG x_den, LONG y_num, LONG y_den )
02074     {
02075         emr.iType = EMR_SCALEWINDOWEXTEx;
02076         emr.nSize = sizeof( ::EMRSCALEWINDOWEXTEx );
02077         xNum = x_num;
02078         xDenom = x_den;
02079         yNum = y_num;
02080         yDenom = y_den;
02081     }
02086     EMRSCALEWINDOWEXTEx ( DATASTREAM& ds )
02087     {
02088         ds >> emr >> xNum >> xDenom >> yNum >> yDenom;
02089     }
02093     bool serialize ( DATASTREAM ds )
02094     {
02095         ds << emr << xNum << xDenom << yNum << yDenom;
02096         return true;
02097     }
02101     int size ( void ) const { return emr.nSize; }
02107     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02108     {
02109         EMF_UNUSED(source);
02110         ScaleWindowExtEx( dc, xNum, xDenom, yNum, yDenom, 0 );
02111     }
02112 #ifdef ENABLE_EDITING
02116     void edit ( void ) const
02117     {
02118         #if defined(__LP64__)
02119             const char* FMT0 = "\txNum\t: %d\n";
02120             const char* FMT1 = "\txDenom\t: %d\n";
02121             const char* FMT2 = "\tyNum\t: %d\n";
02122             const char* FMT3 = "\tyDenom\t: %d\n";
02123         #else
02124             const char* FMT0 = "\txNum\t: %ld\n";
02125             const char* FMT1 = "\txDenom\t: %ld\n";
02126             const char* FMT2 = "\tyNum\t: %ld\n";
02127             const char* FMT3 = "\tyDenom\t: %ld\n";
02128         #endif
02129         printf( "*SCALEWINDOWEXTEx*\n" );
02130         printf( FMT0, xNum );
02131         printf( FMT1, xDenom );
02132         printf( FMT2, yNum );
02133         printf( FMT3, yDenom );
02134     }
02135 #endif /* ENABLE_EDITING */
02136 };
02137
02139
02145 class EMRMODIFYWORLDTRANSFORM : public METARECORD, ::EMRMODIFYWORLDTRANSFORM {
02146 public:
02152     EMRMODIFYWORLDTRANSFORM ( const XFORM* transform, DWORD mode )
02153     {
02154         emr.iType = EMR_MODIFYWORLDTRANSFORM;
02155         emr.nSize = sizeof( ::EMRMODIFYWORLDTRANSFORM );
02156         xform = *transform;
02157         iMode = mode;
02158     }
02163     EMRMODIFYWORLDTRANSFORM ( DATASTREAM& ds )
02164     {
02165         ds >> emr >> xform >> iMode;
02166     }
02170     bool serialize ( DATASTREAM ds )
02171     {
02172         ds << emr << xform << iMode;
02173         return true;
02174     }
02178     int size ( void ) const { return emr.nSize; }
02184     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02185     {
02186         EMF_UNUSED(source);
02187         ModifyWorldTransform( dc, &xform, iMode );
02188     }
02189 #ifdef ENABLE_EDITING
02193     void edit ( void ) const
02194     {
02195         #if defined(__LP64__)
02196             const char* FMT = "unknown(%d)\n";
02197         #else
02198             const char* FMT = "unknown(%ld)\n";

```

```

02199 #endif /* __x86_64__ */
02200     printf( " *MODIFYWORLDTRANSFORM*\n" );
02201     edit_xform( "xform", xform );
02202     printf( "\tiMode\t\t: " );
02203     switch ( iMode ) {
02204     case MWT_IDENTITY: printf( "MWT_IDENTITY\n" ); break;
02205     case MWT_LEFTMULTIPLY: printf( "MWT_LEFTMULTIPLY\n" ); break;
02206     case MWT_RIGHTMULTIPLY: printf( "MWT_RIGHTMULTIPLY\n" ); break;
02207     default: printf( FMT, iMode );
02208     }
02209 }
02210 #endif /* ENABLE_EDITING */
02211 };
02212
02214
02220 class EMRSETWORLDTRANSFORM : public METARECORD, ::EMRSETWORLDTRANSFORM {
02221 public:
02222     EMRSETWORLDTRANSFORM ( const XFORM* transform )
02223     {
02224         emr.iType = EMR_SETWORLDTRANSFORM;
02225         emr.nSize = sizeof( ::EMRSETWORLDTRANSFORM );
02226         xform = *transform;
02227     }
02228     EMRSETWORLDTRANSFORM ( DATASTREAM& ds )
02229     {
02230         ds » emr » xform;
02231     }
02232     bool serialize ( DATASTREAM ds )
02233     {
02234         ds « emr « xform;
02235         return true;
02236     }
02237     int size ( void ) const { return emr.nSize; }
02238     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02239     {
02240         EMF_UNUSED(source);
02241         SetWorldTransform( dc, &xform );
02242     }
02243 #ifdef ENABLE_EDITING
02244     void edit ( void ) const
02245     {
02246         printf( " *SETWORLDTRANSFORM*\n" );
02247         edit_xform( "xform", xform );
02248     }
02249 #endif /* ENABLE_EDITING */
02250 };
02251
02254
02257 class EMRSETTEXTALIGN : public METARECORD, ::EMRSETTEXTALIGN {
02258 public:
02259     EMRSETTEXTALIGN ( UINT mode )
02260     {
02261         emr.iType = EMR_SETTEXTALIGN;
02262         emr.nSize = sizeof( ::EMRSETTEXTALIGN );
02263         iMode = mode;
02264     }
02265     EMRSETTEXTALIGN ( DATASTREAM& ds )
02266     {
02267         ds » emr » iMode;
02268     }
02269     bool serialize ( DATASTREAM ds )
02270     {
02271         ds « emr « iMode;
02272         return true;
02273     }
02274     int size ( void ) const { return emr.nSize; }
02275     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02276     {
02277         EMF_UNUSED(source);
02278         SetTextAlign( dc, iMode );
02279     }
02280 #ifdef ENABLE_EDITING
02281     void edit ( void ) const
02282     {
02283         #if defined(__LP64__)
02284             const char* FMT = "| unknown bits(0x%x)";
02285         #else
02286             const char* FMT = "| unknown bits(0x%lx)";
02287         #endif /* __x86_64__ */
02288         unsigned int known_bits = TA_BASELINE+TA_CENTER+TA_UPDATECP+TA_RTLCREADING;
02289         unsigned int unknown_bits = ~known_bits;
02290
02291         printf( " *SETTEXTALIGN*\n" );
02292         printf( "\tiMode\t\t: " );
02293         if ( iMode & TA_UPDATECP )
02294             printf( "TA_UPDATECP" );
02295         else

```

```

02337     printf( "TA_NOUPDATECP" );
02338     if ( iMode & TA_CENTER )
02339         printf( " | TA_CENTER" );
02340     else if ( iMode & TA_RIGHT )
02341         printf( " | TA_RIGHT" );
02342     else
02343         printf( " | TA_LEFT" );
02344     if ( iMode & TA_BASELINE )
02345         printf( " | TA_BASELINE" );
02346     else if ( iMode & TA_BOTTOM )
02347         printf( " | TA_BOTTOM" );
02348     else
02349         printf( " | TA_TOP" );
02350     if ( iMode & TA_RTLREADING )
02351         printf( " | TA_RTLREADING" );
02352     if ( iMode & unknown_bits )
02353         printf( FMT, iMode & unknown_bits );
02354     printf( "\n" );
02355 }
02356 #endif /* ENABLE_EDITING */
02357 };
02358
02360
02363 class EMRSETTEXTCOLOR : public METARECORD, ::EMRSETTEXTCOLOR {
02364 public:
02365     EMRSETTEXTCOLOR ( COLORREF color )
02366     {
02367         emr.iType = EMR_SETTEXTCOLOR;
02368         emr.nSize = sizeof( ::EMRSETTEXTCOLOR );
02369         crColor = color;
02370     }
02371     EMRSETTEXTCOLOR ( DATASTREAM& ds )
02372     {
02373         ds » emr » crColor;
02374     }
02375     bool serialize ( DATASTREAM ds )
02376     {
02377         ds « emr « crColor;
02378         return true;
02379     }
02380     int size ( void ) const { return emr.nSize; }
02381     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02382     {
02383         EMF_UNUSED(source);
02384         SetTextColor( dc, crColor );
02385     }
02386 #ifdef ENABLE_EDITING
02387     void edit ( void ) const
02388     {
02389         printf( "*SETTEXTCOLOR*\n" );
02390         edit_color( "crColor", crColor );
02391     }
02392 #endif /* ENABLE_EDITING */
02393 };
02394
02397 class EMRSETBKCOLOR : public METARECORD, ::EMRSETBKCOLOR {
02398 public:
02399     EMRSETBKCOLOR ( COLORREF color )
02400     {
02401         emr.iType = EMR_SETBKCOLOR;
02402         emr.nSize = sizeof( ::EMRSETBKCOLOR );
02403         crColor = color;
02404     }
02405     EMRSETBKCOLOR ( DATASTREAM& ds )
02406     {
02407         ds » emr » crColor;
02408     }
02409     bool serialize ( DATASTREAM ds )
02410     {
02411         ds « emr « crColor;
02412         return true;
02413     }
02414     int size ( void ) const { return emr.nSize; }
02415     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02416     {
02417         EMF_UNUSED(source);
02418         SetBkColor( dc, crColor );
02419     }
02420 #ifdef ENABLE_EDITING
02421     void edit ( void ) const
02422     {
02423         printf( "*SETBKCOLOR*\n" );
02424         edit_color( "crColor", crColor );
02425     }
02426 #endif /* ENABLE_EDITING */
02427 };

```

```

02472
02473
02478 class EMRSETBKMODE : public METARECORD, ::EMRSETBKMODE {
02479 public:
02483     EMRSETBKMODE ( DWORD mode )
02484     {
02485         emr.iType = EMR_SETBKMODE;
02486         emr.nSize = sizeof( ::EMRSETBKMODE );
02487         iMode = mode;
02488     }
02493     EMRSETBKMODE ( DATASTREAM& ds )
02494     {
02495         ds » emr » iMode;
02496     }
02500     bool serialize ( DATASTREAM ds )
02501     {
02502         ds « emr « iMode;
02503         return true;
02504     }
02508     int size ( void ) const { return emr.nSize; }
02514     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02515     {
02516         EMF_UNUSED(source);
02517         SetBkMode( dc, iMode );
02518     }
02519 #ifdef ENABLE_EDITING
02523     void edit ( void ) const
02524     {
02525 #if defined(__LP64__)
02526         const char* FMT = "unknown(%d)\n";
02527 #else
02528         const char* FMT = "unknown(%ld)\n";
02529 #endif /* __x86_64__ */
02530         printf( "*SETBKMODE*\n" );
02531         printf( "\tiMode\t: " );
02532         switch ( iMode ) {
02533             case TRANSPARENT: printf( "TRANSPARENT\n" ); break;
02534             case OPAQUE: printf( "OPAQUE\n" ); break;
02535             default: printf( FMT, iMode );
02536         }
02537     }
02538 #endif /* ENABLE_EDITING */
02539 };
02540
02542
02545 class EMRSETPOLYFILLMODE : public METARECORD, ::EMRSETPOLYFILLMODE {
02546 public:
02550     EMRSETPOLYFILLMODE ( DWORD mode )
02551     {
02552         emr.iType = EMR_SETPOLYFILLMODE;
02553         emr.nSize = sizeof( ::EMRSETPOLYFILLMODE );
02554         iMode = mode;
02555     }
02560     EMRSETPOLYFILLMODE ( DATASTREAM& ds )
02561     {
02562         ds » emr » iMode;
02563     }
02567     bool serialize ( DATASTREAM ds )
02568     {
02569         ds « emr « iMode;
02570         return true;
02571     }
02575     int size ( void ) const { return emr.nSize; }
02581     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02582     {
02583         EMF_UNUSED(source);
02584         SetPolyFillMode( dc, iMode );
02585     }
02586 #ifdef ENABLE_EDITING
02590     void edit ( void ) const
02591     {
02592 #if defined(__LP64__)
02593         const char* FMT = "unknown(%d)\n";
02594 #else
02595         const char* FMT = "unknown(%ld)\n";
02596 #endif /* __x86_64__ */
02597         printf( "*SETPOLYFILLMODE*\n" );
02598         printf( "\tiMode: " );
02599         switch ( iMode ) {
02600             case ALTERNATE: printf( "ALTERNATE\n" ); break;
02601             case WINDING: printf( "WINDING\n" ); break;
02602             default: printf( FMT, iMode );
02603         }
02604     }
02605 #endif /* ENABLE_EDITING */
02606 };
02607

```

```

02609
02613 class EMRSETMAPMODE : public METARECORD, ::EMRSETMAPMODE {
02614 public:
02618     EMRSETMAPMODE ( DWORD mode )
02619     {
02620         emr.iType = EMR_SETMAPMODE;
02621         emr.nSize = sizeof( ::EMRSETMAPMODE );
02622         iMode = mode;
02623     }
02628     EMRSETMAPMODE ( DATASTREAM& ds )
02629     {
02630         ds » emr » iMode;
02631     }
02635     bool serialize ( DATASTREAM ds )
02636     {
02637         ds « emr « iMode;
02638         return true;
02639     }
02643     int size ( void ) const { return emr.nSize; }
02649     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02650     {
02651         EMF_UNUSED(source);
02652         SetMapMode( dc, iMode );
02653     }
02654 #ifdef ENABLE_EDITING
02658     void edit ( void ) const
02659     {
02660 #if defined(__LP64__)
02661         const char* FMT = "unknown(%d)\n";
02662 #else
02663         const char* FMT = "unknown(%ld)\n";
02664 #endif /* __x86_64__ */
02665         printf( "*SETMAPMODE*\n" );
02666         printf( "\tiMode\t: " );
02667         switch ( iMode ) {
02668             case MM_TEXT: printf( "MM_TEXT\n" ); break;
02669             case MM_LOMETRIC: printf( "MM_LOMETRIC\n" ); break;
02670             case MM_HIMETRIC: printf( "MM_HIMETRIC\n" ); break;
02671             case MM_LOENGLISH: printf( "MM_LOENGLISH\n" ); break;
02672             case MM_HIENGLISH: printf( "MM_HIENGLISH\n" ); break;
02673             case MM_TWIPS: printf( "MM_TWIPS\n" ); break;
02674             case MM_ISOTROPIC: printf( "MM_ISOTROPIC\n" ); break;
02675             case MM_ANISOTROPIC: printf( "MM_ANISOTROPIC\n" ); break;
02676             default: printf( FMT, iMode );
02677         }
02678     }
02679 #endif /* ENABLE_EDITING */
02680 };
02681
02683
02686 class EMRSELECTOBJECT : public METARECORD, ::EMRSELECTOBJECT {
02687 public:
02691     EMRSELECTOBJECT ( HGDI OBJ object )
02692     {
02693         emr.iType = EMR_SELECTOBJECT;
02694         emr.nSize = sizeof( ::EMRSELECTOBJECT );
02695         ihObject = object;
02696     }
02701     EMRSELECTOBJECT ( DATASTREAM& ds )
02702     {
02703         ds » emr » ihObject;
02704     }
02708     bool serialize ( DATASTREAM ds )
02709     {
02710         ds « emr « ihObject;
02711         return true;
02712     }
02716     int size ( void ) const { return emr.nSize; }
02722     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const;
02723 #ifdef ENABLE_EDITING
02727     void edit ( void ) const
02728     {
02729 #if defined(__LP64__)
02730         const char* FMT = "\tihObject\t: 0x%x\n";
02731 #else
02732         const char* FMT = "\tihObject\t: 0x%lx\n";
02733 #endif /* __x86_64__ */
02734         printf( "*SELECTOBJECT*\n" );
02735         printf( FMT, ihObject );
02736     }
02737 #endif /* ENABLE_EDITING */
02738 };
02739
02741
02744 class EMRDELETEOBJECT : public METARECORD, ::EMRDELETEOBJECT {
02745 public:
02749     EMRDELETEOBJECT ( HGDI OBJ object )

```

```

02750     {
02751         emr.iType = EMR_DELETEOBJECT;
02752         emr.nSize = sizeof( ::EMRDELETEOBJECT );
02753         ihObject = object;
02754     }
02755     EMRDELETEOBJECT ( DATASTREAM& ds )
02756     {
02757         ds » emr » ihObject;
02758     }
02759     bool serialize ( DATASTREAM ds )
02760     {
02761         ds « emr « ihObject;
02762         return true;
02763     }
02764     int size ( void ) const { return emr.nSize; }
02765     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const;
02766 #ifdef ENABLE_EDITING
02767     void edit ( void ) const
02768     {
02769         #if defined(__LP64__)
02770             const char* FMT = "\\tihObject\\t: 0x%x\\n";
02771         #else
02772             const char* FMT = "\\tihObject\\t: 0x%lx\\n";
02773         #endif /* __x86_64__ */
02774         printf( "*DELETEOBJECT*\\n" );
02775         printf( FMT, ihObject );
02776     }
02777 #endif /* ENABLE_EDITING */
02778 };
02779
02780 class EMRMOVETOEX : public METARECORD, ::EMRMOVETOEX {
02781 public:
02782     EMRMOVETOEX ( INT x, INT y )
02783     {
02784         emr.iType = EMR_MOVETOEX;
02785         emr.nSize = sizeof( ::EMRMOVETOEX );
02786         ptl.x = x;
02787         ptl.y = y;
02788     }
02789     EMRMOVETOEX ( DATASTREAM& ds )
02790     {
02791         ds » emr » ptl;
02792     }
02793     bool serialize ( DATASTREAM ds )
02794     {
02795         ds « emr « ptl;
02796         return true;
02797     }
02798     int size ( void ) const { return emr.nSize; }
02799     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02800     {
02801         EMF_UNUSED(source);
02802         MoveToEx( dc, ptl.x, ptl.y, 0 );
02803     }
02804 #ifdef ENABLE_EDITING
02805     void edit ( void ) const
02806     {
02807         printf( "*MOVETOEX*\\n" );
02808         edit_pointl( "ptl", ptl );
02809     }
02810 #endif /* ENABLE_EDITING */
02811 };
02812
02813 class EMRLINETO : public METARECORD, ::EMRLINETO {
02814 public:
02815     EMRLINETO ( INT x, INT y )
02816     {
02817         emr.iType = EMR_LINETO;
02818         emr.nSize = sizeof( ::EMRLINETO );
02819         ptl.x = x;
02820         ptl.y = y;
02821     }
02822     EMRLINETO ( DATASTREAM& ds )
02823     {
02824         ds » emr » ptl;
02825     }
02826     bool serialize ( DATASTREAM ds )
02827     {
02828         ds « emr « ptl;
02829         return true;
02830     }
02831     int size ( void ) const { return emr.nSize; }
02832     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02833     {
02834         EMF_UNUSED(source);

```

```

02902         LineTo( dc, ptl.x, ptl.y );
02903     }
02904 #ifdef ENABLE_EDITING
02908     void edit ( void ) const
02909     {
02910         printf( ".*LINETO*\n" );
02911         edit_point1( "ptl", ptl );
02912     }
02913 #endif /* ENABLE_EDITING */
02914 };
02915
02917
02920     class EMRARC : public METARECORD, ::EMRARC {
02921     public:
02923         EMRARC ( INT left, INT top, INT right, INT bottom, INT xstart,
02924                 INT ystart, INT xend, INT yend )
02925         {
02926             emr.iType = EMR_ARC;
02927             emr.nSize = sizeof( ::EMRARC );
02928             rclBox.left = left;
02929             rclBox.right = right;
02930             rclBox.bottom = bottom;
02931             rclBox.top = top;
02932             ptlStart.x = xstart;
02933             ptlStart.y = ystart;
02934             ptlEnd.x = xend;
02935             ptlEnd.y = yend;
02936         }
02937         EMRARC ( DATASTREAM& ds )
02938         {
02939             ds » emr » rclBox » ptlStart » ptlEnd;
02940         }
02941         bool serialize ( DATASTREAM ds )
02942         {
02943             ds « emr « rclBox « ptlStart « ptlEnd;
02944             return true;
02945         }
02946         int size ( void ) const { return emr.nSize; }
02947         void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02948         {
02949             EMF_UNUSED(source);
02950             Arc( dc, rclBox.left, rclBox.top, rclBox.right, rclBox.bottom,
02951                 ptlStart.x, ptlStart.y, ptlEnd.x, ptlEnd.y );
02952         }
02953 #ifdef ENABLE_EDITING
02954         void edit ( void ) const
02955         {
02956             printf( ".*ARC*\n" );
02957             edit_rect1( "rclBox\t", rclBox );
02958             edit_point1( "ptlStart", ptlStart );
02959             edit_point1( "ptlEnd\t", ptlEnd );
02960         }
02961 #endif /* ENABLE_EDITING */
02962 };
02963
02966     class EMRARCTO : public METARECORD, ::EMRARCTO {
02967     public:
02969         EMRARCTO ( INT left, INT top, INT right, INT bottom, INT xstart,
02970                  INT ystart, INT xend, INT yend )
02971         {
02972             emr.iType = EMR_ARCTO;
02973             emr.nSize = sizeof( ::EMRARCTO );
02974             rclBox.left = left;
02975             rclBox.right = right;
02976             rclBox.bottom = bottom;
02977             rclBox.top = top;
02978             ptlStart.x = xstart;
02979             ptlStart.y = ystart;
02980             ptlEnd.x = xend;
02981             ptlEnd.y = yend;
02982         }
02983         EMRARCTO ( DATASTREAM& ds )
02984         {
02985             ds » emr » rclBox » ptlStart » ptlEnd;
02986         }
02987         bool serialize ( DATASTREAM ds )
02988         {
02989             ds « emr « rclBox « ptlStart « ptlEnd;
02990             return true;
02991         }
02992         int size ( void ) const { return emr.nSize; }
02993         void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
02994         {
02995             EMF_UNUSED(source);
02996             ArcTo( dc, rclBox.left, rclBox.top, rclBox.right, rclBox.bottom,
02997                   ptlStart.x, ptlStart.y, ptlEnd.x, ptlEnd.y );
02998         }
02999     };

```

```

03053     }
03054 #ifdef ENABLE_EDITING
03055     void edit ( void ) const
03056     {
03057         printf( "*ARCTO*\n" );
03058         edit_rectl( "rclBox\t", rclBox );
03059         edit_pointl( "ptlStart", ptlStart );
03060         edit_pointl( "ptlEnd\t", ptlEnd );
03061     }
03062 #endif /* ENABLE_EDITING */
03063 };
03064
03065 class EMRRECTANGLE : public METARECORD, ::EMRRECTANGLE {
03066 public:
03067     EMRRECTANGLE ( INT left, INT top, INT right, INT bottom )
03068     {
03069         emr.iType = EMR_RECTANGLE;
03070         emr.nSize = sizeof( ::EMRRECTANGLE );
03071         rclBox.left = left;
03072         rclBox.right = right;
03073         rclBox.bottom = bottom;
03074         rclBox.top = top;
03075     }
03076     EMRRECTANGLE ( DATASTREAM& ds )
03077     {
03078         ds » emr » rclBox;
03079     }
03080     bool serialize ( DATASTREAM ds )
03081     {
03082         ds « emr « rclBox;
03083         return true;
03084     }
03085     int size ( void ) const { return emr.nSize; }
03086     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
03087     {
03088         EMF_UNUSED(source);
03089         Rectangle( dc, rclBox.left, rclBox.top, rclBox.right, rclBox.bottom );
03090     }
03091 #ifdef ENABLE_EDITING
03092     void edit ( void ) const
03093     {
03094         printf( "*RECTANGLE*\n" );
03095         edit_rectl( "rclBox", rclBox );
03096     }
03097 #endif /* ENABLE_EDITING */
03098 };
03099
03100 class EMRELLIPSE : public METARECORD, ::EMRELLIPSE {
03101 public:
03102     EMRELLIPSE ( INT left, INT top, INT right, INT bottom )
03103     {
03104         emr.iType = EMR_ELLIPSE;
03105         emr.nSize = sizeof( ::EMRELLIPSE );
03106         rclBox.left = left;
03107         rclBox.right = right;
03108         rclBox.bottom = bottom;
03109         rclBox.top = top;
03110     }
03111     EMRELLIPSE ( DATASTREAM& ds )
03112     {
03113         ds » emr » rclBox;
03114     }
03115     bool serialize ( DATASTREAM ds )
03116     {
03117         ds « emr « rclBox;
03118         return true;
03119     }
03120     int size ( void ) const { return emr.nSize; }
03121     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
03122     {
03123         EMF_UNUSED(source);
03124         Ellipse( dc, rclBox.left, rclBox.top, rclBox.right, rclBox.bottom );
03125     }
03126 #ifdef ENABLE_EDITING
03127     void edit ( void ) const
03128     {
03129         printf( "*ELLIPSE*\n" );
03130         edit_rectl( "rclBox", rclBox );
03131     }
03132 #endif /* ENABLE_EDITING */
03133 };
03134
03135 class EMRPOLYLINE : public METARECORD, ::EMRPOLYLINE {
03136     POINTL* lpoints{nullptr};

```

```

03201 public:
03202     EMRPOLYLINE ( const RECTL* bounds, const POINT* points, INT n )
03203     {
03204         cptl = n;
03205         aptl[0].x = 0;           // Really unused
03206         aptl[0].y = 0;
03207
03208         emr.iType = EMR_POLYLINE;
03209         // The (cptl - 1) below is to account for aptl, which isn't written out
03210         emr.nSize = sizeof( ::EMRPOLYLINE ) + sizeof( POINTL ) * ( cptl - 1 );
03211
03212         lpoints = new POINTL[cptl];
03213
03214         for (int i=0; i<n; i++) {
03215             lpoints[i].x = points[i].x;
03216             lpoints[i].y = points[i].y;
03217         }
03218
03219         rclBounds = *bounds;
03220     }
03221 ~EMRPOLYLINE ( )
03222 {
03223     if ( lpoints ) delete[] lpoints;
03224 }
03225 EMRPOLYLINE ( DATASTREAM& ds )
03226 {
03227     ds » emr » rclBounds » cptl;
03228
03229     if ( emr.nSize - (sizeof(::EMRPOLYLINE)-sizeof(POINTL) ) <
03230         sizeof(POINTL) * cptl ) {
03231         throw std::runtime_error( "Invalid record size" );
03232     }
03233
03234     std::unique_ptr<POINTL[]> buffer( new POINTL[cptl] );
03235     POINTLARRAY points( buffer.get(), cptl );
03236
03237     ds » points;
03238
03239     lpoints = buffer.release();
03240 }
03241 bool serialize ( DATASTREAM ds )
03242 {
03243     ds « emr « rclBounds « cptl « POINTLARRAY( lpoints, cptl );
03244     return true;
03245 }
03246 int size ( void ) const { return emr.nSize; }
03247 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
03248 {
03249     EMF_UNUSED(source);
03250     // According to the wine windef.h header, POINT and POINTL are equivalent
03251     Polyline( dc, (POINT*)lpoints, cptl );
03252 }
03253 #ifdef ENABLE_EDITING
03254 void edit ( void ) const
03255 {
03256     printf( "*POLYLINE*\n" );
03257     edit_rectl( "rclBounds", rclBounds );
03258     edit_pointlarray( "\t", cptl, lpoints );
03259 }
03260 #endif /* ENABLE_EDITING */
03261 };
03262
03263 class EMRPOLYLINE16 : public METARECORD, ::EMRPOLYLINE16 {
03264     POINT16* lpoints{ nullptr };
03265 public:
03266     EMRPOLYLINE16 ( const RECTL* bounds, const POINT16* points, INT n )
03267     {
03268         cpts = n;
03269         apts[0].x = 0;           // Really unused
03270         apts[0].y = 0;
03271
03272         emr.iType = EMR_POLYLINE16;
03273         // The (cptl - 1) below is to account for aptl, which isn't written out
03274         emr.nSize = sizeof( ::EMRPOLYLINE16 ) + sizeof( POINT16 ) * ( cpts - 1 );
03275
03276         lpoints = new POINT16[cpts];
03277
03278         for (int i=0; i<n; i++) {
03279             lpoints[i].x = points[i].x;
03280             lpoints[i].y = points[i].y;
03281         }
03282
03283         rclBounds = *bounds;
03284     }
03285     EMRPOLYLINE16 ( const RECTL* bounds, const POINT* points, INT n )
03286     {

```

```

03328     cpts = n;
03329     apts[0].x = 0;           // Really unused
03330     apts[0].y = 0;
03331
03332     emr.iType = EMR_POLYLINE16;
03333     // The (cptl - 1) below is to account for aptl, which isn't written out
03334     emr.nSize = sizeof( ::EMRPOLYLINE16 ) + sizeof( POINT16 ) * ( cpts - 1 );
03335
03336     lpoints = new POINT16[cpts];
03337
03338     for (int i=0; i<n; i++) {
03339         lpoints[i].x = points[i].x;
03340         lpoints[i].y = points[i].y;
03341     }
03342
03343     rclBounds = *bounds;
03344 }
03345 ~EMRPOLYLINE16 ( )
03346 {
03347     if ( lpoints ) delete[] lpoints;
03348 }
03349 EMRPOLYLINE16 ( DATASTREAM& ds )
03350 {
03351     ds » emr » rclBounds » cpts;
03352
03353     if ( emr.nSize - (sizeof(::EMRPOLYLINE16)-sizeof(POINT16)) <
03354           sizeof(POINT16) * cpts ) {
03355         throw std::runtime_error( "Invalid record size" );
03356     }
03357
03358     std::unique_ptr<POINT16[]> buffer( new POINT16[cpts] );
03359
03360     POINT16ARRAY points( buffer.get(), cpts );
03361
03362     ds » points;
03363
03364     lpoints = buffer.release();
03365 }
03366 bool serialize ( DATASTREAM ds )
03367 {
03368     ds « emr « rclBounds « cpts « POINT16ARRAY( lpoints, cpts );
03369     return true;
03370 }
03371 int size ( void ) const { return emr.nSize; }
03372 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
03373 {
03374     EMF_UNUSED(source);
03375     // According to the wine windef.h header, POINT and POINTL are equivalent
03376     Polyline16( dc, lpoints, cpts );
03377 }
03378 #ifdef ENABLE_EDITING
03379 void edit ( void ) const
03380 {
03381     printf( "*POLYLINE16*\n" );
03382     edit_rectl( "rclBounds", rclBounds );
03383     edit_point16array( "\t", cpts, lpoints );
03384 }
03385 #endif /* ENABLE_EDITING */
03386 };
03387
03388 class EMRPOLYGON : public METARECORD, ::EMRPOLYGON {
03389     POINTL* lpoints{ nullptr };
03390 public:
03391     EMRPOLYGON ( const RECTL* bounds, const POINT* points, INT n )
03392     {
03393         cptl = n;
03394         aptl[0].x = 0;           // Really unused
03395         aptl[0].y = 0;
03396
03397         emr.iType = EMR_POLYGON;
03398         // The (cptl-1) below is to account for aptl, which isn't written out
03399         emr.nSize = sizeof( ::EMRPOLYGON ) + sizeof( POINTL ) * (cptl-1);
03400
03401         lpoints = new POINTL[cptl];
03402
03403         for (int i=0; i<n; i++) {
03404             lpoints[i].x = points[i].x;
03405             lpoints[i].y = points[i].y;
03406         }
03407
03408         rclBounds = *bounds;
03409     }
03410     EMRPOLYGON ( DATASTREAM& ds )
03411     {
03412         ds » emr » rclBounds » cptl;
03413     }

```

```

03448     if ( emr.nSize - (sizeof(::EMRPOLYGON) - sizeof(POINTL)) <
03449         cptl * sizeof(POINTL) ) {
03450         throw std::runtime_error( "Invalid record size" );
03451     }
03452
03453     std::unique_ptr<POINTL[]> buffer( new POINTL[cptl] );
03454
03455     POINTLARRAY points( buffer.get(), cptl );
03456
03457     ds » points;
03458
03459     lpoints = buffer.release();
03460 }
03461 ~EMRPOLYGON ( )
03462 {
03463     if ( lpoints ) delete[] lpoints;
03464 }
03465 bool serialize ( DATASTREAM ds )
03466 {
03467     ds « emr « rclBounds « cptl « POINTLARRAY( lpoints, cptl );
03468     return true;
03469 }
03470 int size ( void ) const { return emr.nSize; }
03471 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
03472 {
03473     EMF_UNUSED(source);
03474     // According to the wine windef.h header, POINT and POINTL are equivalent
03475     Polygon( dc, (POINT*)lpoints, cptl );
03476 }
03477 #ifdef ENABLE_EDITING
03478 void edit ( void ) const
03479 {
03480     printf( "POLYGON\n" );
03481     edit_rectl( "rclBounds", rclBounds );
03482     edit_pointlarray( "\t", cptl, lpoints );
03483 }
03484 #endif /* ENABLE_EDITING */
03485 };
03486
03487 class EMRPOLYGON16 : public METARECORD, ::EMRPOLYGON16 {
03488     POINT16* lpoints{ nullptr };
03489 public:
03490     EMRPOLYGON16 ( const RECTL* bounds, const POINT* points, INT16 n )
03491     {
03492         cpts = n;
03493         apts[0].x = 0;           // Really unused
03494         apts[0].y = 0;
03495
03496         emr.iType = EMR_POLYGON16;
03497         // The (cptl-1) below is to account for aptl, which isn't written out
03498         emr.nSize = sizeof( ::EMRPOLYGON16 ) + sizeof( POINT16 ) * (cpts-1);
03499
03500         lpoints = new POINT16[cpts];
03501
03502         for (int i=0; i<n; i++) {
03503             lpoints[i].x = points[i].x;
03504             lpoints[i].y = points[i].y;
03505         }
03506
03507         rclBounds = *bounds;
03508     }
03509     EMRPOLYGON16 ( const RECTL* bounds, const POINT16* points, INT16 n )
03510     {
03511         cpts = n;
03512         apts[0].x = 0;           // Really unused
03513         apts[0].y = 0;
03514
03515         emr.iType = EMR_POLYGON16;
03516         // The (cptl-1) below is to account for aptl, which isn't written out
03517         emr.nSize = sizeof( ::EMRPOLYGON16 ) + sizeof( POINT16 ) * (cpts-1);
03518
03519         lpoints = new POINT16[cpts];
03520
03521         for (int i=0; i<n; i++) {
03522             lpoints[i].x = points[i].x;
03523             lpoints[i].y = points[i].y;
03524         }
03525
03526         rclBounds = *bounds;
03527     }
03528     EMRPOLYGON16 ( DATASTREAM& ds )
03529     {
03530         ds » emr » rclBounds » cpts;
03531
03532         if ( emr.nSize - (sizeof(::EMRPOLYGON16) - sizeof(POINT16)) <
03533             cpts * sizeof(POINT16) ) {

```

```

03570         throw std::runtime_error( "Invalid record size" );
03571     }
03572
03573     std::unique_ptr<POINT16[]> buffer( new POINT16[cpts] );
03574
03575     POINT16ARRAY points( buffer.get(), cpts );
03576
03577     ds » points;
03578
03579     lpoints = buffer.release();
03580 }
03581 ~EMRPOLYGON16 ( )
03582 {
03583     if ( lpoints ) delete[] lpoints;
03584 }
03585 bool serialize ( DATASTREAM ds )
03586 {
03587     ds « emr « rclBounds « cpts « POINT16ARRAY( lpoints, cpts );
03588     return true;
03589 }
03590 int size ( void ) const { return emr.nSize; }
03591 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
03592 {
03593     EMF_UNUSED(source);
03594     // According to the wine windef.h header, POINT and POINTL are equivalent
03595     Polygon16( dc, lpoints, cpts );
03596 }
03597 #ifdef ENABLE_EDITING
03598 void edit ( void ) const
03599 {
03600     printf( "*POLYGON16*\n" );
03601     edit_rectl( "rclBounds", rclBounds );
03602     edit_point16array( "\t", cpts, lpoints );
03603 }
03604 #endif /* ENABLE_EDITING */
03605 };
03606
03607 class EMRPOLYPOLYGON : public METARECORD, ::EMRPOLYPOLYGON {
03608     DWORD* lcounts{ nullptr };
03609     POINTL* lpoints{ nullptr };
03610 public:
03611     EMRPOLYPOLYGON ( const RECTL* bounds, const POINT* points, const INT* counts,
03612                     UINT polygons )
03613     {
03614         nPolys = polygons;
03615         // Count the number of points in points
03616         int n = 0;
03617         for ( unsigned int i = 0; i < nPolys; i++ )
03618             n += counts[i];
03619
03620         cptl = n;
03621         aPolyCounts[0] = 0; // Really unused
03622         aptl[0].x = 0;
03623         aptl[0].y = 0;
03624
03625         emr.iType = EMR_POLYPOLYGON;
03626         // The (#-1)'s below are to account for aPolyCounts[0] and aptl[0], which
03627         // aren't directly written out
03628         emr.nSize = sizeof( ::EMRPOLYPOLYGON ) + sizeof( POINTL ) * (cptl-1)
03629         + sizeof( DWORD ) * (nPolys-1);
03630
03631         lcounts = new DWORD[nPolys];
03632
03633         for ( unsigned int i = 0; i < nPolys; i++ )
03634             lcounts[i] = counts[i];
03635
03636         lpoints = new POINTL[cptl];
03637
03638         for (int i=0; i<n; i++) {
03639             lpoints[i].x = points[i].x;
03640             lpoints[i].y = points[i].y;
03641         }
03642
03643         rclBounds = *bounds;
03644     }
03645     ~EMRPOLYPOLYGON ( )
03646     {
03647         if ( lcounts ) delete[] lcounts;
03648         if ( lpoints ) delete[] lpoints;
03649     }
03650     EMRPOLYPOLYGON ( DATASTREAM& ds )
03651     {
03652         ds » emr » rclBounds » nPolys » cptl;
03653
03654         if ( emr.nSize - ( sizeof( ::EMRPOLYPOLYGON ) - sizeof(POINTL) - sizeof(DWORD) ) <
03655             sizeof( POINTL ) * cptl + sizeof( DWORD ) * nPolys ) {

```

```

03690         throw std::runtime_error( "Invalid record size" );
03691     }
03692
03693     std::unique_ptr<DWORD[]> cbuffer( new DWORD[nPolys] );
03694
03695     DWORDARRAY counts( cbuffer.get(), nPolys );
03696
03697     ds » counts;
03698
03699     // Counts have to add up to less than the number of points
03700     // we have. DWORD is unsigned so we most care about overflow.
03701     DWORD n{0}, n_old{0};
03702     for ( DWORD c{0}; c < nPolys; ++c ) {
03703         n_old = n;
03704         n += cbuffer[c];
03705         if ( n < n_old ) {
03706             throw std::runtime_error( "Unsigned overflow" );
03707         }
03708     }
03709     if ( n > cptl ) {
03710         throw std::runtime_error( "Too few points" );
03711     }
03712
03713     std::unique_ptr<POINTL[]> pBuffer( new POINTL[cptl] );
03714
03715     POINTLARRAY points( pBuffer.get(), cptl );
03716
03717     ds » points;
03718
03719     // Don't do this until we won't have any more exceptions.
03720     lcounts = cbuffer.release();
03721     lpoints = pBuffer.release();
03722 }
03726 bool serialize ( DATASTREAM ds )
03727 {
03728     ds « emr « rclBounds « nPolys « cptl « DWORDARRAY( lcounts, nPolys )
03729     « POINTLARRAY( lpoints, cptl );
03730     return true;
03731 }
03735 int size ( void ) const { return emr.nSize; }
03741 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
03742 {
03743     EMF_UNUSED(source);
03744     // According to the wine windef.h header, POINT and POINTL are equivalent
03745     // (but DWORD and INT are not)
03746     std::vector<INT> countsv( lcounts, lcounts + nPolys );
03747
03748     PolyPolygon( dc, (POINT*)lpoints, countsv.data(), nPolys );
03749 }
03750 #ifdef ENABLE_EDITING
03754 void edit ( void ) const
03755 {
03756     #if defined(__LP64__)
03757         const char* FMT0 = "\tnPolys\t\t: %d\n";
03758         const char* FMT1 = "\tcptl\t\t: %d\n";
03759         const char* FMT2 = "%d\n";
03760         const char* FMT3 = "\t\t\t\t %d\n";
03761         const char* FMT4 = "%d, %d\n";
03762         const char* FMT5 = "\t\t\t\t %d, %d\n";
03763     #else
03764         const char* FMT0 = "\tnPolys\t\t: %ld\n";
03765         const char* FMT1 = "\tcptl\t\t: %ld\n";
03766         const char* FMT2 = "%ld\n";
03767         const char* FMT3 = "\t\t\t\t %ld\n";
03768         const char* FMT4 = "%ld, %ld\n";
03769         const char* FMT5 = "\t\t\t\t %ld, %ld\n";
03770     #endif /* __x86_64__ */
03771     printf( "*POLYPOLYGON*\n" );
03772     edit_rectl( "rclBounds", rclBounds );
03773     printf( FMT0, nPolys );
03774     printf( FMT1, cptl );
03775     printf( "\taPolyCounts\t: " );
03776     if ( nPolys > 0 )
03777         printf( FMT2, lcounts[0] );
03778     else
03779         puts( "" );
03780     for ( unsigned int i = 1; i < nPolys; i++ )
03781         printf( FMT3, lcounts[i] );
03782     printf( "\taps\t\t: " );
03783     if ( cptl > 0 )
03784         printf( FMT4, lpoints[0].x, lpoints[0].y );
03785     else
03786         puts( "" );
03787     for ( unsigned int i = 1; i < cptl; i++ )
03788         printf( FMT5, lpoints[i].x, lpoints[i].y );
03789 }
03790 #endif /* ENABLE_EDITING */

```

```

03791     };
03792
03793
03794
03797     class EMRPOLYPOLYGON16 : public METARECORD, ::EMRPOLYPOLYGON16 {
03798     DWORD* lcounts{ nullptr };
03799     POINT16* lpoints{ nullptr };
03800 public:
03807     EMRPOLYPOLYGON16 ( const RECTL* bounds, const POINT* points,
03808                       const INT* counts, UINT polygons )
03809     {
03810         nPolys = polygons;
03811         // Count the number of points in points
03812         int n = 0;
03813         for ( unsigned int i = 0; i < nPolys; i++ )
03814             n += counts[i];
03815
03816         cpts = n;
03817         aPolyCounts[0] = 0;    // Really unused
03818         apts[0].x = 0;
03819         apts[0].y = 0;
03820
03821         emr.iType = EMR_POLYPOLYGON16;
03822         // The (#-1)'s below are to account for aPolyCounts[0] and aptl[0], which
03823         // aren't directly written out
03824         emr.nSize = sizeof( ::EMRPOLYPOLYGON16 ) + sizeof( POINT16 ) * (cpts-1)
03825         + sizeof( DWORD ) * (nPolys-1);
03826
03827         lcounts = new DWORD[nPolys];
03828
03829         for ( unsigned int i = 0; i < nPolys; i++ )
03830             lcounts[i] = counts[i];
03831
03832         lpoints = new POINT16[cpts];
03833
03834         for (int i=0; i<n; i++) {
03835             lpoints[i].x = points[i].x;
03836             lpoints[i].y = points[i].y;
03837         }
03838
03839         rclBounds = *bounds;
03840     }
03841     EMRPOLYPOLYGON16 ( const RECTL* bounds, const POINT16* points,
03842                       const INT* counts, UINT16 polygons )
03843     {
03844         nPolys = polygons;
03845         // Count the number of points in points
03846         int n = 0;
03847         for ( unsigned int i = 0; i < nPolys; i++ )
03848             n += counts[i];
03849
03850         cpts = n;
03851         aPolyCounts[0] = 0;    // Really unused
03852         apts[0].x = 0;
03853         apts[0].y = 0;
03854
03855         emr.iType = EMR_POLYPOLYGON16;
03856         // The (#-1)'s below are to account for aPolyCounts[0] and aptl[0], which
03857         // aren't directly written out
03858         emr.nSize = sizeof( ::EMRPOLYPOLYGON16 ) + sizeof( POINT16 ) * (cpts-1)
03859         + sizeof( DWORD ) * (nPolys-1);
03860
03861         lcounts = new DWORD[nPolys];
03862
03863         for ( unsigned int i = 0; i < nPolys; i++ )
03864             lcounts[i] = counts[i];
03865
03866         lpoints = new POINT16[cpts];
03867
03868         for (int i=0; i<n; i++) {
03869             lpoints[i].x = points[i].x;
03870             lpoints[i].y = points[i].y;
03871         }
03872
03873         rclBounds = *bounds;
03874     }
03875     ~EMRPOLYPOLYGON16 ( )
03876     {
03877         if ( lcounts ) delete[] lcounts;
03878         if ( lpoints ) delete[] lpoints;
03879     }
03880     EMRPOLYPOLYGON16 ( DATASTREAM& ds )
03881     {
03882         ds » emr » rclBounds » nPolys » cpts;
03883
03884         if ( emr.nSize - ( sizeof( ::EMRPOLYPOLYGON16 ) - sizeof(POINT16) - sizeof(DWORD) ) <
03885             sizeof( POINT16 ) * cpts + sizeof( DWORD ) * nPolys ) {
03886             throw std::runtime_error( "Invalid record size " );
03887         }
03888     }
03889
03890
03891

```

```

03901     }
03902
03903     std::unique_ptr<DWORD[]> cbuffer( new DWORD[nPolys] );
03904
03905     DWORDARRAY counts( cbuffer.get(), nPolys );
03906
03907     ds » counts;
03908
03909     // Counts have to add up to less than the number of points
03910     // we have. DWORD is unsigned so we must care about overflow.
03911     DWORD n{0}, n_old{0};
03912     for ( DWORD c{0}; c < nPolys; ++c ) {
03913         n_old = n;
03914         n += cbuffer[c];
03915         if ( n < n_old ) {
03916             throw std::runtime_error( "Unsigned overflow" );
03917         }
03918     }
03919     if ( n > cpts ) {
03920         throw std::runtime_error( "Too few points" );
03921     }
03922
03923     std::unique_ptr<POINT16[]> pBuffer( new POINT16[cpts] );
03924
03925     POINT16ARRAY points( pBuffer.get(), cpts );
03926
03927     ds » points;
03928
03929     lcounts = cbuffer.release();
03930     lpoints = pBuffer.release();
03931 }
03932
03933 bool serialize ( DATASTREAM ds )
03934 {
03935     ds « emr « rclBounds « nPolys « cpts « DWORDARRAY( lcounts, nPolys )
03936     « POINT16ARRAY( lpoints, cpts );
03937     return true;
03938 }
03939
03940 int size ( void ) const { return emr.nSize; }
03941 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
03942 {
03943     EMF_UNUSED(source);
03944     // According to the wine windef.h header, POINT and POINTL are equivalent
03945     // (but DWORD and INT are not)
03946     std::vector<INT> counts( lcounts, lcounts + nPolys );
03947
03948     PolyPolygon16( dc, lpoints, counts.data(), nPolys );
03949 }
03950
03951 #ifdef ENABLE_EDITING
03952 void edit ( void ) const
03953 {
03954     #if defined(__LP64__)
03955         const char* FMT0 = "\tnPolys\t\t: %d\n";
03956         const char* FMT1 = "\tcptl\t\t: %d\n";
03957         const char* FMT2 = "%d\n";
03958         const char* FMT3 = "\t\t\t %d\n";
03959     #else
03960         const char* FMT0 = "\tnPolys\t\t: %ld\n";
03961         const char* FMT1 = "\tcptl\t\t: %ld\n";
03962         const char* FMT2 = "%ld\n";
03963         const char* FMT3 = "\t\t\t %ld\n";
03964     #endif /* __x86_64__ */
03965     printf( "*POLYPOLYGON16*\n" );
03966     edit_rectl( "rclBounds", rclBounds );
03967     printf( FMT0, nPolys );
03968     printf( FMT1, cpts );
03969     printf( "\taPolyCounts\t: " );
03970     if ( nPolys > 0 )
03971         printf( FMT2, lcounts[0] );
03972     else
03973         puts( "" );
03974     for ( unsigned int i = 1; i < nPolys; i++ )
03975         printf( FMT3, lcounts[i] );
03976     printf( "\taps\t\t: " );
03977     if ( cpts > 0 )
03978         printf( "%d, %d\n", lpoints[0].x, lpoints[0].y );
03979     else
03980         puts( "" );
03981     for ( unsigned int i = 1; i < cpts; i++ )
03982         printf( "\t\t\t %d, %d\n", lpoints[i].x, lpoints[i].y );
03983 }
03984 #endif /* ENABLE_EDITING */
03985 };
03986
03987
03988
03989
04000 class EMRPOLYBEZIER : public METARECORD, ::EMRPOLYBEZIER {
04001     POINTL* lpoints{ nullptr };
04002 public:

```

```

04010     EMRPOLYBEZIER ( const RECTL* bounds, const POINT* points, INT n )
04011     {
04012         cptl = n;
04013         aptl[0].x = 0;           // Really unused
04014         aptl[0].y = 0;
04015
04016         emr.iType = EMR_POLYBEZIER;
04017         // The (cptl-1) below is to account for aptl, which isn't written out
04018         emr.nSize = sizeof( ::EMRPOLYBEZIER ) + sizeof( POINTL ) * (cptl-1);
04019
04020         lpoints = new POINTL[cptl];
04021
04022         for (int i=0; i<n; i++) {
04023             lpoints[i].x = points[i].x;
04024             lpoints[i].y = points[i].y;
04025         }
04026
04027         rclBounds = *bounds;
04028     }
04029     EMRPOLYBEZIER ( DATASTREAM& ds )
04030     {
04031         ds » emr » rclBounds » cptl;
04032
04033         if ( emr.nSize - (sizeof( ::EMRPOLYBEZIER ) - sizeof(POINTL)) <
04034             sizeof( POINTL ) * cptl ) {
04035             throw std::runtime_error( "Invalid record size " );
04036         }
04037
04038         std::unique_ptr<POINTL[]> buffer( new POINTL[cptl] );
04039
04040         POINTLARRAY points( buffer.get(), cptl );
04041
04042         ds » points;
04043
04044         lpoints = buffer.release();
04045     }
04046     ~EMRPOLYBEZIER ( )
04047     {
04048         if ( lpoints ) delete[] lpoints;
04049     }
04050     bool serialize ( DATASTREAM ds )
04051     {
04052         ds « emr « rclBounds « cptl « POINTLARRAY( lpoints, cptl );
04053         return true;
04054     }
04055     int size ( void ) const { return emr.nSize; }
04056     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
04057     {
04058         EMF_UNUSED(source);
04059         // According to the wine windef.h header, POINT and POINTL are equivalent
04060         PolyBezier( dc, (POINT*)lpoints, cptl );
04061     }
04062 #ifdef ENABLE_EDITING
04063     void edit ( void ) const
04064     {
04065         printf( "%POLYBEZIER*\n" );
04066         edit_rectl( "rclBounds", rclBounds );
04067         edit_pointlarray( "\t", cptl, lpoints );
04068     }
04069 #endif /* ENABLE_EDITING */
04070 };
04071
04072 class EMRPOLYBEZIER16 : public METARECORD, ::EMRPOLYBEZIER16 {
04073     POINT16* lpoints{ nullptr };
04074 public:
04075     EMRPOLYBEZIER16 ( const RECTL* bounds, const POINT16* points, INT n )
04076     {
04077         cpts = n;
04078         apts[0].x = 0;           // Really unused
04079         apts[0].y = 0;
04080
04081         emr.iType = EMR_POLYBEZIER16;
04082         // The (cptl-1) below is to account for aptl, which isn't written out
04083         emr.nSize = sizeof( ::EMRPOLYBEZIER16 ) + sizeof( POINT16 ) * (cpts-1);
04084
04085         lpoints = new POINT16[cpts];
04086
04087         for (int i=0; i<n; i++) {
04088             lpoints[i].x = points[i].x;
04089             lpoints[i].y = points[i].y;
04090         }
04091
04092         rclBounds = *bounds;
04093     }
04094     EMRPOLYBEZIER16 ( const RECTL* bounds, const POINT* points, INT n )
04095     {

```

```

04132     cpts = n;
04133     apts[0].x = 0;           // Really unused
04134     apts[0].y = 0;
04135
04136     emr.iType = EMR_POLYBEZIER16;
04137     // The (cptl-1) below is to account for aptl, which isn't written out
04138     emr.nSize = sizeof( ::EMRPOLYBEZIER16 ) + sizeof( POINT16 ) * (cpts-1);
04139
04140     lpoints = new POINT16[cpts];
04141
04142     for (int i=0; i<n; i++) {
04143         lpoints[i].x = points[i].x;
04144         lpoints[i].y = points[i].y;
04145     }
04146
04147     rclBounds = *bounds;
04148 }
04153 EMRPOLYBEZIER16 ( DATASTREAM& ds )
04154 {
04155     ds » emr » rclBounds » cpts;
04156
04157     if ( emr.nSize - (sizeof( ::EMRPOLYBEZIER16 ) - sizeof(POINT16)) <
04158         sizeof( POINT16 ) * cpts ) {
04159         throw std::runtime_error( "Invalid record size " );
04160     }
04161
04162     std::unique_ptr<POINT16[]> buffer( new POINT16[cpts] );
04163
04164     POINT16ARRAY points( buffer.get(), cpts );
04165
04166     ds » points;
04167
04168     lpoints = buffer.release();
04169 }
04173 ~EMRPOLYBEZIER16 ( )
04174 {
04175     if ( lpoints ) delete[] lpoints;
04176 }
04180 bool serialize ( DATASTREAM ds )
04181 {
04182     ds « emr « rclBounds « cpts « POINT16ARRAY( lpoints, cpts );
04183     return true;
04184 }
04188 int size ( void ) const { return emr.nSize; }
04194 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
04195 {
04196     EMF_UNUSED(source);
04197     // According to the wine windef.h header, POINT and POINTL are equivalent
04198     PolyBezier16( dc, lpoints, cpts );
04199 }
04200 #ifdef ENABLE_EDITING
04204 void edit ( void ) const
04205 {
04206     printf( "*POLYBEZIER16*\n" );
04207     edit_rectl( "rclBounds", rclBounds );
04208     edit_point16array( "\t", cpts, lpoints );
04209 }
04210 #endif /* ENABLE_EDITING */
04211 };
04212
04214
04217 class EMRPOLYBEZIERTO : public METARECORD, ::EMRPOLYBEZIER {
04218     POINTL* lpoints{ nullptr };
04219 public:
04225     EMRPOLYBEZIERTO ( const RECTL* bounds, const POINT* points, INT n )
04226     {
04227         cptl = n;
04228         aptl[0].x = 0;           // Really unused
04229         aptl[0].y = 0;
04230
04231         emr.iType = EMR_POLYBEZIERTO;
04232         // The (cptl-1) below is to account for aptl, which isn't written out
04233         emr.nSize = sizeof( ::EMRPOLYBEZIERTO ) + sizeof( POINTL ) * (cptl-1);
04234
04235         lpoints = new POINTL[cptl];
04236
04237         for (int i=0; i<n; i++) {
04238             lpoints[i].x = points[i].x;
04239             lpoints[i].y = points[i].y;
04240         }
04241
04242         rclBounds = *bounds;
04243     }
04248     EMRPOLYBEZIERTO ( DATASTREAM& ds )
04249     {
04250         ds » emr » rclBounds » cptl;
04251

```

```

04252     if ( emr.nSize - (sizeof( ::EMRPOLYBEZIERTO ) - sizeof(POINTL)) <
04253         sizeof( POINTL ) * cptl ) {
04254         throw std::runtime_error( "Invalid record size" );
04255     }
04256
04257     std::unique_ptr<POINTL[]> buffer( new POINTL[cptl] );
04258
04259     POINTLARRAY points( buffer.get(), cptl );
04260
04261     ds » points;
04262
04263     lpoints = buffer.release();
04264 }
04265 ~EMRPOLYBEZIERTO ( )
04266 {
04267     if ( lpoints ) delete[] lpoints;
04271 }
04275 bool serialize ( DATASTREAM ds )
04276 {
04277     ds « emr « rclBounds « cptl « POINTLARRAY( lpoints, cptl );
04278     return true;
04279 }
04283 int size ( void ) const { return emr.nSize; }
04289 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
04290 {
04291     EMF_UNUSED(source);
04292     // According to the wine windef.h header, POINT and POINTL are equivalent
04293     PolyBezierTo( dc, (POINT*)lpoints, cptl );
04294 }
04295 #ifdef ENABLE_EDITING
04299 void edit ( void ) const
04300 {
04301     printf( "*POLYBEZIERTO*\n" );
04302     edit_rectl( "rclBounds", rclBounds );
04303     edit_pointlarray( "\t", cptl, lpoints );
04304 }
04305 #endif /* ENABLE_EDITING */
04306 };
04307
04309
04312 class EMRPOLYBEZIERTO16 : public METARECORD, ::EMRPOLYBEZIER16 {
04313     POINT16* lpoints{ nullptr };
04314 public:
04320     EMRPOLYBEZIERTO16 ( const RECTL* bounds, const POINT16* points, INT n )
04321     {
04322         cpts = n;
04323         apts[0].x = 0;           // Really unused
04324         apts[0].y = 0;
04325
04326         emr.iType = EMR_POLYBEZIERTO16;
04327         // The (cptl-1) below is to account for aptl, which isn't written out
04328         emr.nSize = sizeof( ::EMRPOLYBEZIERTO16 ) + sizeof( POINT16 ) * (cpts-1);
04329
04330         lpoints = new POINT16[cpts];
04331
04332         for (int i=0; i<n; i++) {
04333             lpoints[i].x = points[i].x;
04334             lpoints[i].y = points[i].y;
04335         }
04336
04337         rclBounds = *bounds;
04338     }
04345     EMRPOLYBEZIERTO16 ( const RECTL* bounds, const POINT* points, INT n )
04346     {
04347         cpts = n;
04348         apts[0].x = 0;           // Really unused
04349         apts[0].y = 0;
04350
04351         emr.iType = EMR_POLYBEZIERTO16;
04352         // The (cptl-1) below is to account for aptl, which isn't written out
04353         emr.nSize = sizeof( ::EMRPOLYBEZIERTO16 ) + sizeof( POINT16 ) * (cpts-1);
04354
04355         lpoints = new POINT16[cpts];
04356
04357         for (int i=0; i<n; i++) {
04358             lpoints[i].x = points[i].x;
04359             lpoints[i].y = points[i].y;
04360         }
04361
04362         rclBounds = *bounds;
04363     }
04368     EMRPOLYBEZIERTO16 ( DATASTREAM& ds )
04369     {
04370         ds » emr » rclBounds » cpts;
04371
04372         if ( emr.nSize - (sizeof( ::EMRPOLYBEZIERTO16 ) - sizeof(POINT16)) <
04373             sizeof( POINT16 ) * cpts ) {

```

```

04374         throw std::runtime_error( "Invalid record size" );
04375     }
04376
04377     std::unique_ptr<POINT16[]> buffer( new POINT16[cpts] );
04378
04379     POINT16ARRAY points( buffer.get(), cpts );
04380
04381     ds » points;
04382
04383     lpoints = buffer.release();
04384 }
04385 ~EMRPOLYBEZIERTO16 ( )
04386 {
04387     if ( lpoints ) delete[] lpoints;
04388 }
04389 bool serialize ( DATASTREAM ds )
04390 {
04391     ds « emr « rclBounds « cpts « POINT16ARRAY( lpoints, cpts );
04392     return true;
04393 }
04394 int size ( void ) const { return emr.nSize; }
04395 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
04396 {
04397     EMF_UNUSED(source);
04398     // According to the wine windef.h header, POINT and POINTL are equivalent
04399     PolyBezierTo16( dc, lpoints, cpts );
04400 }
04401 #ifdef ENABLE_EDITING
04402 void edit ( void ) const
04403 {
04404     printf( "POLYBEZIERTO16*\n" );
04405     edit_rectl( "rclBounds", rclBounds );
04406     edit_pointlarray( "\t", cpts, lpoints );
04407 }
04408 #endif /* ENABLE_EDITING */
04409 };
04410
04411 class EMRPOLYLINETO : public METARECORD, ::EMRPOLYLINETO {
04412     POINTL* lpoints{ nullptr };
04413 public:
04414     EMRPOLYLINETO ( const RECTL* bounds, const POINT* points, INT n )
04415     {
04416         cptl = n;
04417         aptl[0].x = 0;
04418         aptl[0].y = 0;
04419
04420         emr.iType = EMR_POLYLINETO;
04421         // The (cptl-1) below is to account for aptl, which isn't written out
04422         emr.nSize = sizeof( ::EMRPOLYLINETO ) + sizeof( POINTL ) * (cptl-1);
04423
04424         lpoints = new POINTL[cptl];
04425
04426         for (int i=0; i<n; i++) {
04427             lpoints[i].x = points[i].x;
04428             lpoints[i].y = points[i].y;
04429         }
04430
04431         rclBounds = *bounds;
04432     }
04433     EMRPOLYLINETO ( DATASTREAM& ds )
04434     {
04435         ds » emr » rclBounds » cptl;
04436
04437         if ( emr.nSize - (sizeof( ::EMRPOLYLINETO ) - sizeof(POINTL)) <
04438             sizeof( POINTL ) * cptl ) {
04439             throw std::runtime_error( "Invalid record size" );
04440         }
04441
04442         std::unique_ptr<POINTL[]> buffer( new POINTL[cptl] );
04443
04444         POINTLARRAY points( buffer.get(), cptl );
04445
04446         ds » points;
04447
04448         lpoints = buffer.release();
04449     }
04450     ~EMRPOLYLINETO ( )
04451     {
04452         if ( lpoints ) delete[] lpoints;
04453     }
04454     bool serialize ( DATASTREAM ds )
04455     {
04456         ds « emr « rclBounds « cptl « POINTLARRAY( lpoints, cptl );
04457         return true;
04458     }
04459     int size ( void ) const { return emr.nSize; }

```

```

04504 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
04505 {
04506     EMF_UNUSED(source);
04507     // According to the wine windef.h header, POINT and POINTL are equivalent
04508     PolylineTo( dc, (POINT*)lpoints, cpts );
04509 }
04510 #ifdef ENABLE_EDITING
04511 void edit ( void ) const
04512 {
04513     printf( "*POLYLINETO*\n" );
04514     edit_rectl( "rclBounds", rclBounds );
04515     edit_pointlarray( "\t", cpts, lpoints );
04516 }
04517 #endif /* ENABLE_EDITING */
04518 };
04519
04520 class EMRPOLYLINETO16 : public METARECORD, ::EMRPOLYLINETO16 {
04521     POINT16* lpoints{ nullptr };
04522 public:
04523     EMRPOLYLINETO16 ( const RECTL* bounds, const POINT16* points, INT n )
04524     {
04525         cpts = n;
04526         apts[0].x = 0;
04527         apts[0].y = 0;
04528
04529         emr.iType = EMR_POLYLINETO16;
04530         // The (cpts-1) below is to account for aptl, which isn't written out
04531         emr.nSize = sizeof( ::EMRPOLYLINETO16 ) + sizeof( POINT16 ) * (cpts-1);
04532
04533         lpoints = new POINT16[cpts];
04534
04535         for (int i=0; i<n; i++) {
04536             lpoints[i].x = points[i].x;
04537             lpoints[i].y = points[i].y;
04538         }
04539
04540         rclBounds = *bounds;
04541     }
04542     EMRPOLYLINETO16 ( const RECTL* bounds, const POINT* points, INT n )
04543     {
04544         cpts = n;
04545         apts[0].x = 0;
04546         apts[0].y = 0;
04547
04548         emr.iType = EMR_POLYLINETO16;
04549         // The (cpts-1) below is to account for aptl, which isn't written out
04550         emr.nSize = sizeof( ::EMRPOLYLINETO16 ) + sizeof( POINT16 ) * (cpts-1);
04551
04552         lpoints = new POINT16[cpts];
04553
04554         for (int i=0; i<n; i++) {
04555             lpoints[i].x = points[i].x;
04556             lpoints[i].y = points[i].y;
04557         }
04558
04559         rclBounds = *bounds;
04560     }
04561     EMRPOLYLINETO16 ( DATASTREAM& ds )
04562     {
04563         ds » emr » rclBounds » cpts;
04564
04565         if ( emr.nSize - (sizeof( ::EMRPOLYLINETO16 ) - sizeof(POINT16)) <
04566             sizeof( POINT16 ) * cpts ) {
04567             throw std::runtime_error( "Invalid record size" );
04568         }
04569
04570         std::unique_ptr<POINT16[]> buffer( new POINT16[cpts] );
04571
04572         POINT16ARRAY points( buffer.get(), cpts );
04573
04574         ds » points;
04575
04576         lpoints = buffer.release();
04577     }
04578     ~EMRPOLYLINETO16 ( )
04579     {
04580         if ( lpoints ) delete[] lpoints;
04581     }
04582     bool serialize ( DATASTREAM ds )
04583     {
04584         ds « emr « rclBounds « cpts « POINT16ARRAY( lpoints, cpts );
04585         return true;
04586     }
04587     int size ( void ) const { return emr.nSize; }
04588     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
04589     {

```

```

04626     EMF_UNUSED(source);
04627     // According to the wine windef.h header, POINT and POINTL are equivalent
04628     PolylineTo16( dc, lpoints, cpts );
04629 }
04630 #ifndef ENABLE_EDITING
04631 void edit ( void ) const
04632 {
04633     printf( "*POLYLINE16*\n" );
04634     edit_rectl( "rclBounds", rclBounds );
04635     edit_pointl6array( "\t", cpts, lpoints );
04636 }
04637 #endif /* ENABLE_EDITING */
04638 };
04639
04640 class EMREXTTEXTOUTA : public METARECORD, ::EMREXTTEXTOUTA {
04641     PSTR string_a{ nullptr };
04642     int string_size;
04643
04644     INT* dx_i{ nullptr };
04645 public:
04646     EMREXTTEXTOUTA ( const RECTL* bounds, DWORD graphicsMode, FLOAT xScale,
04647                     FLOAT yScale, const PEMRTEXT text, LPCSTR string,
04648                     const INT* dx )
04649     {
04650         emr.iType = EMR_EXTTEXTOUTA;
04651         emr.nSize = sizeof( ::EMREXTTEXTOUTA );
04652
04653         rclBounds = *bounds;
04654
04655         iGraphicsMode = graphicsMode;
04656         exScale = xScale;
04657         eyScale = yScale;
04658
04659         emrtext = *text;
04660
04661         string_size = ROUND_TO_LONG( emrtext.nChars );
04662
04663         string_a = new CHAR[ string_size ];
04664
04665         memset( string_a, 0, sizeof(CHAR) * string_size );
04666
04667         for ( unsigned int i=0; i<emrtext.nChars; i++ )
04668             string_a[i] = *string++;
04669
04670         emrtext.offString = emr.nSize;
04671         emr.nSize += string_size * sizeof(CHAR);
04672 #if 0
04673 /*
04674 Test only - Problem: Windows requires this dx to be set - at least from 2K on
04675 but to calculate real dx values is hard
04676 For pstoeedit - this is "fixed" now by estimating dx in pstoeedit
04677 */
04678         if ( !dx ) {
04679             int * dxn = new int [string_size];
04680             for (unsigned int i=0; i < string_size; i++) dxn[i] = 10;
04681             dx = dxn;
04682         }
04683 #endif
04684
04685         if ( dx ) {
04686             dx_i = new INT[ emrtext.nChars ];
04687
04688             for ( unsigned int i=0; i<emrtext.nChars; i++ )
04689                 dx_i[i] = *dx++;
04690
04691             emrtext.offDx = emr.nSize;
04692             emr.nSize += emrtext.nChars * sizeof(INT);
04693         }
04694         else {
04695             emrtext.offDx = 0;
04696             dx_i = 0;
04697         }
04698     }
04699     EMREXTTEXTOUTA ( DATASTREAM& ds )
04700     {
04701         ds » emr » rclBounds » iGraphicsMode » exScale » eyScale » emrtext;
04702
04703         if ( emrtext.nChars > 0 and emrtext.offString == 0 ) {
04704             throw std::runtime_error( "Invalid text specification" );
04705         }
04706
04707         if ( emrtext.nChars > emr.nSize - emrtext.offString ) {
04708             throw std::runtime_error( "Invalid text specification" );
04709         }
04710     }
04711 }

```

```

04734     std::unique_ptr<char[]> cbuffer;
04735     std::unique_ptr<INT[]> ibuffer;
04736
04737     if ( emrtext.offString != 0 ) {
04738     string_size = ROUND_TO_LONG( emrtext.nChars );
04739
04740         cbuffer.reset( new char[string_size] );
04741
04742     memset( cbuffer.get(), 0, sizeof(CHAR) * string_size );
04743
04744     CHARSTR string( cbuffer.get(), string_size );
04745
04746     ds » string;
04747     }
04748
04749     if ( emrtext.offDx ) {
04750         ibuffer.reset( new INT[emrtext.nChars] );
04751
04752         INTARRAY dx_is( ibuffer.get(), emrtext.nChars );
04753
04754     ds » dx_is;
04755     }
04756
04757     string_a = cbuffer.release();
04758     dx_i      = ibuffer.release();
04759 }
04760 ~EMREXTTEXTOUTA ( )
04761 {
04762     if ( string_a ) delete[] string_a;
04763     if ( dx_i ) delete[] dx_i;
04764 }
04765 bool serialize ( DATASTREAM ds )
04766 {
04767     ds « emr « rclBounds « iGraphicsMode « exScale « eyScale
04768     « emrtext « CHARSTR( string_a, string_size );
04769     if ( dx_i )
04770     ds « INTARRAY( dx_i, emrtext.nChars );
04771     return true;
04772 }
04773 int size ( void ) const { return emr.nSize; }
04774 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
04775 {
04776     EMF_UNUSED(source);
04777     RECT rect;
04778     rect.left = emrtext.rcl.left;
04779     rect.top = emrtext.rcl.top;
04780     rect.right = emrtext.rcl.right;
04781     rect.bottom = emrtext.rcl.bottom;
04782
04783     ExtTextOutA( dc, emrtext.ptlReference.x, emrtext.ptlReference.y,
04784         emrtext.fOptions, &rect, string_a, emrtext.nChars,
04785         dx_i );
04786 }
04787 #ifdef ENABLE_EDITING
04788 void edit ( void ) const
04789 {
04790     #if defined(__LP64__)
04791     const char* FMT0 = "unknown(%d)\n";
04792     const char* FMT1 = "\tptlReference\t: (%d,%d)\n";
04793     const char* FMT2 = "\tnChars\t\t: %d\n";
04794     const char* FMT3 = "\toffString\t: %d\n";
04795     const char* FMT4 = "\toffDx\t\t: %d\n";
04796     #else
04797     const char* FMT0 = "unknown(%ld)\n";
04798     const char* FMT1 = "\tptlReference\t: (%ld,%ld)\n";
04799     const char* FMT2 = "\tnChars\t\t: %ld\n";
04800     const char* FMT3 = "\toffString\t: %ld\n";
04801     const char* FMT4 = "\toffDx\t\t: %ld\n";
04802     #endif /* __x86_64__ */
04803     printf( "*EXTTEXTOUTA*\n" );
04804     edit_rectl( "rclBounds", rclBounds );
04805     printf( "\tiGraphicsMode\t: " );
04806     switch ( iGraphicsMode ) {
04807     case GM_COMPATIBLE: printf( "GM_COMPATIBLE\n" ); break;
04808     case GM_ADVANCED: printf( "GM_ADVANCED\n" ); break;
04809     default: printf( FMT0, iGraphicsMode );
04810     }
04811     printf( "\texScale\t\t: %f\n", exScale );
04812     printf( "\teyScale\t\t: %f\n", eyScale );
04813     printf( FMT1, emrtext.ptlReference.x, emrtext.ptlReference.y );
04814     printf( FMT2, emrtext.nChars );
04815     printf( FMT3, emrtext.offString );
04816     printf( "\tfOptions\t: " );
04817     if ( emrtext.fOptions == 0 )
04818     printf( "None" );
04819     else {
04820     if ( emrtext.fOptions & ETO_GRAYED ) {

```

```

04839     printf( "ETO_GRAYED" );
04840     if ( emrtext.fOptions & ~ETO_GRAYED )
04841         printf( " | " );
04842     }
04843     if ( emrtext.fOptions & ETO_OPAQUE ) {
04844         printf( "ETO_OPAQUE" );
04845         if ( emrtext.fOptions & ~(ETO_GRAYED | ETO_OPAQUE) )
04846             printf( " | " );
04847     }
04848     if ( emrtext.fOptions & ETO_CLIPPED ) {
04849         printf( "ETO_CLIPPED" );
04850         if ( emrtext.fOptions & ~(ETO_GRAYED | ETO_OPAQUE | ETO_CLIPPED ) )
04851             printf( " | " );
04852     }
04853     if ( emrtext.fOptions & ETO_GLYPH_INDEX ) {
04854         printf( "ETO_GLYPH_INDEX" );
04855         if ( emrtext.fOptions &
04856             ~(ETO_GRAYED | ETO_OPAQUE | ETO_CLIPPED | ETO_GLYPH_INDEX) )
04857             printf( " | " );
04858     }
04859     if ( emrtext.fOptions & ETO_RTLREADING ) {
04860         printf( "ETO_RTLREADING" );
04861         if ( emrtext.fOptions &
04862             ~(ETO_GRAYED | ETO_OPAQUE | ETO_CLIPPED | ETO_GLYPH_INDEX |
04863               ETO_RTLREADING) )
04864             printf( " | " );
04865     }
04866     if ( emrtext.fOptions & ETO_IGNORELANGUAGE )
04867         printf( "ETO_IGNORELANGUAGE" );
04868     }
04869     printf( "\n" );
04870     edit_rectl( "rcl\t", emrtext.rcl );
04871     printf( FMT4, emrtext.offDx );
04872     printf( "\tString:\n\t\t" );
04873     if ( emrtext.nChars > 0 ) {
04874         for ( DWORD i = 0; i < emrtext.nChars; ++i ) {
04875             putchar( string_a[i] );
04876         }
04877     }
04878     else {
04879         printf( "<empty>" );
04880     }
04881     putchar( '\n' );
04882     if ( emrtext.offDx != 0 ) {
04883         printf( "\tOffsets:\n\t\t" );
04884         for ( unsigned int i = 0; i < emrtext.nChars; i++ )
04885             printf( "%d ", dx_i[i] );
04886         printf( "\n" );
04887     }
04888     }
04889 #endif /* ENABLE_EDITING */
04890 };
04891
04892 class EMREXTTEXTOUTW : public METARECORD, ::EMREXTTEXTOUTW {
04893     PWSTR string_a{ nullptr };
04894     int string_size;
04895
04896     INT* dx_i{ nullptr };
04897 public:
04898     EMREXTTEXTOUTW ( const RECTL* bounds, DWORD graphicsMode, FLOAT xScale,
04899                     FLOAT yScale, const PEMRTEXT text, LPCWSTR string,
04900                     const INT* dx )
04901     {
04902         emr.iType = EMR_EXTTEXTOUTW;
04903         emr.nSize = sizeof( ::EMREXTTEXTOUTW );
04904
04905         rclBounds = *bounds;
04906
04907         iGraphicsMode = graphicsMode;
04908         exScale = xScale;
04909         eyScale = yScale;
04910
04911         emrtext = *text;
04912
04913         string_size = ROUND_TO_LONG( emrtext.nChars );
04914
04915         string_a = new WCHAR[ string_size ];
04916
04917         memset( string_a, 0, sizeof(WCHAR) * string_size );
04918
04919         for ( unsigned int i=0; i<emrtext.nChars; i++ )
04920             string_a[i] = *string++;
04921
04922         emrtext.offString = emr.nSize;
04923         emr.nSize += string_size * sizeof(WCHAR);
04924     }
04925 #if 0
04926     /*

```

```

04940 Test only - Problem: Windows requires this dx to be set - at least from 2K on
04941 but to calculate real dx values is hard
04942 For pstoeedit - this is "fixed" now by estimating dx in pstoeedit
04943 */
04944     if ( !dx ) {
04945         int * dxn = new int [string_size];
04946         for (unsigned int i=0; i < string_size; i++) dxn[i] = 10;
04947         dx = dxn;
04948     }
04949 #endif
04950
04951     if ( dx ) {
04952         dx_i = new INT[ emrtext.nChars ];
04953
04954         for ( unsigned int i=0; i<emrtext.nChars; i++ )
04955             dx_i[i] = *dx++;
04956
04957         emrtext.offDx = emr.nSize;
04958         emr.nSize += emrtext.nChars * sizeof(INT);
04959     }
04960     else {
04961         emrtext.offDx = 0;
04962         dx_i = 0;
04963     }
04964 }
04965
04966 EMREXTTEXTOUTW ( DATASTREAM& ds )
04967 {
04968     ds » emr » rclBounds » iGraphicsMode » exScale » eyScale » emrtext;
04969
04970     if ( emrtext.nChars > 0 and emrtext.offString == 0 ) {
04971         throw std::runtime_error( "Invalid text specification" );
04972     }
04973
04974     if ( emrtext.nChars > emr.nSize - emrtext.offString ) {
04975         throw std::runtime_error( "Invalid text specification" );
04976     }
04977
04978     std::unique_ptr<WCHAR[]> cbuffer;
04979     std::unique_ptr<INT[]> ibuffer;
04980
04981     if ( emrtext.offString != 0 ) { // So, what is the point of this check?
04982         string_size = ROUND_TO_LONG( emrtext.nChars );
04983
04984         cbuffer.reset( new WCHAR[string_size] );
04985
04986         memset( cbuffer.get(), 0, sizeof(WCHAR) * string_size );
04987
04988         WCHARSTR string( cbuffer.get(), string_size );
04989
04990         ds » string;
04991     }
04992
04993     if ( emrtext.offDx ) {
04994         ibuffer.reset( new INT[ emrtext.nChars ] );
04995
04996         INTARRAY dx_is( ibuffer.get(), emrtext.nChars );
04997
04998         ds » dx_is;
04999     }
05000
05001     string_a = cbuffer.release();
05002     dx_i = ibuffer.release();
05003 }
05004
05005 ~EMREXTTEXTOUTW ( )
05006 {
05007     if ( string_a ) delete[] string_a;
05008     if ( dx_i ) delete[] dx_i;
05009 }
05010
05011 bool serialize ( DATASTREAM ds )
05012 {
05013     ds « emr « rclBounds « iGraphicsMode « exScale « eyScale
05014     « emrtext « WCHARSTR( string_a, string_size );
05015     if ( dx_i )
05016         ds « INTARRAY( dx_i, emrtext.nChars );
05017     return true;
05018 }
05019
05020 int size ( void ) const { return emr.nSize; }
05021
05022 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
05023 {
05024     EMF_UNUSED(source);
05025     RECT rect;
05026     rect.left = emrtext.rcl.left;
05027     rect.top = emrtext.rcl.top;
05028     rect.right = emrtext.rcl.right;
05029     rect.bottom = emrtext.rcl.bottom;
05030 }

```

```

05046     ExtTextOutW( dc, emrtext.ptlReference.x, emrtext.ptlReference.y,
05047         emrtext.fOptions, &rect, string_a, emrtext.nChars,
05048         dx_i );
05049 }
05050 #ifdef ENABLE_EDITING
05054 void edit ( void ) const
05055 {
05056     #if defined(__LP64__)
05057         const char* FMT0 = "unknown(%d)\n";
05058         const char* FMT1 = "\tptlReference\t: (%d,%d)\n";
05059         const char* FMT2 = "\tnChars\t\t: %d\n";
05060         const char* FMT3 = "\toffString\t: %d\n";
05061         const char* FMT4 = "\toffDx\t\t: %d\n";
05062     #else
05063         const char* FMT0 = "unknown(%ld)\n";
05064         const char* FMT1 = "\tptlReference\t: (%ld,%ld)\n";
05065         const char* FMT2 = "\tnChars\t\t: %ld\n";
05066         const char* FMT3 = "\toffString\t: %ld\n";
05067         const char* FMT4 = "\toffDx\t\t: %ld\n";
05068     #endif /* __x86_64__ */
05069     printf( "*EXTTEXTOUTW*\n" );
05070     edit_rectl( "rclBounds", rclBounds );
05071     printf( "\tiGraphicsMode\t: " );
05072     switch ( iGraphicsMode ) {
05073     case GM_COMPATIBLE: printf( "GM_COMPATIBLE\n" ); break;
05074     case GM_ADVANCED: printf( "GM_ADVANCED\n" ); break;
05075     default: printf( FMT0, iGraphicsMode );
05076     }
05077     printf( "\texScale\t\t: %f\n", exScale );
05078     printf( "\teyScale\t\t: %f\n", eyScale );
05079     printf( FMT1, emrtext.ptlReference.x, emrtext.ptlReference.y );
05080     printf( FMT2, emrtext.nChars );
05081     printf( FMT3, emrtext.offString );
05082     printf( "\tfOptions\t: " );
05083     if ( emrtext.fOptions == 0 )
05084         printf( "None" );
05085     else {
05086         if ( emrtext.fOptions & ETO_GRAYED ) {
05087             printf( "ETO_GRAYED" );
05088             if ( emrtext.fOptions & ~ETO_GRAYED )
05089                 printf( " | " );
05090         }
05091         if ( emrtext.fOptions & ETO_OPAQUE ) {
05092             printf( "ETO_OPAQUE" );
05093             if ( emrtext.fOptions & ~(ETO_GRAYED | ETO_OPAQUE) )
05094                 printf( " | " );
05095         }
05096         if ( emrtext.fOptions & ETO_CLIPPED ) {
05097             printf( "ETO_CLIPPED" );
05098             if ( emrtext.fOptions & ~(ETO_GRAYED | ETO_OPAQUE | ETO_CLIPPED) )
05099                 printf( " | " );
05100         }
05101         if ( emrtext.fOptions & ETO_GLYPH_INDEX ) {
05102             printf( "ETO_GLYPH_INDEX" );
05103             if ( emrtext.fOptions &
05104                 ~(ETO_GRAYED | ETO_OPAQUE | ETO_CLIPPED | ETO_GLYPH_INDEX) )
05105                 printf( " | " );
05106         }
05107         if ( emrtext.fOptions & ETO_RTLREADING ) {
05108             printf( "ETO_RTLREADING" );
05109             if ( emrtext.fOptions &
05110                 ~(ETO_GRAYED | ETO_OPAQUE | ETO_CLIPPED | ETO_GLYPH_INDEX |
05111                   ETO_RTLREADING) )
05112                 printf( " | " );
05113         }
05114         if ( emrtext.fOptions & ETO_IGNORELANGUAGE )
05115             printf( "ETO_IGNORELANGUAGE" );
05116     }
05117     printf( "\n" );
05118     edit_rectl( "rcl\t", emrtext.rcl );
05119     printf( FMT4, emrtext.offDx );
05120
05121     if ( emrtext.nChars > 0 ) {
05122         // iconv_open arguments are TO, FROM (not the other way around).
05123         iconv_t cvt = iconv_open( "UTF-8", "UTF-16LE" );
05124         std::vector<char> utf8_buffer( emrtext.nChars );
05125         // Cannot predict the space necessary to hold the converted
05126         // string. So, we loop until conversion is complete.
05127         size_t size = emrtext.nChars;
05128         size_t in_bytes_left = emrtext.nChars * sizeof(*string_a);
05129         size_t converted = 0;
05130         char* in_buffer = (char*)string_a;
05131         while ( 1 ) {
05132             char* out_buffer = &utf8_buffer[converted];
05133             size_t out_bytes_left = size - converted;
05134
05135             size_t n = iconv( cvt, &in_buffer, &in_bytes_left,

```

```

05136             &out_buffer, &out_bytes_left );
05137
05138     converted = size - out_bytes_left;
05139
05140     if ( n == (size_t)-1 ) {
05141         if ( errno == E2BIG ) {
05142             size_t new_size = 2 * utf8_buffer.size();
05143             utf8_buffer.resize( new_size );
05144             size = utf8_buffer.size();
05145         }
05146         else {
05147             // Real conversion error.
05148             break;
05149         }
05150     }
05151     else {
05152         break;
05153     }
05154 }
05155
05156 iconv_close( cvt );
05157
05158 if ( converted == utf8_buffer.size() )
05159     utf8_buffer.push_back( '\\0' );
05160 else
05161     utf8_buffer[converted] = '\\0';
05162
05163 printf( "\\tString:\\n\\t\\t%s\\n", utf8_buffer.data() );
05164 }
05165 else {
05166     puts( "\\tString:\\n\\t\\t<empty>\\n" );
05167 }
05168
05169 if ( emrtext.offDx != 0 and emrtext.nChars > 0 ) {
05170     printf( "\\tOffsets:\\n\\t\\t" );
05171     for ( unsigned int i = 0; i < emrtext.nChars; i++ )
05172         printf( "%d ", dx_i[i] );
05173     printf( "\\n" );
05174 }
05175 }
05176 #endif /* ENABLE_EDITING */
05177 };
05178
05180
05183 class EMRSETPIXELV : public METARECORD, ::EMRSETPIXELV {
05184 public:
05190     EMRSETPIXELV ( INT x, INT y, COLORREF color )
05191     {
05192         emr.iType = EMR_SETPIXELV;
05193         emr.nSize = sizeof( ::EMRSETPIXELV );
05194         ptlPixel.x = x;
05195         ptlPixel.y = y;
05196         crColor = color;
05197     }
05202     EMRSETPIXELV ( DATASTREAM& ds )
05203     {
05204         ds » emr » ptlPixel » crColor;
05205     }
05209     bool serialize ( DATASTREAM ds )
05210     {
05211         ds « emr « ptlPixel « crColor;
05212         return true;
05213     }
05217     int size ( void ) const { return emr.nSize; }
05223     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
05224     {
05225         EMF_UNUSED(source);
05226         SetPixel( dc, ptlPixel.x, ptlPixel.y, crColor );
05227     }
05228 #ifdef ENABLE_EDITING
05232     void edit ( void ) const
05233     {
05234         printf( "*SETPIXELV*\\n" );
05235         edit_pointl( "ptlPixel", ptlPixel );
05236         edit_color( "crColor\\t", crColor );
05237     }
05238 #endif /* ENABLE_EDITING */
05239 };
05240
05241 class PEN;
05242 class EXTPEN;
05243 class BRUSH;
05244 class FONT;
05245 class PALETTE;
05246
05248
05251 class EMRCREATEPEN : public METARECORD, public ::EMRCREATEPEN

```

```

05252 {
05253 public:
05258     EMRCREATEPEN ( PEN* pen, HGDIOBJ handle );
05263     EMRCREATEPEN ( DATASTREAM& ds );
05267     bool serialize ( DATASTREAM ds )
05268     {
05269         ds << emr << ihPen << lopn;
05270         return true;
05271     }
05275     int size ( void ) const { return emr.nSize; }
05281     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const;
05282 #ifdef ENABLE_EDITING
05286     void edit ( void ) const
05287     {
05288 #if defined(__LP64__)
05289         const char* FMT0 = "\tihPen\t\t: 0x%x\n";
05290         const char* FMT1 = "\tlopn.lopnWidth\t: %d, %d\n";
05291 #else
05292         const char* FMT0 = "\tihPen\t\t: 0x%x\n";
05293         const char* FMT1 = "\tlopn.lopnWidth\t: %ld, %ld\n";
05294 #endif /* __x86_64__ */
05295         printf( "CREATEPEN*\n" );
05296         printf( FMT0, ihPen );
05297         edit_pen_style( "lopn.lopnStyle", lopn.lopnStyle );
05298         printf( FMT1, lopn.lopnWidth.x, lopn.lopnWidth.y );
05299         edit_color( "lopn.lopnColor", lopn.lopnColor );
05300     }
05301 #endif /* ENABLE_EDITING */
05302 };
05303
05305
05309 class EMREXTCREATEPEN : public METARECORD, public ::EMREXTCREATEPEN
05310 {
05311 public:
05316     EMREXTCREATEPEN ( EXTPEN* pen, HGDIOBJ handle );
05321     EMREXTCREATEPEN ( DATASTREAM& ds );
05325     bool serialize ( DATASTREAM ds )
05326     {
05327         ds << emr << ihPen << offBmi << cbBmi << offBits << cbBits << elp;
05328         return true;
05329     }
05333     int size ( void ) const { return emr.nSize; }
05339     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const;
05340 #ifdef ENABLE_EDITING
05344     void edit ( void ) const
05345     {
05346 #if defined(__LP64__)
05347         const char* FMT0 = "\tihPen\t\t\t: 0x%x\n";
05348         const char* FMT1 = "\toffBmi\t\t\t: %d\n";
05349         const char* FMT2 = "\tcbBmi\t\t\t: %d\n";
05350         const char* FMT3 = "\toffBits\t\t\t: %d\n";
05351         const char* FMT4 = "\tcbBits\t\t\t: %d\n";
05352         const char* FMT5 = "\telp.elpWidth\t\t: %d\n";
05353         const char* FMT6 = "\telp.elpNumEntries\t: %d\n";
05354 #else
05355         const char* FMT0 = "\tihPen\t\t\t: 0x%x\n";
05356         const char* FMT1 = "\toffBmi\t\t\t: %ld\n";
05357         const char* FMT2 = "\tcbBmi\t\t\t: %ld\n";
05358         const char* FMT3 = "\toffBits\t\t\t: %ld\n";
05359         const char* FMT4 = "\tcbBits\t\t\t: %ld\n";
05360         const char* FMT5 = "\telp.elpWidth\t\t: %ld\n";
05361         const char* FMT6 = "\telp.elpNumEntries\t: %ld\n";
05362 #endif /* __x86_64__ */
05363         printf( "EXTCREATEPEN*\n" );
05364         printf( FMT0, ihPen );
05365         printf( FMT1, offBmi );
05366         printf( FMT2, cbBmi );
05367         printf( FMT3, offBits );
05368         printf( FMT4, cbBits );
05369         edit_pen_style( "elp.elpPenStyle\t", elp.elpPenStyle );
05370         printf( FMT5, elp.elpWidth );
05371         edit_brush_style( "elp.elpBrushStyle", elp.elpBrushStyle );
05372         edit_color( "elp.elpColor\t", elp.elpColor );
05373         edit_brush_hatch( "elp.elpHatch\t", elp.elpHatch );
05374         printf( FMT6, elp.elpNumEntries );
05375     }
05376 #endif /* ENABLE_EDITING */
05377 };
05378
05380
05383 class EMRCREATEBRUSHINDIRECT : public METARECORD, public ::EMRCREATEBRUSHINDIRECT
05384 {
05385 public:
05390     EMRCREATEBRUSHINDIRECT ( BRUSH* brush, HGDIOBJ handle );
05395     EMRCREATEBRUSHINDIRECT ( DATASTREAM& ds );
05399     bool serialize ( DATASTREAM ds )
05400     {

```

```

05401     ds << emr << ihBrush << lb;
05402     return true;
05403 }
05407 int size ( void ) const { return emr.nSize; }
05413 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const;
05414 #ifdef ENABLE_EDITING
05418 void edit ( void ) const
05419 {
05420 #if defined(__LP64__)
05421     const char* FMT = "\\tihBrush\\t\\t: 0x%x\\n";
05422 #else
05423     const char* FMT = "\\tihBrush\\t\\t: 0x%lx\\n";
05424 #endif /* __x86_64__ */
05425     printf( "*CREATEBRUSHINDIRECT*\\n" );
05426     printf( FMT, ihBrush );
05427     edit_brush_style( "lb.lbStyle", lb.lbStyle );
05428     edit_color( "lb.lbColor", lb.lbColor );
05429     edit_brush_hatch( "lb.lbHatch", lb.lbHatch );
05430 }
05431 #endif /* ENABLE_EDITING */
05432 };
05433
05435 class EMREXTCREATEFONTINDIRECTW : public METARECORD, public ::EMREXTCREATEFONTINDIRECTW
05436 {
05437 public:
05438     EMREXTCREATEFONTINDIRECTW ( FONT* font, HGDI OBJ handle );
05439     EMREXTCREATEFONTINDIRECTW ( DATASTREAM& ds );
05440     bool serialize ( DATASTREAM ds )
05441     {
05442         // Since EMF records have to be multiples of 4 bytes, this
05443         // should perhaps be a general thing, but we know it's currently
05444         // only a problem for this structure.
05445
05446         ds << emr << ihFont << elfw << PADDING( 2 );
05447
05448         return true;
05449     }
05450     int size ( void ) const { return emr.nSize; }
05451     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const;
05452 #ifdef ENABLE_EDITING
05453     void edit ( void ) const
05454     {
05455 #if defined(__LP64__)
05456         const char* FMT0 = "\\tihFont\\t\\t\\t: %d\\n";
05457         const char* FMT1 = "\\tlfHeight\\t\\t: %d\\n";
05458         const char* FMT2 = "\\tlfWidth\\t\\t\\t: %d\\n";
05459         const char* FMT3 = "\\tlfEscapement\\t\\t: %d\\n";
05460         const char* FMT4 = "\\tlfOrientation\\t\\t: %d\\n";
05461         const char* FMT5 = "\\telfVersion\\t\\t: %d\\n";
05462         const char* FMT6 = "\\telfStyleSize\\t\\t: %d\\n";
05463         const char* FMT7 = "\\telfMatch\\t\\t: %d\\n";
05464         const char* FMT8 = "\\telfCulture\\t\\t: %d\\n";
05465 #else
05466         const char* FMT0 = "\\tihFont\\t\\t\\t: %ld\\n";
05467         const char* FMT1 = "\\tlfHeight\\t\\t: %ld\\n";
05468         const char* FMT2 = "\\tlfWidth\\t\\t\\t: %ld\\n";
05469         const char* FMT3 = "\\tlfEscapement\\t\\t: %ld\\n";
05470         const char* FMT4 = "\\tlfOrientation\\t\\t: %ld\\n";
05471         const char* FMT5 = "\\telfVersion\\t\\t: %ld\\n";
05472         const char* FMT6 = "\\telfStyleSize\\t\\t: %ld\\n";
05473         const char* FMT7 = "\\telfMatch\\t\\t: %ld\\n";
05474         const char* FMT8 = "\\telfCulture\\t\\t: %ld\\n";
05475 #endif /* __x86_64__ */
05476         printf( "*EXTCREATEFONTINDIRECTW*\\n" );
05477         printf( FMT0, ihFont );
05478         printf( FMT1, elfw.elfLogFont.lfHeight );
05479         printf( FMT2, elfw.elfLogFont.lfWidth );
05480         printf( FMT3, elfw.elfLogFont.lfEscapement );
05481         printf( FMT4, elfw.elfLogFont.lfOrientation );
05482         printf( "\\tlfWeight\\t\\t: " );
05483         switch ( elfw.elfLogFont.lfWeight ) {
05484             case FW_DONTCARE: printf( "FW_DONTCARE\\n" ); break;
05485             case FW_THIN: printf( "FW_THIN\\n" ); break;
05486             case FW_EXTRALIGHT: printf( "FW_EXTRALIGHT\\n" ); break;
05487             case FW_LIGHT: printf( "FW_LIGHT\\n" ); break;
05488             case FW_NORMAL: printf( "FW_NORMAL\\n" ); break;
05489             case FW_MEDIUM: printf( "FW_MEDIUM\\n" ); break;
05490             case FW_SEMIBOLD: printf( "FW_SEMIBOLD\\n" ); break;
05491             case FW_BOLD: printf( "FW_BOLD\\n" ); break;
05492             case FW_EXTRABOLD: printf( "FW_EXTRABOLD\\n" ); break;
05493             case FW_BLACK: printf( "FW_BLACK\\n" ); break;
05494         }
05495         printf( "\\tlfItalic\\t\\t: %d\\n", elfw.elfLogFont.lfItalic );
05496         printf( "\\tlfUnderline\\t\\t: %d\\n", elfw.elfLogFont.lfUnderline );
05497         printf( "\\tlfStrikeOut\\t\\t: %d\\n", elfw.elfLogFont.lfStrikeOut );
05498         printf( "\\tlfCharSet\\t\\t: %d\\n", elfw.elfLogFont.lfCharSet );

```

```

05524     printf( "\\tlfOutPrecision\\t\\t: %d\\n", elfw.elfLogFont.lfOutPrecision );
05525     printf( "\\tlfClipPrecision\\t\\t: %d\\n", elfw.elfLogFont.lfClipPrecision );
05526     printf( "\\tlfQuality\\t\\t: %d\\n", elfw.elfLogFont.lfQuality );
05527     printf( "\\tlfPitchAndFamily\\t: %d\\n", elfw.elfLogFont.lfPitchAndFamily );
05528     int i = 0;
05529     printf( "\\tlfFaceName\\t\\t: ' ' );
05530     while ( elfw.elfLogFont.lfFaceName[i] != 0 && i < LF_FACESIZE ) {
05531     putchar( elfw.elfLogFont.lfFaceName[i] );
05532     i++;
05533     }
05534     puts( " " );
05535
05536     i = 0;
05537     printf( "\\telfFullName\\t\\t: ' ' );
05538     while ( elfw.elfFullName[i] != 0 && i < LF_FULLFACESIZE ) {
05539     putchar( elfw.elfFullName[i] );
05540     i++;
05541     }
05542     puts( " " );
05543
05544     i = 0;
05545     printf( "\\telfStyle\\t\\t: ' ' );
05546     while ( elfw.elfStyle[i] != 0 && i < LF_FACESIZE ) {
05547     putchar( elfw.elfStyle[i] );
05548     i++;
05549     }
05550     puts( " " );
05551
05552     printf( FMT5, elfw.elfVersion );
05553     printf( FMT6, elfw.elfStyleSize );
05554     printf( FMT7, elfw.elfMatch );
05555     printf( "\\telfVendorId\\t\\t: '%s'\\n", elfw.elfVendorId );
05556     printf( FMT8, elfw.elfCulture );
05557     printf( "\\telfPanose\\t\\t:\\n" );
05558     printf( "\\t\\tbFamilyType\\t\\t: %d\\n", elfw.elfPanose.bFamilyType );
05559     printf( "\\t\\tbSerifStyle\\t\\t: %d\\n", elfw.elfPanose.bSerifStyle );
05560     printf( "\\t\\tbWeight\\t\\t\\t: %d\\n", elfw.elfPanose.bWeight );
05561     printf( "\\t\\tbProportion\\t\\t: %d\\n", elfw.elfPanose.bProportion );
05562     printf( "\\t\\tbContrast\\t\\t: %d\\n", elfw.elfPanose.bContrast );
05563     printf( "\\t\\tbStrokeVariation\\t: %d\\n", elfw.elfPanose.bStrokeVariation );
05564     printf( "\\t\\tbArmStyle\\t\\t: %d\\n", elfw.elfPanose.bArmStyle );
05565     printf( "\\t\\tbLetterform\\t\\t: %d\\n", elfw.elfPanose.bLetterform );
05566     printf( "\\t\\tbMidline\\t\\t: %d\\n", elfw.elfPanose.bMidline );
05567     printf( "\\t\\tbXHeight\\t\\t: %d\\n", elfw.elfPanose.bXHeight );
05568     }
05569 #endif /* ENABLE_EDITING */
05570 };
05571
05572
05573
05574 class EMRCREATEPALETTE : public METARECORD, public ::EMRCREATEPALETTE
05575 {
05576 public:
05577     EMRCREATEPALETTE ( PALETTE* palette, HGDIOBJ handle );
05578     EMRCREATEPALETTE ( DATASTREAM& ds );
05579     bool serialize ( DATASTREAM ds )
05580     {
05581         ds << emr << ihPal << lgpl;
05582         return true;
05583     }
05584     int size ( void ) const { return emr.nSize; }
05585     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const;
05586 #ifdef ENABLE_EDITING
05587     void edit ( void ) const
05588     {
05589         printf( "*CREATEPALETTE* (not really handled by libEMF)\\n" );
05590     }
05591 #endif /* ENABLE_EDITING */
05592 };
05593
05594
05595
05596 class EMRFILLPATH : public METARECORD, ::EMRFILLPATH {
05597 public:
05598     EMRFILLPATH ( const RECTL* bounds )
05599     {
05600         emr.iType = EMR_FILLPATH;
05601         emr.nSize = sizeof( ::EMRFILLPATH );
05602         rclBounds = *bounds;
05603     }
05604     EMRFILLPATH ( DATASTREAM& ds )
05605     {
05606         ds >> emr >> rclBounds;
05607     }
05608     bool serialize ( DATASTREAM ds )
05609     {
05610         ds << emr << rclBounds;
05611         return true;
05612     }
05613 };

```

```

05652     int size ( void ) const { return emr.nSize; }
05658     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
05659     {
05660         EMF_UNUSED(source);
05661         FillPath( dc );
05662     }
05663 #ifdef ENABLE_EDITING
05667     void edit ( void ) const
05668     {
05669         printf( "*FILLPATH*\n" );
05670         edit_rectl( "rclBounds", rclBounds );
05671     }
05672 #endif /* ENABLE_EDITING */
05673 };
05675
05678 class EMRSTROKEPATH : public METARECORD, ::EMRSTROKEPATH {
05679 public:
05683     EMRSTROKEPATH ( const RECTL* bounds )
05684     {
05685         emr.iType = EMR_STROKEPATH;
05686         emr.nSize = sizeof( ::EMRSTROKEPATH );
05687         rclBounds = *bounds;
05688     }
05693     EMRSTROKEPATH ( DATASTREAM& ds )
05694     {
05695         ds » emr » rclBounds;
05696     }
05700     bool serialize ( DATASTREAM ds )
05701     {
05702         ds « emr « rclBounds;
05703         return true;
05704     }
05708     int size ( void ) const { return emr.nSize; }
05714     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
05715     {
05716         EMF_UNUSED(source);
05717         StrokePath( dc );
05718     }
05719 #ifdef ENABLE_EDITING
05723     void edit ( void ) const
05724     {
05725         printf( "*STROKEPATH*\n" );
05726         edit_rectl( "rclBounds", rclBounds );
05727     }
05728 #endif /* ENABLE_EDITING */
05729 };
05731
05734 class EMRSTROKEANDFILLPATH : public METARECORD, ::EMRSTROKEANDFILLPATH {
05735 public:
05739     EMRSTROKEANDFILLPATH ( const RECTL* bounds )
05740     {
05741         emr.iType = EMR_STROKEANDFILLPATH;
05742         emr.nSize = sizeof( ::EMRSTROKEANDFILLPATH );
05743         rclBounds = *bounds;
05744     }
05749     EMRSTROKEANDFILLPATH ( DATASTREAM& ds )
05750     {
05751         ds » emr » rclBounds;
05752     }
05756     bool serialize ( DATASTREAM ds )
05757     {
05758         ds « emr « rclBounds;
05759         return true;
05760     }
05764     int size ( void ) const { return emr.nSize; }
05770     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
05771     {
05772         EMF_UNUSED(source);
05773         StrokeAndFillPath( dc );
05774     }
05775 #ifdef ENABLE_EDITING
05779     void edit ( void ) const
05780     {
05781         printf( "*STROKEANDFILLPATH*\n" );
05782         edit_rectl( "rclBounds", rclBounds );
05783     }
05784 #endif /* ENABLE_EDITING */
05785 };
05787
05790 class EMRBEGINPATH : public METARECORD, ::EMRBEGINPATH {
05791 public:
05795     EMRBEGINPATH ( void )
05796     {
05797         emr.iType = EMR_BEGINPATH;
05798         emr.nSize = sizeof( ::EMRBEGINPATH );
05799     }
05804     EMRBEGINPATH ( DATASTREAM& ds )

```

```

05805     {
05806         ds » emr;
05807     }
05811     bool serialize ( DATASTREAM ds )
05812     {
05813         ds « emr;
05814         return true;
05815     }
05819     int size ( void ) const { return emr.nSize; }
05825     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
05826     {
05827         EMF_UNUSED(source);
05828         BeginPath( dc );
05829     }
05830 #ifdef ENABLE_EDITING
05834     void edit ( void ) const
05835     {
05836         printf( "*BEGINPATH*\n" );
05837     }
05838 #endif /* ENABLE_EDITING */
05839 };
05841
05844     class EMRENDPATH : public METARECORD, ::EMRENDPATH {
05845     public:
05849         EMRENDPATH ( void )
05850         {
05851             emr.iType = EMR_ENDPATH;
05852             emr.nSize = sizeof( ::EMRENDPATH );
05853         }
05858         EMRENDPATH ( DATASTREAM& ds )
05859         {
05860             ds » emr;
05861         }
05865         bool serialize ( DATASTREAM ds )
05866         {
05867             ds « emr;
05868             return true;
05869         }
05873         int size ( void ) const { return emr.nSize; }
05879         void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
05880         {
05881             EMF_UNUSED(source);
05882             EndPath( dc );
05883         }
05884 #ifdef ENABLE_EDITING
05888         void edit ( void ) const
05889         {
05890             printf( "*ENDPATH*\n" );
05891         }
05892 #endif /* ENABLE_EDITING */
05893 };
05895
05898     class EMRCLOSEFIGURE : public METARECORD, ::EMRCLOSEFIGURE {
05899     public:
05903         EMRCLOSEFIGURE ( void )
05904         {
05905             emr.iType = EMR_CLOSEFIGURE;
05906             emr.nSize = sizeof( ::EMRCLOSEFIGURE );
05907         }
05912         EMRCLOSEFIGURE ( DATASTREAM& ds )
05913         {
05914             ds » emr;
05915         }
05919         bool serialize ( DATASTREAM ds )
05920         {
05921             ds « emr;
05922             return true;
05923         }
05927         int size ( void ) const { return emr.nSize; }
05933         void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
05934         {
05935             EMF_UNUSED(source);
05936             CloseFigure( dc );
05937         }
05938 #ifdef ENABLE_EDITING
05942         void edit ( void ) const
05943         {
05944             printf( "*CLOSEFIGURE*\n" );
05945         }
05946 #endif /* ENABLE_EDITING */
05947 };
05949
05953     class EMRSAVEDC : public METARECORD, ::EMRSAVEDC {
05954     public:
05958         EMRSAVEDC ( void )
05959         {
05960             emr.iType = EMR_SAVEDC;

```

```

05961     emr.nSize = sizeof( ::EMRSAVEDC );
05962 }
05963 EMRSAVEDC ( DATASTREAM& ds )
05964 {
05965     ds » emr;
05966 }
05967 bool serialize ( DATASTREAM ds )
05968 {
05969     ds « emr;
05970     return true;
05971 }
05972 int size ( void ) const { return emr.nSize; }
05973 void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
05974 {
05975     EMF_UNUSED(source);
05976     SaveDC( dc );
05977 }
05978 #ifdef ENABLE_EDITING
05979 void edit ( void ) const
05980 {
05981     printf( "%sAVEDC*\n" );
05982 }
05983 #endif /* ENABLE_EDITING */
05984 };
05985
05986 class EMRRESTOREDC : public METARECORD, ::EMRRESTOREDC {
05987 public:
05988     EMRRESTOREDC ( INT n )
05989     {
05990         emr.iType = EMR_RESTOREDC;
05991         emr.nSize = sizeof( ::EMRRESTOREDC );
05992         iRelative = n;
05993     }
05994     EMRRESTOREDC ( DATASTREAM& ds )
05995     {
05996         ds » emr » iRelative;
05997     }
05998     bool serialize ( DATASTREAM ds )
05999     {
06000         ds « emr « iRelative;
06001         return true;
06002     }
06003     int size ( void ) const { return emr.nSize; }
06004     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
06005     {
06006         EMF_UNUSED(source);
06007         RestoreDC( dc, iRelative );
06008     }
06009 #ifdef ENABLE_EDITING
06010     void edit ( void ) const
06011     {
06012         #if defined(__LP64__)
06013             const char* FMT = "\tiRelative: %d\n";
06014         #else
06015             const char* FMT = "\tiRelative: %ld\n";
06016         #endif /* __x86_64__ */
06017         printf( "%sRESTOREDC*\n" );
06018         printf( FMT, iRelative );
06019     }
06020 #endif /* ENABLE_EDITING */
06021 };
06022
06023 class EMRSETMETARGN : public METARECORD, ::EMRSETMETARGN {
06024 public:
06025     EMRSETMETARGN ( void )
06026     {
06027         emr.iType = EMR_SETMETARGN;
06028         emr.nSize = sizeof( ::EMRSETMETARGN );
06029     }
06030     EMRSETMETARGN ( DATASTREAM& ds )
06031     {
06032         ds » emr;
06033     }
06034     bool serialize ( DATASTREAM ds )
06035     {
06036         ds « emr;
06037         return true;
06038     }
06039     int size ( void ) const { return emr.nSize; }
06040     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
06041     {
06042         EMF_UNUSED(source);
06043         SetMetaRgn( dc );
06044     }
06045 #ifdef ENABLE_EDITING
06046     void edit ( void ) const
06047     {
06048

```

```

06114     printf( "SETMETARGN*\n" );
06115 }
06116 #endif /* ENABLE_EDITING */
06117 };
06118
06120
06123 class PEN : public GRAPHICSOBJECT, public LOGPEN {
06124 public:
06128     PEN ( const LOGPEN* lpen )
06129     {
06130         lopnStyle = lpen->lopnStyle;
06131         lopnWidth = lpen->lopnWidth;
06132         lopnColor = lpen->lopnColor;
06133     }
06137     OBJECTTYPE getType ( void ) const { return O_PEN; }
06144     METARECORD* newEMR ( HDC dc, HGDIOBJ emf_handle )
06145     {
06146         contexts[dc] = emf_handle;
06147         return new EMRCREATEPEN( this, emf_handle );
06148     }
06149 };
06150
06152
06155 class EXTPEN : public GRAPHICSOBJECT, public EXTLOGPEN {
06156 public:
06160     EXTPEN ( const EXTLOGPEN* lpen )
06161     {
06162         elpPenStyle = lpen->elpPenStyle;
06163         elpWidth = lpen->elpWidth;
06164         elpBrushStyle = lpen->elpBrushStyle;
06165         elpColor = lpen->elpColor;
06166         elpHatch = lpen->elpHatch;
06167         elpNumEntries = 0;
06168         elpStyleEntry[0] = 0;
06169     }
06173     OBJECTTYPE getType ( void ) const { return O_EXTPEN; }
06180     METARECORD* newEMR ( HDC dc, HGDIOBJ emf_handle )
06181     {
06182         contexts[dc] = emf_handle;
06183         return new EMREXTCREATEPEN( this, emf_handle );
06184     }
06185 };
06186
06188
06191 class BRUSH : public GRAPHICSOBJECT, public LOGBRUSH {
06192 public:
06196     BRUSH ( const LOGBRUSH* lbrush )
06197     {
06198         lbStyle = lbrush->lbStyle;
06199         lbColor = lbrush->lbColor;
06200         lbHatch = lbrush->lbHatch;
06201     }
06205     OBJECTTYPE getType ( void ) const { return O_BRUSH; }
06212     METARECORD* newEMR ( HDC dc, HGDIOBJ emf_handle )
06213     {
06214         contexts[dc] = emf_handle;
06215         return new EMRCREATEBRUSHINDIRECT( this, emf_handle );
06216     }
06217 };
06218
06220
06223 class FONT : public GRAPHICSOBJECT, public EXTLOGFONTW {
06224 public:
06228     FONT ( const LOGFONTW* lfont )
06229     {
06230         this->elfLogFont = *lfont;
06231         // There are a lot more entries in the EXTLOGFONTW structure than
06232         // the API has values for, so we invent them here
06233         memset( &elfFullName, 0, sizeof elfFullName );
06234         memset( &elfStyle, 0, sizeof elfStyle );
06235         elfVersion = ELF_VERSION;
06236         elfStyleSize = 0;
06237         elfMatch = 0;
06238         elfReserved = 0;
06239         memset( &elfVendorId, 0, sizeof elfVendorId );
06240         elfCulture = ELF_CULTURE_LATIN;
06241         memset( &elfPanose, 1, sizeof(PANOSE) );
06242     }
06246     OBJECTTYPE getType ( void ) const { return O_FONT; }
06253     METARECORD* newEMR ( HDC dc, HGDIOBJ emf_handle )
06254     {
06255         contexts[dc] = emf_handle;
06256         return new EMREXTCREATEFONTINDIRECTW( this, emf_handle );
06257     }
06258 };
06259
06261

```

```

06264 class PALETTE : public GRAPHICSOBJECT, public LOGPALETTE {
06265 public:
06266     PALETTE ( const LOGPALETTE* lpalette )
06270     {
06271         EMF_UNUSED(lpalette);
06272         palVersion = 0;
06273         palNumEntries = 0;
06274         PALETTEENTRY zero_entry = { 0, 0, 0, 0 };
06275         palPalEntry[0] = zero_entry;
06276     }
06280     OBJECTTYPE getType ( void ) const { return O_PALETTE; }
06287     METARECORD* newEMR ( HDC dc, HGDIOBJ emf_handle )
06288     {
06289         contexts[dc] = emf_handle;
06290         return new EMRCREATEPALETTE( this, emf_handle );
06291     }
06292 };
06293
06295
06298 class EMRSETMITERLIMIT : public METARECORD, ::EMRSETMITERLIMIT {
06299 public:
06303     EMRSETMITERLIMIT ( FLOAT limit )
06304     {
06305         emr.iType = EMR_SETMITERLIMIT;
06306         emr.nSize = sizeof( ::EMRSETMITERLIMIT );
06307         eMiterLimit = limit;
06308     }
06313     EMRSETMITERLIMIT ( DATASTREAM& ds )
06314     {
06315         int miter_limit;
06316         ds » emr » miter_limit;
06317         eMiterLimit = float(miter_limit);
06318     }
06322     bool serialize ( DATASTREAM ds )
06323     {
06324         ds « emr « (int)eMiterLimit;
06325         return true;
06326     }
06330     int size ( void ) const { return emr.nSize; }
06336     void execute ( METAFILEDEVICECONTEXT* source, HDC dc ) const
06337     {
06338         EMF_UNUSED(source);
06339         SetMiterLimit( dc, eMiterLimit, 0 );
06340     }
06341 #ifdef ENABLE_EDITING
06345     void edit ( void ) const
06346     {
06347         printf( "*SETMITERLIMIT*\n" );
06348         printf( "\teMiterLimit\t: %f\n", eMiterLimit );
06349     }
06350 #endif /* ENABLE_EDITING */
06351 };
06352
06354
06360 class METAFILEDEVICECONTEXT : public OBJECT {
06368     void init ( const RECT* size, LPCWSTR description_w ) {
06369
06370         // Evidently, metafile handles are numbered from 1, so don't
06371         // ever use 0.
06372
06373         handles.push_back( true );
06374
06375         // Keep some of our graphics state in a header record
06376
06377         header = new ENHMETAHEADER ( description_w );
06378         records.push_back( header );
06379
06380         // Compute the size and position of the metafile on the "page"
06381
06382         if ( size ) {
06383             update_frame = false;
06384
06385             header->rclFrame.left = size->left;
06386             header->rclFrame.top = size->top;
06387             header->rclFrame.right = size->right;
06388             header->rclFrame.bottom = size->bottom;
06389
06390             header->rclBounds.left =
06391                 size->left * header->szlDevice.cx / ( header->szlMillimeters.cx * 100 );
06392             header->rclBounds.top =
06393                 size->top * header->szlDevice.cy / ( header->szlMillimeters.cy * 100 );
06394             header->rclBounds.right =
06395                 size->right * header->szlDevice.cx / ( header->szlMillimeters.cx * 100 );
06396             header->rclBounds.bottom =
06397                 size->bottom * header->szlDevice.cy / ( header->szlMillimeters.cy * 100 );
06398         }
06399         else {

```

```

06400     update_frame = true;
06401
06402     header->rclBounds.left = -10;
06403     header->rclBounds.top = -10;
06404     header->rclBounds.right = 10;
06405     header->rclBounds.bottom = 10;
06406
06407     header->rclFrame.left = (LONG)floor( (float)header->rclBounds.left *
06408         header->szlMillimeters.cx * 100 / header->szlDevice.cx );
06409     header->rclFrame.top = (LONG)floor( (float)header->rclBounds.top *
06410         header->szlMillimeters.cy * 100 / header->szlDevice.cy );
06411     header->rclFrame.right = (LONG)ceil( (float)header->rclBounds.right *
06412         header->szlMillimeters.cx * 100 / header->szlDevice.cx );
06413     header->rclFrame.bottom = (LONG)ceil( (float)header->rclBounds.bottom *
06414         header->szlMillimeters.cy * 100 / header->szlDevice.cy );
06415     }
06416
06417     // Some default graphics state (are they really, though?)
06418
06419     SIZEL default_resolution = { RESOLUTION, RESOLUTION };
06420     resolution = default_resolution;
06421     SIZEL default_viewport_ext = { 1, 1 };
06422     viewport_ext = default_viewport_ext;
06423     POINT default_viewport_org = { 0, 0 };
06424     viewport_org = default_viewport_org;
06425     SIZEL default_window_ext = { 1, 1 };
06426     window_ext = default_window_ext;
06427     POINT default_window_org = { 0, 0 };
06428     window_org = default_window_org;
06429
06430     min_device_point = viewport_org;
06431     max_device_point = viewport_org;
06432
06433     pen = (PEN*)globalObjects.find( BLACK_PEN | ENHMETA_STOCK_OBJECT );
06434     brush = (BRUSH*)globalObjects.find( BLACK_BRUSH | ENHMETA_STOCK_OBJECT );
06435     font = (FONT*)globalObjects.find( DEVICE_DEFAULT_FONT | ENHMETA_STOCK_OBJECT );
06436     palette = (PALETTE*)globalObjects.find( DEFAULT_PALETTE | ENHMETA_STOCK_OBJECT );
06437
06438     text_alignment = TA_BASELINE;
06439     text_color = RGB(0,0,0);
06440     bk_color = RGB(0xff,0xff,0xff);
06441     bk_mode = OPAQUE;
06442     polyfill_mode = ALTERNATE;
06443     map_mode = MM_TEXT;
06444     miter_limit = 10.f;
06445
06446     handle = globalObjects.add( this );
06447 }
06448
06449 public:
06450     ::FILE* fp;
06451     DATASTREAM ds;
06452     ENHMETAHEADER* header;
06453     std::vector< EMF::METARECORD* > records;
06454
06455     // Keep a small set of graphics state information
06456     SIZEL resolution;
06457     SIZEL viewport_ext;
06458     POINT viewport_org;
06459     SIZEL window_ext;
06460     POINT window_org;
06461     bool update_frame;
06462     POINT min_device_point;
06463     POINT max_device_point;
06464     POINT point;
06465     PEN* pen;
06466     BRUSH* brush;
06467     FONT* font;
06468     PALETTE* palette;
06469     UINT text_alignment;
06470     COLORREF text_color;
06471     COLORREF bk_color;
06472     INT bk_mode;
06473     INT polyfill_mode;
06474     INT map_mode;
06475     FLOAT miter_limit;
06476
06477     std::vector< bool > handles;
06478
06479     std::map< HGDIOBJ, HGDIOBJ > emf_handles;
06480
06481     METAFILEDEVICECONTEXT ( FILE* fp_, const RECT* size,
06482         LPCWSTR description_w )
06483     : fp(fp_), ds( fp_ )
06484     {
06485         init( size, description_w );
06486     }

```

```

06524     virtual ~METAFILEDEVICECONTEXT ( )
06525     {
06526         // Purge all the metarecords (if there are any) {this include the
06527         // header record, too}
06528         if ( records.size() > 0 )
06529             deleteMetafile();
06530     }
06531     OBJECTTYPE getType ( void ) const { return O_METAFILEDEVICECONTEXT; }
06532     DWORD nextHandle ( void )
06533     {
06534         for ( unsigned int i = 1; i < handles.size(); i++ ) {
06535             if ( !handles[i] ) {
06536                 handles[i] = true;
06537                 return i;
06538             }
06539         }
06540         handles.push_back( true );
06541         // Well, it appears that even StockObject handles count for something.
06542         // Not sure what the right value here is, then.
06543         header->nHandles = handles.size();
06544         return handles.size()-1;
06545     }
06546     void clearHandle ( DWORD handle )
06547     {
06548         if ( handle < handles.size() ) {
06549             handles[handle] = false;
06550         }
06551     }
06552     void appendRecord ( METARECORD* record )
06553     {
06554         records.push_back( record );
06555         header->nBytes += record->size();
06556         header->nRecords++;
06557     }
06558     void appendHandle ( METARECORD* record )
06559     {
06560         records.push_back( record );
06561         header->nBytes += record->size();
06562         header->nRecords++;
06563     }
06564     void deleteMetafile ( void )
06565     {
06566         for ( auto r = records.begin(); r != records.end(); r++ ) {
06567             delete *r;
06568         }
06569         records.clear();
06570     }
06571     void mergePoint ( const LONG& x, const LONG& y )
06572     {
06573         POINT p;
06574         p.x = x;
06575         p.y = y;
06576         mergePoint( p );
06577     }
06578     void mergePoint( const POINT& p )
06579     {
06580         POINT device_point;
06581
06582         // *** Note, it's possible for the global transformation matrix to
06583         // affect this too. ***
06584
06585         int window_width = window_ext.cx <= 0 ? 1 : window_ext.cx;
06586         int window_height = window_ext.cy <= 0 ? 1 : window_ext.cy;
06587
06588         device_point.x = (LONG)( (float)( p.x - window_org.x ) / window_width *
06589             viewport_ext.cx + viewport_org.x );
06590
06591         device_point.y = (LONG)( (float)( p.y - window_org.y ) / window_height *
06592             viewport_ext.cy + viewport_org.y );
06593
06594         // If the user didn't specify a bounding rectangle in the constructor,
06595         // compute one from this data.
06596         if ( device_point.x < min_device_point.x ) {
06597             min_device_point.x = device_point.x;
06598             if ( update_frame ) {
06599                 header->rclBounds.left = min_device_point.x - 10;
06600                 int device_width = header->szlDevice.cx <= 0 ? 1 : header->szlDevice.cx;
06601                 header->rclFrame.left = (LONG)floor( (float)header->rclBounds.left *
06602                     header->szlMillimeters.cx * 100 / device_width );
06603             }
06604         }
06605         else if ( device_point.x > max_device_point.x ) {
06606             max_device_point.x = device_point.x;
06607             if ( update_frame ) {
06608                 header->rclBounds.right = max_device_point.x + 10;

```

```
06643         int device_width = header->szlDevice.cx <= 0 ? 1 : header->szlDevice.cx;
06644         header->rclFrame.right = (LONG)ceil( (float)header->rclBounds.right *
06645         header->szlMillimeters.cx * 100 / device_width );
06646     }
06647 }
06648
06649     if ( device_point.y < min_device_point.y ) {
06650 min_device_point.y = device_point.y;
06651     if ( update_frame ) {
06652         header->rclBounds.top = min_device_point.y - 10;
06653         int device_height = header->szlDevice.cy <= 0 ? 1 : header->szlDevice.cy;
06654         header->rclFrame.top = (LONG)floor( (float)header->rclBounds.top *
06655         header->szlMillimeters.cy * 100 / device_height );
06656     }
06657 }
06658     else if ( device_point.y > max_device_point.y ) {
06659 max_device_point.y = device_point.y;
06660     if ( update_frame ) {
06661         header->rclBounds.bottom = max_device_point.y + 10;
06662         int device_height = header->szlDevice.cy <= 0 ? 1 : header->szlDevice.cy;
06663         header->rclFrame.bottom = (LONG)ceil( (float)header->rclBounds.bottom *
06664         header->szlMillimeters.cy * 100 / device_height );
06665     }
06666 }
06667 }
06668 };
06669
06670 } // close EMF namespace
06671
06672 #undef EMF_UNUSED
06673 #endif /* _LIBEMF_H */
```

## Index

- ~EMREXTTEXTOUTA
  - EMF::EMREXTTEXTOUTA, [64](#)
- ~EMREXTTEXTOUTW
  - EMF::EMREXTTEXTOUTW, [67](#)
- ~EMRPOLYBEZIER
  - EMF::EMRPOLYBEZIER, [77](#)
- ~EMRPOLYBEZIER16
  - EMF::EMRPOLYBEZIER16, [80](#)
- ~EMRPOLYBEZIERTO
  - EMF::EMRPOLYBEZIERTO, [82](#)
- ~EMRPOLYBEZIERTO16
  - EMF::EMRPOLYBEZIERTO16, [85](#)
- ~EMRPOLYGON
  - EMF::EMRPOLYGON, [88](#)
- ~EMRPOLYGON16
  - EMF::EMRPOLYGON16, [90](#)
- ~EMRPOLYLINE
  - EMF::EMRPOLYLINE, [92](#)
- ~EMRPOLYLINE16
  - EMF::EMRPOLYLINE16, [95](#)
- ~EMRPOLYLINETO
  - EMF::EMRPOLYLINETO, [98](#)
- ~EMRPOLYLINETO16
  - EMF::EMRPOLYLINETO16, [100](#)
- ~EMRPOLYPOLYGON
  - EMF::EMRPOLYPOLYGON, [103](#)
- ~EMRPOLYPOLYGON16
  - EMF::EMRPOLYPOLYGON16, [106](#)
- ~ENHMETAHEADER
  - EMF::ENHMETAHEADER, [153](#)
- ~METAFILEDEVICECONTEXT
  - EMF::METAFILEDEVICECONTEXT, [168](#)
- ~METARECORD
  - EMF::METARECORD, [173](#)
- add
  - EMF::GLOBALOBJECTS, [162](#)
- appendHandle
  - EMF::METAFILEDEVICECONTEXT, [169](#)
- appendRecord
  - EMF::METAFILEDEVICECONTEXT, [169](#)
- basetsd.h, [184](#)
- begin
  - EMF::GLOBALOBJECTS, [162](#)
- BRUSH
  - EMF::BRUSH, [8](#)
- BYTEARRAY
  - EMF::BYTEARRAY, [10](#)
- CHARSTR
  - EMF::CHARSTR, [11](#)
- clearHandle
  - EMF::METAFILEDEVICECONTEXT, [169](#)
- contexts
  - EMF::GRAPHICSOBJECT, [165](#)
- DATASTREAM
  - EMF::DATASTREAM, [13](#)
- deleteMetafile
  - EMF::METAFILEDEVICECONTEXT, [169](#)
- ds
  - EMF::METAFILEDEVICECONTEXT, [170](#)
- DWORDARRAY
  - EMF::DWORDARRAY, [35](#)
- emf.h, [183](#)
- EMF::BRUSH, [7](#)
  - BRUSH, [8](#)
  - getType, [9](#)
  - newEMR, [9](#)
- EMF::BYTEARRAY, [9](#)
  - BYTEARRAY, [10](#)
- EMF::CHARSTR, [10](#)
  - CHARSTR, [11](#)
- EMF::DATASTREAM, [11](#)
  - DATASTREAM, [13](#)
  - operator<<, [13–23](#)
  - operator>>, [23–33](#)
  - setStream, [33](#)
- EMF::DWORDARRAY, [34](#)
  - DWORDARRAY, [35](#)
- EMF::EMRARC, [35](#)
  - EMRARC, [36](#)
  - execute, [37](#)
  - serialize, [37](#)
  - size, [37](#)
- EMF::EMRARCTO, [37](#)
  - EMRARCTO, [38](#)
  - execute, [39](#)
  - serialize, [39](#)
  - size, [39](#)
- EMF::EMRBEGINPATH, [40](#)
  - EMRBEGINPATH, [40](#)
  - execute, [41](#)
  - serialize, [41](#)
  - size, [41](#)
- EMF::EMRCLOSEFIGURE, [42](#)
  - EMRCLOSEFIGURE, [42](#)
  - execute, [43](#)
  - serialize, [43](#)
  - size, [43](#)
- EMF::EMRCREATEBRUSHINDIRECT, [44](#)
  - EMRCREATEBRUSHINDIRECT, [44](#)
  - execute, [45](#)
  - serialize, [45](#)
  - size, [45](#)
- EMF::EMRCREATEPALETTE, [46](#)
  - EMRCREATEPALETTE, [46](#)
  - execute, [47](#)
  - serialize, [47](#)
  - size, [47](#)

- EMF::EMRCREATEPEN, 48
  - EMRCREATEPEN, 48
  - execute, 49
  - serialize, 49
  - size, 49
- EMF::EMRDELETEOBJECT, 50
  - EMRDELETEOBJECT, 50
  - execute, 51
  - serialize, 51
  - size, 51
- EMF::EMRELLIPSE, 52
  - EMRELLIPSE, 52, 53
  - execute, 53
  - serialize, 53
  - size, 53
- EMF::EMRENDPATH, 54
  - EMRENDPATH, 55
  - execute, 55
  - serialize, 55
  - size, 56
- EMF::EMREOF, 56
  - EMREOF, 57
  - execute, 57
  - serialize, 57
  - size, 58
- EMF::EMREXTCREATEFONTINDIRECTW, 58
  - EMREXTCREATEFONTINDIRECTW, 59
  - execute, 59
  - serialize, 60
  - size, 60
- EMF::EMREXTCREATEPEN, 60
  - EMREXTCREATEPEN, 61
  - execute, 62
  - serialize, 62
  - size, 62
- EMF::EMREXTTEXTOUTA, 63
  - ~EMREXTTEXTOUTA, 64
  - EMREXTTEXTOUTA, 63, 64
  - execute, 64
  - serialize, 65
  - size, 65
- EMF::EMREXTTEXTOUTW, 65
  - ~EMREXTTEXTOUTW, 67
  - EMREXTTEXTOUTW, 66
  - execute, 67
  - serialize, 67
  - size, 67
- EMF::EMRFILLPATH, 68
  - EMRFILLPATH, 68
  - execute, 69
  - serialize, 69
  - size, 69
- EMF::EMRLINETO, 70
  - EMRLINETO, 70
  - execute, 71
  - serialize, 71
  - size, 71
- EMF::EMRMODIFYWORLDTRANSFORM, 72
  - EMRMODIFYWORLDTRANSFORM, 72
  - execute, 73
  - serialize, 73
  - size, 73
- EMF::EMRMOVETOEX, 74
  - EMRMOVETOEX, 74
  - execute, 75
  - serialize, 75
  - size, 75
- EMF::EMRPOLYBEZIER, 76
  - ~EMRPOLYBEZIER, 77
  - EMRPOLYBEZIER, 76, 77
  - execute, 77
  - serialize, 77
  - size, 78
- EMF::EMRPOLYBEZIER16, 78
  - ~EMRPOLYBEZIER16, 80
  - EMRPOLYBEZIER16, 79
  - execute, 80
  - serialize, 80
  - size, 80
- EMF::EMRPOLYBEZIERTO, 81
  - ~EMRPOLYBEZIERTO, 82
  - EMRPOLYBEZIERTO, 82
  - execute, 83
  - serialize, 83
  - size, 83
- EMF::EMRPOLYBEZIERTO16, 83
  - ~EMRPOLYBEZIERTO16, 85
  - EMRPOLYBEZIERTO16, 84, 85
  - execute, 85
  - serialize, 86
  - size, 86
- EMF::EMRPOLYGON, 86
  - ~EMRPOLYGON, 88
  - EMRPOLYGON, 87
  - execute, 88
  - serialize, 88
  - size, 88
- EMF::EMRPOLYGON16, 89
  - ~EMRPOLYGON16, 90
  - EMRPOLYGON16, 90
  - execute, 91
  - serialize, 91
  - size, 91
- EMF::EMRPOLYLINE, 92
  - ~EMRPOLYLINE, 92
  - EMRPOLYLINE, 92, 93
  - execute, 93
  - serialize, 93
  - size, 93
- EMF::EMRPOLYLINE16, 94
  - ~EMRPOLYLINE16, 95
  - EMRPOLYLINE16, 95
  - execute, 96
  - serialize, 96
  - size, 96
- EMF::EMRPOLYLINETO, 97

- ~EMRPOLYLINETO, 98
- EMRPOLYLINETO, 97
- execute, 98
- serialize, 98
- size, 98
- EMF::EMRPOLYLINETO16, 99
  - ~EMRPOLYLINETO16, 100
  - EMRPOLYLINETO16, 100
  - execute, 101
  - serialize, 101
  - size, 101
- EMF::EMRPOLYPOLYGON, 102
  - ~EMRPOLYPOLYGON, 103
  - EMRPOLYPOLYGON, 102, 103
  - execute, 103
  - serialize, 103
  - size, 104
- EMF::EMRPOLYPOLYGON16, 104
  - ~EMRPOLYPOLYGON16, 106
  - EMRPOLYPOLYGON16, 105, 106
  - execute, 106
  - serialize, 106
  - size, 107
- EMF::EMRRECTANGLE, 107
  - EMRRECTANGLE, 108
  - execute, 109
  - serialize, 109
  - size, 109
- EMF::EMRRESTOREDC, 109
  - EMRRESTOREDC, 110
  - execute, 111
  - serialize, 111
  - size, 111
- EMF::EMRSAVEDC, 111
  - EMRSAVEDC, 112
  - execute, 113
  - serialize, 113
  - size, 113
- EMF::EMRSCALEVIEWPORTEXT, 113
  - EMRSCALEVIEWPORTEXT, 114
  - execute, 115
  - serialize, 115
  - size, 115
- EMF::EMRSCALEWINDOWEXT, 116
  - EMRSCALEWINDOWEXT, 116, 117
  - execute, 117
  - serialize, 117
  - size, 117
- EMF::EMRSELECTOBJECT, 118
  - EMRSELECTOBJECT, 119
  - execute, 119
  - serialize, 119
  - size, 120
- EMF::EMRSETBKCOLOR, 120
  - EMRSETBKCOLOR, 121
  - execute, 121
  - serialize, 122
  - size, 122
- EMF::EMRSETBKMODE, 122
  - EMRSETBKMODE, 123
  - execute, 124
  - serialize, 124
  - size, 124
- EMF::EMRSETMAPMODE, 124
  - EMRSETMAPMODE, 125
  - execute, 126
  - serialize, 126
  - size, 126
- EMF::EMRSETMETARGN, 126
  - EMRSETMETARGN, 127
  - execute, 128
  - serialize, 128
  - size, 128
- EMF::EMRSETMITERLIMIT, 128
  - EMRSETMITERLIMIT, 129
  - execute, 130
  - serialize, 130
  - size, 130
- EMF::EMRSETPIXELV, 130
  - EMRSETPIXELV, 131
  - execute, 132
  - serialize, 132
  - size, 132
- EMF::EMRSETPOLYFILLMODE, 133
  - EMRSETPOLYFILLMODE, 133
  - execute, 134
  - serialize, 134
  - size, 134
- EMF::EMRSETTEXTALIGN, 135
  - EMRSETTEXTALIGN, 135
  - execute, 136
  - serialize, 136
  - size, 136
- EMF::EMRSETTEXTCOLOR, 137
  - EMRSETTEXTCOLOR, 137
  - execute, 138
  - serialize, 138
  - size, 138
- EMF::EMRSETVIEWPORTEXT, 139
  - EMRSETVIEWPORTEXT, 139
  - execute, 140
  - serialize, 140
  - size, 140
- EMF::EMRSETVIEWPORTORGEX, 141
  - EMRSETVIEWPORTORGEX, 141
  - execute, 142
  - serialize, 142
  - size, 142
- EMF::EMRSETWINDOWEXT, 143
  - EMRSETWINDOWEXT, 143
  - execute, 144
  - serialize, 144
  - size, 144
- EMF::EMRSETWINDOWORGEX, 145
  - EMRSETWINDOWORGEX, 145
  - execute, 146

- serialize, 146
- size, 146
- EMF::EMRSETWORLDTRANSFORM, 147
  - EMRSETWORLDTRANSFORM, 147
  - execute, 148
  - serialize, 148
  - size, 148
- EMF::EMRSTROKEANDFILLPATH, 149
  - EMRSTROKEANDFILLPATH, 149
  - execute, 150
  - serialize, 150
  - size, 150
- EMF::EMRSTROKEPATH, 151
  - EMRSTROKEPATH, 151
  - execute, 152
  - serialize, 152
  - size, 152
- EMF::ENHMETAHEADER, 153
  - ~ENHMETAHEADER, 153
  - ENHMETAHEADER, 153
  - execute, 154
  - serialize, 154
  - size, 154
  - unserialize, 154
- EMF::EXTPEN, 155
  - EXTPEN, 156
  - getType, 156
  - newEMR, 156
- EMF::FONT, 157
  - FONT, 158
  - getType, 158
  - newEMR, 158
- EMF::GLOBALOBJECTS, 159
  - add, 162
  - begin, 162
  - end, 162
  - find, 162
  - newRecord, 162
  - remove, 163
- EMF::GRAPHICSOBJECT, 163
  - contexts, 165
  - newEMR, 164
- EMF::INTARRAY, 165
  - INTARRAY, 166
- EMF::METAFILEDEVICECONTEXT, 166
  - ~METAFILEDEVICECONTEXT, 168
  - appendHandle, 169
  - appendRecord, 169
  - clearHandle, 169
  - deleteMetafile, 169
  - ds, 170
  - emf\_handles, 170
  - fp, 171
  - getType, 169
  - handles, 171
  - header, 171
  - mergePoint, 170
  - METAFILEDEVICECONTEXT, 168
  - nextHandle, 170
  - records, 171
- EMF::METARECORD, 172
  - ~METARECORD, 173
  - execute, 173
  - serialize, 173
  - size, 174
- EMF::OBJECT, 174
  - getType, 175
  - handle, 175
  - OBJECT, 175
- EMF::PADDING, 176
  - PADDING, 176
- EMF::PALETTE, 177
  - getType, 178
  - newEMR, 178
  - PALETTE, 178
- EMF::PEN, 179
  - getType, 180
  - newEMR, 180
  - PEN, 180
- EMF::POINT16ARRAY, 180
  - POINT16ARRAY, 181
- EMF::POINTLARRAY, 181
  - POINTLARRAY, 182
- EMF::WCHARSTR, 182
  - WCHARSTR, 183
- emf\_handles
  - EMF::METAFILEDEVICECONTEXT, 170
- EMRARC
  - EMF::EMRARC, 36
- EMRARCTO
  - EMF::EMRARCTO, 38
- EMRBEGINPATH
  - EMF::EMRBEGINPATH, 40
- EMRCLOSEFIGURE
  - EMF::EMRCLOSEFIGURE, 42
- EMRCREATEBRUSHINDIRECT
  - EMF::EMRCREATEBRUSHINDIRECT, 44
- EMRCREATEPALETTE
  - EMF::EMRCREATEPALETTE, 46
- EMRCREATEPEN
  - EMF::EMRCREATEPEN, 48
- EMRDELETEOBJECT
  - EMF::EMRDELETEOBJECT, 50
- EMRELLIPSE
  - EMF::EMRELLIPSE, 52, 53
- EMRENDPATH
  - EMF::EMRENDPATH, 55
- EMREOF
  - EMF::EMREOF, 57
- EMREXTCREATEFONTINDIRECTW
  - EMF::EMREXTCREATEFONTINDIRECTW, 59
- EMREXTCREATEPEN
  - EMF::EMREXTCREATEPEN, 61
- EMREXTTEXTOUTA
  - EMF::EMREXTTEXTOUTA, 63, 64
- EMREXTTEXTOUTW

EMF::EMREXTTEXTOUTW, 66  
EMRFILLPATH  
EMF::EMRFILLPATH, 68  
EMRLINETO  
EMF::EMRLINETO, 70  
EMRMODIFYWORLDTRANSFORM  
EMF::EMRMODIFYWORLDTRANSFORM, 72  
EMRMOVETOEX  
EMF::EMRMOVETOEX, 74  
EMRPOLYBEZIER  
EMF::EMRPOLYBEZIER, 76, 77  
EMRPOLYBEZIER16  
EMF::EMRPOLYBEZIER16, 79  
EMRPOLYBEZIERTO  
EMF::EMRPOLYBEZIERTO, 82  
EMRPOLYBEZIERTO16  
EMF::EMRPOLYBEZIERTO16, 84, 85  
EMRPOLYGON  
EMF::EMRPOLYGON, 87  
EMRPOLYGON16  
EMF::EMRPOLYGON16, 90  
EMRPOLYLINE  
EMF::EMRPOLYLINE, 92, 93  
EMRPOLYLINE16  
EMF::EMRPOLYLINE16, 95  
EMRPOLYLINETO  
EMF::EMRPOLYLINETO, 97  
EMRPOLYLINETO16  
EMF::EMRPOLYLINETO16, 100  
EMRPOLYPOLYGON  
EMF::EMRPOLYPOLYGON, 102, 103  
EMRPOLYPOLYGON16  
EMF::EMRPOLYPOLYGON16, 105, 106  
EMRRECTANGLE  
EMF::EMRRECTANGLE, 108  
EMRRESTOREDC  
EMF::EMRRESTOREDC, 110  
EMRSAVEDC  
EMF::EMRSAVEDC, 112  
EMRSCALEVIEWPORTEXT  
EMF::EMRSCALEVIEWPORTEXT, 114  
EMRSCALEWINDOWEXT  
EMF::EMRSCALEWINDOWEXT, 116, 117  
EMRSELECTOBJECT  
EMF::EMRSELECTOBJECT, 119  
EMRSETBKCOLOR  
EMF::EMRSETBKCOLOR, 121  
EMRSETBKMODE  
EMF::EMRSETBKMODE, 123  
EMRSETMAPMODE  
EMF::EMRSETMAPMODE, 125  
EMRSETMETARGN  
EMF::EMRSETMETARGN, 127  
EMRSETMITERLIMIT  
EMF::EMRSETMITERLIMIT, 129  
EMRSETPIXELV  
EMF::EMRSETPIXELV, 131  
EMRSETPOLYFILLMODE  
EMF::EMRSETPOLYFILLMODE, 133  
EMRSETTEXTALIGN  
EMF::EMRSETTEXTALIGN, 135  
EMRSETTEXTCOLOR  
EMF::EMRSETTEXTCOLOR, 137  
EMRSETVIEWPORTEXT  
EMF::EMRSETVIEWPORTEXT, 139  
EMRSETVIEWPORTORGE  
EMF::EMRSETVIEWPORTORGE, 141  
EMRSETWINDOWEXT  
EMF::EMRSETWINDOWEXT, 143  
EMRSETWINDOWORGE  
EMF::EMRSETWINDOWORGE, 145  
EMRSETWORLDTRANSFORM  
EMF::EMRSETWORLDTRANSFORM, 147  
EMRSTROKEANDFILLPATH  
EMF::EMRSTROKEANDFILLPATH, 149  
EMRSTROKEPATH  
EMF::EMRSTROKEPATH, 151  
end  
EMF::GLOBALOBJECTS, 162  
ENHMETAHEADER  
EMF::ENHMETAHEADER, 153  
execute  
EMF::EMRARC, 37  
EMF::EMRARCTO, 39  
EMF::EMRBEGINPATH, 41  
EMF::EMRCLOSEFIGURE, 43  
EMF::EMRCREATEBRUSHINDIRECT, 45  
EMF::EMRCREATEPALETTE, 47  
EMF::EMRCREATEPEN, 49  
EMF::EMRDELETEOBJECT, 51  
EMF::EMRELLIPSE, 53  
EMF::EMRENDPATH, 55  
EMF::EMREOF, 57  
EMF::EMREXTCREATEFONTINDIRECTW, 59  
EMF::EMREXTCREATEPEN, 62  
EMF::EMREXTTEXTOUTA, 64  
EMF::EMREXTTEXTOUTW, 67  
EMF::EMRFILLPATH, 69  
EMF::EMRLINETO, 71  
EMF::EMRMODIFYWORLDTRANSFORM, 73  
EMF::EMRMOVETOEX, 75  
EMF::EMRPOLYBEZIER, 77  
EMF::EMRPOLYBEZIER16, 80  
EMF::EMRPOLYBEZIERTO, 83  
EMF::EMRPOLYBEZIERTO16, 85  
EMF::EMRPOLYGON, 88  
EMF::EMRPOLYGON16, 91  
EMF::EMRPOLYLINE, 93  
EMF::EMRPOLYLINE16, 96  
EMF::EMRPOLYLINETO, 98  
EMF::EMRPOLYLINETO16, 101  
EMF::EMRPOLYPOLYGON, 103  
EMF::EMRPOLYPOLYGON16, 106  
EMF::EMRRECTANGLE, 109  
EMF::EMRRESTOREDC, 111  
EMF::EMRSAVEDC, 113

- EMF::EMRSCALEVIEWPORTEXT, 115
- EMF::EMRSCALEWINDOWEXT, 117
- EMF::EMRSELECTOBJECT, 119
- EMF::EMRSETBKCOLOR, 121
- EMF::EMRSETBKMODE, 124
- EMF::EMRSETMAPMODE, 126
- EMF::EMRSETMETARGN, 128
- EMF::EMRSETMITERLIMIT, 130
- EMF::EMRSETPIXELV, 132
- EMF::EMRSETPOLYFILLMODE, 134
- EMF::EMRSETTEXTALIGN, 136
- EMF::EMRSETTEXTCOLOR, 138
- EMF::EMRSETVIEWPORTEXT, 140
- EMF::EMRSETVIEWPORTORGEX, 142
- EMF::EMRSETWINDOWEXT, 144
- EMF::EMRSETWINDOWORGEX, 146
- EMF::EMRSETWORLDTRANSFORM, 148
- EMF::EMRSTROKEANDFILLPATH, 150
- EMF::EMRSTROKEPATH, 152
- EMF::ENHMETAHEADER, 154
- EMF::METARECORD, 173
- EXTPEN
  - EMF::EXTPEN, 156
- find
  - EMF::GLOBALOBJECTS, 162
- FONT
  - EMF::FONT, 158
- fp
  - EMF::METAFILEDEVICECONTEXT, 171
- getType
  - EMF::BRUSH, 9
  - EMF::EXTPEN, 156
  - EMF::FONT, 158
  - EMF::METAFILEDEVICECONTEXT, 169
  - EMF::OBJECT, 175
  - EMF::PALETTE, 178
  - EMF::PEN, 180
- guiddef.h, 186
- handle
  - EMF::OBJECT, 175
- handles
  - EMF::METAFILEDEVICECONTEXT, 171
- header
  - EMF::METAFILEDEVICECONTEXT, 171
- INTARRAY
  - EMF::INTARRAY, 166
- libemf.h, 377
- mergePoint
  - EMF::METAFILEDEVICECONTEXT, 170
- METAFILEDEVICECONTEXT
  - EMF::METAFILEDEVICECONTEXT, 168
- newEMR
  - EMF::BRUSH, 9
- EMF::EXTPEN, 156
- EMF::FONT, 158
- EMF::GRAPHICSOBJECT, 164
- EMF::PALETTE, 178
- EMF::PEN, 180
- newRecord
  - EMF::GLOBALOBJECTS, 162
- nextHandle
  - EMF::METAFILEDEVICECONTEXT, 170
- OBJECT
  - EMF::OBJECT, 175
- operator<<
  - EMF::DATASTREAM, 13–23
- operator>>
  - EMF::DATASTREAM, 23–33
- PADDING
  - EMF::PADDING, 176
- PALETTE
  - EMF::PALETTE, 178
- PEN
  - EMF::PEN, 180
- POINT16ARRAY
  - EMF::POINT16ARRAY, 181
- POINTLARRAY
  - EMF::POINTLARRAY, 182
- poppack.h, 187
- pshpack2.h, 187
- pshpack4.h, 188
- records
  - EMF::METAFILEDEVICECONTEXT, 171
- remove
  - EMF::GLOBALOBJECTS, 163
- serialize
  - EMF::EMRARC, 37
  - EMF::EMRARCTO, 39
  - EMF::EMRBEGINPATH, 41
  - EMF::EMRCLOSEFIGURE, 43
  - EMF::EMRCREATEBRUSHINDIRECT, 45
  - EMF::EMRCREATEPALETTE, 47
  - EMF::EMRCREATEPEN, 49
  - EMF::EMRDELETEOBJECT, 51
  - EMF::EMRELLIPSE, 53
  - EMF::EMRENDPATH, 55
  - EMF::EMREOF, 57
  - EMF::EMREXTCREATEFONTINDIRECTW, 60
  - EMF::EMREXTCREATEPEN, 62
  - EMF::EMREXTTEXTOUTA, 65
  - EMF::EMREXTTEXTOUTW, 67
  - EMF::EMRFILLPATH, 69
  - EMF::EMRLINETO, 71
  - EMF::EMRMODIFYWORLDTRANSFORM, 73
  - EMF::EMRMOVETOEX, 75
  - EMF::EMRPOLYBEZIER, 77
  - EMF::EMRPOLYBEZIER16, 80
  - EMF::EMRPOLYBEZIERTO, 83

- EMF::EMRPOLYBEZIERTO16, [86](#)
- EMF::EMRPOLYGON, [88](#)
- EMF::EMRPOLYGON16, [91](#)
- EMF::EMRPOLYLINE, [93](#)
- EMF::EMRPOLYLINE16, [96](#)
- EMF::EMRPOLYLINETO, [98](#)
- EMF::EMRPOLYLINETO16, [101](#)
- EMF::EMRPOLYPOLYGON, [103](#)
- EMF::EMRPOLYPOLYGON16, [106](#)
- EMF::EMRRECTANGLE, [109](#)
- EMF::EMRRESTOREDC, [111](#)
- EMF::EMRSAVEDC, [113](#)
- EMF::EMRSCALEVIEWPORTEXTEX, [115](#)
- EMF::EMRSCALEWINDOWEXTTEX, [117](#)
- EMF::EMRSELECTOBJECT, [119](#)
- EMF::EMRSETBKCOLOR, [122](#)
- EMF::EMRSETBKMODE, [124](#)
- EMF::EMRSETMAPMODE, [126](#)
- EMF::EMRSETMETARGN, [128](#)
- EMF::EMRSETMITERLIMIT, [130](#)
- EMF::EMRSETPIXELV, [132](#)
- EMF::EMRSETPOLYFILLMODE, [134](#)
- EMF::EMRSETTEXTALIGN, [136](#)
- EMF::EMRSETTEXTCOLOR, [138](#)
- EMF::EMRSETVIEWPORTEXTEX, [140](#)
- EMF::EMRSETVIEWPORTORGEX, [142](#)
- EMF::EMRSETWINDOWEXTTEX, [144](#)
- EMF::EMRSETWINDOWORGEX, [146](#)
- EMF::EMRSETWORLDTRANSFORM, [148](#)
- EMF::EMRSTROKEANDFILLPATH, [150](#)
- EMF::EMRSTROKEPATH, [152](#)
- EMF::ENHMETAHEADER, [154](#)
- EMF::METARECORD, [173](#)
- setStream
  - EMF::DATASTREAM, [33](#)
- size
  - EMF::EMRARC, [37](#)
  - EMF::EMRARCTO, [39](#)
  - EMF::EMRBEGINPATH, [41](#)
  - EMF::EMRCLOSEFIGURE, [43](#)
  - EMF::EMRCREATEBRUSHINDIRECT, [45](#)
  - EMF::EMRCREATEPALETTE, [47](#)
  - EMF::EMRCREATEPEN, [49](#)
  - EMF::EMRDELETEOBJECT, [51](#)
  - EMF::EMRELLIPSE, [53](#)
  - EMF::EMRENDPATH, [56](#)
  - EMF::EMREOF, [58](#)
  - EMF::EMREXTCREATEFONTINDIRECTW, [60](#)
  - EMF::EMREXTCREATEPEN, [62](#)
  - EMF::EMREXTTEXTOUTA, [65](#)
  - EMF::EMREXTTEXTOUTW, [67](#)
  - EMF::EMRFILLPATH, [69](#)
  - EMF::EMRLINETO, [71](#)
  - EMF::EMRMODIFYWORLDTRANSFORM, [73](#)
  - EMF::EMRMOVETOEX, [75](#)
  - EMF::EMRPOLYBEZIER, [78](#)
  - EMF::EMRPOLYBEZIER16, [80](#)
  - EMF::EMRPOLYBEZIERTO, [83](#)
  - EMF::EMRPOLYBEZIERTO16, [86](#)
  - EMF::EMRPOLYGON, [88](#)
  - EMF::EMRPOLYGON16, [91](#)
  - EMF::EMRPOLYLINE, [93](#)
  - EMF::EMRPOLYLINE16, [96](#)
  - EMF::EMRPOLYLINETO, [98](#)
  - EMF::EMRPOLYLINETO16, [101](#)
  - EMF::EMRPOLYPOLYGON, [104](#)
  - EMF::EMRPOLYPOLYGON16, [107](#)
  - EMF::EMRRECTANGLE, [109](#)
  - EMF::EMRRESTOREDC, [111](#)
  - EMF::EMRSAVEDC, [113](#)
  - EMF::EMRSCALEVIEWPORTEXTEX, [115](#)
  - EMF::EMRSCALEWINDOWEXTTEX, [117](#)
  - EMF::EMRSELECTOBJECT, [120](#)
  - EMF::EMRSETBKCOLOR, [122](#)
  - EMF::EMRSETBKMODE, [124](#)
  - EMF::EMRSETMAPMODE, [126](#)
  - EMF::EMRSETMETARGN, [128](#)
  - EMF::EMRSETMITERLIMIT, [130](#)
  - EMF::EMRSETPIXELV, [132](#)
  - EMF::EMRSETPOLYFILLMODE, [134](#)
  - EMF::EMRSETTEXTALIGN, [136](#)
  - EMF::EMRSETTEXTCOLOR, [138](#)
  - EMF::EMRSETVIEWPORTEXTEX, [140](#)
  - EMF::EMRSETVIEWPORTORGEX, [142](#)
  - EMF::EMRSETWINDOWEXTTEX, [144](#)
  - EMF::EMRSETWINDOWORGEX, [146](#)
  - EMF::EMRSETWORLDTRANSFORM, [148](#)
  - EMF::EMRSTROKEANDFILLPATH, [150](#)
  - EMF::EMRSTROKEPATH, [152](#)
  - EMF::ENHMETAHEADER, [154](#)
  - EMF::METARECORD, [174](#)
- unserialize
  - EMF::ENHMETAHEADER, [154](#)
- w16.h, [188](#)
- WCHARSTR
  - EMF::WCHARSTR, [183](#)
- winbase.h, [189](#)
- windef.h, [210](#)
- winerror.h, [213](#)
- wingdi.h, [235](#)
- winnt.h, [273](#)
- winuser.h, [329](#)