[MS-DOC]:

Word (.doc) Binary File Format

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Copyright © 2025 Microsoft Corporation

Revision Summary

Date	Revision History	Revision Class	Comments
6/27/2008	1.0	New	First release
1/16/2009	1.01	Minor	Updated IP Notice
7/13/2009	1.02	Major	Changes made for template compliance
8/28/2009	1.03	Editorial	Revised and edited the technical content
11/6/2009	1.04	Editorial	Revised and edited the technical content
2/19/2010	2.0	Editorial	Revised and edited the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Major	Updated and revised the technical content
6/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.05	Editorial	Changed language and formatting in the technical content.
11/15/2010	2.05	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.05	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.05	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.0	Major	Significantly changed the technical content.
4/11/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	3.1	Minor	Clarified the meaning of the technical content.
10/8/2012	3.2	Minor	Clarified the meaning of the technical content.
2/11/2013	3.3	Minor	Clarified the meaning of the technical content.
7/30/2013	3.3	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	3.3	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	3.3	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	3.4	Minor	Clarified the meaning of the technical content.
7/31/2014	4.0	Major	Significantly changed the technical content.

Copyright © 2025 Microsoft Corporation

Date	Revision History	Revision Class	Comments
10/30/2014	4.1	Minor	Clarified the meaning of the technical content.
3/16/2015	5.0	Major	Significantly changed the technical content.
9/4/2015	6.0	Major	Significantly changed the technical content.
7/15/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.
9/29/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.
10/17/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.
1/12/2017	6.1	Minor	Clarified the meaning of the technical content.
4/27/2018	7.0	Major	Significantly changed the technical content.
8/28/2018	8.0	Major	Significantly changed the technical content.
12/11/2018	8.1	Minor	Clarified the meaning of the technical content.
3/19/2019	8.1	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2019	8.1	None	No changes to the meaning, language, or formatting of the technical content.
2/16/2021	8.2	Minor	Clarified the meaning of the technical content.
4/22/2021	9.0	Major	Significantly changed the technical content.
8/17/2021	10.0	Major	Significantly changed the technical content.
11/16/2021	10.0	None	No changes to the meaning, language, or formatting of the technical content.
2/15/2022	10.0	None	No changes to the meaning, language, or formatting of the technical content.
5/17/2022	10.1	Minor	Clarified the meaning of the technical content.
11/15/2022	10.2	Minor	Clarified the meaning of the technical content.
4/16/2024	11.0	Major	Significantly changed the technical content.
5/21/2024	11.1	Minor	Clarified the meaning of the technical content.
8/20/2024	12.0	Major	Significantly changed the technical content.
11/12/2024	12.1	Minor	Clarified the meaning of the technical content.
2/18/2025	12.2	Minor	Clarified the meaning of the technical content.

Copyright © 2025 Microsoft Corporation

Table of Contents

1 Introduction 14
1.1 Glossary 14
1.2 References 23
1.2.1 Normative References 23
1.2.2 Informative References 24
1.3 Overview 25
1.3.1 Characters 25
1.3.2 PLCs 25
1.3.3 Formatting 25
1.3.4 Tables 25
1.3.5 Pictures 26
1.3.6 The FIB 26
1.3.7 Byte Ordering 26
1.3.8 General Organization of This Documentation 26
1.4 Relationship to Protocols and Other Structures 27
1.5 Applicability Statement 27
1.6 Versioning and Localization 27
1.7 Vendor-Extensible Fields 28

2 Structures 29
2.1 File Structure 29
2.1.1 WordDocument Stream 29
2.1.2 1Table Stream or 0Table Stream 29
2.1.3 Data Stream 29
2.1.4 ObjectPool Storage 29
2.1.4.1 ObjInfo Stream 30
2.1.4.2 Print Stream 30
2.1.4.3 EPrint Stream 30
2.1.5 Custom XML Data Storage 30
2.1.6 Summary Information Stream 30
2.1.7 Document Summary Information Stream 30
2.1.8 Encryption Stream 30
2.1.9 Macros Storage 31
2.1.10 XML Signatures Storage 31
2.1.11 Signatures Stream 31
2.1.12 Information Rights Management Data Space Storage 31
2.1.13 Protected Content Stream 31
2.2 Fundamental Concepts 31
2.2.1 Character Position (CP) 31
2.2.2 PLC 32
2.2.3 Valid Selection 32
2.2.4 STTB 33
2.2.5 Property Storage 35
2.2.5.1 Sprm 36
2.2.5.2 Prl 37
2.2.6 Encryption and Obfuscation (Password to Open) 37
2.2.6.1 XOR Obfuscation 38
2.2.6.2 Office Binary Document RC4 Encryption 38
2.2.6.3 Office Binary Document RC4 CryptoAPI Encryption 38
2.3 Document Parts 39
2.3.1 Main Document 39
2.3.2 Footnotes 39
2.3.3 Headers 39
2.3.4 Comments 40
2.3.5 Endnotes 41

Copyright © 2025 Microsoft Corporation

2.3.6 Textboxes 41
2.3.7 Header Textboxes 41
2.4 Document Content 41
2.4.1 Retrieving Text 41
2.4.2 Determining Paragraph Boundaries 42
2.4.3 Overview of Tables 43
2.4.4 Determining Cell Boundaries 46
2.4.5 Determining Row Boundaries 47
2.4.6 Applying Properties 48
2.4.6.1 Direct Paragraph Formatting 48
2.4.6.2 Direct Character Formatting 48
2.4.6.3 Determining List Formatting of a Paragraph 49
2.4.6.4 Determining Level Number of a Paragraph 50
2.4.6.5 Determining Properties of a Style 51
2.4.6.6 Determining Formatting Properties 52
2.4.7 Application Data For VtHyperlink 55
2.5 The File Information Block 55
2.5.1 Fib 55
2.5.2 FibBase 57
2.5.3 FibRgW97 59
2.5.4 FibRgLw97 60
2.5.5 FibRgFcLcb 62
2.5.6 FibRgFcLcb97 62
2.5.7 FibRgFcLcb2000 82
2.5.8 FibRgFcLcb2002 85
2.5.9 FibRgFcLcb2003 92
2.5.10 FibRgFcLcb2007 99
2.5.11 FibRgCswNew 102
2.5.12 FibRgCswNewData2000 103
2.5.13 FibRgCswNewData2007 103
2.5.14 Determining the nFib 104
2.5.15 How to read the FIB 104
2.6 Single Property Modifiers 104
2.6.1 Character Properties 105
2.6.2 Paragraph Properties 120
2.6.3 Table Properties 131
2.6.4 Section Properties 140
2.6.5 Picture Properties 148
2.7 Document Properties 148
2.7.1 Dop 148
2.7.2 DopBase 149
2.7.3 Dop95 155
2.7.4 Dop97 156
2.7.5 Dop2000 159
2.7.6 Dop2002 163
2.7.7 Dop2003 166
2.7.8 Dop2007 168
2.7.9 Dop2010 170
2.7.10 Dop2013 170
2.7.11 Copts60 171
2.7.12 Copts80 172
2.7.13 Copts 173
2.7.14 Asumyi 176
2.7.15 Dogrid 177
2.7.16 DopTypography 178
2.7.17 DopMth 180
2.8 PLCs 182
2.8.1 Plcbkf 182

Copyright © 2025 Microsoft Corporation

2.8.2 Plcbkfd 183
2.8.3 Plcbkl 184
2.8.4 Plcbkld 184
2.8.5 PlcBteChpx 185
2.8.6 PlcBtePapx 185
2.8.7 PlcfandRef 186
2.8.8 PlcfandTxt 186
2.8.9 PlcfAsumy 187
2.8.10 Plcfbkf 187
2.8.11 Plcfbkfd 188
2.8.12 Plcfbkl 189
2.8.13 Plcfbkld 189
2.8.14 Plcfcookie 189
2.8.15 PlcfcookieOld 190
2.8.16 PlcfendRef 190
2.8.17 PlcfendTxt 191
2.8.18 Plcffactoid 191
2.8.19 PlcffndRef 192
2.8.20 PlcffndTxt 192
2.8.21 Plcfgram 193
2.8.22 Plcfhdd 193
2.8.23 PlcfHdrtxbxTxt 194
2.8.24 Plcflad 194
2.8.25 Plcfld 195
2.8.26 PlcfSed 196
2.8.27 PlcfSpa 196
2.8.28 Plcfspl 197
2.8.29 PlcfTch 197
2.8.30 PlcfTxbxBkd 198
2.8.31 PlcfTxbxHdrBkd 199
2.8.32 PlcftxbxTxt 199
2.8.33 Plcfuim 200
2.8.34 PlcfWKB 200
2.8.35 PlcPcd 201
2.9 Basic Types 202
2.9.1 Acd 202
2.9.2 Afd 203
2.9.3 ASUMY 204
2.9.4 ATNBE 204
2.9.5 AtrdExtra 204
2.9.6 ATRDPost10 205
2.9.7 ATRDPre10 205
2.9.8 BKC 206
2.9.9 BKF 207
2.9.10 BKFD 207
2.9.11 BKL 208
2.9.12 BKLD 208
2.9.13 BlockSel 209
2.9.14 Bool16 209
2.9.15 Bool8 209
2.9.16 Brc 209
2.9.17 Brc80 210
2.9.18 Brc80MayBeNil 210
2.9.19 BrcCvOperand 210
2.9.20 BrcMayBeNil 211
2.9.21 BrcOperand 211
2.9.22 BrcType 211
2.9.23 BxPap 218

Copyright © 2025 Microsoft Corporation

2.9.24 CAPI 218
2.9.25 CDB 219
2.9.26 CellHideMarkOperand 220
2.9.27 CellRangeFitText 220
2.9.28 CellRangeNoWrap 220
2.9.29 CellRangeTextFlow 221
2.9.30 CellRangeVertAlign 221
2.9.31 CFitTextOperand 221
2.9.32 Chpx 222
2.9.33 ChpxFkp 222
2.9.34 Cid 223
2.9.35 CidAllocated 223
2.9.36 CidFci 223
2.9.37 CidMacro 226
2.9.38 Clx 227
2.9.39 CMajorityOperand 227
2.9.40 Cmt 227
2.9.41 CNFOperand 228
2.9.42 CNS 228
2.9.43 COLORREF 229
2.9.44 COSL 229
2.9.45 CSSA 230
2.9.46 CSSAOperand 230
2.9.47 CSymbolOperand 231
2.9.48 CTB 231
2.9.49 CTBWRAPPER 233
2.9.50 Customization 233
2.9.51 DCS 234
2.9.52 DefTableShd80Operand 235
2.9.53 DefTableShdOperand 235
2.9.54 DispFldRmOperand 235
2.9.55 Dofr 236
2.9.56 DofrFsn 236
2.9.57 DofrFsnFnm 237
2.9.58 DofrFsnName 238
2.9.59 DofrFsnp 238
2.9.60 DofrFsnSpbd 238
2.9.61 Dofrh 239
2.9.62 DofrRglstsf 239
2.9.63 Dofrt 240
2.9.64 DPCID 240
2.9.65 DTTM 241
2.9.66 FACTOIDINFO 242
2.9.67 FactoidSpls 242
2.9.68 FarEastLayoutOperand 242
2.9.69 Fatl 243
2.9.70 FBKF 244
2.9.71 FBKFD 244
2.9.72 FBKLD 244
2.9.73 FcCompressed 245
2.9.74 FCCT 246
2.9.75 Fci 247
2.9.76 FCKS 315
2.9.77 FCKSOLD 316
2.9.78 FFData 317
2.9.79 FFDataBits 319
2.9.80 FFID 320
2.9.81 FFM 321

Copyright © 2025 Microsoft Corporation

2.9.82 FFN 321
2.9.83 FieldMapBase 323
2.9.84 FieldMapDataItem 323
2.9.85 FieldMapInfo 324
2.9.86 FieldMapTerminator 325
2.9.87 FilterDataItem 325
2.9.88 Fld 326
2.9.89 fldch 326
2.9.90 flt 326
2.9.91 FNFB 329
2.9.92 FNIF 330
2.9.93 FNPI 330
2.9.94 FOBJH 331
2.9.95 FrameTextFlowOperand 331
2.9.96 FSDAP 332
2.9.97 Fsnk 332
2.9.98 Fssd 332
2.9.99 FssUnits 333
2.9.100 FTO 333
2.9.101 Fts 333
2.9.102 FtsWWidth_Indent 334
2.9.103 FtsWWidth_Table 334
2.9.104 FtsWWidth_TablePart 335
2.9.105 FTXBXNonReusable 335
2.9.106 FTXBXS 336
2.9.107 FTXBXSReusable 337
2.9.108 GOSL 337
2.9.109 GrammarSpls 338
2.9.110 grffldEnd 338
2.9.111 grfhic 339
2.9.112 GRFSTD 340
2.9.113 GrLPUpxSw 341
2.9.114 GrpPrlAndIstd 341
2.9.115 HFD 341
2.9.116 HFDBits 342
2.9.117 Hplxsdr 342
2.9.118 HresiOperand 343
2.9.119 Ico 343
2.9.120 IDPCI 344
2.9.121 Ipat 345
2.9.122 IScrollType 349
2.9.123 ItcFirstLim 349
2.9.124 Kcm 349
2.9.125 Kme 350
2.9.126 Kt 350
2.9.127 Kul 351
2.9.128 LadSpls 351
2.9.129 LBCOperand 352
2.9.130 LEGOXTR_V11 352
2.9.131 LFO 353
2.9.132 LFOData 354
2.9.133 LFOLVL 354
2.9.134 LID 355
2.9.135 LPStd 355
2.9.136 LPStshi 355
2.9.137 LPStshiGrpPrl 355
2.9.138 LPUpxChpx 356
2.9.139 LPUpxChpxRM 356

Copyright © 2025 Microsoft Corporation

2.9.140 LPUpxPapx 356
2.9.141 LPUpxPapxRM 357
2.9.142 LPUpxRm 357
2.9.143 LPUpxTapx 357
2.9.144 LPXCharBuffer9 358
2.9.145 LSD 358
2.9.146 LSPD 359
2.9.147 LSTF 359
2.9.148 Lstsf 360
2.9.149 LVL 360
2.9.150 LVLF 361
2.9.151 MacroName 363
2.9.152 MacroNames 364
2.9.153 MathPrOperand 364
2.9.154 Mcd 364
2.9.155 MDP 365
2.9.156 MFPF 365
2.9.157 NilBrc 366
2.9.158 NilPICFAndBinData 366
2.9.159 NumRM 367
2.9.160 NumRMOperand 369
2.9.161 OcxInfo 369
2.9.162 ODSOPropertyBase 370
2.9.163 ODSOPropertyLarge 372
2.9.164 ODSOPropertyStandard 372
2.9.165 ODT 372
2.9.166 ODTPersist1 373
2.9.167 ODTPersist2 374
2.9.168 OfficeArtClientAnchor 375
2.9.169 OfficeArtClientData 375
2.9.170 OfficeArtClientTextbox 375
2.9.171 OfficeArtContent 376
2.9.172 OfficeArtWordDrawing 376
2.9.173 PANOSE 377
2.9.174 PapxFkp 381
2.9.175 PapxInFkp 382
2.9.176 PbiGrfOperand 382
2.9.177 Pcd 383
2.9.178 Pcdt 383
2.9.179 PChgTabsAdd 384
2.9.180 PChgTabsDel 384
2.9.181 PChgTabsDelClose 384
2.9.182 PChgTabsOperand 385
2.9.183 PChgTabsPapxOperand 386
2.9.184 PgbApplyTo 386
2.9.185 PgbOffsetFrom 386
2.9.186 PgbPageDepth 386
2.9.187 PGPArray 387
2.9.188 PGPInfo 387
2.9.189 PGPOptions 388
2.9.190 PICF 389
2.9.191 PICF_Shape 390
2.9.192 PICFAndOfficeArtData 390
2.9.193 PICMID 391
2.9.194 PlcfGlsy 393
2.9.195 PlfAcd 393
2.9.196 PlfCosl 393
2.9.197 PlfGosl 394

Copyright © 2025 Microsoft Corporation

2.9.198 PlfguidUim 394
2.9.199 PlfKme 395
2.9.200 PlfLfo 395
2.9.201 PlfLst 395
2.9.202 PlfMcd 396
2.9.203 PLRSID 396
2.9.204 Pmfs 397
2.9.205 Pms 399
2.9.206 PnFkpChpx 401
2.9.207 PnFkpPapx 401
2.9.208 PositionCodeOperand 401
2.9.209 Prc 402
2.9.210 PrcData 402
2.9.211 PrDrvr 402
2.9.212 PrEnvLand 403
2.9.213 PrEnvPort 403
2.9.214 Prm 403
2.9.215 Prm0 403
2.9.216 Prm1 405
2.9.217 PropRMark 405
2.9.218 PropRMarkOperand 406
2.9.219 ProtectionType 406
2.9.220 PRTI 406
2.9.221 PTIstdInfoOperand 407
2.9.222 Rca 407
2.9.223 RecipientBase 408
2.9.224 RecipientDataItem 408
2.9.225 RecipientInfo 409
2.9.226 RecipientTerminator 410
2.9.227 Rfs 410
2.9.228 RgCdb 411
2.9.229 RgxOcxInfo 411
2.9.230 RmdThreading 412
2.9.231 Rnc 416
2.9.232 RouteSlip 417
2.9.233 RouteSlipInfo 418
2.9.234 RouteSlipProtectionEnum 419
2.9.235 SBkcOperand 419
2.9.236 SBOrientationOperand 419
2.9.237 SClmOperand 419
2.9.238 SDmBinOperand 420
2.9.239 SDTI 420
2.9.240 SDTT 421
2.9.241 SDxaColSpacingOperand 421
2.9.242 SDxaColWidthOperand 421
2.9.243 Sed 422
2.9.244 Selsf 422
2.9.245 Sepx 424
2.9.246 SFpcOperand 425
2.9.247 Shd 425
2.9.248 Shd80 426
2.9.249 SHDOperand 427
2.9.250 SLncOperand 427
2.9.251 SmartTagData 427
2.9.252 SortColumnAndDirection 428
2.9.253 Spa 428
2.9.254 SpellingSpls 430
2.9.255 SPgbPropOperand 431

Copyright © 2025 Microsoft Corporation

2.9.256 SPLS 431
2.9.257 SPPOperand 432
2.9.258 STD 432
2.9.259 Stdf 433
2.9.260 StdfBase 433
2.9.261 StdfPost2000 435
2.9.262 StdfPost2000OrNone 436
2.9.263 StkCharGRLPUPX 436
2.9.264 StkCharLPUpxGrLPUpxRM 437
2.9.265 StkCharUpxGrLPUpxRM 437
2.9.266 StkListGRLPUPX 437
2.9.267 StkParaGRLPUPX 438
2.9.268 StkParaLPUpxGrLPUpxRM 438
2.9.269 StkParaUpxGrLPUpxRM 439
2.9.270 StkTableGRLPUPX 439
2.9.271 STSH 440
2.9.272 STSHI 441
2.9.273 STSHIB 441
2.9.274 Stshif 442
2.9.275 StshiLsd 443
2.9.276 SttbfAssoc 443
2.9.277 SttbfAtnBkmk 444
2.9.278 SttbfAutoCaption 445
2.9.279 SttbfBkmk 446
2.9.280 SttbfBkmkBPRepairs 450
2.9.281 SttbfBkmkFactoid 451
2.9.282 SttbfBkmkFcc 452
2.9.283 SttbfBkmkProt 453
2.9.284 SttbfBkmkSdt 454
2.9.285 SttbfCaption 455
2.9.286 SttbfFfn 456
2.9.287 SttbfGlsy 456
2.9.288 SttbFnm 457
2.9.289 SttbfRfs 458
2.9.290 SttbfRMark 459
2.9.291 SttbGlsyStyle 460
2.9.292 SttbListNames 461
2.9.293 SttbProtUser 462
2.9.294 SttbRgtplc 463
2.9.295 SttbSavedBy 463
2.9.296 SttbTtmbd 464
2.9.297 SttbW6 465
2.9.298 StwUser 465
2.9.299 Sty 466
2.9.300 TabJC 467
2.9.301 TabLC 467
2.9.302 TableBordersOperand 468
2.9.303 TableBordersOperand80 469
2.9.304 TableBrc80Operand 469
2.9.305 TableBrcOperand 470
2.9.306 TableCellWidthOperand 471
2.9.307 TableSel 471
2.9.308 TableShadeOperand 472
2.9.309 TBC 472
2.9.310 TBD 472
2.9.311 TBDelta 473
2.9.312 Tbkd 475
2.9.313 TC80 475

Copyright © 2025 Microsoft Corporation

2.9.314 TCellBrcTypeOperand 476
2.9.315 Tcg 476
2.9.316 Tcg255 477
2.9.317 TCGRF 477
2.9.318 TcgSttbf 478
2.9.319 TcgSttbfCore 478
2.9.320 Tch 479
2.9.321 TDefTableOperand 480
2.9.322 TDxaColOperand 480
2.9.323 TextFlow 481
2.9.324 TInsertOperand 481
2.9.325 TIQ 481
2.9.326 TLP 482
2.9.327 ToggleOperand 482
2.9.328 Tplc 483
2.9.329 TplcBuildIn 483
2.9.330 TplcUser 484
2.9.331 Ttmbd 484
2.9.332 UFEL 485
2.9.333 UID 486
2.9.334 UidSel 486
2.9.335 UIM 486
2.9.336 UpxChpx 487
2.9.337 UPXPadding 488
2.9.338 UpxPapx 488
2.9.339 UpxRm 489
2.9.340 UpxTapx 490
2.9.341 VerticalAlign 492
2.9.342 VerticalMergeFlag 492
2.9.343 VertMergeOperand 492
2.9.344 Vjc 493
2.9.345 WHeightAbs 493
2.9.346 WKB 493
2.9.347 Wpms 494
2.9.348 Wpmsdt 495
2.9.349 XAS 495
2.9.350 XAS_nonNeg 495
2.9.351 XAS_plusOne 495
2.9.352 XSDR 496
2.9.353 Xst 496
2.9.354 Xstz 497
2.9.355 YAS 497
2.9.356 YAS_nonNeg 497
2.9.357 YAS_plusOne 497

3 Structure Examples 498
3.1 Example of a Clx 498
3.2 Example of a section 503
3.3 Example of a Bookmark 508
3.4 Example of a PlcBteChpx 513
3.5 Example of a PlcBtePapx 517
3.6 Example of Table Row Properties 523
3.7 Example of a List 534

4 Security Considerations 545
4.1 Encryption and Obfuscation (Password to Open) 545
4.2 Write Reservation Password 545

5 Appendix A: Product Behavior 546

Copyright © 2025 Microsoft Corporation

6 Change Tracking 565
7 Index 566

Copyright © 2025 Microsoft Corporation

1 Introduction

This document specifies the Word Binary File Format (.doc) Structure, which defines the Word Binary File Format (.doc). The Word Binary File Format is a collection of records and structures that specify text, tables, fields, pictures, embedded XML markup, and other document content. The content can be printed on pages of multiple sizes or displayed on a variety of devices.

The Word Binary File Format begins with a master record named the File Information Block, which references all other data in the file. By following links from the File Information Block, an application can locate all text and other objects in the file and compute the properties of those objects.

Sections 1.7 and 2 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

accelerator key: Any combination of keys that are pressed simultaneously to run a command.

allocated command: A built-in command that requires the user to specify a value for a parameter when customizing the command.

anchor: A set of qualifiers and quantifiers that specifies the location of an element or object within a document. These values are typically relative to another element or known location in the document, such as the edge of a page or margin.

annotation bookmark: An entity in a document that is used to denote the range of content to which a comment applies.

ASCII: The American Standard Code for Information Interchange (ASCII) is an 8-bit character-encoding scheme based on the English alphabet. ASCII codes represent text in computers, communications equipment, and other devices that work with text. ASCII refers to a single 8-bit ASCII character or an array of 8-bit ASCII characters with the high bit of each character set to zero.

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [RFC5234].

auto spacing: A condition in which space is inserted automatically before and after a series of consecutive paragraphs that do not have breaks or other items between them.

AutoCaption: A feature that adds a **caption** to an object automatically when the object is inserted in a document.

AutoCorrect: A feature that corrects errors and makes other substitutions in a document automatically by using default and user-defined settings.

auto-hyphenated: A condition of content where the distance between the text is measured and maintained to force breaks automatically in elongated words that would not otherwise end correctly on a line.

automark file: A file that stores the text, location, and index level of a set of characters that were marked for inclusion in a document index.

Copyright © 2025 Microsoft Corporation

AutoSummary: A process in which key points are identified in selected text by analyzing document content. A score is assigned to each sentence; sentences that contain frequently used words are given a higher score.

AutoText: A storage location for text and graphics, such as a standard contract clause, that can be used multiple times in one or more documents. Each selection of text or graphics is recorded as an AutoText entry and assigned a unique name.

bar tab: A tab that specifies where to draw a vertical line or bar in a paragraph. It neither affects the position of characters nor creates a custom tab stop in a paragraph.

bidirectional compatibility: The ability to display and process text in two directions, right-to-left and left-to-right.

big-endian: Multiple-byte values that are byte-ordered with the most significant byte stored in the memory location with the lowest address.

bookmark: An entity that is used in a document to denote the beginning and ending character positions of specific text in the document, and optionally, metadata about that text or its relationship to other referenced parts of the document.

caption: One or more characters that can be used as a label for display purposes or as an identifier.

cascading style sheet (CSS): An extension to **HTML** that enables authors and users of HTML documents to attach style sheets to those documents, as described in [CSS-LEVEL1] and [CSS-LEVEL2]. A style sheet includes typographical information about the appearance of a page, including the font for text on the page.

cell: A box that is formed by the intersection of a row and a column in a worksheet or a table. A cell can contain numbers, strings, and formulas, and various formats can be applied to that data.

cell margin: A measurement of the distance between the border of a cell and the nearest pixel in a character or digit of data in the cell. There are top, bottom, right, and left margins. See also **cell spacing**.

cell spacing: A measurement of the distance between the cells of a table or worksheet. Most tables and worksheets are implemented with contiguous cells, in which case the cell spacing value is 0 (zero). See also **cell margin**.

CGAPI: An API that is implemented by **grammar checkers** that have been licensed to Microsoft Corporation by external vendors.

chapter numbering: A page numbering format in which pages are numbered relative to the beginning of a chapter within a document instead of the beginning of the document. The chapter number is typically included in a page number; for example "3 – 2," where "3" is the chapter number and "2" is the number of that page within that chapter.

character pitch: A quality that measures the number of characters that can be printed in a horizontal inch. Pitch is typically used to measure monospace fonts.

character set: A mapping between the characters of a written language and the values that are used to represent those characters to a computer.

character unit: A horizontal unit of measurement that is relative to the document grid and is used to position content in a document.

class identifier (CLSID): A GUID that identifies a software component; for instance, a DCOM object class or a COM class.

Copyright © 2025 Microsoft Corporation

code page: An ordered set of characters of a specific script in which a numerical index (code-point value) is associated with each character. Code pages are a means of providing support for **character sets** and keyboard layouts used in different countries. Devices such as the display and keyboard can be configured to use a specific code page and to switch from one code page (such as the United States) to another (such as Portugal) at the user's request.

Component Object Model (COM): An object-oriented programming model that defines how objects interact within a single process or between processes. In **COM**, clients have access to an object through interfaces implemented on the object. For more information, see [MS-DCOM].

connection string: A series of arguments, delimited by a semicolon, that defines the location of a database and how to connect to it.

custom toolbar: A type of toolbar that contains a user-defined set of controls and is not included in an application by default. A custom toolbar has a toolbar identifier value of "1".

custom toolbar control: A user-defined control that can be added to a toolbar. A custom toolbar control has a **toolbar control identifier (TCID)** value of "1" and can be one of the following types of controls: ActiveX, Button, ComboBox, DropDown, Edit, or Popup.

deletion point: A position between two existing characters, or a position before or after a character, where text was removed. If a caret is positioned at a deletion point, the point can retain unique formatting and that formatting can be reapplied to any text that is inserted at the deletion point.

digital signature: A value that is generated by using a digital signature algorithm, taking as input a private key and an arbitrary-length string, such that a specific verification algorithm is satisfied by the value, the input string, and the public key corresponding to the input private key.

document: An object in a content database such as a file, folder, list, or site. Each object is identified by a **URI**.

document grid: A feature that enables the precise layout of full-width East Asian language characters by specifying the number of characters per line and the number of lines per page.

document template: A file that serves as the basis for new documents.

East Asian character: A character that is part of the Simplified Chinese, Traditional Chinese, Japanese, or Korean **character set**.

East Asian language: A spoken or written communication that consists of words that are used within the grammatical and syntactic structure of Simplified Chinese, Traditional Chinese, Japanese, or Korean.

East Asian line breaking rules: A set of algorithms that define how text is parsed and displayed to ensure that line breaks and word wraps follow the rules of various East Asian languages, including Simplified Chinese, Traditional Chinese, Japanese, and Korean.

end of cell mark: A character with a hexadecimal value of "0x07" that is used to indicate the end of a cell in a table.

end of row mark: The combination of a character, hexadecimal value of "0x07", and a paragraph property, sprmPFTtp, that is used to indicate the end of a row in a table.

endnote: A note that appears at the end of a section or document and that is referenced by text in the main body of the document. An endnote consists of two linked parts, a reference mark within the main body of text and the corresponding text of the note.

endnote continuation notice: A set of characters indicating that an endnote continues to the next page. The default notice is blank.

Copyright © 2025 Microsoft Corporation

endnote continuation separator: A set of characters that indicates the end of document text on a page and the beginning of endnotes that continue from the preceding page.

endnote separator: A set of characters that separates document text from endnotes about that text. The default separator is a horizontal line.

field: An element or attribute in a data source that can contain data.

field type: A name that identifies the action or effect that a field has within a document. Examples of field types are Author, Page, Comments, and Date.

file allocation table (FAT): A data structure that the operating system creates when a volume is formatted by using **FAT** or FAT32 file systems. The operating system stores information about each file in the **FAT** so that it can retrieve the file later.

footer: One or more lines of text in the bottom margin area of a page in a document or a slide in a presentation. A footer typically contains elements such as the page number and the name of the file.

footnote: A note that appears at the end of a page, section, chapter, or publication. It explains, comments on, or provides references for text in the main body of a document. A footnote consists of two linked parts, a reference mark within the main body of the document and the corresponding text of the note.

footnote continuation notice: A set of characters indicating that a footnote continues to the next page. The default notice is blank.

footnote continuation separator: A set of characters that indicates the end of document text on a page and the beginning of footnotes that continue from the preceding page.

footnote separator: A set of characters that separates document text from footnotes about that text. The default separator is a horizontal line.

form field: A data-entry area on a webpage, document, or form.

format consistency checker: An application that applies a wavy blue underline to text where the formatting is similar, but not identical, to comparable text in a document.

format consistency-checker bookmark: An entity in a document that is used to denote text where the formatting is similar, but not identical, to comparable text in the document, and the user indicated that the formatting inconsistency is not to be flagged.

frame: A space, displayed onscreen as a box, that contains a specific element of a publication.

full save: A process in which an existing file is overwritten with all of the additions, changes, and other content in a document.

full screen view: A document view that expands the display of a document to fill the computer screen. The view hides menus, toolbars, and taskbars.

grammar checker: An application that uses default or user-defined settings to search for grammatical errors in a document.

grammar checker cookie: An entity in a document that a grammar checker uses to denote a possible grammatical error in the document and data about that error.

gutter: An area above a column heading and to the left of a row heading. A gutter typically displays outline symbols that are used to expand and collapse groups of cells.

gutter margin: A margin setting that adds extra space to the side or top margin of a document that will be printed and bound. A gutter margin ensures that text is not obscured by the binding.

Copyright © 2025 Microsoft Corporation

Hangul-Hanja converter (HHC): A collection of dictionaries that readers can use to search for and select a Hanja word that corresponds to a specified Hangul word, or a Hangul word that corresponds to a specified Hanja word.

header: A line, or lines, of content in the top margin area of a page in a document or a slide in a presentation. A header typically contains elements such as the title of the chapter, the title of the document, a page number, or the name of the author.

heading style: A type of paragraph style that also specifies a heading level. There are as many as nine built-in heading styles, Heading 1 through Heading 9.

horizontal band: A set of rows in a table that are treated as a single unit, typically to ensure the consistency of the layout and the format.

HTML image map: An image that contains more than one hyperlink on a webpage. Clicking various parts of the image links the user to other resources on another part of the page, a different page, or a file.

hybrid list: A nine-level list that is exposed in the user interface as a collection of nine, one-level lists, instead of a single nine-level list.

Hyperlink view: A document view that displays a document as it would appear as a webpage.

Hypertext Markup Language (HTML): An application of the Standard Generalized Markup Language (SGML) that uses tags to mark elements in a document, as described in [HTML].

incremental save: A process in which an existing file is modified to reflect only additions or changes to a document, while maintaining all other existing content in the file.

Input Method Editor (IME): An application that is used to enter characters in written Asian languages by using a standard 101-key keyboard. An IME consists of both an engine that converts keystrokes into phonetic and ideographic characters and a dictionary of commonly used ideographic words.

insertion point: A position between two existing characters, or a position before or after a character, where text can be inserted. If a caret is positioned at an insertion point, the point can have unique formatting, which is applied to any text that is inserted at the insertion point.

kinsoku: A rule set in the Japanese language that is used to determine characters that are not permitted at the beginning or end of a line.

Kumimoji: A text layout setting that displays annotative characters inline next to the text to which they apply. It is typically used with East Asian text to indicate pronunciation.

labels document: A document that stores label design and printing information in conjunction with a mail merge document.

language auto-detection: A process that automatically determines the language code identifier (LCID) for text in a document.

left-to-right: A reading order in which characters in words are read from left to right, and words are read from left to right in sentences.

line numbers: A formatting property in which each line of text is prefixed with a sequential number as part of a larger collection of lines on a page.

line unit: A vertical unit of measurement that is relative to the document grid and is used to position content in a document.

list level: A condition of a paragraph that specifies which numbering system and indentation to use, relative to other paragraphs in a bulleted or numbered list.

Copyright © 2025 Microsoft Corporation

list tab: A tab stop that is between a list number or bullet and the text of that list item.

little-endian: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

logical left: A position that is relative to the language orientation of a document. Logical left means left, except in a right-to-left language where it means right. Also referred to as leading edge.

logical right: A position that is relative to the language orientation of a document. Logical right means right, except in a right-to-left language where it means left. Also referred to as trailing edge.

macro: A set of instructions that are recorded or written, and then typically saved to a file. When a macro is run, all of the instructions are performed automatically.

mail merge: The process of merging information into a document from a data source, such as an address book or database, to create customized documents, such as form letters or mailing labels.

mail merge data source: A file or address book that contains the information to be merged into a document during a mail merge operation.

mail merge header document: A file that contains the names of the fields in a mail merge data source.

mail merge main document: A document that contains the text and graphics that are the same for each version of the merged document, such as the return address or salutation in a form letter.

manifest: A file that stores metadata about an expansion pack, such as the name of the expansion pack, the files and resources that are included in the expansion pack, and the dependencies that it has on other files and components.

master document: A document that refers to or contains one or more other documents, which are referred to as subdocuments. A master document can be used to configure and manage a multipart document, such as a book with multiple chapters.

menu toolbar: A type of toolbar that is displayed in an application window, typically at the top, and provides a set of menu controls from which the user can select. Activating a control on the toolbar displays a list of commands in that menu, and the menu remains open until the user closes it or chooses a menu command.

message identifier: A string that uniquely identifies an email message.

NLCheck: An API that is implemented by **grammar checkers** that were developed by Microsoft Corporation.

Normal template: The default global template that is used for any type of document. Users can modify this template to change default document formatting, or content for any new document.

Normal view: A document view that displays text formatting and a simplified page layout of a document. The Normal view hides some layout elements such as the header and footer. Referred to as Draft view in Microsoft Office Word 2007 and Microsoft Word 2010.

NT file system (NTFS): A proprietary Microsoft file system. For more information, see [MSFT-NTFS].

number text: A string that is calculated automatically and represents the numbering scheme and position of a paragraph in a bulleted or numbered list.

Copyright © 2025 Microsoft Corporation

Object Linking and Embedding (OLE): A technology for transferring and sharing information between applications by inserting a file or part of a file into a compound document. The inserted file can be either embedded or linked. See also embedded object and linked object.

OLE compound file: A form of structured storage, as described in [MS-CFB]. A compound file allows independent storages and streams to exist within a single file.

OLE control: A reusable software component that is designed to work in containers that support **Object Linking and Embedding (OLE)** 2.0.

OLE object: An object that supports the **Object Linking and Embedding (OLE)** protocol.

outline level: A type of paragraph formatting that can be used to assign a hierarchical level, Level 1 through Level 9, to paragraphs in a document. After outline levels are assigned, an outline of a document can be viewed by using Outline view, the document map, or the navigation pane.

page border: A line that can be applied to the outer edge of a page in a document. A page border can be formatted for style, color, and thickness.

paragraph mark: An entity in a document that is used to denote the end of a paragraph and has a Unicode character code of 13.

paragraph style: A combination of character- and paragraph-formatting characteristics that are named and stored as a set. Users can select a paragraph and use a paragraph style to apply all of the formatting characteristics to the paragraph simultaneously.

personal style: A list of formatting settings that is applied to a document or an Internet message when it is opened or created by a specific user on a specific computer. The settings are associated with a user and a computer.

physical left: A leftward position that is not relative to the language orientation of document content. See also **logical left**.

physical right: A rightward position that is not relative to the language orientation of document content. See also **logical right**.

point: A unit of measurement for fonts and spacing. A point is equal to 1/72 of an inch.

policy labels: A set of fields that stores metadata about a document and is defined by an information management policy.

primary shortcut key: The default combination of keys that are pressed simultaneously to execute a command. See also **secondary shortcut key**.

Print Preview view: A document view that displays a document as it will appear on a printed page.

ProgID: An identifier that is used by the Windows registry to uniquely identify an object and is in the form OLEServerName.ObjectName, for example, "Excel.Sheet" or "PowerPoint.Slide."

property revision mark: A type of revision mark indicating that one or more formatting properties, such as bold, indentation, or spacing, changed.

range-level protection: A mechanism that permits users to change only specific parts of a protected document while restricting access to all other parts of the document. See also **range-level protection bookmark**.

range-level protection bookmark: An entity in a document that is used to denote a range of content that is an exception to a document-level protection setting.

Copyright © 2025 Microsoft Corporation

Reading Layout view: A document view that displays a document as it will appear on a printed page and is optimized for reading a document on a computer screen. Two pages are displayed simultaneously, side-by-side.

repair bookmark: An entity in a document that is used to denote text that was changed automatically during a document repair operation.

rich text: Text that is formatted in the Rich Text Format, as described in [MSFT-RTF].

Rich Text Format (RTF): Text with formatting as described in [MSFT-RTF].

right-to-left: A reading and display order that is optimized for right-to-left languages.

Ruby: A text layout setting that displays annotative characters above or to the right of the text to which it applies. It is typically used in East Asian documents to indicate pronunciation or to provide a brief annotation.

ScreenTip: A small pop-up window that provides brief context-sensitive help when users point to an item.

secondary shortcut key: A user-defined combination of keys that are pressed simultaneously to execute a command. See also **primary shortcut key**.

section: A portion of a document that is terminated by a section break or the end of the document. A section can store unique, page-level formatting, such as page size and orientation, and other formatting features such as headers and footers.

section break: A special character that terminates a section and acts as a repository for the properties of the specified section.

shading pattern: A background color pattern against which characters and graphics are displayed, typically in tables. The color can be no color or it can be a specific color with a transparency or pattern value.

smart tag: A feature that adds the ability to recognize and label specific data types, such as people's names, within a document and displays an action button that enables users to perform common tasks for that data type.

smart tag bookmark: An entity in a document that is used to denote the location and presence of a smart tag.

smart tag recognizer: An add-in that can interpret a specific type of smart tag, such as an address or a financial symbol, in a document and display an action button that enables users to perform common tasks for that data type.

South Asian language: A spoken or written communication consisting of words that are used within the grammatical and syntactic structure of a language of southern Asia, such as Hindi, Urdu, or Tamil.

structured document tag: An entity in a document that is used to denote content that is stored as **XML** data.

structured document tag bookmark: An entity in a document that is used to denote the location and presence of a **structured document tag**.

style: A set of formatting options that is applied to text, tables, charts, and other objects in a document.

subdocument: A document that can be referred to or inserted into another document. Subdocuments can be referenced by master documents and other subdocuments.

Copyright © 2025 Microsoft Corporation

table depth: An indicator that specifies how tables are nested and how to display paragraphs within those tables. The depth is derived from values that are applied to paragraph marks, cell marks, or table-terminating paragraph marks. A paragraph that is not in a table has a table depth of "0" (zero); a nested table has a table depth of one greater than the cell that contains it.

table style: A set of formatting options, such as font, border formatting, and row banding, that are applied to a table. The regions of a table, such as the header row, header column, and data area, can be variously formatted.

Tatenakayoko: A text layout setting that displays a range of text perpendicular (horizontal) to the flow of other text (vertical).

toolbar: A row, column, or block of controls that represent tasks or commands within an application. A toolbar can be either a menu toolbar, which provides access to menu commands, or a basic toolbar, which contains buttons that provide shortcuts to tasks that are frequently accessed from menus.

toolbar control: An object that appears on a toolbar and enables user interaction or input, typically to initiate an action, display information, or set values.

toolbar control identifier (TCID): An integer that identifies a specific control on a toolbar.

toolbar delta: A file component that stores a modification that a user made to a built-in toolbar. Stored modifications include adding, changing, or removing a control from a built-in toolbar.

TrueType font: A type of computer font that can be scaled to any size. TrueType fonts are clear and readable in all sizes and can be sent to any printer or other output device.

twip: A unit of measurement that is used in typesetting and desktop publishing. It equals one-twentieth of a printer's point, or 1/1440 of an inch.

Unicode: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The **Unicode** standard [UNICODE5.0.0/2007] provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

Universal Input Method (UIM): An application or service that provides multilingual support and delivers text services such as keyboard processors, handwriting recognition, and speech recognition

Vector Markup Language (VML): A system of marking up or tagging two-dimensional vector graphics for publication on the World Wide Web. VML graphics are scalable and editable, and typically require less disk space and less time to download.

vertical band: A set of columns in a table that are treated as a single unit, typically for the purpose of layout and formatting consistency.

virtual key code: A symbolic constant name, hexadecimal value, or mouse or keyboard equivalent that provides a hardware- and language-independent method of identifying keyboard keys. Each virtual key code represents a unique keyboard key and also identifies the purpose of that key. The keyboard driver provides one or more keyboard layouts that maps keyboard scan codes to the appropriate virtual key codes.

Visual Basic for Applications (VBA): A macro-based programming language that derives from Microsoft Visual Basic and can be used to customize and extend an application. Unlike Visual

Copyright © 2025 Microsoft Corporation

Basic, Microsoft Visual Basic for Applications (VBA) code and macros can be run only from within a host application that supports VBA.

Warichu: A text layout setting that creates two sublines within a line and stacks text equally between those sublines. One subline contains the text proper and the other subline contains comments, notes, and annotations about that text.

Web Layout view: A view of a document as it might appear in a web browser. For example, the document appears as only one page, without page breaks.

word wrap: The process of breaking lines of text automatically to stay within the page margins of a document or window boundaries.

Word97 compatibility mode: An application mode that prevents users from applying formatting and other document features and settings that are not supported in Microsoft Word 97 or earlier versions of Word.

write-reservation password: A sequence of characters that need to be entered to modify a document.

XML: The Extensible Markup Language, as described in [XML1.0].

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring XML schemas.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[ECMA-376] ECMA International, "Office Open XML File Formats", https://www.ecma-international.org/publications-and-standards/standards/ecma-376/

[Embed-Open-Type-Format] Nelson, P., "Embedded OpenType (EOT) File Format", W3C Member Submission, March 2008, http://www.w3.org/Submission/2008/SUBM-EOT-20080305/

[MC-CPB] Microsoft Corporation, "Code Page Bitfields", https://learn.microsoft.com/en-us/windows/win32/intl/code-page-bitfields

[MC-FONTSIGNATURE] Microsoft Corporation, "FONTSIGNATURE", http://msdn.microsoft.com/en-us/library/dd318064.aspx

[MC-USB] Microsoft Corporation, "Unicode Subset Bitfields", http://msdn.microsoft.com/en-us/library/ms776439.aspx

Copyright © 2025 Microsoft Corporation

[MS-CFB] Microsoft Corporation, "Compound File Binary File Format".

[MS-CTDOC] Microsoft Corporation, "Word Custom Toolbar Binary File Format".

[MS-DOCX] Microsoft Corporation, "Word Extensions to the Office Open XML (.docx) File Format".

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-EMF] Microsoft Corporation, "Enhanced Metafile Format".

[MS-LCID] Microsoft Corporation, "Windows Language Code Identifier (LCID) Reference".

[MS-ODRAW] Microsoft Corporation, "Office Drawing Binary File Format".

[MS-OE376] Microsoft Corporation, "Office Implementation Information for ECMA-376 Standards Support".

[MS-OFFCRYPTO] Microsoft Corporation, "Office Document Cryptography Structure".

[MS-OLEPS] Microsoft Corporation, "Object Linking and Embedding (OLE) Property Set Data Structures".

[MS-OSHARED] Microsoft Corporation, "Office Common Data Types and Objects Structures".

[MS-OVBA] Microsoft Corporation, "Office VBA File Format Structure".

[MS-WMF] Microsoft Corporation, "Windows Metafile Format".

[PANOSE] Hewlett-Packard Corporation, "PANOSE Classification Metrics Guide", February 1997, http://www.panose.com

[RFC1950] Deutsch, P., and Gailly, J-L., "ZLIB Compressed Data Format Specification version 3.3", RFC 1950, May 1996, http://www.ietf.org/rfc/rfc1950.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, https://www.rfc-editor.org/info/rfc2119

[RFC2822] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001, https://www.rfc-editor.org/info/rfc2822

[RFC4234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, https://www.rfc-editor.org/info/rfc4234

1.2.2 Informative References

[MS-OLEDS] Microsoft Corporation, "Object Linking and Embedding (OLE) Data Structures".

[MSDN-FONTS] Microsoft Corporation, "About Fonts", http://msdn.microsoft.com/en-us/library/dd162470(VS.85).aspx

[MSDOCS-IViewObject] Microsoft Corporation, "IViewObject interface", https://learn.microsoft.com/en-us/windows/win32/api/oleidl/nn-oleidl-iviewobject

Copyright © 2025 Microsoft Corporation

1.3 Overview

1.3.1 Characters

The fundamental unit of a Word binary file is a character. This includes visual characters such as letters, numbers, and punctuation. It also includes formatting characters such as **paragraph marks**, **end of cell marks**, line breaks, or **section breaks**. Finally, it includes **anchor** characters such as footnote reference characters, picture anchors, and comment anchors.

Characters are indexed by their zero-based **Character Position**, or **CP** (section 2.2.1). This documentation is generally concerned with **CPs** (section 2.2.1), not with the underlying text. Section 2.4.1 specifies an algorithm for determining the text at a particular **CP** (section 2.2.1), but this is just one of many pieces of information an application might look for. The reader needs to understand that this documentation is much more about logical characters in a document than about physical bytes in a file.

1.3.2 PLCs

Many features of the Word Binary File Format pertain to a range of **CPs** (section 2.2.1). For example, a **bookmark** is a range of **CPs** (section 2.2.1) that is named by the document author. As another example, a field is made up of three control characters with ranges of arbitrary document content between them.

The Word Binary File Format uses a **PLC** structure (section 2.2.2) to specify these and other kinds of ranges of **CPs** (section 2.2.1). A **PLC** (section 2.2.2) is simply a mapping from **CPs** (section 2.2.1) to other, arbitrary data.

1.3.3 Formatting

The formatting of characters, paragraphs, sections, tables, and pictures is specified as a set of differences in formatting from the default formatting for these objects. Modifications to individual properties are expressed using a Prl. A Prl is a Single Property Modifier, or Sprm, and an operand that specifies the new value for the property. Each property has (at least) one unique Sprm that modifies it. For example, sprmCFBold modifies the bold formatting of text, and sprmPDxaLeft modifies the **logical left** indent of a paragraph.

The final set of properties for text, paragraphs, and tables comes from a hierarchy of styles and from Prl elements applied directly (for example, by the user selecting some text and clicking the Bold button in the user interface). Styles allow complex sets of properties to be specified in a compact way. They also allow the user to change the appearance of a document without visiting every place in the document where a change is necessary. The style sheet for a document is specified by a STSH, as defined in section 2.9.271.

See section 2.4.6.6 for the algorithm that determines the complete set of formatting for a character, paragraph, table, or picture.

See section 2.8.26 for the structure used to determine the boundaries of sections and the location of their properties.

See section 2.6 for the complete list of Sprms.

1.3.4 Tables

A table consists of a set of paragraphs that has a particular set of properties applied. There are special characters that denote the ends of table **cells** and the ends of table rows, but there are no characters to denote the beginning of a table cell or the end of the table as a whole. Tables can be nested inside other tables.

Copyright © 2025 Microsoft Corporation

Section 2.4.3 provides an overview of tables, and Sections 2.4.4 and 2.4.5 specify algorithms for determining the boundaries of a table cell and table row, respectively.

1.3.5 Pictures

Pictures in the Word Binary File format can be either inline or floating. An inline picture is represented by a character whose **Unicode** value is 0x0001 and has sprmCFSpec applied with a value of 1 and sprmCPicLocation applied to specify the location of the picture data. A floating picture is represented by an **anchor** character with a Unicode value of 0x0008 with sprmCFSpec applied with a value of 1. In addition, floating pictures are referenced by a PlcfSpa structure which contains additional data about the picture. A floating picture can appear anywhere on the same page as its anchor. The document author can choose to have the floating picture rearrange the text in various ways or to leave the text as is.

1.3.6 The FIB

The main stream of the Word Binary File Format begins with a File Information Block, or FIB. The FIB specifies the locations of all other data in the file. The locations are specified by a pair of integers, the first of which specifies the location and the second of which specifies the size. These integers appear in substructures of the FIB such as the FibRgFcLcb97. The location names are prefixed with **fc**; the size names are prefixed with **lcb**.

1.3.7 Byte Ordering

Some computer architectures number bytes in a binary word from left to right, which is referred to as **big-endian**. The bit diagram for this documentation is big-endian. Other architectures number the bytes in a binary word from right to left, which is referred to as **little-endian**. The underlying file format enumerations, objects, and records are little-endian.

Using big-endian and little-endian methods, the number 0x12345678 would be stored as shown in the following table.

Byte order	Byte 0	Byte 1	Byte 2	Byte 3
Big-endian	0x12	0x34	0x56	0x78
Little-endian	0x78	0x56	0x34	0x12

Unless otherwise specified, all data in the Word Binary File Format is stored in little-endian format.

1.3.8 General Organization of This Documentation

Section 2 of this documentation is arranged with high-level overviews followed by detailed specifications.

Sections 2.1 through 2.4 provide general specifications of structures and concepts that recur in this documentation. Read these sections from beginning to end before delving deeper into section 2. The most important part of this documentation is section 2.4, which specifies algorithms for retrieving document content and determining its properties.

Section 2.5 provides a complete specification of the FIB, including links to all referenced data structures.

Section 2.6 provides a complete list of Sprm elements and their operands; it can be considered a complete list of the character, paragraph, table, and section properties supported by the Word Binary File Format. Note that most picture properties are not represented by Sprm elements. [MS-ODRAW]

Copyright © 2025 Microsoft Corporation

specifies most picture properties. Each Sprm definition specifies the default value for the property that it modifies.

Section 2.7 provides a specification of document-level properties

Section 2.8 provides a complete specification of all PLC types. Finally, section 2.9 provides specifications for data types referenced by previous sections. Sections 2.8 and 2.9 are intended to be read as the destination of links from other sections; they are not intended to be read straight through.

Section 3 provides examples that relate to the algorithms in section 2.4 and examples of bookmarks (1) and sections. These examples are intended to illustrate the concept of property storage, PLCs, and numbering, and to demonstrate the mapping between **CP** (section 2.2.1) and underlying text (as specified in section 2.4.1).

Section 4 discusses encryption, obfuscation, and other security issues relating to the Word Binary File Format.

Section 5 is a list of version-specific behaviors. It is intended to be read in the context of specifications in section 2, not as a stand-alone section. Specifications in section 2 provide links to the relevant items in section 5.

1.4 Relationship to Protocols and Other Structures

The Word Binary File Format is an **OLE compound file** as specified in [MS-CFB]. It is dependent on the structures defined in the following references:

- [MS-ODRAW] for the persistence format for shapes.
- [MS-OVBA] for the persistence format for macros.
- [MS-OFFCRYPTO] for the persistence format for document signing, information rights management, document encryption, and obfuscation.
- [MS-OSHARED] for the persistence format for additional common structures.

The Word Binary File Format is superseded by [ECMA-376].

1.5 Applicability Statement

This document specifies a persistence format for word processing document content and templates, which can include text, images, tables, custom XML schemas applied to the content, and page layout information. This persistence format is applicable when the document content is intended to flow across a set of pages as necessary for a particular media, and when the document can be printed. This persistence format is not applicable when exact reproduction of a specific representation of the content across various media and devices is desired.

This persistence format is applicable for use as a stand-alone document, and for containment within other documents as an embedded object as specified by [MS-OLEDS].

This persistence format provides interoperability with applications that create or read documents conforming to this structure.

1.6 Versioning and Localization

This document covers versioning issues in the following areas:

Structure Versions: There is only one version of the Word Binary File Format structure.

Localization: This structure defines no general locale-specific processes or data. Locale-specific variations for specific field values within the structure are specified in the definition of the affected field in Section 2.

Copyright © 2025 Microsoft Corporation

1.7 Vendor-Extensible Fields

This persistence format can be extended by storing information in streams and storages that are not specified in section 2. Implementations are not required to preserve or remove additional streams or storages when modifying an existing document.

Copyright © 2025 Microsoft Corporation

2 Structures

2.1 File Structure

A Word Binary File is an **OLE compound file** as specified by [MS-CFB]. The file consists of the following storages and streams.

2.1.1 WordDocument Stream

The WordDocument stream MUST be present in the file and MUST have an **FIB** at offset 0. It also contains the document text and other information referenced from other parts of the file. The stream has no predefined structure other than the **FIB** at the beginning.

In the context of Word Binary Files, the delay stream that is referenced in [MS-ODRAW] is the WordDocument stream.

The WordDocument stream MUST NOT be larger than 0x7FFFFFFF bytes.

2.1.2 1Table Stream or 0Table Stream

Either the 1Table stream or the 0Table stream MUST be present in the file. If the **FIB** at offset 0 in the WordDocument stream has **base.fWhichTblStm** set to 1, this stream is called 1Table. Otherwise, it is called 0Table.

If the document is encrypted as specified in section 2.2.6, this stream MUST have an **EncryptionHeader** at offset 0, as specified in section 2.2.6. If the document is not encrypted, this stream has no predefined structure. Other than the possible **EncryptionHeader**, this stream contains the data that is referenced from the **FIB** or from other parts of the file.

This documentation refers to this stream as the *Table Stream*.

If a file contains both a 1Table and a 0Table stream, only the stream that is referenced by **base.fWhichTblStm** is used. The unreferenced stream MUST be ignored.

The Table Stream MUST NOT be larger than 0x7FFFFFFF bytes.

2.1.3 Data Stream

The Data stream has no predefined structure. It contains data that is referenced from the **FIB** or from other parts of the file. This stream need not be present if there are no references to it.

The Data stream MUST NOT be larger than 0x7FFFFFFF bytes.

2.1.4 ObjectPool Storage

The Object Pool storage contains storages for embedded **OLE objects**. This storage need not be present if there are no embedded OLE objects in the document.

The names of these storages are formed by locating the field of an OLE object by field type (2.9.90 flt) or CP (character position), following (2.4.6.2 Direct Character Formatting) to retrieve the sprmCPicLocation property (2.6.1 Character Properties) for the field separator character (0x14) (2.8.25 Plcfld), converting that value to a string and prepending an underscore "_".

Copyright © 2025 Microsoft Corporation

2.1.4.1 ObjInfo Stream

Each storage within the ObjectPool storage contains a stream whose name is "\003ObjInfo" where \003 is the character with value 0x0003, not the string literal "\003". This stream contains an **ODT** structure which specifies information about that embedded **OLE object**.

2.1.4.2 Print Stream

Each storage within the ObjectPool storage optionally contains a stream whose name is "\003PRINT" where \003 is the character with value 0x0003, not the string literal "\003". This stream contains an MFPF followed immediately by a Metafile as specified in [MS-WMF]. If no PRINT or EPRINT stream is present, then the object does not have a print presentation distinct from its screen presentation.

2.1.4.3 EPrint Stream

Each storage within the ObjectPool storage optionally contains a stream whose name is "\003EPRINT" where \003 is the character with value 0x0003, not the string literal "\003".<1> This stream contains an Enhanced Metafile, as specified in [MS-EMF], to be used when printing the object. If no EPRINT or PRINT stream is present, then the object does not have a print presentation distinct from its screen presentation.

2.1.5 Custom XML Data Storage

The Custom XML Data storage is an optional storage whose name MUST be "MsoDataStore".

The contents of the storage are specified in [MS-OSHARED] section 2.3.6.

2.1.6 Summary Information Stream

The Summary Information stream is an optional stream whose name MUST be "\005SummaryInformation", where \005 is the character with value 0x0005, and not the string literal "\005".

The contents of this stream are specified in [MS-OSHARED] section 2.3.3.2.1.

2.1.7 Document Summary Information Stream

The Document Summary Information stream is an optional stream whose name MUST be "\005DocumentSummaryInformation", where \005 is the character with value 0x0005, not the string literal "\005".

The contents of this stream are specified in [MS-OSHARED] section 2.3.3.2.2.

2.1.8 Encryption Stream

The Encryption stream is an optional stream whose name MUST be "encryption". The format of this stream is specified in section 2.2.6.3. This stream MUST NOT be present unless both of the following conditions are met:

- The document is encrypted with Office Binary Document RC4 CryptoAPI Encryption (section 2.2.6.3).
- The **fDocProps** value is set in the **EncryptionHeader**.**Flags**.

Copyright © 2025 Microsoft Corporation

2.1.9 Macros Storage

The Macros storage is an optional storage that contains the macros for the file. If present, it MUST be a Project Root Storage as defined in [MS-OVBA] section 2.2.1.

2.1.10 XML Signatures Storage

The **XML** signatures storage is an optional storage whose name MUST be "_xmlsignatures". This storage contains **digital signatures** as specified in [MS-OFFCRYPTO] section 2.5.2.4. This storage MAY<2> be ignored.

2.1.11 Signatures Stream

The signatures stream is an optional stream whose name MUST be "_signatures". This stream contains **digital signatures** as specified in [MS-OFFCRYPTO] section 2.5.1. This stream MAY<3> be ignored.

2.1.12 Information Rights Management Data Space Storage

The Information Rights Management Data Space storage is an optional storage whose name MUST be "\006DataSpaces", where \006 is the character with value 0x0006, and not the string literal "\006". This storage is specified in [MS-OFFCRYPTO] section 2.2.

If this storage is present, the Protected Content Stream MUST also be present.

If this storage is present, all specified streams and storages other than this storage and the Protected Content Stream SHOULD<4> be read from the Protected Content Stream as specified in [MS-OFFCRYPTO] section 1.3.2 and if any of those streams and storages exist outside of the Protected Content Stream, they SHOULD<5> be ignored.

2.1.13 Protected Content Stream

The Protected Content Stream is an optional stream whose name MUST be "\009DRMContent", where \009 is the character with value 0x0009, and not the string literal "\009". This storage is specified in [MS-OFFCRYPTO] section 2.2.10.

If this stream is present, the Information Rights Management Data Space Storage MUST also be present.

2.2 Fundamental Concepts

2.2.1 Character Position (CP)

A character position, which is also known as a CP, is an unsigned 32-bit integer that serves as the zero-based index of a character in the document text. There is no requirement that the text at consecutive character positions be at adjacent locations in the file. The size of each character in the file also varies. The location and size of each character in the file can be computed using the algorithm in section 2.4.1 (Retrieving Text).

Characters include the text of the document, anchors for objects such as footnotes or textboxes, and control characters such as paragraph marks and table cell marks.

Unless otherwise specified by a particular usage, a CP MUST be greater than or equal to zero and less than 0x7FFFFFFF. The range of valid character positions in a particular document is given by the algorithm in section 2.4.1 (Retrieving Text).

Copyright © 2025 Microsoft Corporation

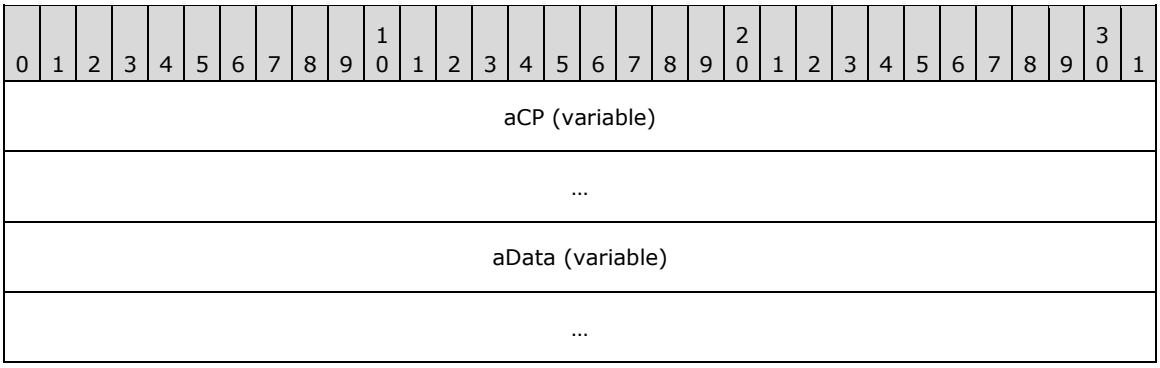
2.2.2 PLC

The **PLC** structure is an array of character positions followed by an array of data elements. The data elements for any **PLC** MUST be the same size of zero or more bytes. The number of CPs MUST be one more than the number of data elements. The CPs MUST appear in ascending order. There are different types of **PLC** structures, as specified in section 2.8. Each type specifies whether duplicate CPs are allowed for that type.

If the total size of a **PLC** (cbPlc) and the size of a single data element (cbData) are known, the number of data elements in that **PLC** (*n*) is given by the following expression:

$$n = \frac{\text{cbPlc} - 4}{4 + \text{cbData}}$$

The preceding expression MUST yield a whole number for *n*.



aCP (variable length): An array of CP elements. Each type of **PLC** structure specifies the meaning of the CP elements and the allowed range.

aData (variable length): Each type of **PLC** structure specifies the structure and meaning of the data elements, any restrictions on the number of data elements, and any restrictions on the data contained therein. It also specifies the relationship between the data elements and the corresponding CPs.

2.2.3 Valid Selection

Many constructs in file types described by this document refer to ranges of CPs. When such ranges specify that they are restricted to a valid selection, the following rules apply.

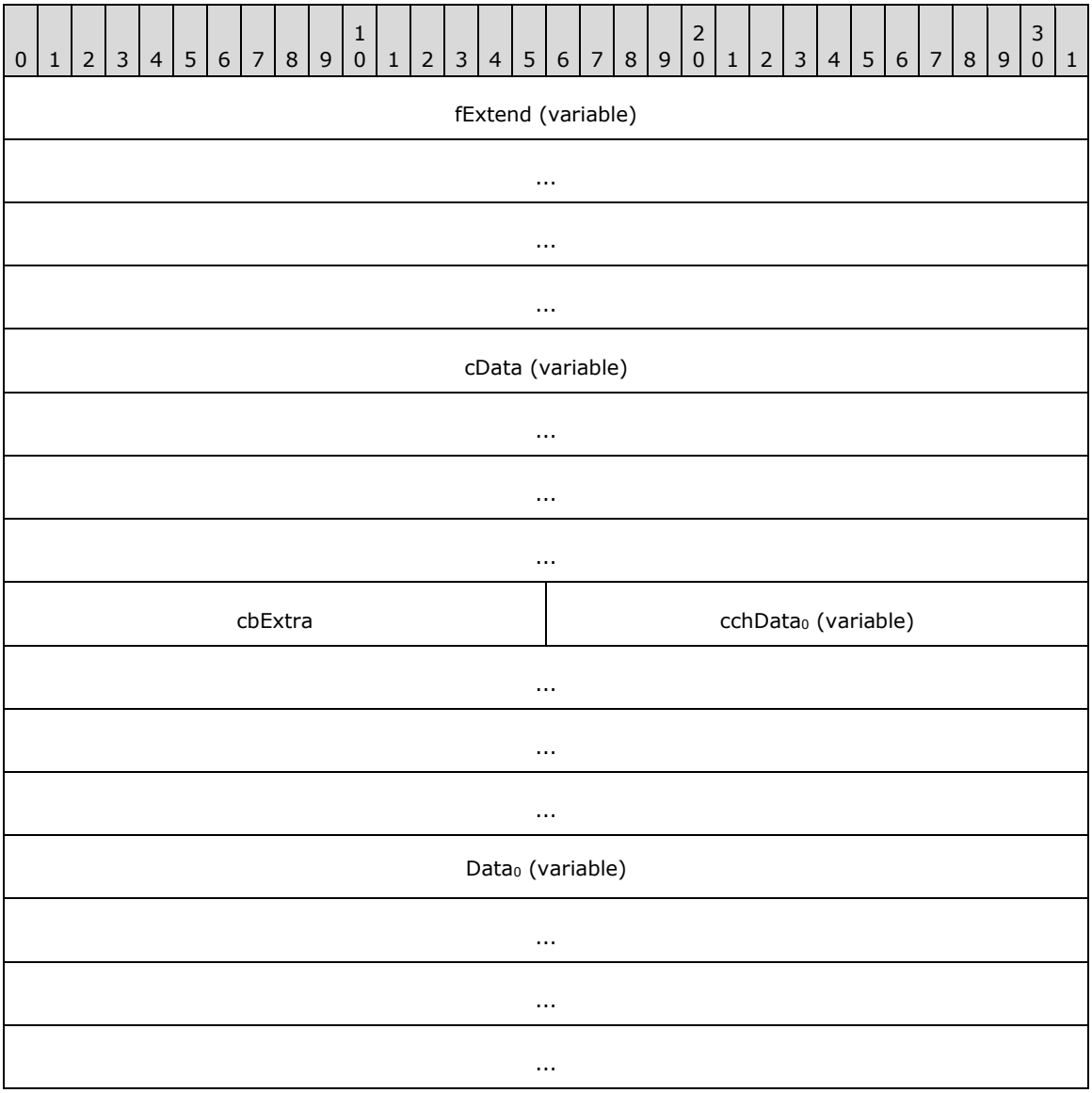
- If the range contains content from more than one table cell at a particular table depth, then it MUST contain only whole table rows at that table depth. For further specification, see Overview of Tables (section 2.4.3).
- If the range contains a field begin character, field separator character, or field end character, then it MUST contain the entire field. For further specification, see **Plcfld** (section 2.8.25).
- Both ends of the range MUST be in the same document part.
- If the range is in the footnote document, then both ends MUST be in the same footnote. For further specification, see **PlcffndTxt** (section 2.8.20).
- If the range is in the header document, then both ends MUST be in the same header or footer. For further specification, see **Plcfhdd** (section 2.8.22).

Copyright © 2025 Microsoft Corporation

- If the range is in the comment document, both ends MUST be in the same comment. For further specification, see **PlcfandTxt** (section 2.8.8).
- If the range is in the endnote document, then both ends MUST be in the same end note. For further specification, see **PlcfendTxt** (section 2.8.17).
- If the range is in the textbox document, then both ends MUST be in the same textbox. For further specification, see **PlcftxbxTxt** (section 2.8.32).
- If the range is in the header textbox document, then both ends MUST be in the same textbox. For further specification, see **PlcfHdrtxbxTxt** (section 2.8.23).

2.2.4 STTB

The **STTB** is a string table that is made up of a header that is followed by an array of elements. The **cData** value specifies the number of elements that are contained in the array.



Copyright © 2025 Microsoft Corporation

$ExtraData_0$ (variable)
...
...
...
$cchData_1$ (variable)
...
...
...
$Data_1$ (variable)
...
...
...
$ExtraData_1$ (variable)
...
...
...
...
$cchData_{cData-1}$ (variable)
...
...
...
$Data_{cData-1}$ (variable)
...
...
...

Copyright © 2025 Microsoft Corporation

ExtraData$_{cData-1}$ (variable)
...
...
...

The header consists of the following.

fExtend (variable): Optional. If the first two bytes of the **STTB** are equal to 0xFFFF, this is a 2-byte **fExtend** field that specifies, by its existence, that the **Data** fields in this **STTB** contain extended (2-byte) characters and that the **cchData** fields are 2 bytes in size. If the first two bytes of the **STTB** are not equal to 0xFFFF, this **fExtend** field does not exist, which specifies, by its nonexistence, that the **Data** fields in this **STTB** contain nonextended (1-byte) characters and that the **cchData** fields are 1 byte in size.

cData (variable): A 2-byte unsigned integer or a 4-byte signed integer that specifies the count of elements in this **STTB**. If this is a 2-byte unsigned integer, it MUST be less than 0xFFFF. If this is a 4-byte signed integer, it MUST be greater than zero. Unless otherwise specified, this is a 2-byte unsigned integer.

cbExtra (2 bytes): An unsigned integer that specifies the size, in bytes, of the **ExtraData** fields in this **STTB**.

The array of elements consists of the following.

cchData (variable): An unsigned integer that specifies the count of characters in the **Data** field following this field. If this **STTB** is using extended characters as defined by **fExtend**, the size of **cchData** is 2 bytes. If this **STTB** is not using extended characters, the size of **cchData** is 1 byte.

Data (variable): The definition of each **STTB** specifies the meaning of this field. If this **STTB** uses extended characters, the size of this field is 2×**cchData** bytes and it is a **Unicode** string unless otherwise specified by the **STTB** definition. If this **STTB** does not use extended characters, then the size of this field is **cchData** bytes and it is an ANSI string, unless otherwise specified by the **STTB** definition.

ExtraData (variable): The definition of each **STTB** specifies the structure and meaning of this field. The size of this field is **cbExtra** bytes.

2.2.5 Property Storage

Files in Word Binary File Format store the properties of characters, paragraphs, tables, pictures, and sections as lists of differences from the default. A **Prl** specifies each difference. It consists of a Single Property Modifier (**Sprm**) and its operand. An application can determine the final set of properties by applying lists of **Prl**s in the order that is specified in section 2.4.6 (Applying Properties).

An application SHOULD<6> skip any **Prl** that corresponds to a property or feature not present in the application by using **Sprm.spra** to determine the size of the **Prl** to skip.

The definition of each **Sprm** in section 2.6 specifies the default value for the corresponding property.

If multiple **Prl**s modify the same property, the last one that is applied determines the final value of that property unless otherwise specified in a **Sprm** definition in section 2.6.

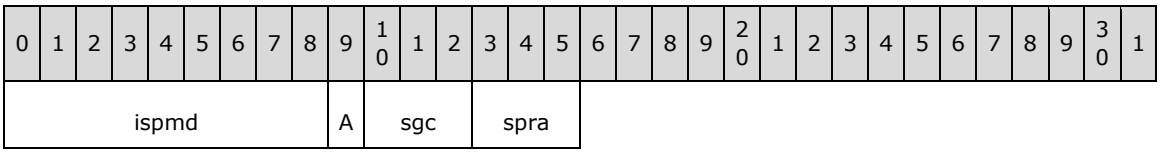
Any restrictions on the ordering of **Prl**s are included in the specifications of the individual **Sprm**s involved in the restriction. See sprmTDelete as an example.

Copyright © 2025 Microsoft Corporation

In cases where multiple **Sprm**s modify the same property, but are supported by different application versions, an application generating a file MUST first emit the **Sprm** that has the lower **ispmd**, followed by the **Sprm** that has the higher **ispmd**. For example, sprmPBrcTop80 and sprmPBrcTop both modify the top border of a paragraph, but sprmPBrcTop can express more colors. If an application emits only sprmPBrcTop, applications that support only sprmPBrcTop80 do not display a top border.

2.2.5.1 Sprm

The **Sprm** structure specifies a modification to a property of a character, paragraph, table, or section.



ispmd (9 bits): An unsigned integer that, when combined with **fSpec**, specifies the property being modified. See the tables in the Single Property Modifiers section (2.6) for the complete list of valid **ispmd**, **fSpec**, **spra** combinations for each **sgc**.

A - fSpec (1 bit): When combined with **ispmd**, specifies the property being modified. See the tables in the Single Property Modifiers section (2.6) for the complete list of valid **ispmd**, **fSpec**, **spra** combinations for each **sgc**.

sgc (3 bits): An unsigned integer that specifies the kind of document content to which this **Sprm** applies. The following table specifies the valid values and their meanings.

Sgc	Meaning
1	**Sprm** is modifying a paragraph property.
2	**Sprm** is modifying a character property.
3	**Sprm** is modifying a picture property.
4	**Sprm** is modifying a section property.
5	**Sprm** is modifying a table property.

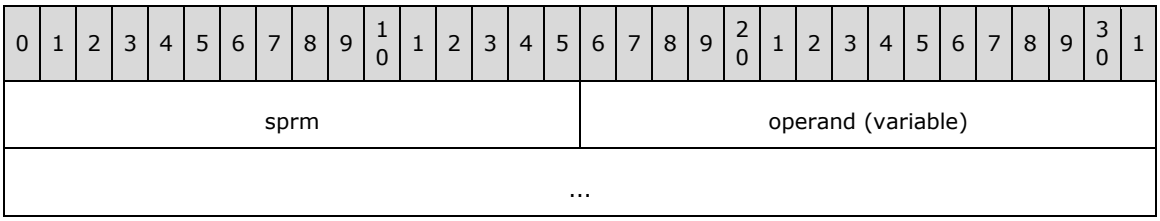
Copyright © 2025 Microsoft Corporation

spra (3 bits): An unsigned integer that specifies the size of the operand of this **Sprm**. The following table specifies the valid values and their meanings.

Spra	Meaning
0	Operand is a ToggleOperand (which is 1 byte in size).
1	Operand is 1 byte.
2	Operand is 2 bytes.
3	Operand is 4 bytes.
4	Operand is 2 bytes.
5	Operand is 2 bytes.
6	Operand is of variable length. The first byte of the operand indicates the size of the rest of the operand, except in the cases of sprmTDefTable and sprmPChgTabs.
7	Operand is 3 bytes.

2.2.5.2 Prl

The **Prl** structure is a **Sprm** that is followed by an operand. The **Sprm** specifies a property to modify, and the operand specifies the new value.



sprm (2 bytes): A **Sprm** which specifies the property to be modified.

operand (variable): A variable-length operand for the **sprm**. The size of the operand is specified by **sprm.spra**. The meaning of the operand depends on the **sprm**, see section 2.6 (Single Property Modifiers).

2.2.6 Encryption and Obfuscation (Password to Open)

A file in Word Binary File Format can be password protected by using one of the following mechanisms:

- XOR obfuscation (section 2.2.6.1)
- Office binary document RC4 encryption (section 2.2.6.2)
- Office binary document RC4 CryptoAPI encryption<7> (section 2.2.6.3)

If **FibBase.fEncrypted** and **FibBase.fObfuscated** are both 1, the file is obfuscated by using XOR obfuscation (section 2.2.6.1) as specified in section 2.2.6.1.

If **FibBase.fEncrypted** is 1 and **FibBase.fObfuscated** is 0, the file is encrypted by using either Office Binary Document RC4 Encryption (section 2.2.6.2) or Office Binary Document RC4 CryptoAPI Encryption (section 2.2.6.3), with the **EncryptionHeader** stored in the first **FibBase.lKey** bytes of the Table stream. The **EncryptionHeader.EncryptionVersionInfo** specifies which encryption mechanism was used to encrypt the file.

See Security Considerations for information about security concerns relating to file obfuscation and encryption for this file format.

Copyright © 2025 Microsoft Corporation

2.2.6.1 XOR Obfuscation

In a file that is password protected by using XOR obfuscation, **FibBase.fEncrypted** and **FibBase.fObfuscated** MUST both be 1.

The password verifier computed from the password as specified in Binary Document Password Verifier Derivation Method 2 in [MS-OFFCRYPTO] section 2.3.7.4 MUST be stored in FibBase.**lKey**.

The WordDocument stream, the Table stream, and the Data stream MUST be obfuscated using XOR Data Transformation Method 2 as specified in [MS-OFFCRYPTO] section 2.3.7.6. All other streams and storages MUST NOT be obfuscated.

The byte transformation specified in [MS-OFFCRYPTO] section 2.3.7.6 MUST be carried out in the WordDocument stream relative to the beginning of the stream, but the initial 68 bytes MUST be written out with their untransformed values.

2.2.6.2 Office Binary Document RC4 Encryption

In a file that is password protected by using Office binary document RC4 encryption as specified in [MS-OFFCRYPTO] section 2.3.6, **FibBase.fEncrypted** MUST be 1 and **FibBase.fObfuscated** MUST be 0.

The **EncryptionHeader**, as specified in [MS-OFFCRYPTO] section 2.3.6.1, MUST be written in unencrypted form in the first **FibBase.lKey** bytes of the Table stream. The remainder of the Table stream, the WordDocument stream beyond the initial 68 bytes, and the entire Data stream MUST be encrypted.

These three streams of data MUST be encrypted in 512-byte blocks. The block number MUST be set to zero at the beginning of the stream and MUST be incremented at each 512-byte boundary. The encryption algorithm MUST be carried out at the beginning of the Table stream and the WordDocument stream even though some of the bytes are written in unencrypted form.

All other streams and storages MUST NOT be encrypted.

2.2.6.3 Office Binary Document RC4 CryptoAPI Encryption

In a file that is password protected by using Office binary document RC4 CryptoAPI encryption as specified in [MS-OFFCRYPTO] section 2.3.5, **FibBase.fEncrypted** MUST be 1 and **FibBase.fObfuscated** MUST be 0.

The **EncryptionHeader** as specified in [MS-OFFCRYPTO] section 2.3.5.1 MUST be written in unencrypted form in the first **FibBase.lKey** bytes of the Table stream. The remainder of the Table stream, the WordDocument stream beyond the initial 68 bytes, and the entire Data stream MUST be encrypted.

These three streams of data MUST be encrypted in 512-byte blocks. The block number MUST be set to zero at the beginning of the stream and MUST be incremented at each 512 byte boundary. The encryption algorithm MUST be carried out at the beginning of the Table stream and the WordDocument stream even though some of the bytes are written in unencrypted form.

The ObjectPool storage MUST NOT be present and if the file contains **OLE objects**, the storage objects for the OLE objects MUST be stored in the Data stream as specified in sprmCPicLocation.

If **fDocProps** is set in the **EncryptionHeader**.**Flags**, the Encryption stream MUST be present, the Summary Information stream MUST NOT be present, and a placeholder Document Summary Information stream MUST be present as specified in [MS-OFFCRYPTO] section 2.3.5.4.

If **fDocProps** is not set in the **EncryptionHeader**.**Flags**, the Document Summary Information stream and the Summary Information stream MUST NOT be encrypted.

Copyright © 2025 Microsoft Corporation

All other streams and storages MUST NOT be encrypted<8>.

2.3 Document Parts

The range of CPs in a document is separated into multiple logical parts. Many features operate within the individual parts and use CPs relative to the beginning of the part in which they operate rather than relative to the beginning of the document. This section defines the document parts and specifies the corresponding range of CPs.

All documents MUST include a non-empty Main Document part. In addition, if any of the other document parts are non-empty, the document MUST include one additional **paragraph mark** character (**Unicode** 0x000D) beyond the end of the last non-empty document part. That character is not displayed to or editable by the user, because it is outside of any document part.

2.3.1 Main Document

The main document contains all content outside any of the specialized document parts, including **anchors** that specify where content from the other document parts appears.

The main document begins at CP zero, and is **FibRgLw97.ccpText** characters long.

The last character in the main document MUST be a **paragraph mark** (**Unicode** 0x000D).

2.3.2 Footnotes

The footnote document contains all of the content in the footnotes. It begins at the CP immediately following the Main Document, and is **FibRgLw97.ccpFtn** characters long.

The locations of individual footnotes within the footnote document are specified by a PlcffndTxt whose location is specified by the **fcPlcffndTxt** member of **FibRgFcLcb97**. The locations of the footnote reference characters in the Main Document are specified by a **PlcffndRef** whose location is specified by the **fcPlcffndRef** member of **FibRgFcLcb97**.

2.3.3 Headers

The header document contains all content in headers and footers as well as the footnote and endnote separators. It begins immediately after the footnote document and is **FibRgLw97.ccpHdd** characters long.

The header document is split into text ranges called stories, as specified by **PlcfHdd**. Each story specifies the contents of a single header, footer, or footnote/endnote separator. If a story is non-empty, it MUST end with a **paragraph mark** that serves as a guard between stories. This paragraph mark is not considered part of the story contents (that is, if the story contents require a paragraph mark themselves, a second paragraph mark MUST be used).

Stories are considered empty if they have no contents and no guard paragraph mark. Thus, an empty story is indicated by the beginning CP, as specified in **PlcfHdd**, being the same as the next CP in **PlcfHdd**.

If the header document exists, as indicated by **FibRgLw97.ccpHdd** and **FibRgFcLcb97.lcbPlcfHdd** being nonzero, its first six stories specify footnote and endnote separators, in this order.

Story number	Contents
0	**Footnote separator**
1	**Footnote continuation separator**

Copyright © 2025 Microsoft Corporation

Story number	Contents
2	**Footnote continuation notice**
3	**Endnote separator**
4	**Endnote continuation separator**
5	**Endnote continuation notice**

The footnote and endnote separator stories do not need to contain whole paragraphs—that is, they do not necessarily need to have paragraph marks in their contents. However, they MUST have the guard paragraph marks if they are non-empty.

Following the footnote and endnote separator stories are the stories that contain the contents of headers and footers. Six such stories MUST exist for every **section** of the Main Document. The first such group of stories specifies the contents of the headers and footers for the first section. The second group specifies the contents of the headers and footers for the second section, and so on. The stories within each group MUST appear in the following order.

Story number in group	Contents
0	Even page header. This MUST be non-empty if different even and odd headers and footers are enabled for the section.
1	Odd page header. If different even and odd headers and footers are not enabled for the section, the odd page header MUST be used on both even and odd pages.
2	Even page footer. This MUST be non-empty if different even and odd headers and footers are enabled for the section.
3	Odd page footer. If different even and odd headers and footers are not enabled for the section, the odd page footer MUST be used on both even and odd pages.
4	First page header. This MUST be non-empty if different first page headers and footers are enabled for the section.
5	First page footer. This MUST be non-empty if different first page headers and footers are enabled for the section.

Non-empty header and footer stories MUST contain whole paragraphs and thus MUST end with a paragraph mark. Therefore, non-empty header and footer stories MUST have two paragraph marks at their ends, one as part of the content followed by a separate guard paragraph mark.

An empty header or footer story specifies that the header or footer of the corresponding type of the previous section is used. For the first section, an empty header or footer story specifies that it does not have a header or footer of this type.

2.3.4 Comments

The comment document contains all of the content in the comments. It begins at the CP immediately following the Header Document and is **FibRgLw97.ccpAtn** characters long.

The locations of individual comments within the comment document are specified by a **PlcfandTxt** whose location is specified by the **fcPlcfandTxt** member of **FibRgFcLcb97**. The locations of the comment reference characters in the Main Document are specified by a **PlcfandRef** whose location is specified by the **fcPlcfandRef** member of **FibRgFcLcb97**.

Copyright © 2025 Microsoft Corporation

2.3.5 Endnotes

The endnote document contains all of the content in the endnotes. It begins at the CP that immediately follows the Comment Document and is **FibRgLw97.ccpEdn** characters long.

The locations of individual endnotes within the endnote document are specified by a PlcfendTxt whose location is specified by the **fcPlcfendTxt** member of **FibRgFcLcb97**. The locations of the endnote reference characters in the Main Document are specified by a PlcfendRef whose location is specified by the **fcPlcfendRef** member of **FibRgFcLcb97**.

2.3.6 Textboxes

The textbox document contains all of the content in the textboxes whose anchors are in the Main Document. It begins at the CP immediately following the Endnote Document and is **FibRgLw97.ccpTxbx** characters long.

The locations of individual textboxes within the textbox document are specified by a PlcftxbxTxt whose location is specified by the **fcPlcftxbxTxt** member of the **FibRgFcLcb97**. The locations of the textbox **anchors** in the Main Document are specified by a **plcfSpa** whose location is specified by the **fcPlcSpaMom** member of the **FibRgFcLcb97**.

Not all members of a **plcfSpa** specify the location of a textbox. The **lid** member of the FTXBXS structure specifies the relationship between shape anchors and textbox anchors.

2.3.7 Header Textboxes

The header textbox document contains all of the content in the textboxes whose anchors are in the Header Document. It begins at the CP immediately following the Textbox Document and is **FibRgLw97.ccpHdrTxbx** characters long.

The locations of individual textboxes within the header textbox document are specified by a **PlcfHdrtxbxTxt** whose location is specified by the **fcPlcfHdrtxbxTxt** member of the **FibRgFcLcb97**. The locations of the textbox **anchors** in the Header Document are specified by a **plcfSpa** whose location is specified by the **fcPlcSpaHdr** member of the **FibRgFcLcb97**.

Not all members of a **plcfSpa** specify the location of a textbox. The **lid** member of the **FTXBXS** structure specifies the relationship between shape anchors and textbox anchors.

2.4 Document Content

This section specifies algorithms that are used to analyze document content and determine its properties. These algorithms take CPs as input and return some piece of information about the document content at that location. For example, the algorithm in section 2.4.1 returns the text at that CP.

Collectively, these algorithms specify relationships among data structures in the file types that are specified in this documentation. These relationships MUST be maintained. These algorithms are not examples, but definitions of how to interpret these data structures.

These algorithms can derive significant performance benefits from common programming practices such as caching the results from previous input.

2.4.1 Retrieving Text

The following algorithm specifies how to find the text at a particular character position (**cp**). Negative character positions are not valid.

1. Read the **FIB** from offset zero in the WordDocument Stream.

Copyright © 2025 Microsoft Corporation

2. All versions of the **FIB** contain exactly one **FibRgFcLcb97**, though it can be nested in a larger structure. **FibRgFcLcb97.fcClx** specifies the offset in the Table Stream of a **Clx**. **FibRgFcLcb97.lcbClx** specifies the size, in bytes, of that **Clx**. Read the **Clx** from the Table Stream.

3. The **Clx** contains a **Pcdt**, and the **Pcdt** contains a **PlcPcd**. Find the largest *i* such that **PlcPcd.aCp[*i*]** ≤ **cp**. As with all **Plc**s, the elements of **PlcPcd.aCp** are sorted in ascending order. Recall from the definition of a **Plc** that the **aCp** array has one more element than the **aPcd** array. Thus, if the last element of **PlcPcd.aCp** is less than or equal to **cp**, **cp** is outside the range of valid character positions in this document.

4. **PlcPcd.aPcd[*i*]** is a **Pcd**. **Pcd.fc** is an **FcCompressed** that specifies the location in the WordDocument Stream of the text at character position **PlcPcd.aCp[*i*]**.

5. If **FcCompressed.fCompressed** is zero, the character at position **cp** is a 16-bit **Unicode** character at offset **FcCompressed.fc** + 2(**cp** - **PlcPcd**.**aCp[*i*]**) in the WordDocument Stream. This is to say that the text at character position **PlcPcd.aCP[*i*]** begins at offset **FcCompressed.fc** in the WordDocument Stream and each character occupies two bytes.

6. If **FcCompressed.fCompressed** is 1, the character at position **cp** is an 8-bit ANSI character at offset (**FcCompressed.fc** / 2) + (**cp** - **PlcPcd**.**aCp[*i*]**) in the WordDocument Stream, unless it is one of the special values in the table defined in the description of **FcCompressed.fc**. This is to say that the text at character position **PlcPcd.aCP[*i*]** begins at offset **FcCompressed.fc** / 2 in the WordDocument Stream and each character occupies one byte.

2.4.2 Determining Paragraph Boundaries

This section specifies how to find the beginning and end character positions of the paragraph that contains a given character position. The character at the end character position of a paragraph MUST be a **paragraph mark**, an end-of-section character, a cell mark, or a TTP mark (See Overview of Tables). Negative character positions are not valid.

To find the character position of the first character in the paragraph that contains a given character position **cp**:

1. Follow the algorithm from Retrieving Text up to and including step 3 to find *i*. Also remember the **FibRgFcLcb97** and **PlcPcd** found in step 1 of Retrieving Text. If the algorithm from Retrieving Text specifies that **cp** is invalid, leave the algorithm.

2. Let **pcd** be **PlcPcd.aPcd[*i*].**

3. Let **fcPcd** be **Pcd.fc.fc.** Let **fc** be **fcPcd** + 2(**cp** – **PlcPcd.aCp[*i*]**). If **Pcd.fc.fCompressed** is one, set **fc** to **fc** / 2, and set **fcPcd** to **fcPcd**/2.

4. Read a **PlcBtePapx** at offset **FibRgFcLcb97**.**fcPlcfBtePapx** in the Table Stream, and of size **FibRgFcLcb97**.**lcbPlcfBtePapx**. Let **fcLast** be the last element of **plcbtePapx.aFc**. If **fcLast** is less than or equal to **fc**, examine **fcPcd**. If **fcLast** is less than **fcPcd**, go to step 8. Otherwise, set **fc** to **fcLast**. If **Pcd.fc.fCompressed** is one, set **fcLast** to **fcLast** / 2. Set **fcFirst** to **fcLast** and go to step 7.

5. Find the largest *j* such that **plcbtePapx.aFc[*j*]** ≤ **fc**. Read a **PapxFkp** at offset **aPnBtePapx[*j*].pn** *512 in the WordDocument Stream.

6. Find the largest *k* such that **PapxFkp.rgfc[*k*]** ≤ **fc**. If the last element of **PapxFkp.rgfc** is less than or equal to **fc**, then **cp** is outside the range of character positions in this document, and is not valid. Let **fcFirst** be **PapxFkp.rgfc[*k*]**.

Copyright © 2025 Microsoft Corporation

7. If **fcFirst** is greater than **fcPcd**, then let **dfc** be **(fcFirst – fcPcd)**. If **Pcd**.**fc**.**fCompressed** is zero, then set **dfc** to **dfc** / 2. The first character of the paragraph is at character position **PlcPcd.aCp[*i*] + dfc**. Leave the algorithm.

8. If **PlcPcd.aCp[*i*]** is 0, then the first character of the paragraph is at character position 0. Leave the algorithm.

9. Set **cp** to **PlcPcd.aCp[*i*]**. Set *i* to *i* - 1. Go to step 2.

To find the character position of the last character in the paragraph that contains a given character position **cp**:

1. Follow the algorithm from Retrieving Text up to and including step 3 to find *i*. Also remember the **FibRgFcLcb97**, and **PlcPcd** found in step 1 of Retrieving Text. If the algorithm from Retrieving Text specifies that **cp** is invalid, leave the algorithm.

2. Let **pcd** be **PlcPcd.aPcd[*i*]**.

3. Let **fcPcd** be **Pcd.fc.fc**. Let **fc** be **fcPcd** + 2(**cp** – **PlcPcd.aCp[*i*]**). Let **fcMac** be **fcPcd** + 2(**PlcPcd.aCp**[*i*+1] - **PlcPcd.aCp[*i*]**). If **Pcd.fc.fCompressed** is one, set **fc** to **fc**/2, set **fcPcd** to **fcPcd** /2 and set **fcMac** to **fcMac**/2.

4. Read a **PlcBtePapx** at offset **FibRgFcLcb97**.**fcPlcfBtePapx** in the Table Stream, and of size **FibRgFcLcb97**.**lcbPlcfBtePapx**. Then find the largest *j* such that **plcbtePapx.aFc**[*j*] ≤ **fc**. If the last element of **plcbtePapx.aFc** is less than or equal to **fc**, then go to step 7. Read a **PapxFkp** at offset **aPnBtePapx**[*j*]**.pn** *512 in the WordDocument Stream.

5. Find largest *k* such that **PapxFkp.rgfc[*k*]** ≤ **fc**. If the last element of **PapxFkp.rgfc** is less than or equal to **fc**, then **cp** is outside the range of character positions in this document, and is not valid. Let **fcLim** be **PapxFkp.rgfc[*k*+*1*]**.

6. If **fcLim** ≤ **fcMac**, then let **dfc** be **(fcLim – fcPcd)**. If **Pcd**.**fc**.**fCompressed** is zero, then set **dfc** to **dfc** / 2. The last character of the paragraph is at character position **PlcPcd.aCp[*i*] + dfc –** 1. Leave the algorithm.

7. Set **cp** to **PlcPcd.aCp[***i***+1]**. Set *i* to *i* + 1. Go to step 2.

2.4.3 Overview of Tables

A table cell consists of one or more paragraphs at the same nonzero table depth and, optionally, one or more tables whose table depth is one greater than that of the containing cell. The last paragraph in a table cell is terminated by a cell mark. If the table depth is 1, the cell mark MUST be character **Unicode** 0x0007. If the table depth is greater than 1, the cell mark MUST be a **paragraph mark** (Unicode 0x000D) with sprmPFInnerTableCell applied with a value of 1.

A table row has between 1 and 63 table cells, each at the same table depth, followed by a Table Terminating Paragraph mark (TTP mark, also called a row mark), also at the same table depth. If the table depth is 1, then the TTP mark MUST be a character Unicode 0x0007 with sprmPFTtp applied with a value of 1. If the table depth is greater than 1, then the TTP mark MUST be a paragraph mark (Unicode 0x000D) with sprmPFInnerTtp applied with a value of 1.

The table depth of a paragraph, table cell, or table row, is derived from the values of sprmPFInTable, sprmPItap, and sprmPDtap applied as direct paragraph properties to the paragraph mark, cell mark, or TTP mark. See section 2.4.6.1, Direct Paragraph Formatting for further specifications. Paragraphs that are not in a table have a table depth of zero.

The following [ABNF] rulelist defines a table at depth *N* (TableN) in terms of paragraphs at depth *N* (ParaN), cell marks at depth *N* (CellMarkN), TTP marks at depth *N* (TTPN), and tables at depth *N*+1 (TableN1). ABNF is specified in [RFC4234].

Copyright © 2025 Microsoft Corporation

```
CellN    =  *(TableN1 / ParaN)  CellMarkN
RowN     =  1*63CellN           TTPN
TableN   =  1*RowN
```

Two adjacent table rows of the same table depth are considered part of the same table unless they differ in one of the following properties:

- The operand to sprmTIpgp
- The table style, as specified by sprmTIstd
- The table directionality as specified by section sprmTFBidi or section sprmTFBidi90
- The table position and wrapping as specified by sprmTPc, sprmTFNoAllowOverlap, sprmTDxaAbs, sprmTDyaAbs, sprmTDxaFromText, sprmTDyafromText, sprmTDxaFromTextRight, and sprmTDyaFromTextBottom

If neither table row specifies nondefault values for the preceding table position and wrapping properties, then two adjacent table rows of the same table depth are considered different tables if the first paragraphs of the first cells of the rows differ in any of the paragraph frame properties specified by sprmPPc, sprmPDxaAbs, sprmPDyaAbs, sprmPDxaWidth, sprmPWHeightAbs, sprmPDcs, sprmPWr, sprmPDxaFromText, sprmPDyaFromText, sprmPFLocked, sprmPFNoAllowOverlap, and sprmPFrameTextFlow.

In addition, two table rows are considered part of different tables if a **range-level protection bookmark** is present whose type, as specified by the **sdtt** member of the corresponding SDTI, is sdttPara and that bookmark (1) contains content from more than one table cell but does not contain the entirety of both rows.

The properties of each row mark MUST define the cells for that table row. SprmTDefTable and sprmTInsert are used to create cell definitions, and sprmTDelete is used to remove them. The number of cell definitions applied to the row mark MUST be equal to the number of cells in the row. There is no requirement that each row of a table have the same number of cells.

An application SHOULD<9> use sprmTDefTable to define table cells for applications that do not process sprmPTableProps, and at the same time use sprmTInsert for applications that do process sprmPTableProps.

The following diagram shows several elements of a table and gives examples of **Sprm**s that can be used to modify each. The table in this example includes spacing between cells to demonstrate borders and shading. It includes a nested table to demonstrate table depth.

Copyright © 2025 Microsoft Corporation

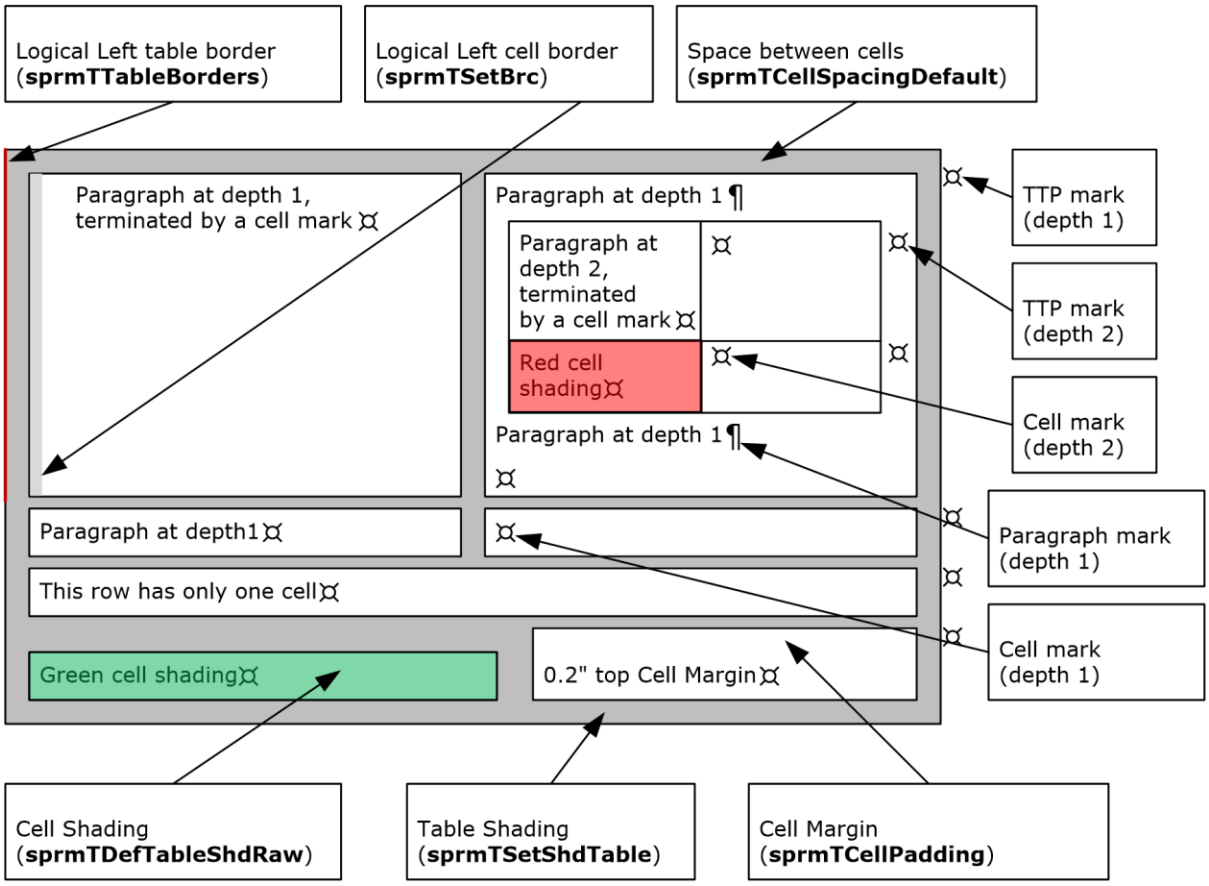


Figure 1: A sample table

To determine which borders are displayed, see the following sections from [ECMA-376] Part 1:

- Section 17.4.66 tcBorders (Table Cell Borders)
- Section 17.4.39 tblBorders (Table Border Exceptions)
- Section 17.4.38 tblBorders (Table Borders)

Cells can be vertically merged to create the appearance of a single cell spanning multiple rows. The cell mark characters for the merged cells MUST still appear in the file. The second and subsequent cells in the merged group MUST NOT contain any content other than their cell marks. The following diagram shows a table with vertically merged cells. It uses inside borders to demonstrate that the vertically merged cells act as one cell.

Copyright © 2025 Microsoft Corporation

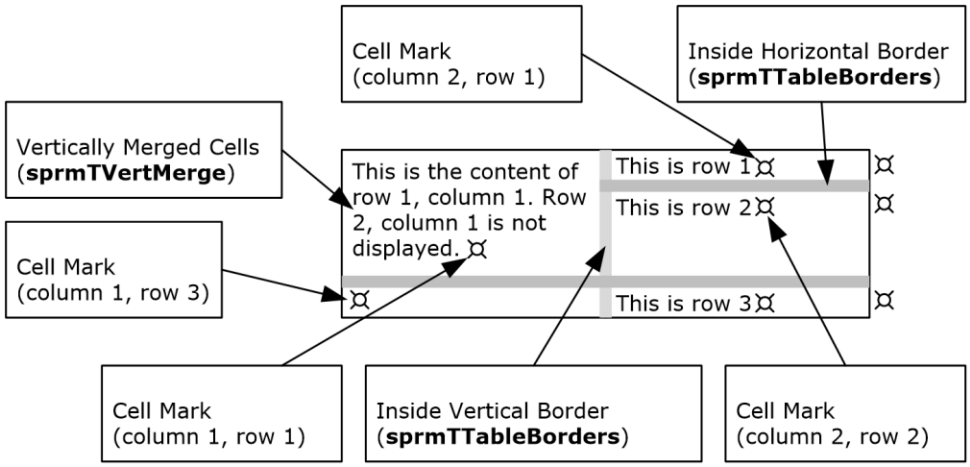


Figure 2: A table with vertically merged cells

2.4.4 Determining Cell Boundaries

This section describes an algorithm to find the boundaries of the innermost table cell containing a given character position or to determine that the given character position is not in a table cell. Every valid character position in a document belongs to a paragraph, so table depth can be computed for each paragraph. If a paragraph is found to be at depth zero, that paragraph is not in a table cell.

Given character position **cp**, use the following algorithm to determine if **cp** is in a table cell.

1. Follow the procedure from Direct Paragraph Formatting to find the paragraph properties for the paragraph that contains **cp**. Apply the properties, and determine the table depth as specified in Overview of Tables. Call this **itapOrig**.
2. If **itapOrig** is 0, then this paragraph is not in a table cell, so the following algorithms do not apply. Leave this algorithm. Otherwise, **cp** is in a table.
3. If the character at character position **cp** is not a TTP mark as specified in Overview of Tables, then leave this algorithm.
4. If **itapOrig** is 1, then the **cp** is not in a table cell. Leave this algorithm. Otherwise this TTP mark is in a cell itself, to determine the boundaries of the containing cell set **itapOrig** to **itapOrig** – 1 in the following algorithms.

Given a character position **cp** known to be at table depth **itapOrig**, follow this procedure to determine the character position of the last character in the innermost table cell that contains **cp**.

1. Set **itap** to **itapOrig**.
2. Determine the character position of the last character in the paragraph that contains **cp**, as specified in Determining Paragraph Boundaries. Let this position be called **cpLast**.
3. Follow the procedure from Direct Paragraph Formatting to find the paragraph properties for the paragraph that contains **cpLast**. Apply the properties, and determine the table depth as specified in Overview of Tables. Call this **itap'**. It is invalid for **itap'** to be less than **itap**. If **itap'** is less than **itap**, leave the algorithm.
4. If **itap'** is equal to **itap**, determine the text at character position **cpLast**, as specified in Retrieving Text. If this character is a cell mark, as specified in Overview of Tables, then **cpLast** is the desired output. Leave the algorithm.
5. Let **cp** be **cpLast** + 1, and go to step 2.

Given a character position **cp** that is known to be at table depth **itapOrig,** follow this procedure to determine the character position of the first character in the innermost table cell that contains **cp**.

1. Set **itap** to **itapOrig**.

Copyright © 2025 Microsoft Corporation

2. Determine the character position of the first character in the paragraph that contains **cp**, as specified in Determining Paragraph Boundaries. Let this character position be called **cpFirst**.
3. If **cpFirst** is zero, then this is the desired output. Leave the algorithm. Negative values for **cpFirst** are invalid. If **cpFirst** is negative, leave the algorithm.
4. Let **cpPrev** be **cpFirst** – 1. Follow the procedure from Direct Paragraph Formatting to find the paragraph properties for the paragraph that contains **cpPrev**. Apply the properties, and determine the table depth as specified in Overview of Tables. Call this **itapPrev**.
5. If **itapPrev** is less than **itap**, then **cpFirst** is the desired output. Leave the algorithm.
6. If **itapPrev** is equal to **itap**, determine the text at character position **cpPrev**, as specified in Retrieving Text. If this character is a cell mark or a TTP mark, then **cpFirst** is the desired output. Leave the algorithm.
7. Set **cp** to **cpPrev**. Go to step 2.

2.4.5 Determining Row Boundaries

This section describes an algorithm to find the boundaries of the innermost table row containing a given character position or to determine that the given character position is not in a table row. Every valid character position in a document belongs to a paragraph, so table depth can be computed for each paragraph. If a paragraph is found to be at depth zero, then that paragraph is not in a table row.

This algorithm is the same as Determining Cell Boundaries except that only TTP marks cause a termination, not cell marks.

Given character position **cp**, use the following algorithm to determine if **cp** is in a table.

1. Follow the procedure from Direct Paragraph Formatting to find the paragraph properties for the paragraph that contains **cp**. Apply the properties and determine the table depth as specified in Overview of Tables. Call this **itap**.
2. If **itap** is zero, then this paragraph is not in a table row. Leave the algorithm.

Given a character position **cp** known to be at table depth **itap**, which is greater than 0, follow this procedure to determine the character position of the TTP mark of the row that contains **cp**.

1. Determine the character position of the last character in the paragraph that contains **cp**, as specified in Determining Paragraph Boundaries. Let this position be called **cpLast**.
2. Follow the procedure from Direct Paragraph Formatting to find the paragraph properties for the paragraph that contains **cpLast**. Apply the properties and determine the table depth as specified in Overview of Tables. Call this **itap'**. It is invalid for **itap'** to be less than **itap**. If **itap'** is less than **itap**, leave the algorithm.
3. If **itap'** is equal to **itap**, determine the text at character position **cpLast**, as specified in Retrieving Text. If this character is a TTP mark as specified in Overview of Tables, then **cpLast** is the desired output. Leave the algorithm.
4. Let **cp** be **cpLast** + 1 and go to step 1.

Given a character position **cp** known to be at table depth **itap**, which is greater than 0, follow this procedure to determine the character position of the first character in the innermost table row that contains **cp**.

1. Determine the character position of the first character in the paragraph that contains **cp** as specified in Determining Paragraph Boundaries. Let this character position be called **cpFirst**.
2. If **cpFirst** is zero, then this is the desired output. Leave the algorithm. Negative values for **cpFirst** are invalid. If **cpFirst** is negative leave the algorithm.
3. Let **cpPrev** be **cpFirst** – 1. Follow the procedure from Direct Paragraph Formatting to find the paragraph properties for the paragraph that contains **cpPrev**. Apply the properties, and determine the table depth as specified in Overview of Tables. Call this **itapPrev**.
4. If **itapPrev** is less than **itap**, then **cpFirst** is the desired output. Leave the algorithm.
5. If **itapPrev** is equal to **itap**, determine the text at character position **cpPrev**, as specified in Retrieving Text. If this character is a TTP mark as specified in Overview of Tables, then **cpFirst** is the desired output. Leave the algorithm.

Copyright © 2025 Microsoft Corporation

6. Set **cp** to **cpPrev**. Go to step 1.

2.4.6 Applying Properties

This section specifies algorithms for determining the properties of text, paragraphs, lists, and tables. The final two subsections (Determining Properties of a Style and Determining Formatting Properties) specify the order in which the arrays of Prls are combined to compute the final property set. Recall from section 2.2.5 (Property Storage) that it is valid for multiple Prls to modify the same property. In this event, the last Prl applied determines the value of that property, unless otherwise specified in the specification of a particular Sprm. Thus, an application MUST process the arrays of Prls in the order specified in section 2.4.6.6, Determining Formatting Properties, to arrive at the correct property set.

Recall also from section 2.2.5 (Property Storage) that a Prl MAY<10> be ignored by applications that do not support the features represented by the Prl.

2.4.6.1 Direct Paragraph Formatting

This section explains how to find the properties applied directly (as opposed to through a style, for example) to a paragraph, given a character position **cp** within it. The properties are found as an array of **Prl** elements.

1. Follow the algorithm from Determining Paragraph Boundaries for finding the character position of the last character in the paragraph to completion. From step 5, remember the **PapxFkp** and *k*. From step 4, remember the offset in the WordDocument Stream at which **PapxFkp** was read. Let this offset be called **of**. From step 2 remember the **Pcd**. If the algorithm from Determining Paragraph Boundaries specifies that **cp** is invalid, leave the algorithm.

2. Find a **BxPap** at **PapxFkp**.**rgbx[***k***]**. Find a **PapxInFkp** at offset **of** + 2***BxPap.bOffset**

3. Find a **GrpprlAndIstd** in the **PapxInFkp** from step 2. The offset and size of the **GrpprlAndIstd** is instructed by the first byte of the **PapxInFkp**, as detailed at **PapxInFkp**.

4. Find the **grpprl** within the **GrpprlAndIstd**. This is an array of **Prl** elements that specifies the direct properties of this paragraph.

5. Finally **Pcd.Prm** specifies further property modifications that apply to this paragraph. If **Pcd**.**Prm** is a **Prm0** and the **Sprm** specified within **Prm0** modifies a paragraph property, append to the array of **Prl** elements from the previous step a single **Prl** made of the **Sprm** and value in **Prm0**. if **Pcd**.**Prm** is a **Prm1**, append to the array of **Prl** elements from the previous step any **Sprm** structures that modify paragraph properties within the array of **Prl** elements specified by **Prm1**.

2.4.6.2 Direct Character Formatting

This section specifies how to find the properties applied directly to a given character position **cp**. The result will be an array of **Prl** elements that specify the property modifications to be applied.

Additional formatting and properties can affect that **cp** as well, if a style is applied. To determine the full set of properties, including those from styles, see section 2.4.6.6 Determining Formatting Properties.

1. Follow the algorithm from Retrieving Text. From step 5 or 6, determine the offset in the WordDocument Stream where text was found. Call this offset **fc**. Also remember from step 4, the Pcd. If the algorithm from Retrieving Text specifies **cp** is invalid, leave the algorithm.

2. Read a PlcBteChpx at offset FibRgFcLcb97.**fcPlcfBteChpx** in the Table Stream, and of size FibRgFcLcb97.**lcbPlcfBteChpx**.

Copyright © 2025 Microsoft Corporation

3. Find the largest *i* such that plcbteChpx**.aFc**[*i*] ≤ **fc**. If the last element of plcbteChpx**.aFc** is less than or equal to **fc**, then **cp** is outside the range of character positions in this document, and is not valid. Read a ChpxFkp at offset **aPnBteChpx**[*i*]**.pn** *512 in the WordDocument Stream.

4. Find the largest *j* such that ChpxFkp**.rgfc[***j***]** ≤ **fc**. If the last element of ChpxFkp**.rgfc** is less than or equal to **fc**, then **cp** is outside the range of character positions in this document, and is not valid. Find a Chpx at offset ChpxFkp.**rgb**[*i*] in ChpxFkp.

5. The **grpprl** within the Chpx is an array of Prls that specifies the direct properties of this character.

6. Additionally, apply Pcd**.**Prm which specifies additional properties for this text. If Pcd.Prm is a Prm0 and the Sprm specified within Prm0 modifies a character property (a Sprm with an **sgc** value of 2), append a single Prl made of the Sprm and value in that Prm0 to the array of Prls from the previous step. If Pcd.Prm is a Prm1, append any Sprms that modify character properties from the array of Prls specified by Prm1.

2.4.6.3 Determining List Formatting of a Paragraph

A list in an MS-DOC file consists of one or more paragraphs. Each paragraph in a list has a nonzero **iLfo** property (see sprmPIlfo) and an **iLvl** property (see sprmPIlvl), which are used to determine the information that is necessary to format the paragraph as a member in a specific list. Paragraphs that share the same **iLfo** property, and exist in a range of text that constitutes a Valid Selection, are considered to be part of the same list. Paragraphs in a list do not need to be consecutive, and a list can overlap with other lists. This section describes an algorithm to add list formatting to a paragraph containing a given character position.

Given character position **cp**, use the following three-part algorithm to add list formatting to the paragraph containing **cp**.

Part 1

1. Follow the procedure for determining formatting properties, as specified in section 2.4.6.6, to find the paragraph properties for the paragraph that **cp** belongs to.

2. Let *iLfoCur* and *iLvlCur* be the **iLfo** (see sprmPIlfo) and **iLvl** (see sprmPIlvl) properties of the paragraph, respectively. If *iLfoCur* is zero, the paragraph is not part of a list, and the algorithm ends.

3. Let **lfo** be the LFO at PlfLfo.**rgLfo**[*iLfoCur* -1]. If there is no such LFO, the file is invalid and the algorithm ends.

4. Let **lstf** be the LSTF in PlfLst.**rgLstf** such that **lstf**.**lsid** equals **lfo**.**lsid**. If there is no such LSTF, the file is invalid and the algorithm ends.

5. Let **lfodata** be the LFOData at PlfLfo.**rgLfoData**[*iLfoCur* -1].

6. Let **lfolvl** be the LFOLVL in **lfodata**.**rgLfoLvl** such that **lfolvl**.**iLvl** equals *iLvlCur*, if such an LFOLVL exists. If there is no such LFOLVL, go to part 1 step 8.

7. If **lfolvl**.**fFormatting** is nonzero, let **lvl** be **lfolvl.lvl** and go to part 2 step 1.

8. Let *i* be 0. For each LSTF in PlfLst.**rgLstf** prior to **lstf**, if LSTF.**fSimpleList** is zero, let *i* = *i* + 9, if LSTF.**fSimpleList** is nonzero, let *i* be *i* + 1.

9. Let *i* be *i* + *iLvlCur*.

Copyright © 2025 Microsoft Corporation

10. Let **lvl** be the *i*th LVL in the array of LVLs appended to PlfLst (see the **fcPlfLst** field of FibRgFcLcb97).

Part 2

After the **lstf** and **lvl** are determined, the next step is to determine the number text of the paragraph.

1. Let *xstNumberText* be a copy of **lvl.xst**.

2. If **lvl.lvlf.nfc** is not equal to 0x17, go to part 2 step 4. If **lvl.lvlf.nfc** is equal to 0x17, the paragraph is in a bulleted level.

3. Let *xchBullet* be the 16-bit character at *xstNumberText***.rgtchar**[0]. If *xchBullet* & 0xF000 is nonzero, let *xstNumberText***.rgtchar**[0] equal *xchBullet* & 0x0FFF. Go to part 3 step 1.

4. For each entry *j* in **lvl**.**lvlf**.**rgbxchNums** such that **lvl**.**lvlf**.**rgbxchNums**[*j*] is nonzero, let *iLvlTemp* be the 16-bit integer stored at **lvl**.**xst.rgtchar**[**lvl**.**lvlf**.**rgbxchNums**[*j*] - 1]. If *iLvlTemp* == *iLvlCur*, replace the *iLvlTemp* placeholder in *xstNumberText* with the level number of the current paragraph. If *iLvlTemp* < *iLvlCur*, replace the *iLvlTemp* placeholder in *xstNumberText* with the level number of the closest previous paragraph in the list that has an **iLvl** property that equals *iLvlTemp*. If *iLvlTemp* > *iLvlCur*, the file is invalid and the algorithm ends. If **lvl.lvlf.fLegal** is nonzero, each of these level numbers MUST be reformatted as according to the **fLegal** field description in LVLF before they replace their respective placeholders.

Part 3

After the number text of the paragraphs is determined, the final step is to format the paragraph and the number text.

1. If **lstf**.**rgistdPara**[*iLvlCur*] != 0x0FFF, apply the style specified by **lstf**.**rgistdPara**[*iLvlCur*] to both the paragraph and *xstNumberText*.

2. Apply the character properties specified by **lvl**.**grpprlChpx** to *xstNumberText*.

3. Append the character specified by **lvl**.**lvlf**.**ixchFollow** to *xstNumberText*. *xstNumberText* is now the number text that will be displayed at the beginning of the paragraph.

4. Apply the paragraph properties specified by **lvl**.**grpprlPapx** to the paragraph, including *xstNumberText*.

5. Justify only the *xstNumberText* according to the justification specified by **lvl**.**lvlf**.**jc**.

The paragraph is now formatted as part of a list.

2.4.6.4 Determining Level Number of a Paragraph

The level number of a paragraph is the number in the number sequence of the level that corresponds to that paragraph, formatted according to an MSONFC (as specified in [MS-OSHARED] section 2.2.1.3). The number sequence of a level begins at a specified value and increments by 1 for each paragraph in the level. Also, the number sequence of a level can restart when certain other levels are encountered. See the specification of LVLF for more information. This section describes an algorithm to determine the level number of a paragraph containing a given character position.

Given character position **cp**, use the following algorithm to determine the level number of the paragraph containing **cp**:

Copyright © 2025 Microsoft Corporation

1. Follow steps 1 thru 10 of Determining List Formatting of a Paragraph to get the *iLvlCur*, **lvl** that correspond to the paragraph that **cp** belongs to. Also let *lsidCur* be **lfo.lsid** from step 4 of that algorithm.

2. Let *nfcCur* be **lvl.lvlf.nfc**. If *nfcCur* is equal to 0xFF or 0x17, this level has no number sequence, and the level number of the paragraph is an empty string. In this case, let *xsLevelNumber* be an empty string, and the algorithm ends.

3. Let *iStartAt* be **lvl.lvlf.iStartAt**.

4. If **lvl.lvlf.fNoRestart** is nonzero, let *iLvlRestartLim* be **lvl.lvlf.iLvlRestartLim**. Otherwise, let *iLvlRestartLim* be *iLvlCur*.

5. Let *numCur* be *iStartAt*.

6. For each paragraph *p* that has a nonzero **iLfo** property and that is in the same Valid Selection as **cp**, beginning with the paragraph starting at the lowest character position up to but not including the paragraph containing **cp**, carry out steps 7 to 12 below.

7. Determine **lfo.lsid** and **lfolvl** for paragraph *p* using steps 1 through 6 of Determining List Formatting of a Paragraph. If **lfo.lsid** is not equal to *lsidCur*, continue to the next paragraph.

8. If the **iLvl** property of the paragraph *p* is less than *iLvlRestartNum*, let *numCur* by *iStartAt*.

9. If the **iLvl** property of the paragraph *p* is not equal to *iLvlCur*, continue to the next paragraph.

10. If the value of the **iLfo** property of the paragraph *p* has not been encountered before in the execution of this algorithm and **lvolvl.fStartAt** is nonzero, then let *numCur* be **lfolvl.iStartAt** and continue to the next paragraph.

11. Let *numCur* be *numCur* + 1.

12. If the paragraph *p* does not contain **cp**, return to step 6.

13. Let *xsLevelNumber* be a string containing the number specified by *numCur* formatted according to the MSONFC (as specified in [MS-OSHARED] section 2.2.1.3) specified by *nfcCur*.

xsLevelNumber is now the level number of the paragraph.

2.4.6.5 Determining Properties of a Style

This section specifies an algorithm to determine the set of properties to apply to text, a paragraph, a table, or a list when a particular style is applied to it. Given an **istd**, one or more arrays of **Prl** can be derived that express the differences from defaults for this style. Depending on its **stk**, a style can specify properties for any combination of tables, paragraphs, and characters.

Given an **istd**:

1. Read the **FIB** from offset zero in the WordDocument Stream.

2. All versions of the **FIB** contain exactly one **FibRgFcLcb97** though it can be nested in a larger structure. Read a **STSH** from offset **FibRgFcLcb97.fcStshf** in the Table Stream with size **FibRgFcLcb97.lcbStshf**.

3. The given **istd** is a zero-based index into **STSH.rglpstd**. Read an **LPStd** at **STSH.rglpstd**[**istd**].

Copyright © 2025 Microsoft Corporation

4. Read the **STD** structure as **LPStd.std**, of length **LPStd.cbStd** bytes.

5. From the **STD.stdf.stdfBase** obtain **istdBase**. If **istdBase** is any value other than 0x0FFF, then this style is based on another style. Recursively apply this algorithm using **istdBase** as the starting **istd** to obtain one or more arrays of **Prl**s as the properties for tables, paragraphs and characters from the base style.

6. From the **STD.stdf.stdfBase** obtain **stk**. For more information, see the description of the **cupx** member of **StdfBase**. Read an **STD.grLPUpxSw**. Based on the **stk**, **grLPUpxSw** contains one of the following structures: StkParaGRLPUPX, StkCharGRLPUPX, StkTableGRLPUPX, StkListGRLPUPX.

7. Each of the preceding structures contains one or more of the following: LPUpxPapx, LPUpxChpx, LPUpxTapx. Each of the latter structures leads to one or more arrays of **Prl** that specify properties. For more information, see the sections documenting these structures for how to obtain these arrays.

8. For each array obtained in step 7 that specifies properties of a table, paragraph, or characters, append to the beginning of the corresponding array from step 5, if any. The resulting arrays of **Prl** are the desired output. Leave the algorithm.

2.4.6.6 Determining Formatting Properties

This section specifies an algorithm for how to combine properties from various sources that influence the properties of a character position to obtain the final formatting.

Character, paragraph, and table properties of the text at any given character position are specified by lists of differences from the defaults. Property Storage explains how to determine defaults and how to apply property differences. This section further specifies which lists of property differences are applicable and the order in which they apply.

In general, the differences from defaults are specified by one or more styles as well as any directly applied property modifications. Multiple styles can influence the properties at a given character position. A table style, for example, can specify paragraph properties that apply to some or all paragraphs within that table. A paragraph in such a table can itself have a paragraph style, in which case two different lists of differences modify the properties of said paragraph.

Given character position **cp**, use the following algorithm to determine the properties of text at **cp**:

Part 1:

1. Determine defaults for all properties the application is interested in. For further specification, see Property Storage.

2. Split the properties into three groups based on the objects they apply to: paragraph properties, character properties, and table properties as specified by Single Property Modifies. These are the set of properties which will be modified throughout the algorithm to arrive at the desired properties.

3. All versions of the **FIB** contain exactly one **FibRgFcLcb97** though it can be nested in a larger structure. Read an **STSH** from offset **FibRgFcLcb97.fcStshf** in the Table Stream, with size **FibRgFcLcb97.lcbStshf**. From the **STSH**, obtain an LPStshi and from that obtain an **STSHI**.

4. Apply the property modifications specified by the **ftcAsci**, **ftcFE** and **ftcOther** members of the **STSHI.Stshif** along with the **ftcBi** member of **STSHI** if specified.

5. Determine whether **cp** is in a table or not. For further specification, see Determining Cell Boundaries. If **cp** is not in a table, go to step 1 of part 2.

6. Determine the table style that is applied to the innermost row that contains **cp** as follows:

Copyright © 2025 Microsoft Corporation

1. Apply the algorithm from Determining Row Boundaries to obtain the character position of the TTP mark of the innermost row that contains **cp**. Call this **cpTtp**.

2. Apply the algorithm from Direct Paragraph Formatting on **cpTtp**.

3. Apply the array of **Prl** elements that was obtained to the table row and determine the **istd** of the table style applied to this table row using sprmTIstd. Call it **istdTable**. If no table style is applied, go to step 1 of part 2.

7. Using the algorithm from Determining Properties of a Style, obtain a **grpprlPapx**, **grpprlChpx**, and a **grpprlTapx** (if available) from the **istdTable**. Apply any property modifications specified in **grpprlChpx**, **grpprlPapx**, and **grpprlTapx** to the character, paragraph, and table properties, respectively.

8. Find the position of the innermost cell that contains **cp** within the innermost table that contains **cp** by applying the algorithm from Determining Row Boundaries and Determining Cell Boundaries as appropriate. Specifically, determine if the innermost cell that contains **cp** belongs to the first row, first column, last row, or last column of the innermost table that contains **cp**. Also, determine whether the innermost cell that contains **cp** is in an even or an odd **horizontal band** based on horizontal banding applied in **grpprlTapx** with sprmTCHorzBands and, similarly, if it is in an even or an odd **vertical band** based on vertical banding applied in **grpprlTapx** with sprmTCVertBands. Note that if sprmTTlp.**grfatl** specifies that the top row of the table receives special formatting, then the top row of the table and any row with sprmTTableHeader applied with a value of 0x01 is not counted when determining odd or even horizontal banding. Similarly, if sprmTTlp.**grfatl** specifies that the logically leftmost column of the table receives special formatting, then that column is not counted when determining odd or even vertical banding.

9. Next, using the array of Prls obtained in step 6, determine if additional property differences need to be applied to **cp** based on its location in the table as specified by sprmTTlp.**grfatl**. If additional property differences need to be applied, look for sprmPCnfs within the **grpprlPapx** from step 7, sprmCCnfs within **grpprlChpx** from step 7, and sprmTCnfs within **grpprlTapx** from step 7 whose CNFC, see CNFOperand.**cnfc**, matches the position information found in step 8. The following table specifies which CNFC values match which position information.

CNFC Value	Matches …
0x0001	Any cell in the top row or with sprmTTableHeader applied with a value of 0x01 if sprmTTlp.**grfatl** specifies that top row of the table receives special formatting.
0x0002	Any cell in the bottom row if sprmTTlp.**grfatl** specifies that bottom row of the table receives special formatting and the cell does not match CNFC value 0x0001.
0x0004	Any cell in the logically leftmost column if sprmTTlp.**grfatl** specifies that the logically leftmost column receives special formatting.
0x0008	Any cell in the logically rightmost column if sprmTTlp.**grfatl** specifies that the logically rightmost column receives special formatting and the cell does not match CNFC value 0x0004.
0x0010	Any cell in an odd numbered vertical band if sprmTTlp.**grfatl** specifies that odd numbered vertical bands receive special formatting and the cell does not match CNFC values 0x0004 or 0x0008.
0x0020	Any cell in an even numbered vertical band if sprmTTlp.**grfatl** specifies that even numbered vertical bands receive special formatting, and the cell does not match CNFC values 0x0004 or 0x0008.
0x0040	Any cell in an odd numbered horizontal band if sprmTTlp.**grfatl** specifies that odd numbered horizontal bands receive special formatting, and the cell does not match CNFC values 0x0001 or 0x0002.
0x0080	Any cell in an even numbered horizontal band if sprmTTlp.**grfatl** specifies that even numbered horizontal bands receive special formatting, and the cell does not match CNFC values 0x0001 or 0x0002.
0x0100	The logically rightmost cell on the top row of the table if sprmTTlp.**grfatl** specifies that both the top row and the logically rightmost column receive special formatting and the cell does not match

Copyright © 2025 Microsoft Corporation

CNFC Value	Matches …
	CNFC value 0x200.
0x0200	The logically leftmost cell on the top row of the table if sprmTTlp.**grfatl** specifies that both the top row and the logically leftmost column receive special formatting.
0x0400	The logically rightmost cell on the bottom row of the table if sprmTTlp.**grfatl** specifies that both the bottom row and the logically rightmost column receive special formatting and the cell does not match CNFC value 0x0100, 0x0200, or 0x0800.
0x0800	The logically leftmost cell on the bottom row of the table if sprmTTlp.**grfatl** specifies that both the bottom row and the logically leftmost column receive special formatting and the cell does not match CNFC value 0x0100 or 0x0200.

A single cell position can match multiple CNFC values. For example the logically rightmost cell on the top row could match all of these CNFC values: 0x0100, 0x0008, 0x0001. Apply conditional formatting in the following order.

CNFC Values	Conditional Formatting Type
0x0040 or 0x0080	Odd or even horizontal banding
0x0010 or 0x0020	Odd or even vertical banding
0x0004 or 0x0008	First or last column
0x0001 or 0x0002	First or last row
0x0100, 0x0200, 0x0400, or 0x0800	Corner cell

Apply any property modifications specified in a matching sprmCCnf, if one exists, to the character properties. Apply any property modifications specified in a matching sprmPCnf, if one exists, to paragraph properties. Apply any property modifications specified in a matching sprmTCnf, if one exists, to table properties.

Part 2:

1. Apply the algorithm from Direct Paragraph Formatting up to and including step 4. The remaining steps of that algorithm are applied later. Obtain **GrpprlAndIstd**. Using the algorithm from Determining Properties of a Style, obtain any paragraph property modifications that are specified by **GrpprlAndIstd**.**istd**.

2. Apply any paragraph property modifications obtained from **GrpprlAndIstd**.**istd** in the previous step. Next, apply any paragraph property modifications found in **GrpprlAndIstd**.**grpprl**. Finally, finish the remaining steps in the algorithm from Direct Paragraph Formatting that was started in the previous step.

3. If the paragraph that contains **cp** belongs to a list, apply any further paragraph property modifications specified by the list. For information about how to determine whether a paragraph belongs to a list and how to obtain the property modifications specified by the list, see Determining List Formatting of a Paragraph. At this point the paragraph properties reflect those of the paragraph that contains **cp**. The remaining steps determine the character properties.

4. Using the algorithm from Determining Properties of a Style, obtain any character property modifications specified by **GrpprlAndIstd**.**istd** from step 1 of part 2 or the value of the last sprmPIstdPermute if any in **GrpprlAndIstd**.**grpprl**. Apply any character property modifications obtained from the style to the character properties.

5. Finally, using the algorithm from Direct Character Formatting, obtain any property modifications to be applied to character properties and apply them.

Copyright © 2025 Microsoft Corporation

2.4.7 Application Data For VtHyperlink

The following algorithm specifies how hyperlink properties, as specified in [MS-OSHARED] section 2.3.3.1.18, are associated with content in a document construct their **dwApp** field value.

- If the hyperlink is associated with an OfficeArtFSP shape, as specified in [MS-ODRAW] section 2.2.40, the **dwApp** value MUST be 0xFFFFFFFF. Otherwise the hyperlink MUST be associated with a picture, an external link to a picture source, or other document content.
- If the hyperlink is associated directly with a picture, as opposed to the hyperlink field associated with the picture, or an external link to a picture source, the **dwApp** value MUST be set to an FcCompressed structure that specifies the starting offset of the *field result* in the WordDocument Stream associated with the picture. For further specification on *field results*, see **Plcfld**.
- If the hyperlink is associated with any other type of document content, including the hyperlink field of a WordArt shape or picture, the **dwApp** value MUST be set to an unsigned 4-byte integer that specifies the index into a **Plcfld**. The specified **Plcfld** item corresponds to the field begin character of the hyperlink field in the document content associated with the hyperlink property.

  The hyperlink properties that have **dwApp** set to an index into a **Plcfld** MUST conform to a specific ordering relative to each other when written. They MUST be written within the property set hyperlink property array **VtHyperlinks**, as specified in [MS-OSHARED] section 2.3.3.1.21, grouped according to the document **Plcfld** to which the indices apply, in the following order:

1. Main Document links.
2. Footnote Document links.
3. Header Document links.
4. Comment Document links.
5. Endnote Document links.
6. Textbox Document links.
7. Header Textbox Document links.

   Within these groupings the hyperlink properties MUST be ordered from largest index to smallest index.

   Example:

   A document contains two hyperlink fields in the Main Document, and two hyperlink fields in the Footnote Document. The field indices for the hyperlinks (h1M, and h2M) in the Main Document are 1 and 4 respectively. The field indices for the hyperlinks (h1F, and h2F) in the Footnote Document are 0 and 3 respectively.

   The hyperlink properties in this example MUST be written in the order: h2M, h1M, h2F, h1F.

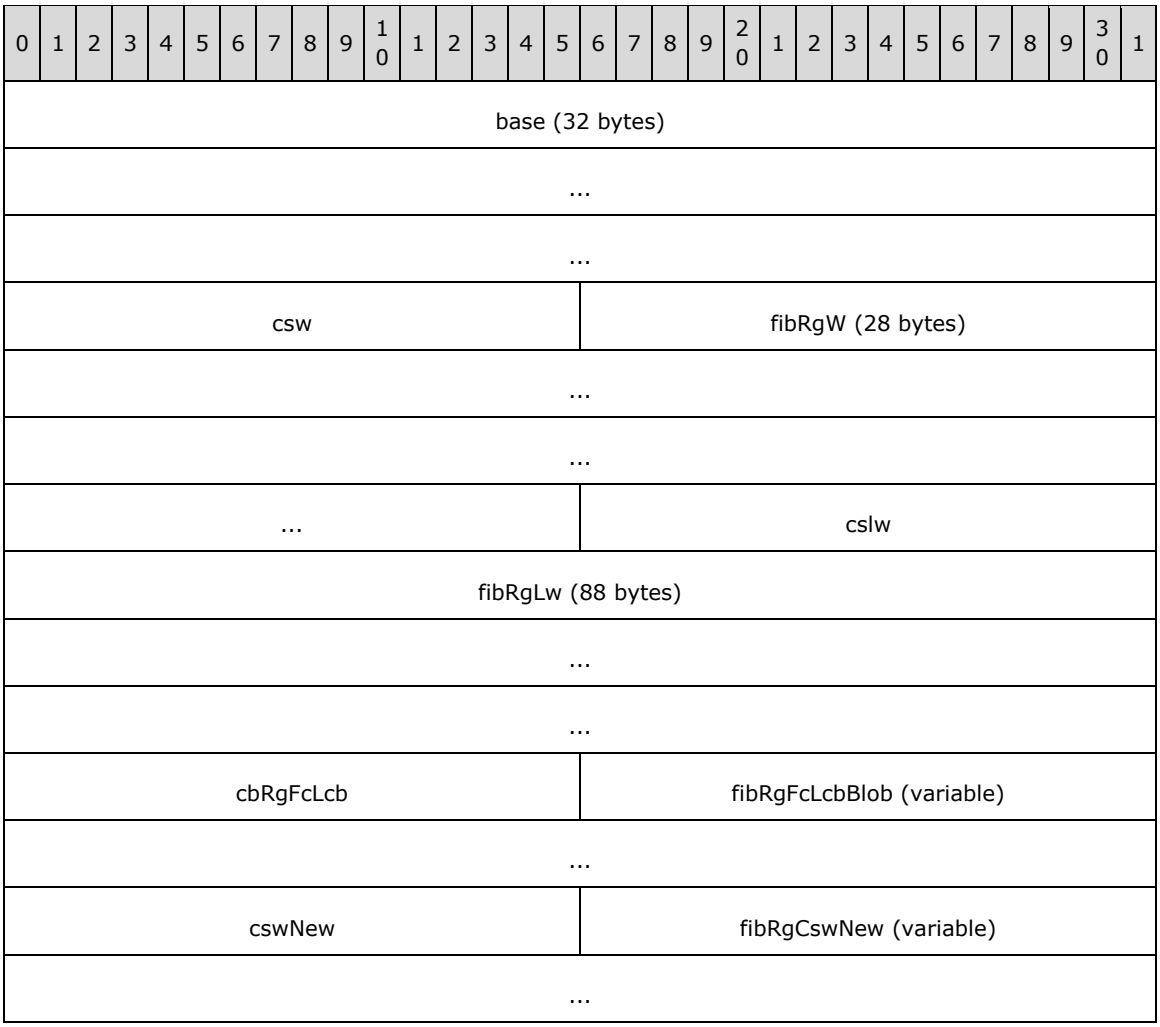
2.5 The File Information Block

2.5.1 Fib

The **Fib** structure contains information about the document and specifies the file pointers to various portions that make up the document.

The **Fib** is a variable length structure. With the exception of the base portion which is fixed in size, every section is preceded with a count field that specifies the size of the next section.

Copyright © 2025 Microsoft Corporation



base (32 bytes): The **FibBase**.

csw (2 bytes): An unsigned integer that specifies the count of 16-bit values corresponding to **fibRgW** that follow. MUST be 0x000E.

fibRgW (28 bytes): The **FibRgW97**.

cslw (2 bytes): An unsigned integer that specifies the count of 32-bit values corresponding to **fibRgLw** that follow. MUST be 0x0016.

fibRgLw (88 bytes): The **FibRgLw97**.

cbRgFcLcb (2 bytes): An unsigned integer that specifies the count of 64-bit values corresponding to **fibRgFcLcbBlob** that follow. This MUST be one of the following values, depending on the value of **nFib**.

Value of nFib	cbRgFcLcb
0x00C1	0x005D
0x00D9	0x006C
0x0101	0x0088
0x010C	0x00A4
0x0112	0x00B7

Copyright © 2025 Microsoft Corporation

fibRgFcLcbBlob (variable): The **FibRgFcLcb**.

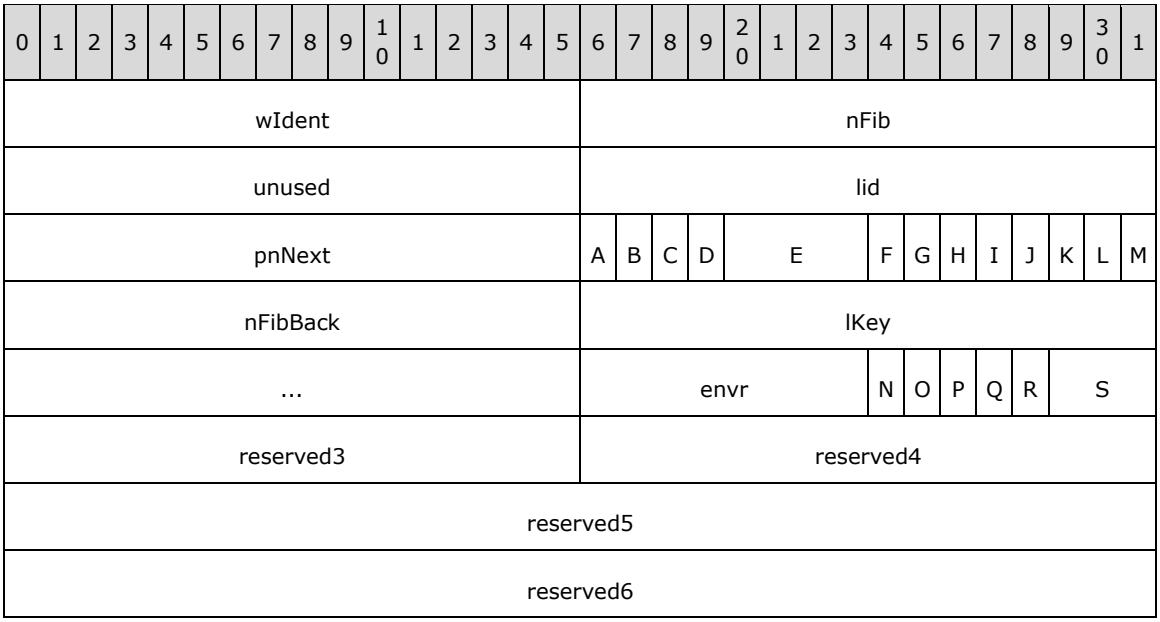
cswNew (2 bytes): An unsigned integer that specifies the count of 16-bit values corresponding to **fibRgCswNew** that follow. This MUST be one of the following values, depending on the value of **nFib**.

Value of nFib	cswNew
0x00C1	0
0x00D9	0x0002
0x0101	0x0002
0x010C	0x0002
0x0112	0x0005

fibRgCswNew (variable): Optional. If **cswNew** is nonzero, this is **fibRgCswNew**. Otherwise, it is not present in the file.

2.5.2 FibBase

The **FibBase** structure is the fixed-size portion of the Fib.



wIdent (2 bytes): An unsigned integer that specifies that this is a Word Binary File. This value MUST be 0xA5EC.

nFib (2 bytes): An unsigned integer that specifies the version number of the file format used. Superseded by **FibRgCswNew**.**nFibNew** if it is present. This value SHOULD<11> be 0x00C1.

unused (2 bytes): This value is undefined and MUST be ignored.

lid (2 bytes): A **LID** that specifies the install language of the application that is producing the document. If nFib is 0x00D9 or greater, then any East Asian install lid or any install lid with a base language of Spanish, German or French MUST be recorded as 0x0409. If the nFib is 0x0101 or

Copyright © 2025 Microsoft Corporation

greater, then any install lid with a base language of Vietnamese, Thai, or Hindi MUST be recorded as 0x0409.

pnNext (2 bytes): An unsigned integer that specifies the offset in the WordDocument stream of the **FIB** for the document which contains all the **AutoText** items. If this value is 0, there are no AutoText items attached. Otherwise the **FIB** is found at file location **pnNext**×512. If **fGlsy** is 1 or **fDot** is 0, this value MUST be 0. If **pnNext** is not 0, each **FIB** MUST share the same values for **FibRgFcLcb97**.**fcPlcfBteChpx, FibRgFcLcb97**.**lcbPlcfBteChpx, FibRgFcLcb97.fcPlcfBtePapx, FibRgFcLcb97.lcbPlcfBtePapx,** and **FibRgLw97**.**cbMac**.

A - fDot (1 bit): Specifies whether this is a **document template**.

B - fGlsy (1 bit): Specifies whether this is a document that contains only AutoText items (see **FibRgFcLcb97**.**fcSttbfGlsy**, **FibRgFcLcb97**.**fcPlcfGlsy** and **FibRgFcLcb97**.**fcSttbGlsyStyle**).

C - fComplex (1 bit): Specifies that the last save operation that was performed on this document was an **incremental save** operation.

D - fHasPic (1 bit): When set to 0, there SHOULD<12> be no pictures in the document.

E - cQuickSaves (4 bits): An unsigned integer. If nFib is less than 0x00D9, then **cQuickSaves** specifies the number of consecutive times this document was incrementally saved. If nFib is 0x00D9 or greater, then **cQuickSaves** MUST be 0xF.

F - fEncrypted (1 bit): Specifies whether the document is encrypted or obfuscated as specified in Encryption and Obfuscation.

G - fWhichTblStm (1 bit): Specifies the Table stream to which the **FIB** refers. When this value is set to 1, use 1Table; when this value is set to 0, use 0Table.

H - fReadOnlyRecommended (1 bit): Specifies whether the document author recommended that the document be opened in read-only mode.

I - fWriteReservation (1 bit): Specifies whether the document has a **write-reservation password**.

J - fExtChar (1 bit): This value MUST be 1.

K - fLoadOverride (1 bit): Specifies whether to override the language information and font that are specified in the paragraph style at **istd** 0 (the normal style) with the defaults that are appropriate for the installation language of the application.

L - fFarEast (1 bit): Specifies whether the installation language of the application that created the document was an **East Asian language**.

M - fObfuscated (1 bit): If **fEncrypted** is 1, this bit specifies whether the document is obfuscated by using XOR obfuscation (section 2.2.6.1); otherwise, this bit MUST be ignored.

nFibBack (2 bytes): This value SHOULD<13> be 0x00BF. This value MUST be 0x00BF or 0x00C1.

lKey (4 bytes): If **fEncrypted** is 1 and **fObfuscated** is 1, this value specifies the XOR obfuscation (section 2.2.6.1) password verifier. If **fEncrypted** is 1 and **fObfuscated** is 0, this value specifies the size of the **EncryptionHeader** that is stored at the beginning of the Table stream as described in Encryption and Obfuscation. Otherwise, this value MUST be 0.

envr (1 byte): This value MUST be 0, and MUST be ignored.

N - fMac (1 bit): This value MUST be 0, and MUST be ignored.

O - fEmptySpecial (1 bit): This value SHOULD<14> be 0 and SHOULD<15> be ignored.

Copyright © 2025 Microsoft Corporation

P - fLoadOverridePage (1 bit): Specifies whether to override the section properties for page size, orientation, and margins with the defaults that are appropriate for the installation language of the application.

Q - reserved1 (1 bit): This value is undefined and MUST be ignored.

R - reserved2 (1 bit): This value is undefined and MUST be ignored.

S - fSpare0 (3 bits): This value is undefined and MUST be ignored.

reserved3 (2 bytes): This value MUST be 0 and MUST be ignored.

reserved4 (2 bytes): This value MUST be 0 and MUST be ignored.

reserved5 (4 bytes): This value is undefined and MUST be ignored.

reserved6 (4 bytes): This value is undefined and MUST be ignored.

2.5.3 FibRgW97

The **FibRgW97** structure is a variable-length portion of the Fib.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
reserved1																reserved2															
reserved3																reserved4															
reserved5																reserved6															
reserved7																reserved8															
reserved9																reserved10															
reserved11																reserved12															
reserved13																lidFE															

reserved1 (2 bytes): This value is undefined and MUST be ignored.

reserved2 (2 bytes): This value is undefined and MUST be ignored.

reserved3 (2 bytes): This value is undefined and MUST be ignored.

reserved4 (2 bytes): This value is undefined and MUST be ignored.

reserved5 (2 bytes): This value SHOULD<16> be zero, and MUST be ignored.

reserved6 (2 bytes): This value SHOULD<17> be zero, and MUST be ignored.

reserved7 (2 bytes): This value SHOULD<18> be zero, and MUST be ignored.

reserved8 (2 bytes): This value SHOULD<19> be zero, and MUST be ignored.

reserved9 (2 bytes): This value SHOULD<20> be zero, and MUST be ignored.

reserved10 (2 bytes): This value SHOULD<21> be zero, and MUST be ignored.

Copyright © 2025 Microsoft Corporation

reserved11 (2 bytes): This value SHOULD<22> be zero, and MUST be ignored.

reserved12 (2 bytes): This value SHOULD<23> be zero, and MUST be ignored.

reserved13 (2 bytes): This value SHOULD<24> be zero, and MUST be ignored.

lidFE (2 bytes): A **LID** whose meaning depends on the **nFib** value, which is one of the following.

nFib value	Meaning
0x00C1	If **FibBase**.**fFarEast** is "true", this is the **LID** of the stored style names. Otherwise it MUST be ignored.
0x00D9 0x0101 0x010C 0x0112	The **LID** of the stored style names (**STD**.**xstzName**)

2.5.4 FibRgLw97

The **FibRgLw97** structure is the third section of the **FIB**. This contains an array of 4-byte values.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
cbMac																															
reserved1																															
reserved2																															
ccpText																															
ccpFtn																															
ccpHdd																															
reserved3																															
ccpAtn																															
ccpEdn																															
ccpTxbx																															
ccpHdrTxbx																															
reserved4																															
reserved5																															
reserved6																															

Copyright © 2025 Microsoft Corporation

reserved7
reserved8
reserved9
reserved10
reserved11
reserved12
reserved13
reserved14

cbMac (4 bytes): Specifies the count of bytes of those written to the WordDocument stream of the file that have any meaning. All bytes in the WordDocument stream at offset **cbMac** and greater MUST be ignored.

reserved1 (4 bytes): This value is undefined and MUST be ignored.

reserved2 (4 bytes): This value is undefined and MUST be ignored.

ccpText (4 bytes): A signed integer that specifies the count of CPs in the main document. This value MUST be 0, 1, or greater.

ccpFtn (4 bytes): A signed integer that specifies the count of CPs in the footnote subdocument. This value MUST be 0, 1, or greater.

ccpHdd (4 bytes): A signed integer that specifies the count of CPs in the header subdocument. This value MUST be 0, 1, or greater.

reserved3 (4 bytes): This value MUST be 0 and MUST be ignored.

ccpAtn (4 bytes): A signed integer that specifies the count of CPs in the comment subdocument. This value MUST be 0, 1, or greater.

ccpEdn (4 bytes): A signed integer that specifies the count of CPs in the endnote subdocument. This value MUST be 0, 1, or greater.

ccpTxbx (4 bytes): A signed integer that specifies the count of CPs in the textbox subdocument of the main document. This value MUST be 0, 1, or greater.

ccpHdrTxbx (4 bytes): A signed integer that specifies the count of CPs in the textbox subdocument of the header. This value MUST be 0, 1, or greater.

reserved4 (4 bytes): This value is undefined and MUST be ignored.

reserved5 (4 bytes): This value is undefined and MUST be ignored.

reserved6 (4 bytes): This value MUST be equal or less than the number of data elements in **PlcBteChpx**, as specified by **FibRgFcLcb97**.**fcPlcfBteChpx** and **FibRgFcLcb97**.**lcbPlcfBteChpx**. This value MUST be ignored.

reserved7 (4 bytes): This value is undefined and MUST be ignored

Copyright © 2025 Microsoft Corporation

reserved8 (4 bytes): This value is undefined and MUST be ignored

reserved9 (4 bytes): This value MUST be less than or equal to the number of data elements in **PlcBtePapx**, as specified by **FibRgFcLcb97**.**fcPlcfBtePapx** and **FibRgFcLcb97**.**lcbPlcfBtePapx**. This value MUST be ignored.

reserved10 (4 bytes): This value is undefined and MUST be ignored.

reserved11 (4 bytes): This value is undefined and MUST be ignored.

reserved12 (4 bytes): This value SHOULD<25> be zero, and MUST be ignored.

reserved13 (4 bytes): This value MUST be 0 and MUST be ignored.

reserved14 (4 bytes): This value MUST be 0 and MUST be ignored.

2.5.5 FibRgFcLcb

The **FibRgFcLcb** structure specifies the file offsets and byte counts for various portions of the data in the document. The structure of **FibRgFcLcb** depends on the value of **nFib**, which is one of the following.

Value	Meaning
0x00C1	fibRgFcLcb97
0x00D9	fibRgFcLcb2000
0x0101	fibRgFcLcb2002
0x010C	fibRgFcLcb2003
0x0112	fibRgFcLcb2007

2.5.6 FibRgFcLcb97

The **FibRgFcLcb97** structure is a variable-length portion of the Fib.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fcStshfOrig																															
lcbStshfOrig																															
fcStshf																															
lcbStshf																															
fcPlcffndRef																															
lcbPlcffndRef																															
fcPlcffndTxt																															
lcbPlcffndTxt																															

Copyright © 2025 Microsoft Corporation

fcPlcfandRef
lcbPlcfandRef
fcPlcfandTxt
lcbPlcfandTxt
fcPlcfSed
lcbPlcfSed
fcPlcPad
lcbPlcPad
fcPlcfPhe
lcbPlcfPhe
fcSttbfGlsy
lcbSttbfGlsy
fcPlcfGlsy
lcbPlcfGlsy
fcPlcfHdd
lcbPlcfHdd
fcPlcfBteChpx
lcbPlcfBteChpx
fcPlcfBtePapx
lcbPlcfBtePapx
fcPlcfSea
lcbPlcfSea
fcSttbfFfn
lcbSttbfFfn
fcPlcfFldMom

Copyright © 2025 Microsoft Corporation

lcbPlcfFldMom
fcPlcfFldHdr
lcbPlcfFldHdr
fcPlcfFldFtn
lcbPlcfFldFtn
fcPlcfFldAtn
lcbPlcfFldAtn
fcPlcfFldMcr
lcbPlcfFldMcr
fcSttbfBkmk
lcbSttbfBkmk
fcPlcfBkf
lcbPlcfBkf
fcPlcfBkl
lcbPlcfBkl
fcCmds
lcbCmds
fcUnused1
lcbUnused1
fcSttbfMcr
lcbSttbfMcr
fcPrDrvr
lcbPrDrvr
fcPrEnvPort
lcbPrEnvPort

Copyright © 2025 Microsoft Corporation

fcPrEnvLand
lcbPrEnvLand
fcWss
lcbWss
fcDop
lcbDop
fcSttbfAssoc
lcbSttbfAssoc
fcClx
lcbClx
fcPlcfPgdFtn
lcbPlcfPgdFtn
fcAutosaveSource
lcbAutosaveSource
fcGrpXstAtnOwners
lcbGrpXstAtnOwners
fcSttbfAtnBkmk
lcbSttbfAtnBkmk
fcUnused2
lcbUnused2
fcUnused3
lcbUnused3
fcPlcSpaMom
lcbPlcSpaMom
fcPlcSpaHdr

Copyright © 2025 Microsoft Corporation

lcbPlcSpaHdr
fcPlcfAtnBkf
lcbPlcfAtnBkf
fcPlcfAtnBkl
lcbPlcfAtnBkl
fcPms
lcbPms
fcFormFldSttbs
lcbFormFldSttbs
fcPlcfendRef
lcbPlcfendRef
fcPlcfendTxt
lcbPlcfendTxt
fcPlcfFldEdn
lcbPlcfFldEdn
fcUnused4
lcbUnused4
fcDggInfo
lcbDggInfo
fcSttbfRMark
lcbSttbfRMark
fcSttbfCaption
lcbSttbfCaption
fcSttbfAutoCaption
lcbSttbfAutoCaption

Copyright © 2025 Microsoft Corporation

fcPlcfWkb
lcbPlcfWkb
fcPlcfSpl
lcbPlcfSpl
fcPlcftxbxTxt
lcbPlcftxbxTxt
fcPlcfFldTxbx
lcbPlcfFldTxbx
fcPlcfHdrtxbxTxt
lcbPlcfHdrtxbxTxt
fcPlcffldHdrTxbx
lcbPlcffldHdrTxbx
fcStwUser
lcbStwUser
fcSttbTtmbd
lcbSttbTtmbd
fcCookieData
lcbCookieData
fcPgdMotherOldOld
lcbPgdMotherOldOld
fcBkdMotherOldOld
lcbBkdMotherOldOld
fcPgdFtnOldOld
lcbPgdFtnOldOld
fcBkdFtnOldOld

Copyright © 2025 Microsoft Corporation

lcbBkdFtnOldOld
fcPgdEdnOldOld
lcbPgdEdnOldOld
fcBkdEdnOldOld
lcbBkdEdnOldOld
fcSttbfIntlFld
lcbSttbfIntlFld
fcRouteSlip
lcbRouteSlip
fcSttbSavedBy
lcbSttbSavedBy
fcSttbFnm
lcbSttbFnm
fcPlfLst
lcbPlfLst
fcPlfLfo
lcbPlfLfo
fcPlcfTxbxBkd
lcbPlcfTxbxBkd
fcPlcfTxbxHdrBkd
lcbPlcfTxbxHdrBkd
fcDocUndoWord9
lcbDocUndoWord9
fcRgbUse
lcbRgbUse

Copyright © 2025 Microsoft Corporation

fcUsp
lcbUsp
fcUskf
lcbUskf
fcPlcupcRgbUse
lcbPlcupcRgbUse
fcPlcupcUsp
lcbPlcupcUsp
fcSttbGlsyStyle
lcbSttbGlsyStyle
fcPlgosl
lcbPlgosl
fcPlcocx
lcbPlcocx
fcPlcfBteLvc
lcbPlcfBteLvc
dwLowDateTime
dwHighDateTime
fcPlcfLvcPre10
lcbPlcfLvcPre10
fcPlcfAsumy
lcbPlcfAsumy
fcPlcfGram
lcbPlcfGram
fcSttbListNames

Copyright © 2025 Microsoft Corporation

lcbSttbListNames
fcSttbfUssr
lcbSttbfUssr

fcStshfOrig (4 bytes): This value is undefined and MUST be ignored.

lcbStshfOrig (4 bytes): This value is undefined and MUST be ignored.

fcStshf (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **STSH** that specifies the style sheet for this document begins at this offset.

lcbStshf (4 bytes): An unsigned integer that specifies the size, in bytes, of the **STSH** that begins at offset **fcStshf** in the Table Stream. This MUST be a nonzero value.

fcPlcffndRef (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcffndRef** begins at this offset and specifies the locations of footnote references in the Main Document, and whether those references use auto-numbering or custom symbols. If **lcbPlcffndRef** is zero, **fcPlcffndRef** is undefined and MUST be ignored.

lcbPlcffndRef (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcffndRef** that begins at offset **fcPlcffndRef** in the Table Stream.

fcPlcffndTxt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcffndTxt** begins at this offset and specifies the locations of each block of footnote text in the Footnote Document. If **lcbPlcffndTxt** is zero, **fcPlcffndTxt** is undefined and MUST be ignored.

lcbPlcffndTxt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcffndTxt** that begins at offset **fcPlcffndTxt** in the Table Stream.

lcbPlcffndTxt MUST be zero if **FibRgLw97.ccpFtn** is zero, and MUST be nonzero if **FibRgLw97.ccpFtn** is nonzero.

fcPlcfandRef (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfandRef** begins at this offset and specifies the dates, user initials, and locations of comments in the Main Document. If **lcbPlcfandRef** is zero, **fcPlcfandRef** is undefined and MUST be ignored.

lcbPlcfandRef (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfandRef** at offset **fcPlcfandRef** in the Table Stream.

fcPlcfandTxt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfandTxt** begins at this offset and specifies the locations of comment text ranges in the Comment Document. If **lcbPlcfandTxt** is zero, **fcPlcfandTxt** is undefined, and MUST be ignored.

lcbPlcfandTxt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfandTxt** at offset **fcPlcfandTxt** in the Table Stream.

lcbPlcfandTxt MUST be zero if **FibRgLw97.ccpAtn** is zero, and MUST be nonzero if **FibRgLw97.ccpAtn** is nonzero.

fcPlcfSed (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfSed** begins at this offset and specifies the locations of property lists for each section in the Main Document. If **lcbPlcfSed** is zero, **fcPlcfSed** is undefined and MUST be ignored.

lcbPlcfSed (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfSed** that begins at offset **fcPlcfSed** in the Table Stream.

fcPlcPad (4 bytes): This value is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

lcbPlcPad (4 bytes): This value MUST be 0, and MUST be ignored.

fcPlcfPhe (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plc** begins at this offset and specifies version-specific information about paragraph height. This **Plc** SHOULD NOT<26> be emitted and SHOULD<27> be ignored.

lcbPlcfPhe (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plc** at offset **fcPlcfPhe** in the Table Stream.

fcSttbfGlsy (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **SttbfGlsy** that contains information about the **AutoText** items that are defined in this document begins at this offset.

lcbSttbfGlsy (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbfGlsy** at offset **fcSttbfGlsy** in the Table Stream. If **base.fGlsy** of the **Fib** that contains this **FibRgFcLcb97** is zero, this value MUST be 0.

fcPlcfGlsy (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfGlsy** that contains information about the AutoText items that are defined in this document begins at this offset.

lcbPlcfGlsy (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfGlsy** at offset **fcPlcfGlsy** in the Table Stream. If **base.fGlsy** of the **Fib** that contains this **FibRgFcLcb97** is zero, this value MUST be 0.

fcPlcfHdd (4 bytes): An unsigned integer that specifies the offset in the Table Stream where a **Plcfhdd** begins. The **Plcfhdd** specifies the locations of each block of header/footer text in the WordDocument Stream. If **lcbPlcfHdd** is 0, **fcPlcfHdd** is undefined and MUST be ignored.

lcbPlcfHdd (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfhdd** at offset **fcPlcfHdd** in the Table Stream. If there is no **Plcfhdd**, this value MUST be 0. A **Plcfhdd** MUST exist if **FibRgLw97**.**ccpHdd** indicates that there are characters in the Header Document (that is, if **FibRgLw97**.**ccpHdd** is greater than 0). Otherwise, the **Plcfhdd** MUST NOT exist.

fcPlcfBteChpx (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcBteChpx** begins at the offset. **fcPlcfBteChpx** MUST be greater than zero, and MUST be a valid offset in the Table Stream.

lcbPlcfBteChpx (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcBteChpx** at offset **fcPlcfBteChpx** in the Table Stream. **lcbPlcfBteChpx** MUST be greater than zero.

fcPlcfBtePapx (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcBtePapx** begins at the offset. **fcPlcfBtePapx** MUST be greater than zero, and MUST be a valid offset in the Table Stream.

lcbPlcfBtePapx (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcBtePapx** at offset **fcPlcfBtePapx** in the Table Stream. **lcbPlcfBteChpx** MUST be greater than zero.

fcPlcfSea (4 bytes): This value is undefined and MUST be ignored.

lcbPlcfSea (4 bytes): This value MUST be 0, and MUST be ignored.

fcSttbfFfn (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **SttbfFfn** begins at this offset. This table specifies the fonts that are used in the document. If **lcbSttbfFfn** is 0, **fcSttbfFfn** is undefined and MUST be ignored.

lcbSttbfFfn (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbfFfn** at offset **fcSttbfFfn** in the Table Stream.

fcPlcfFldMom (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfId** begins at this offset and specifies the locations of field characters in the Main Document. All CPs in

Copyright © 2025 Microsoft Corporation

this **Plcfld** MUST be greater than or equal to 0 and less than or equal to **FibRgLw97.ccpText**. If **lcbPlcfFldMom** is zero, **fcPlcfFldMom** is undefined and MUST be ignored.

lcbPlcfFldMom (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfld** at offset **fcPlcfFldMom** in the Table Stream.

fcPlcfFldHdr (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfld** begins at this offset and specifies the locations of field characters in the Header Document. All CPs in this **Plcfld** are relative to the starting position of the Header Document. All CPs in this **Plcfld** MUST be greater than or equal to zero and less than or equal to **FibRgLw97**.**ccpHdd**. If **lcbPlcfFldHdr** is zero, **fcPlcfFldHdr** is undefined and MUST be ignored.

lcbPlcfFldHdr (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfld** at offset **fcPlcfFldHdr** in the Table Stream.

fcPlcfFldFtn (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfld** begins at this offset and specifies the locations of field characters in the Footnote Document. All CPs in this **Plcfld** are relative to the starting position of the Footnote Document. All CPs in this **Plcfld** MUST be greater than or equal to zero and less than or equal to **FibRgLw97.ccpFtn**. If **lcbPlcfFldFtn** is zero, **fcPlcfFldFtn** is undefined, and MUST be ignored.

lcbPlcfFldFtn (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfld** at offset **fcPlcfFldFtn** in the Table Stream.

fcPlcfFldAtn (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfld** begins at this offset and specifies the locations of field characters in the Comment Document. All CPs in this **Plcfld** are relative to the starting position of the Comment Document. All CPs in this **Plcfld** MUST be greater than or equal to zero and less than or equal to **FibRgLw97.ccpAtn**. If **lcbPlcfFldAtn** is zero, **fcPlcfFldAtn** is undefined and MUST be ignored.

lcbPlcfFldAtn (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfld** at offset **fcPlcfFldAtn** in the Table Stream.

fcPlcfFldMcr (4 bytes): This value is undefined and MUST be ignored.

lcbPlcfFldMcr (4 bytes): This value MUST be 0, and MUST be ignored.

fcSttbfBkmk (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **SttbfBkmk** that contains the names of the **bookmarks** in the document begins at this offset. If **lcbSttbfBkmk** is zero, **fcSttbfBkmk** is undefined and MUST be ignored.

This **SttbfBkmk** is parallel to the **Plcfbkf** at offset **fcPlcfBkf** in the Table Stream. Each string specifies the name of the bookmark that is associated with the data element which is located at the same offset in that **Plcfbkf**. For this reason, the **SttbfBkmk** that begins at offset **fcSttbfBkmk**, and the **Plcfbkf** that begins at offset **fcPlcfBkf**, MUST contain the same number of elements.

lcbSttbfBkmk (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbfBkmk** at offset **fcSttbfBkmk**.

fcPlcfBkf (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfbkf** that contains information about the standard bookmarks in the document begins at this offset. If **lcbPlcfBkf** is zero, **fcPlcfBkf** is undefined and MUST be ignored.

Each data element in the **Plcfbkf** is associated, in a one-to-one correlation, with a data element in the **Plcfbkl** at offset **fcPlcfBkl**. For this reason, the **Plcfbkf** that begins at offset **fcPlcfBkf**, and the Plcfbkl that begins at offset **fcPlcfBkl**, MUST contain the same number of data elements. This **Plcfbkf** is parallel to the **SttbfBkmk** at offset **fcSttbfBkmk** in the Table Stream. Each data element in the **Plcfbkf** specifies information about the bookmark that is associated with the element which is located at the same offset in that **SttbfBkmk**. For this reason, the **Plcfbkf** that

Copyright © 2025 Microsoft Corporation

begins at offset **fcPlcfBkf**, and the **SttbfBkmk** that begins at offset **fcSttbfBkmk**, MUST contain the same number of elements.

The largest value that a CP marking the start or end of a standard bookmark is allowed to have is the CP representing the end of all document parts.

lcbPlcfBkf (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfbkf** at offset **fcPlcfBkf**.

fcPlcfBkl (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfbkl** that contains information about the standard bookmarks in the document begins at this offset. If **lcbPlcfBkl** is zero, **fcPlcfBkl** is undefined and MUST be ignored.

Each data element in the **Plcfbkl** is associated, in a one-to-one correlation, with a data element in the **Plcfbkf** at offset **fcPlcfBkf**. For this reason, the **Plcfbkl** that begins at offset **fcPlcfBkl**, and the **Plcfbkf** that begins at offset **fcPlcfBkf**, MUST contain the same number of data elements.

The largest value that a CP marking the start or end of a standard bookmark is allowed to have is the value of the CP representing the end of all document parts.

lcbPlcfBkl (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfbkl** at offset **fcPlcfBkl**.

fcCmds (4 bytes): An unsigned integer that specifies the offset in the Table Stream of a **Tcg** that specifies command-related customizations. If **lcbCmds** is zero, **fcCmds** is undefined and MUST be ignored.

lcbCmds (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Tcg** at offset **fcCmds**.

fcUnused1 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused1 (4 bytes): This value MUST be 0, and MUST be ignored.

fcSttbfMcr (4 bytes): This value is undefined and MUST be ignored.

lcbSttbfMcr (4 bytes): This value MUST be 0, and MUST be ignored.

fcPrDrvr (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The **PrDrvr**, which contains printer driver information (the names of drivers, port, and so on), begins at this offset. If **lcbPrDrvr** is zero, **fcPrDrvr** is undefined and MUST be ignored.

lcbPrDrvr (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PrDrvr** at offset **fcPrDrvr**.

fcPrEnvPort (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The **PrEnvPort** that is the print environment in portrait mode begins at this offset. If **lcbPrEnvPort** is zero, **fcPrEnvPort** is undefined and MUST be ignored.

lcbPrEnvPort (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PrEnvPort** at offset **fcPrEnvPort**.

fcPrEnvLand (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The **PrEnvLand** that is the print environment in landscape mode begins at this offset. If **lcbPrEnvLand** is zero, **fcPrEnvLand** is undefined and MUST be ignored.

lcbPrEnvLand (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PrEnvLand** at offset **fcPrEnvLand**.

Copyright © 2025 Microsoft Corporation

fcWss (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Selsf** begins at this offset and specifies the last selection that was made in the Main Document. If **lcbWss** is zero, **fcWss** is undefined and MUST be ignored.

lcbWss (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Selsf** at offset **fcWss**.

fcDop (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Dop** begins at this offset.

lcbDop (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Dop** at **fcDop**. This value MUST NOT be zero.

fcSttbfAssoc (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **SttbfAssoc** that contains strings that are associated with the document begins at this offset.

lcbSttbfAssoc (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbfAssoc** at offset **fcSttbfAssoc**. This value MUST NOT be zero.

fcClx (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Clx** begins at this offset.

lcbClx (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Clx** at offset **fcClx** in the Table Stream. This value MUST be greater than zero.

fcPlcfPgdFtn (4 bytes): This value is undefined and MUST be ignored.

lcbPlcfPgdFtn (4 bytes): This value MUST be 0, and MUST be ignored.

fcAutosaveSource (4 bytes): This value is undefined and MUST be ignored.

lcbAutosaveSource (4 bytes): This value MUST be 0 and MUST be ignored.

fcGrpXstAtnOwners (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An array of XSTs begins at this offset. The value of **cch** for all XSTs in this array MUST be less than 56. The number of entries in this array is limited to 0x7FFF. This array contains the names of authors of comments in the document. The names in this array MUST be unique. If no comments are defined, **lcbGrpXstAtnOwners** and **fcGrpXstAtnOwners** MUST be zero and MUST be ignored. If any comments are in the document, **fcGrpXstAtnOwners** MUST point to a valid array of XSTs.

lcbGrpXstAtnOwners (4 bytes): An unsigned integer that specifies the size, in bytes, of the XST array at offset **fcGrpXstAtnOwners** in the Table Stream.

fcSttbfAtnBkmk (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An SttbfAtnBkmk that contains information about the **annotation bookmarks** in the document begins at this offset. If **lcbSttbfAtnBkmk** is zero, **fcSttbfAtnBkmk** is undefined and MUST be ignored.

The SttbfAtnBkmk is parallel to the Plcfbkf at offset **fcPlcfAtnBkf** in the Table Stream. Each element in the SttbfAtnBkmk specifies information about the bookmark which is associated with the data element that is located at the same offset in that Plcfbkf, so the SttbfAtnBkmk beginning at offset **fcSttbfAtnBkmk** and the Plcfbkf beginning at offset **fcPlcfAtnBkf** MUST contain the same number of elements. An additional constraint upon the number of elements in the SttbfAtnBkmk is specified in the description of **fcPlcfAtnBkf**.

lcbSttbfAtnBkmk (4 bytes): An unsigned integer that specifies the size, in bytes, of the SttbfAtnBkmk at offset **fcSttbfAtnBkmk**.

fcUnused2 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused2 (4 bytes): This value MUST be 0, and MUST be ignored.

Copyright © 2025 Microsoft Corporation

fcUnused3 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused3 (4 bytes): This value MUST be 0, and MUST be ignored.

fcPlcSpaMom (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A PlcfSpa begins at this offset. The PlcfSpa contains shape information for the Main Document. All CPs in this PlcfSpa are relative to the starting position of the Main Document and MUST be greater than or equal to zero and less than or equal to **ccpText** in FibRgLw97. The final CP is undefined and MUST be ignored, though it MUST be greater than the previous entry. If there are no shapes in the Main Document, **lcbPlcSpaMom** and **fcPlcSpaMom** MUST be zero and MUST be ignored. If there are shapes in the Main Document, **fcPlcSpaMom** MUST point to a valid PlcfSpa structure.

lcbPlcSpaMom (4 bytes): An unsigned integer that specifies the size, in bytes, of the PlcfSpa at offset **fcPlcSpaMom**.

fcPlcSpaHdr (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A PlcfSpa begins at this offset. The PlcfSpa contains shape information for the Header Document. All CPs in this PlcfSpa are relative to the starting position of the Header Document and MUST be greater than or equal to zero and less than or equal to **ccpHdd** in FibRgLw97. The final CP is undefined and MUST be ignored, though this value MUST be greater than the previous entry. If there are no shapes in the Header Document, **lcbPlcSpaHdr** and **fcPlcSpaHdr** MUST both be zero and MUST be ignored. If there are shapes in the Header Document, **fcPlcSpaHdr** MUST point to a valid PlcfSpa structure.

lcbPlcSpaHdr (4 bytes): An unsigned integer that specifies the size, in bytes, of the PlcfSpa at the offset **fcPlcSpaHdr**.

fcPlcfAtnBkf (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfbkf** that contains information about annotation bookmarks in the document begins at this offset. If **lcbPlcfAtnBkf** is zero, **fcPlcfAtnBkf** is undefined and MUST be ignored.

Each data element in the **Plcfbkf** is associated, in a one-to-one correlation, with a data element in the **Plcfbkl** at offset **fcPlcfAtnBkl**. For this reason, the **Plcfbkf** that begins at offset **fcPlcfAtnBkf**, and the **Plcfbkl** that begins at offset **fcPlcfAtnBkl**, MUST contain the same number of data elements. The **Plcfbkf** is parallel to the **SttbfAtnBkmk** at offset **fcSttbfAtnBkmk** in the Table Stream. Each data element in the **Plcfbkf** specifies information about the bookmark which is associated with the element that is located at the same offset in that SttbfAtnBkmk. For this reason, the **Plcfbkf** that begins at offset **fcPlcfAtnBkf**, and the SttbfAtnBkmk that begins at offset **fcSttbfAtnBkmk**, MUST contain the same number of elements.

The CP range of an annotation bookmark MUST be in the Main Document part.

lcbPlcfAtnBkf (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfbkf** at offset **fcPlcfAtnBkf**.

fcPlcfAtnBkl (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfbkl** that contains information about annotation bookmarks in the document begins at this offset. If **lcbPlcfAtnBkl** is zero, then **fcPlcfAtnBkl** is undefined and MUST be ignored.

Each data element in the **Plcfbkl** is associated, in a one-to-one correlation, with a data element in the **Plcfbkf** at offset **fcPlcfAtnBkf**. For this reason, the **Plcfbkl** that begins at offset **fcPlcfAtnBkl**, and the **Plcfbkf** that begins at offset **fcPlcfAtnBkf**, MUST contain the same number of data elements.

The CP range of an annotation bookmark MUST be in the Main Document part.

lcbPlcfAtnBkl (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfbkl** at offset **fcPlcfAtnBkl**.

Copyright © 2025 Microsoft Corporation

fcPms (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Pms**, which contains the current state of a print merge operation, begins at this offset. If **lcbPms** is zero, **fcPms** is undefined and MUST be ignored.

lcbPms (4 bytes): An unsigned integer which specifies the size, in bytes, of the **Pms** at offset **fcPms**.

fcFormFldSttbs (4 bytes): This value is undefined and MUST be ignored.

lcbFormFldSttbs (4 bytes): This value MUST be 0, and MUST be ignored.

fcPlcfendRef (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfendRef** that begins at this offset specifies the locations of endnote references in the Main Document and whether those references use auto-numbering or custom symbols. If **lcbPlcfendRef** is zero, **fcPlcfendRef** is undefined and MUST be ignored.

lcbPlcfendRef (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfendRef** that begins at offset **fcPlcfendRef** in the Table Stream.

fcPlcfendTxt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfendTxt** begins at this offset and specifies the locations of each block of endnote text in the Endnote Document. If **lcbPlcfendTxt** is zero, **fcPlcfendTxt** is undefined and MUST be ignored.

lcbPlcfendTxt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfendTxt** that begins at offset **fcPlcfendTxt** in the Table Stream.

lcbPlcfendTxt MUST be zero if **FibRgLw97.ccpEdn** is zero, and MUST be nonzero if **FibRgLw97.ccpEdn** is nonzero.

fcPlcfFldEdn (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfld** begins at this offset and specifies the locations of field characters in the Endnote Document. All CPs in this **Plcfld** are relative to the starting position of the Endnote Document. All CPs in this **Plcfld** MUST be greater than or equal to zero and less than or equal to **FibRgLw97.ccpEdn**. If **lcbPlcfFldEdn** is zero, **fcPlcfFldEdn** is undefined and MUST be ignored.

lcbPlcfFldEdn (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfld** at offset **fcPlcfFldEdn** in the Table Stream.

fcUnused4 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused4 (4 bytes): This value MUST be 0, and MUST be ignored.

fcDggInfo (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An OfficeArtContent that contains information about the drawings in the document begins at this offset.

lcbDggInfo (4 bytes): An unsigned integer that specifies the size, in bytes, of the OfficeArtContent at the offset **fcDggInfo**. If **lcbDggInfo** is zero, there MUST NOT be any drawings in the document.

fcSttbfRMark (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **SttbfRMark** that contains the names of authors who have added revision marks or comments to the document begins at this offset. If **lcbSttbfRMark** is zero, **fcSttbfRMark** is undefined and MUST be ignored.

lcbSttbfRMark (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbfRMark** at the offset **fcSttbfRMark**.

fcSttbfCaption (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An SttbfCaption that contains information about the **captions** that are defined in this **document**

Copyright © 2025 Microsoft Corporation

begins at this offset. If **lcbSttbfCaption** is zero, **fcSttbfCaption** is undefined and MUST be ignored. If this document is not the **Normal template**, this value MUST be ignored.

lcbSttbfCaption (4 bytes): An unsigned integer that specifies the size, in bytes, of the SttbfCaption at offset **fcSttbfCaption** in the Table Stream. If **base.fDot** of the Fib that contains this FibRgFcLcb97 is zero, this value MUST be 0.

fcSttbfAutoCaption (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A SttbfAutoCaption that contains information about the **AutoCaption** strings defined in this document begins at this offset. If **lcbSttbfAutoCaption** is zero, **fcSttbfAutoCaption** is undefined and MUST be ignored. If this document is not the Normal template, this value MUST be ignored.

lcbSttbfAutoCaption (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbfAutoCaption** at offset **fcSttbfAutoCaption** in the Table Stream. If **base.fDot** of the **Fib** that contains this **FibRgFcLcb97** is zero, this MUST be zero.

fcPlcfWkb (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfWKB** that contains information about all master documents and subdocuments begins at this offset.

lcbPlcfWkb (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfWKB** at offset **fcPlcfWkb** in the Table Stream. If **lcbPlcfWkb** is zero, **fcPlcfWkb** is undefined and MUST be ignored.

fcPlcfSpl (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfspl**, which specifies the state of the spell checker for each text range, begins at this offset. If **lcbPlcfSpl** is zero, then **fcPlcfSpl** is undefined and MUST be ignored.

lcbPlcfSpl (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfspl** that begins at offset **fcPlcfSpl** in the Table Stream.

fcPlcftxbxTxt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcftxbxTxt** begins at this offset and specifies which ranges of text are contained in which textboxes. If **lcbPlcftxbxTxt** is zero, **fcPlcftxbxTxt** is undefined and MUST be ignored.

lcbPlcftxbxTxt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcftxbxTxt** that begins at offset **fcPlcftxbxTxt** in the Table Stream.

lcbPlcftxbxTxt MUST be zero if **FibRgLw97.ccpTxbx** is zero, and MUST be nonzero if **FibRgLw97.ccpTxbx** is nonzero.

fcPlcfFldTxbx (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfld** begins at this offset and specifies the locations of field characters in the Textbox Document. All CPs in this **Plcfld** are relative to the starting position of the Textbox Document. All CPs in this **Plcfld** MUST be greater than or equal to zero and less than or equal to **FibRgLw97.ccpTxbx**. If **lcbPlcfFldTxbx** is zero, **fcPlcfFldTxbx** is undefined and MUST be ignored.

lcbPlcfFldTxbx (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfld** at offset **fcPlcfFldTxbx** in the Table Stream.

fcPlcfHdrtxbxTxt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfHdrtxbxTxt** begins at this offset and specifies which ranges of text are contained in which header textboxes.

lcbPlcfHdrtxbxTxt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfHdrtxbxTxt** that begins at offset **fcPlcfHdrtxbxTxt** in the Table Stream.

lcbPlcfHdrtxbxTxt MUST be zero if **FibRgLw97.ccpHdrTxbx** is zero, and MUST be nonzero if **FibRgLw97.ccpHdrTxbx** is nonzero.

fcPlcffldHdrTxbx (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfld** begins at this offset and specifies the locations of field characters in the Header Textbox

Copyright © 2025 Microsoft Corporation

Document. All CPs in this **Plcfld** are relative to the starting position of the Header Textbox Document. All CPs in this **Plcfld** MUST be greater than or equal to zero and less than or equal to **FibRgLw97.ccpHdrTxbx**. If **lcbPlcffldHdrTxbx** is zero, **fcPlcffldHdrTxbx** is undefined, and MUST be ignored.

lcbPlcffldHdrTxbx (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfld** at offset **fcPlcffldHdrTxbx** in the Table Stream.

fcStwUser (4 bytes): An unsigned integer that specifies an offset into the Table Stream. An **StwUser** that specifies the user-defined variables and **VBA digital signature**, as specified by [MS-OSHARED] section 2.3.2, begins at this offset. If **lcbStwUser** is zero, **fcStwUser** is undefined and MUST be ignored.

lcbStwUser (4 bytes): An unsigned integer that specifies the size, in bytes, of the **StwUser** at offset **fcStwUser**.

fcSttbTtmbd (4 bytes): An unsigned integer that specifies an offset into the Table Stream. A **SttbTtmbd** begins at this offset and specifies information about the **TrueType font**s that are embedded in the document. If **lcbSttbTtmbd** is zero, **fcSttbTtmbd** is undefined and MUST be ignored.

lcbSttbTtmbd (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbTtmbd** at offset **fcSttbTtmbd**.

fcCookieData (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **RgCdb** begins at this offset. If **lcbCookieData** is zero, **fcCookieData** is undefined and MUST be ignored. Otherwise, **fcCookieData** MAY<28> be ignored.

lcbCookieData (4 bytes): An unsigned integer that specifies the size, in bytes, of the **RgCdb** at offset **fcCookieData** in the Table Stream.

fcPgdMotherOldOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated document page layout cache begins at this offset. Information SHOULD NOT<29> be emitted at this offset and SHOULD<30> be ignored. If **lcbPgdMotherOldOld** is zero, **fcPgdMotherOldOld** is undefined and MUST be ignored.

lcbPgdMotherOldOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated document page layout cache at offset **fcPgdMotherOldOld** in the Table Stream.

fcBkdMotherOldOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated document text flow break cache begins at this offset. Information SHOULD NOT<31> be emitted at this offset and SHOULD<32> be ignored. If **lcbBkdMotherOldOld** is zero, **fcBkdMotherOldOld** is undefined and MUST be ignored.

lcbBkdMotherOldOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated document text flow break cache at offset **fcBkdMotherOldOld** in the Table Stream.

fcPgdFtnOldOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated footnote layout cache begins at this offset. Information SHOULD NOT<33> be emitted at this offset and SHOULD<34> be ignored. If **lcbPgdFtnOldOld** is zero, **fcPgdFtnOldOld** is undefined and MUST be ignored.

lcbPgdFtnOldOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated footnote layout cache at offset **fcPgdFtnOldOld** in the Table Stream.

fcBkdFtnOldOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated footnote text flow break cache begins at this offset. Information SHOULD NOT<35> be emitted at this offset and SHOULD<36> be ignored. If **lcbBkdFtnOldOld** is zero, **fcBkdFtnOldOld** is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

lcbBkdFtnOldOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated footnote text flow break cache at offset **fcBkdFtnOldOld** in the Table Stream.

fcPgdEdnOldOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated endnote layout cache begins at this offset. Information SHOULD NOT<37> be emitted at this offset and SHOULD<38> be ignored. If **lcbPgdEdnOldOld** is zero, **fcPgdEdnOldOld** is undefined and MUST be ignored.

lcbPgdEdnOldOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated endnote layout cache at offset **fcPgdEdnOldOld** in the Table Stream.

fcBkdEdnOldOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated endnote text flow break cache begins at this offset. Information SHOULD NOT<39> be emitted at this offset and SHOULD<40> be ignored. If **lcbBkdEdnOldOld** is zero, **fcBkdEdnOldOld** is undefined and MUST be ignored.

lcbBkdEdnOldOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated endnote text flow break cache at offset **fcBkdEdnOldOld** in the Table Stream.

fcSttbfIntlFld (4 bytes): This value is undefined and MUST be ignored.

lcbSttbfIntlFld (4 bytes): This value MUST be 0, and MUST be ignored.

fcRouteSlip (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **RouteSlip** that specifies the route slip for this document begins at this offset. This value SHOULD<41> be ignored.

lcbRouteSlip (4 bytes): An unsigned integer that specifies the size, in bytes, of the **RouteSlip** at offset **fcRouteSlip** in the Table Stream.

fcSttbSavedBy (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **SttbSavedBy** that specifies the save history of this document begins at this offset. This value SHOULD<42> be ignored.

lcbSttbSavedBy (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbSavedBy** at the offset **fcSttbSavedBy**. This value SHOULD<43> be zero.

fcSttbFnm (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **SttbFnm** that contains information about the external files that are referenced by this document begins at this offset. If **lcbSttbFnm** is zero, **fcSttbFnm** is undefined and MUST be ignored.

lcbSttbFnm (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbFnm** at the offset **fcSttbFnm**.

fcPlfLst (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlfLst** that contains list formatting information begins at this offset. An array of **LVL**s is appended to the **PlfLst**. **lcbPlfLst** does not account for the array of **LVL**s. The size of the array of **LVL**s is specified by the **LSTF**s in **PlfLst**. For each **LSTF** whose **fSimpleList** is set to 0x1, there is one **LVL** in the array of **LVL**s that specifies the level formatting of the single level in the list which corresponds to the **LSTF**. And, for each **LSTF** whose **fSimpleList** is set to 0x0, there are 9 **LVL**s in the array of **LVL**s that specify the level formatting of the respective levels in the list which corresponds to the **LSTF**. This array of **LVL**s is in the same respective order as the **LSTF**s in **PlfLst**. If **lcbPlfLst** is 0, **fcPlfLst** is undefined and MUST be ignored.

lcbPlfLst (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlfLst** at the offset **fcPlfLst**. This does not include the size of the array of **LVL**s that are appended to the **PlfLst**.

fcPlfLfo (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlfLfo** that contains list formatting override information begins at this offset. If **lcbPlfLfo** is zero, **fcPlfLfo** is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

lcbPlfLfo (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlfLfo** at the offset **fcPlfLfo**.

fcPlcfTxbxBkd (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfTxbxBkd** begins at this offset and specifies which ranges of text go inside which textboxes.

lcbPlcfTxbxBkd (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfTxbxBkd** that begins at offset **fcPlcfTxbxBkd** in the Table Stream.

lcbPlcfTxbxBkd MUST be zero if **FibRgLw97.ccpTxbx** is zero, and MUST be nonzero if **FibRgLw97.ccpTxbx** is nonzero.

fcPlcfTxbxHdrBkd (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfTxbxHdrBkd** begins at this offset and specifies which ranges of text are contained inside which header textboxes.

lcbPlcfTxbxHdrBkd (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfTxbxHdrBkd** that begins at offset **fcPlcfTxbxHdrBkd** in the Table Stream.

lcbPlcfTxbxHdrBkd MUST be zero if **FibRgLw97.ccpHdrTxbx** is zero, and MUST be nonzero if **FibRgLw97.ccpHdrTxbx** is nonzero.

fcDocUndoWord9 (4 bytes): An unsigned integer that specifies an offset in the WordDocument Stream. Version-specific undo information begins at this offset. This information SHOULD NOT<44> be emitted and SHOULD<45> be ignored.

lcbDocUndoWord9 (4 bytes): An unsigned integer. If this is nonzero, version-specific undo information exists at offset **fcDocUndoWord9** in the WordDocument Stream.

fcRgbUse (4 bytes): An unsigned integer that specifies an offset in the WordDocument Stream. Version-specific undo information begins at this offset. This information SHOULD NOT<46> be emitted and SHOULD<47> be ignored.

lcbRgbUse (4 bytes): An unsigned integer that specifies the size, in bytes, of the version-specific undo information at offset **fcRgbUse** in the WordDocument Stream.

fcUsp (4 bytes): An unsigned integer that specifies an offset in the WordDocument Stream. Version-specific undo information begins at this offset. This information SHOULD NOT<48> be emitted and SHOULD<49> be ignored.

lcbUsp (4 bytes): An unsigned integer that specifies the size, in bytes, of the version-specific undo information at offset **fcUsp** in the WordDocument Stream.

fcUskf (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Version-specific undo information begins at this offset. This information SHOULD NOT<50> be emitted and SHOULD<51> be ignored.

lcbUskf (4 bytes): An unsigned integer that specifies the size, in bytes, of the version-specific undo information at offset **fcUskf** in the Table Stream.

fcPlcupcRgbUse (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plc** begins at this offset and contains version-specific undo information. This information SHOULD NOT<52> be emitted and SHOULD<53> be ignored.

lcbPlcupcRgbUse (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plc** at offset **fcPlcupcRgbUse** in the Table Stream.

fcPlcupcUsp (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plc** begins at this offset and contains version-specific undo information. This information SHOULD NOT<54> be emitted and SHOULD<55> be ignored.

Copyright © 2025 Microsoft Corporation

lcbPlcupcUsp (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plc** at offset **fcPlcupcUsp** in the Table Stream.

fcSttbGlsyStyle (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **SttbGlsyStyle**, which contains information about the **style**s that are used by the AutoText items which are defined in this document, begins at this offset.

lcbSttbGlsyStyle (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbGlsyStyle** at offset **fcSttbGlsyStyle** in the Table Stream. If **base.fGlsy** of the **Fib** that contains this **FibRgFcLcb97** is zero, this value MUST be 0.

fcPlgosl (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlfGosl** begins at the offset. If **lcbPlgosl** is zero, **fcPlgosl** is undefined and MUST be ignored.

lcbPlgosl (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlfGosl** at offset **fcPlgosl** in the Table Stream.

fcPlcocx (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **RgxOcxInfo** that specifies information about the **OLE controls** in the document begins at this offset. When there are no OLE controls in the document, **fcPlcocx** and **lcbPlcocx** MUST be zero and MUST be ignored. If there are any OLE controls in the document, **fcPlcocx** MUST point to a valid **RgxOcxInfo**.

lcbPlcocx (4 bytes): An unsigned integer that specifies the size, in bytes, of the **RgxOcxInfo** at the offset **fcPlcocx**.

fcPlcfBteLvc (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A deprecated numbering field cache begins at this offset. This information SHOULD NOT<56> be emitted and SHOULD<57> ignored. If **lcbPlcfBteLvc** is zero, **fcPlcfBteLvc** is undefined and MUST be ignored.

lcbPlcfBteLvc (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated numbering field cache at offset **fcPlcfBteLvc** in the Table Stream. This value SHOULD<58> be zero.

dwLowDateTime (4 bytes): The low-order part of a **FILETIME** structure, as specified by [MS-DTYP], that specifies when the document was last saved.

dwHighDateTime (4 bytes): The high-order part of a **FILETIME** structure, as specified by [MS-DTYP], that specifies when the document was last saved.

fcPlcfLvcPre10 (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated list level cache begins at this offset. Information SHOULD NOT<59> be emitted at this offset and SHOULD<60> be ignored. If **lcbPlcfLvcPre10** is zero, **fcPlcfLvcPre10** is undefined and MUST be ignored.

lcbPlcfLvcPre10 (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated list level cache at offset **fcPlcfLvcPre10** in the Table Stream. This value SHOULD<61> be zero.

fcPlcfAsumy (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfAsumy** begins at the offset. If **lcbPlcfAsumy** is zero, **fcPlcfAsumy** is undefined and MUST be ignored.

lcbPlcfAsumy (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfAsumy** at offset **fcPlcfAsumy** in the Table Stream.

fcPlcfGram (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfgram**, which specifies the state of the grammar checker for each text range, begins at this offset. If **lcbPlcfGram** is zero, then **fcPlcfGram** is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

lcbPlcfGram (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfgram** that begins at offset **fcPlcfGram** in the Table Stream.

fcSttbListNames (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **SttbListNames**, which specifies the **LISTNUM** field names of the lists in the document, begins at this offset. If **lcbSttbListNames** is zero, **fcSttbListNames** is undefined and MUST be ignored.

lcbSttbListNames (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbListNames** at the offset **fcSttbListNames**.

fcSttbfUssr (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated, version-specific undo information begins at this offset. This information SHOULD NOT<62> be emitted and SHOULD<63> be ignored.

lcbSttbfUssr (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated, version-specific undo information at offset **fcSttbfUssr** in the Table Stream.

2.5.7 FibRgFcLcb2000

The **FibRgFcLcb2000** structure is a variable-sized portion of the Fib. It extends the FibRgFcLcb97.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
rgFcLcb97 (744 bytes)																															
...																															
...																															
fcPlcfTch																															
lcbPlcfTch																															
fcRmdThreading																															
lcbRmdThreading																															
fcMid																															
lcbMid																															
fcSttbRgtplc																															
lcbSttbRgtplc																															
fcMsoEnvelope																															
lcbMsoEnvelope																															
fcPlcfLad																															
lcbPlcfLad																															

Copyright © 2025 Microsoft Corporation

fcRgDofr
lcbRgDofr
fcPlcosl
lcbPlcosl
fcPlcfCookieOld
lcbPlcfCookieOld
fcPgdMotherOld
lcbPgdMotherOld
fcBkdMotherOld
lcbBkdMotherOld
fcPgdFtnOld
lcbPgdFtnOld
fcBkdFtnOld
lcbBkdFtnOld
fcPgdEdnOld
lcbPgdEdnOld
fcBkdEdnOld
lcbBkdEdnOld

rgFcLcb97 (744 bytes): The contained **FibRgFcLcb97**.

fcPlcfTch (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfTch** begins at this offset and specifies a cache of table characters. Information at this offset SHOULD<64> be ignored. If **lcbPlcfTch** is zero, **fcPlcfTch** is undefined and MUST be ignored.

lcbPlcfTch (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfTch** at offset **fcPlcfTch**.

fcRmdThreading (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **RmdThreading** that specifies the data concerning the e-mail messages and their authors in this document begins at this offset.

lcbRmdThreading (4 bytes): An unsigned integer that specifies the size, in bytes, of the **RmdThreading** at the offset **fcRmdThreading**. This value MUST NOT be zero.

Copyright © 2025 Microsoft Corporation

fcMid (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A double-byte character **Unicode** string that specifies the **message identifier** of the document begins at this offset. This value MUST be ignored.

lcbMid (4 bytes): An unsigned integer that specifies the size, in bytes, of the double-byte character Unicode string at offset **fcMid**. This value MUST be ignored.

fcSttbRgtplc (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **SttbRgtplc** that specifies the styles of lists in the document begins at this offset. If **lcbSttbRgtplc** is zero, **fcSttbRgtplc** is undefined and MUST be ignored.

lcbSttbRgtplc (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbRgtplc** at the offset **fcSttbRgtplc**.

fcMsoEnvelope (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **MsoEnvelopeCLSID**, which specifies the envelope data as specified by [MS-OSHARED] section 2.3.8.1, begins at this offset. If **lcbMsoEnvelope** is zero, **fcMsoEnvelope** is undefined and MUST be ignored.

lcbMsoEnvelope (4 bytes): An unsigned integer that specifies the size, in bytes, of the **MsoEnvelopeCLSID** at the offset **fcMsoEnvelope**.

fcPlcfLad (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcflad** begins at this offset and specifies the language auto-detect state of each text range. If **lcbPlcfLad** is zero, **fcPlcfLad** is undefined and MUST be ignored.

lcbPlcfLad (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcflad** that begins at offset **fcPlcfLad** in the Table Stream.

fcRgDofr (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A variable-length array with elements of type **Dofrh** begins at that offset. The elements of this array are records that support the frame set and list style features. If **lcbRgDofr** is zero, **fcRgDofr** is undefined and MUST be ignored.

lcbRgDofr (4 bytes): An unsigned integer that specifies the size, in bytes, of the array that begins at offset **fcRgDofr** in the Table Stream.

fcPlcosl (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlfCosl** begins at the offset. If **lcbPlcosl** is zero, **fcPlcosl** is undefined and MUST be ignored.

lcbPlcosl (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlfCosl** at offset **fcPlcosl** in the Table Stream.

fcPlcfCookieOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlcfcookieOld** begins at this offset. If **lcbPlcfcookieOld** is zero, **fcPlcfcookieOld** is undefined and MUST be ignored. **fcPlcfcookieOld** MAY<65> be ignored.

lcbPlcfCookieOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcfcookieOld** at offset **fcPlcfcookieOld** in the Table Stream.

fcPgdMotherOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated document page layout cache begins at this offset. Information SHOULD NOT<66> be emitted at this offset and SHOULD<67> be ignored. If **lcbPgdMotherOld** is zero, **fcPgdMotherOld** is undefined and MUST be ignored.

lcbPgdMotherOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated document page layout cache at offset **fcPgdMotherOld** in the Table Stream.

fcBkdMotherOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated document text flow break cache begins at this offset. Information SHOULD NOT<68> be

Copyright © 2025 Microsoft Corporation

emitted at this offset and SHOULD[<69>] be ignored. If **lcbBkdMotherOld** is zero, **fcBkdMotherOld** is undefined and MUST be ignored.

lcbBkdMotherOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated document text flow break cache at offset **fcBkdMotherOld** in the Table Stream.

fcPgdFtnOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated footnote layout cache begins at this offset. Information SHOULD NOT[<70>] be emitted at this offset and SHOULD[<71>] be ignored. If **lcbPgdFtnOld** is zero, **fcPgdFtnOld** is undefined and MUST be ignored.

lcbPgdFtnOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated footnote layout cache at offset **fcPgdFtnOld** in the Table Stream.

fcBkdFtnOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated footnote text flow break cache begins at this offset. Information SHOULD NOT[<72>] be emitted at this offset and SHOULD[<73>] be ignored. If **lcbBkdFtnOld** is zero, **fcBkdFtnOld** is undefined and MUST be ignored.

lcbBkdFtnOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated footnote text flow break cache at offset **fcBkdFtnOld** in the Table Stream.

fcPgdEdnOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated endnote layout cache begins at this offset. Information SHOULD NOT[<74>] be emitted at this offset and SHOULD[<75>] be ignored. If **lcbPgdEdnOld** is zero, **fcPgdEdnOld** is undefined and MUST be ignored.

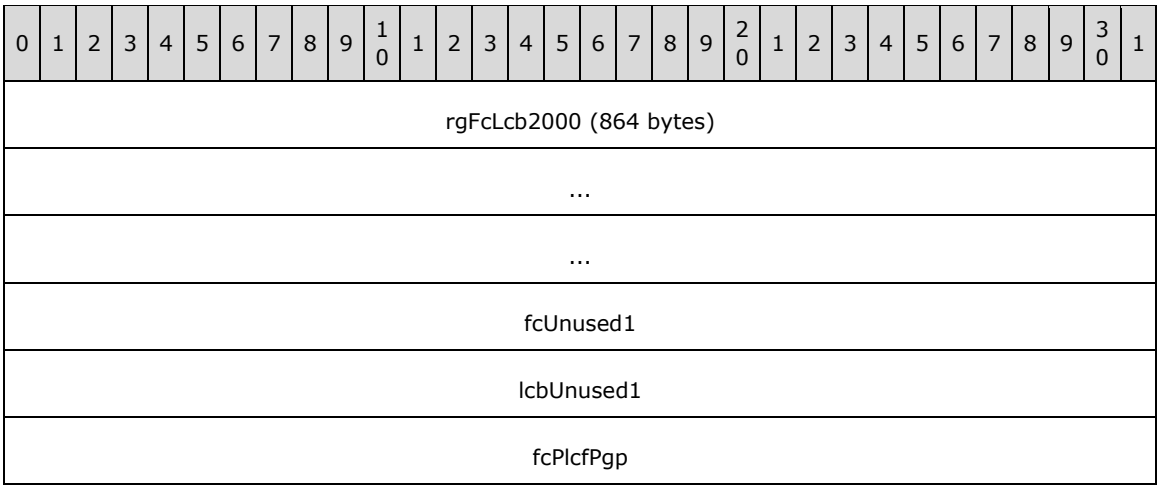
lcbPgdEdnOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated endnote layout cache at offset **fcPgdEdnOld** in the Table Stream.

fcBkdEdnOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated endnote text flow break cache begins at this offset. Information SHOULD NOT[<76>] be emitted at this offset and SHOULD[<77>] be ignored. If **lcbBkdEdnOld** is zero, **fcBkdEdnOld** is undefined and MUST be ignored.

lcbBkdEdnOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated endnote text flow break cache at offset **fcBkdEdnOld** in the Table Stream.

2.5.8 FibRgFcLcb2002

The **FibRgFcLcb2002** structure is a variable-sized portion of the Fib. It extends the FibRgFcLcb2000.



Copyright © 2025 Microsoft Corporation

lcbPlcfPgp
fcPlcfuim
lcbPlcfuim
fcPlfguidUim
lcbPlfguidUim
fcAtrdExtra
lcbAtrdExtra
fcPlrsid
lcbPlrsid
fcSttbfBkmkFactoid
lcbSttbfBkmkFactoid
fcPlcfBkfFactoid
lcbPlcfBkfFactoid
fcPlcfcookie
lcbPlcfcookie
fcPlcfBklFactoid
lcbPlcfBklFactoid
fcFactoidData
lcbFactoidData
fcDocUndo
lcbDocUndo
fcSttbfBkmkFcc
lcbSttbfBkmkFcc
fcPlcfBkfFcc
lcbPlcfBkfFcc

Copyright © 2025 Microsoft Corporation

fcPlcfBklFcc
lcbPlcfBklFcc
fcSttbfbkmkBPRepairs
lcbSttbfbkmkBPRepairs
fcPlcfbkfBPRepairs
lcbPlcfbkfBPRepairs
fcPlcfbklBPRepairs
lcbPlcfbklBPRepairs
fcPmsNew
lcbPmsNew
fcODSO
lcbODSO
fcPlcfpmiOldXP
lcbPlcfpmiOldXP
fcPlcfpmiNewXP
lcbPlcfpmiNewXP
fcPlcfpmiMixedXP
lcbPlcfpmiMixedXP
fcUnused2
lcbUnused2
fcPlcffactoid
lcbPlcffactoid
fcPlcflvcOldXP
lcbPlcflvcOldXP
fcPlcflvcNewXP

Copyright © 2025 Microsoft Corporation

lcbPlcflvcNewXP
fcPlcflvcMixedXP
lcbPlcflvcMixedXP

rgFcLcb2000 (864 bytes): The contained **FibRgFcLcb2000**.

fcUnused1 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused1 (4 bytes): This value MUST be 0, and MUST be ignored

fcPlcfPgp (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PGPArray** begins at this offset. If **lcbPlcfPgp** is 0, **fcPlcfPgp** is undefined and MUST be ignored.

lcbPlcfPgp (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PGPArray** that is stored at offset **fcPlcfPgp**.

fcPlcfuim (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfuim** begins at this offset. If **lcbPlcfuim** is zero, **fcPlcfuim** is undefined and MUST be ignored.

lcbPlcfuim (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfuim** at offset **fcPlcfuim**.

fcPlfguidUim (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PlfguidUim** begins at this offset. If **lcbPlfguidUim** is zero, **fcPlfguidUim** is undefined and MUST be ignored.

lcbPlfguidUim (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlfguidUim** at offset **fcPlfguidUim.**

fcAtrdExtra (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **AtrdExtra** begins at this offset. If **lcbAtrdExtra** is zero, **fcAtrdExtra** is undefined and MUST be ignored.

lcbAtrdExtra (4 bytes): An unsigned integer that specifies the size, in bytes, of the **AtrdExtra** at offset **fcAtrdExtra** in the Table Stream.

fcPlrsid (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **PLRSID** begins at this offset. If **lcbPlrsid** is zero, **fcPlrsid** is undefined and MUST be ignored.

lcbPlrsid (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PLRSID** at offset **fcPlrsid** in the Table Stream.

fcSttbfBkmkFactoid (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **SttbfBkmkFactoid** containing information about **smart tag bookmarks** in the document begins at this offset. If **lcbSttbfBkmkFactoid** is zero, **fcSttbfBkmkFactoid** is undefined and MUST be ignored.

The **SttbfBkmkFactoid** is parallel to the **Plcfbkfd** at offset **fcPlcfBkfFactoid** in the Table Stream. Each element in the **SttbfBkmkFactoid** specifies information about the **bookmark** that is associated with the data element which is located at the same offset in that **Plcfbkfd**. For this reason, the **SttbfBkmkFactoid** that begins at offset **fcSttbfBkmkFactoid**, and the **Plcfbkfd** that begins at offset **fcPlcfBkfFactoid**, MUST contain the same number of elements.

lcbSttbfBkmkFactoid (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbfBkmkFactoid** at offset **fcSttbfBkmkFactoid**.

Copyright © 2025 Microsoft Corporation

fcPlcfBkfFactoid (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfbkfd** that contains information about the smart tag bookmarks in the document begins at this offset. If **lcbPlcfBkfFactoid** is zero, **fcPlcfBkfFactoid** is undefined and MUST be ignored.

Each data element in the **Plcfbkfd** is associated, in a one-to-one correlation, with a data element in the **Plcfbkld** at offset **fcPlcfBklFactoid**. For this reason, the **Plcfbkfd** that begins at offset **fcPlcfBkfFactoid**, and the **Plcfbkld** that begins at offset **fcPlcfBklFactoid**, MUST contain the same number of data elements. The **Plcfbkfd** is parallel to the **SttbfBkmkFactoid** at offset **fcSttbfBkmkFactoid** in the Table Stream. Each data element in the **Plcfbkfd** specifies information about the bookmark that is associated with the element which is located at the same offset in that **SttbfBkmkFactoid**. For this reason, the **Plcfbkfd** that begins at offset **fcPlcfBkfFactoid**, and the **SttbfBkmkFactoid** that begins at offset **fcSttbfBkmkFactoid**, MUST contain the same number of elements.

lcbPlcfBkfFactoid (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfbkfd** at offset **fcPlcfBkfFactoid**.

fcPlcfcookie (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfcookie** begins at this offset. If **lcbPlcfcookie** is zero, **fcPlcfcookie** is undefined and MUST be ignored. **fcPlcfcookie** MAY<78> be ignored.

lcbPlcfcookie (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfcookie** at offset **fcPlcfcookie** in the Table Stream.

fcPlcfBklFactoid (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfbkld** that contains information about the smart tag bookmarks in the document begins at this offset. If **lcbPlcfBklFactoid** is zero, **fcPlcfBklFactoid** is undefined and MUST be ignored.

Each data element in the **Plcfbkld** is associated, in a one-to-one correlation, with a data element in the **Plcfbkfd** at offset **fcPlcfBkfFactoid**. For this reason, the **Plcfbkld** that begins at offset **fcPlcfBklFactoid**, and the **Plcfbkfd** that begins at offset **fcPlcfBkfFactoid**, MUST contain the same number of data elements.

lcbPlcfBklFactoid (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfbkld** at offset **fcPlcfBklFactoid**.

fcFactoidData (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **SmartTagData** begins at this offset and specifies information about the **smart tag recognizers** that are used in this document. If **lcbFactoidData** is zero, **fcFactoidData** is undefined and MUST be ignored.

lcbFactoidData (4 bytes): An unsigned integer that specifies the size, in bytes, of the SmartTagData at offset **fcFactoidData** in the Table Stream.

fcDocUndo (4 bytes): An unsigned integer that specifies an offset in the WordDocument Stream. Version-specific undo information begins at this offset. This information SHOULD NOT<79> be emitted and SHOULD<80> be ignored.

lcbDocUndo (4 bytes): An unsigned integer. If this value is nonzero, version-specific undo information exists at offset **fcDocUndo** in the WordDocument Stream.

fcSttbfBkmkFcc (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **SttbfBkmkFcc** that contains information about the **format consistency-checker bookmarks** in the document begins at this offset. If **lcbSttbfBkmkFcc** is zero, **fcSttbfBkmkFcc** is undefined and MUST be ignored.

The **SttbfBkmkFcc** is parallel to the **Plcfbkfd** at offset **fcPlcfBkfFcc** in the Table Stream. Each element in the **SttbfBkmkFcc** specifies information about the bookmark that is associated with the data element which is located at the same offset in that **Plcfbkfd**. For this reason, the

Copyright © 2025 Microsoft Corporation

SttbfBkmkFcc that begins at offset **fcSttbfBkmkFcc**, and the **Plcfbkfd** that begins at offset **fcPlcfBkfFcc**, MUST contain the same number of elements.

lcbSttbfBkmkFcc (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbfBkmkFcc** at offset **fcSttbfBkmkFcc**.

fcPlcfBkfFcc (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfbkfd** that contains information about format consistency-checker bookmarks in the document begins at this offset. If **lcbPlcfBkfFcc** is zero, **fcPlcfBkfFcc** is undefined and MUST be ignored.

Each data element in the **Plcfbkfd** is associated, in a one-to-one correlation, with a data element in the **Plcfbkld** at offset **fcPlcfBklFcc**. For this reason, the **Plcfbkfd** that begins at offset **fcPlcfBkfFcc** and the **Plcfbkld** that begins at offset **fcPlcfBklFcc** MUST contain the same number of data elements. The **Plcfbkfd** is parallel to the **SttbfBkmkFcc** at offset **fcSttbfBkmkFcc** in the Table Stream. Each data element in the **Plcfbkfd** specifies information about the bookmark that is associated with the element which is located at the same offset in that **SttbfBkmkFcc**. For this reason, the **Plcfbkfd** that begins at offset **fcPlcfBkfFcc** and the **SttbfBkmkFcc** that begins at offset **fcSttbfBkmkFcc** MUST contain the same number of elements.

lcbPlcfBkfFcc (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfbkfd** at offset **fcPlcfBkfFcc**.

fcPlcfBklFcc (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfbkld** that contains information about the format consistency-checker bookmarks in the document begins at this offset. If **lcbPlcfBklFcc** is zero, **fcPlcfBklFcc** is undefined and MUST be ignored.

Each data element in the **Plcfbkld** is associated, in a one-to-one correlation, with a data element in the **Plcfbkfd** at offset **fcPlcfBkfFcc**. For this reason, the **Plcfbkld** that begins at offset **fcPlcfBklFcc**, and the **Plcfbkfd** that begins at offset **fcPlcfBkfFcc**, MUST contain the same number of data elements.

lcbPlcfBklFcc (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfbkld** at offset **fcPlcfBklFcc**.

fcSttbfbkmkBPRepairs (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **SttbfBkmkBPRepairs** that contains information about the **repair bookmarks** in the document begins at this offset. If **lcbSttbfBkmkBPRepairs** is zero, **fcSttbfBkmkBPRepairs** is undefined and MUST be ignored.

The **SttbfBkmkBPRepairs** is parallel to the **Plcfbkf** at offset **fcPlcfBkfBPRepairs** in the Table Stream. Each element in the **SttbfBkmkBPRepairs** specifies information about the bookmark that is associated with the data element which is located at the same offset in that **Plcfbkf**. For this reason, the **SttbfBkmkBPRepairs** that begins at offset **fcSttbfBkmkBPRepairs**, and the **Plcfbkf** that begins at offset **fcPlcfBkfBPRepairs**, MUST contain the same number of elements.

lcbSttbfbkmkBPRepairs (4 bytes): An unsigned integer that specifies the size, in bytes, of the SttbfBkmkBPRepairs at offset **fcSttbfBkmkBPRepairs**.

fcPlcfbkfBPRepairs (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfbkf** that contains information about the repair bookmarks in the document begins at this offset. If **lcbPlcfBkfBPRepairs** is zero, **fcPlcfBkfBPRepairs** is undefined and MUST be ignored.

Each data element in the **Plcfbkf** is associated, in a one-to-one correlation, with a data element in the **Plcfbkl** at offset **fcPlcfBklBPRepairs**. For this reason, the **Plcfbkf** that begins at offset **fcPlcfBkfBPRepairs**, and the **Plcfbkl** that begins at offset **fcPlcfBklBPRepairs**, MUST contain the same number of data elements. The **Plcfbkf** is parallel to the **SttbfBkmkBPRepairs** at offset **fcSttbfBkmkBPRepairs** in the Table Stream. Each data element in the Plcfbkf specifies information about the bookmark that is associated with the element which is located at the same offset in that **SttbfBkmkBPRepairs**. For this reason, the **Plcfbkf** that begins at offset

Copyright © 2025 Microsoft Corporation

fcPlcfbkfBPRepairs, and the **SttbfBkmkBPRepairs** that begins at offset **fcSttbfBkmkBPRepairs**, MUST contain the same number of elements.

The CPs in this **Plcfbkf** MUST NOT exceed the CP that represents the end of the Main Document part.

lcbPlcfbkfBPRepairs (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfbkf** at offset **fcPlcfbkfBPRepairs**.

fcPlcfbklBPRepairs (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcfbkl** that contains information about the repair bookmarks in the document begins at this offset. If **lcbPlcfBklBPRepairs** is zero, **fcPlcfBklBPRepairs** is undefined and MUST be ignored.

Each data element in the **Plcfbkl** is associated, in a one-to-one correlation, with a data element in the **Plcfbkf** at offset **fcPlcfBkfBPRepairs**. For this reason, the **Plcfbkl** that begins at offset **fcPlcfBklBPRepairs**, and the **Plcfbkf** that begins at offset **fcPlcfBkfBPRepairs**, MUST contain the same number of data elements.

The CPs that are contained in this **Plcfbkl** MUST NOT exceed the CP that represents the end of the Main Document part.

lcbPlcfbklBPRepairs (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcfbkl** at offset **fcPlcfBklBPRepairs**.

fcPmsNew (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A new **Pms**, which contains the current state of a print merge operation, begins at this offset. If **lcbPmsNew** is zero, **fcPmsNew** is undefined and MUST be ignored.

lcbPmsNew (4 bytes): An unsigned integer which specifies the size, in bytes, of the **Pms** at offset **fcPmsNew**.

fcODSO (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Office Data Source Object (ODSO) data that is used to perform mail merge begins at this offset. The data is stored in an array of **ODSOPropertyBase** items. The **ODSOPropertyBase** items are of variable size and are stored contiguously. The complete set of properties that are contained in the array is determined by reading each **ODSOPropertyBase**, until a total of **lcbODSO** bytes of data are read. If **lcbODSO** is zero, **fcODSO** is undefined and MUST be ignored.

lcbODSO (4 bytes): An unsigned integer that specifies the size, in bytes, of the Office Data Source Object data at offset **fcODSO** in the Table Stream.

fcPlcfpmiOldXP (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated paragraph mark information cache begins at this offset. Information SHOULD NOT<81> be emitted at this offset and SHOULD<82> be ignored. If **lcbPlcfpmiOldXP** is zero, **fcPlcfpmiOldXP** is undefined and MUST be ignored.

lcbPlcfpmiOldXP (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated paragraph mark information cache at offset **fcPlcfpmiOldXP** in the Table Stream. This value SHOULD<83> be zero.

fcPlcfpmiNewXP (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated paragraph mark information cache begins at this offset. Information SHOULD NOT<84> be emitted at this offset and SHOULD<85> be ignored. If **lcbPlcfpmiNewXP** is zero, **fcPlcfpmiNewXP** is undefined and MUST be ignored.

lcbPlcfpmiNewXP (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated paragraph mark information cache at offset **fcPlcfpmiNewXP** in the Table Stream. This value SHOULD<86> be zero.

Copyright © 2025 Microsoft Corporation

fcPlcfpmiMixedXP (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated paragraph mark information cache begins at this offset. Information SHOULD NOT<87> be emitted at this offset and SHOULD<88> be ignored. If **lcbPlcfpmiMixedXP** is zero, **fcPlcfpmiMixedXP** is undefined and MUST be ignored.

lcbPlcfpmiMixedXP (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated paragraph mark information cache at offset **fcPlcfpmiMixedXP** in the Table Stream. This value SHOULD<89> be zero.

fcUnused2 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused2 (4 bytes): This value MUST be 0, and MUST be ignored.

fcPlcffactoid (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcffactoid**, which specifies the smart tag recognizer state of each text range, begins at this offset. If **lcbPlcffactoid** is zero, **fcPlcffactoid** is undefined and MUST be ignored.

lcbPlcffactoid (4 bytes): An unsigned integer that specifies the size, in bytes of the **Plcffactoid** that begins at offset **fcPlcffactoid** in the Table Stream.

fcPlcflvcOldXP (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated **listnum** field cache begins at this offset. Information SHOULD NOT<90> be emitted at this offset and SHOULD<91> be ignored. If **lcbPlcflvcOldXP** is zero, **fcPlcflvcOldXP** is undefined and MUST be ignored.

lcbPlcflvcOldXP (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated **listnum** field cache at offset **fcPlcflvcOldXP** in the Table Stream. This value SHOULD<92> be zero.

fcPlcflvcNewXP (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated **listnum** field cache begins at this offset. Information SHOULD NOT<93> be emitted at this offset and SHOULD<94> be ignored. If **lcbPlcflvcNewXP** is zero, **fcPlcflvcNewXP** is undefined and MUST be ignored.

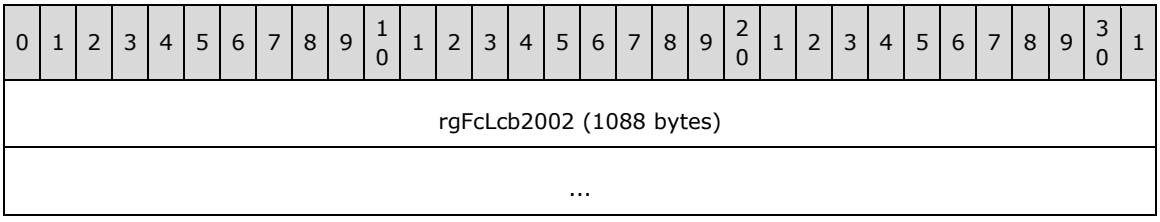
lcbPlcflvcNewXP (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated **listnum** field cache at offset **fcPlcflvcNewXP** in the Table Stream. This value SHOULD<95> be zero.

fcPlcflvcMixedXP (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated **listnum** field cache begins at this offset. Information SHOULD NOT<96> be emitted at this offset and SHOULD<97> be ignored. If **lcbPlcflvcMixedXP** is zero, **fcPlcflvcMixedXP** is undefined and MUST be ignored.

lcbPlcflvcMixedXP (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated **listnum** field cache at offset **fcPlcflvcMixedXP** in the Table Stream. This value SHOULD<98> be zero.

2.5.9 FibRgFcLcb2003

The **FibRgFcLcb2003** structure is a variable-sized portion of the Fib. It extends the FibRgFcLcb2002.



Copyright © 2025 Microsoft Corporation

...
fcHplxsdr
lcbHplxsdr
fcSttbfBkmkSdt
lcbSttbfBkmkSdt
fcPlcfBkfSdt
lcbPlcfBkfSdt
fcPlcfBklSdt
lcbPlcfBklSdt
fcCustomXForm
lcbCustomXForm
fcSttbfBkmkProt
lcbSttbfBkmkProt
fcPlcfBkfProt
lcbPlcfBkfProt
fcPlcfBklProt
lcbPlcfBklProt
fcSttbProtUser
lcbSttbProtUser
fcUnused
lcbUnused
fcPlcfpmiOld
lcbPlcfpmiOld
fcPlcfpmiOldInline
lcbPlcfpmiOldInline

Copyright © 2025 Microsoft Corporation

fcPlcfpmiNew
lcbPlcfpmiNew
fcPlcfpmiNewInline
lcbPlcfpmiNewInline
fcPlcflvcOld
lcbPlcflvcOld
fcPlcflvcOldInline
lcbPlcflvcOldInline
fcPlcflvcNew
lcbPlcflvcNew
fcPlcflvcNewInline
lcbPlcflvcNewInline
fcPgdMother
lcbPgdMother
fcBkdMother
lcbBkdMother
fcAfdMother
lcbAfdMother
fcPgdFtn
lcbPgdFtn
fcBkdFtn
lcbBkdFtn
fcAfdFtn
lcbAfdFtn
fcPgdEdn

Copyright © 2025 Microsoft Corporation

lcbPgdEdn
fcBkdEdn
lcbBkdEdn
fcAfdEdn
lcbAfdEdn
fcAfd
lcbAfd

rgFcLcb2002 (1088 bytes): The contained **FibRgFcLcb2002**.

fcHplxsdr (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **Hplxsdr** structure begins at this offset. This structure specifies information about **XML schema definition (XSD)** references.

lcbHplxsdr (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Hplxsdr** structure at the offset **fcHplxsdr** in the Table Stream. If **lcbHplxsdr** is zero, then **fcHplxsdr** is undefined and MUST be ignored.

fcSttbfBkmkSdt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **SttbfBkmkSdt** that contains information about the **structured document tag bookmarks** in the document begins at this offset. If **lcbSttbfBkmkSdt** is zero, then **fcSttbfBkmkSdt** is undefined and MUST be ignored.

The **SttbfBkmkSdt** is parallel to the **Plcbkfd** at offset **fcPlcfBkfSdt** in the Table Stream. Each element in the **SttbfBkmkSdt** specifies information about the **bookmark** that is associated with the data element which is located at the same offset in that **Plcbkfd**. For this reason, the **SttbfBkmkSdt** that begins at offset **fcSttbfBkmkSdt**, and the **Plcbkfd** that begins at offset **fcPlcfBkfSdt**, MUST contain the same number of elements.

lcbSttbfBkmkSdt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbfBkmkSdt** at offset **fcSttbfBkmkSdt**.

fcPlcfBkfSdt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcbkfd** that contains information about the structured document tag bookmarks in the document begins at this offset. If **lcbPlcfBkfSdt** is zero, **fcPlcfBkfSdt** is undefined and MUST be ignored.

Each data element in the **Plcbkfd** is associated, in a one-to-one correlation, with a data element in the **Plcbkld** at offset **fcPlcfBklSdt**. For this reason, the **Plcbkfd** that begins at offset **fcPlcfBkfSdt**, and the **Plcbkld** that begins at offset **fcPlcfBklSdt**, MUST contain the same number of data elements. The **Plcbkfd** is parallel to the **SttbfBkmkSdt** at offset **fcSttbfBkmkSdt** in the Table Stream. Each data element in the **Plcbkfd** specifies information about the bookmark that is associated with the element which is located at the same offset in that **SttbfBkmkSdt**. For this reason, the **Plcbkfd** that begins at offset **fcPlcfBkfSdt**, and the **SttbfBkmkSdt** that begins at offset **fcSttbfBkmkSdt**, MUST contain the same number of elements.

lcbPlcfBkfSdt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcbkfd** at offset **fcPlcfBkfSdt**.

Copyright © 2025 Microsoft Corporation

fcPlcfBklSdt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcbkld** that contains information about the structured document tag bookmarks in the document begins at this offset. If **lcbPlcfBklSdt** is zero, **fcPlcfBklSdt** is undefined and MUST be ignored.

Each data element in the **Plcbkld** is associated, in a one-to-one correlation, with a data element in the **Plcbkfd** at offset **fcPlcfBkfSdt**. For this reason, the **Plcbkld** that begins at offset **fcPlcfBklSdt**, and the **Plcbkfd** that begins at offset **fcPlcfBkfSdt** MUST contain the same number of data elements.

lcbPlcfBklSdt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcbkld** at offset **fcPlcfBklSdt**.

fcCustomXForm (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An array of 16-bit **Unicode** characters, which specifies the full path and file name of the **XML** Stylesheet to apply when saving this document in XML format, begins at this offset. If **lcbCustomXForm** is zero, **fcCustomXForm** is undefined and MUST be ignored.

lcbCustomXForm (4 bytes): An unsigned integer that specifies the size, in bytes, of the array at offset **fcCustomXForm** in the Table Stream. This value MUST be less than or equal to 4168 and MUST be evenly divisible by two.

fcSttbfBkmkProt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. An **SttbfBkmkProt** that contains information about **range-level protection bookmarks** in the document begins at this offset. If **lcbSttbfBkmkProt** is zero, **fcSttbfBkmkProt** is undefined and MUST be ignored.

The **SttbfBkmkProt** is parallel to the **Plcbkf** at offset **fcPlcfBkfProt** in the Table Stream. Each element in the **SttbfBkmkProt** specifies information about the bookmark that is associated with the data element which is located at the same offset in that **Plcbkf**. For this reason, the **SttbfBkmkProt** that begins at offset **fcSttbfBkmkProt**, and the **Plcbkf** that begins at offset **fcPlcfBkfProt**, MUST contain the same number of elements.

lcbSttbfBkmkProt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbfBkmkProt** at offset **fcSttbfBkmkProt**.

fcPlcfBkfProt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcbkf** that contains information about range-level protection bookmarks in the document begins at this offset. If **lcbPlcfBkfProt** is zero, then **fcPlcfBkfProt** is undefined and MUST be ignored.

Each data element in the **Plcbkf** is associated, in a one-to-one correlation, with a data element in the **Plcbkl** at offset **fcPlcfBklProt**. For this reason, the **Plcbkf** that begins at offset **fcPlcfBkfProt**, and the **Plcbkl** that begins at offset **fcPlcfBklProt**, MUST contain the same number of data elements. The **Plcbkf** is parallel to the **SttbfBkmkProt** at offset **fcSttbfBkmkProt** in the Table Stream. Each data element in the **Plcbkf** specifies information about the bookmark that is associated with the element which is located at the same offset in that **SttbfBkmkProt**. For this reason, the **Plcbkf** that begins at offset **fcPlcfBkfProt**, and the **SttbfBkmkProt** that begins at offset **fcSttbfBkmkProt**, MUST contain the same number of elements.

lcbPlcfBkfProt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcbkf** at offset **fcPlcfBkfProt**.

fcPlcfBklProt (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **Plcbkl** containing information about range-level protection bookmarks in the document begins at this offset. If **lcbPlcfBklProt** is zero, then **fcPlcfBklProt** is undefined and MUST be ignored.

Each data element in the **Plcbkl** is associated in a one-to-one correlation with a data element in the **Plcbkf** at offset **fcPlcfBkfProt**, so the **Plcbkl** beginning at offset **fcPlcfBklProt** and the **Plcbkf** beginning at offset **fcPlcfBkfProt** MUST contain the same number of data elements.

Copyright © 2025 Microsoft Corporation

lcbPlcfBklProt (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Plcbkl** at offset **fcPlcfBklProt**.

fcSttbProtUser (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A **SttbProtUser** that specifies the usernames that are used for **range-level protection** begins at this offset.

lcbSttbProtUser (4 bytes): An unsigned integer that specifies the size, in bytes, of the **SttbProtUser** at the offset **fcSttbProtUser**.

fcUnused (4 bytes): This value MUST be 0, and MUST be ignored.

lcbUnused (4 bytes): This value MUST be 0, and MUST be ignored.

fcPlcfpmiOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated paragraph mark information cache begins at this offset. Information SHOULD NOT<99> be emitted at this offset and SHOULD<100> be ignored. If **lcbPlcfpmiOld** is zero, then **fcPlcfpmiOld** is undefined and MUST be ignored.

lcbPlcfpmiOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated paragraph mark information cache at offset **fcPlcfpmiOld** in the Table Stream. SHOULD<101> be zero.

fcPlcfpmiOldInline (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated paragraph mark information cache begins at this offset. Information SHOULD NOT<102> be emitted at this offset and SHOULD<103> be ignored. If **lcbPlcfpmiOldInline** is zero, then **fcPlcfpmiOldInline** is undefined and MUST be ignored.

lcbPlcfpmiOldInline (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated paragraph mark information cache at offset **fcPlcfpmiOldInline** in the Table Stream. SHOULD<104> be zero.

fcPlcfpmiNew (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated paragraph mark information cache begins at this offset. Information SHOULD NOT<105> be emitted at this offset and SHOULD<106> be ignored. If **lcbPlcfpmiNew** is zero, then **fcPlcfpmiNew** is undefined and MUST be ignored.

lcbPlcfpmiNew (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated paragraph mark information cache at offset **fcPlcfpmiNew** in the Table Stream. SHOULD<107> be zero.

fcPlcfpmiNewInline (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated paragraph mark information cache begins at this offset. Information SHOULD NOT<108> be emitted at this offset and SHOULD<109> be ignored. If **lcbPlcfpmiNewInline** is zero, then **fcPlcfpmiNewInline** is undefined and MUST be ignored.

lcbPlcfpmiNewInline (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated paragraph mark information cache at offset **fcPlcfpmiNewInline** in the Table Stream. SHOULD<110> be zero.

fcPlcflvcOld (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated **listnum** field cache begins at this offset. Information SHOULD NOT<111> be emitted at this offset and SHOULD<112> be ignored. If **lcbPlcflvcOld** is zero, then **fcPlcflvcOld** is undefined and MUST be ignored.

lcbPlcflvcOld (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated **listnum** field cache at offset **fcPlcflvcOld** in the Table Stream. SHOULD<113> be zero.

fcPlcflvcOldInline (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated **listnum** field cache begins at this offset. Information SHOULD NOT<114> be emitted

Copyright © 2025 Microsoft Corporation

at this offset and SHOULD<115> be ignored. If **lcbPlcflvcOldInline** is zero, **fcPlcflvcOldInline** is undefined and MUST be ignored.

lcbPlcflvcOldInline (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated listnum field cache at offset **fcPlcflvcOldInline** in the Table Stream. SHOULD<116> be zero.

fcPlcflvcNew (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated **listnum** field cache begins at this offset. Information SHOULD NOT<117> be emitted at this offset and SHOULD<118> be ignored. If **lcbPlcflvcNew** is zero, **fcPlcflvcNew** is undefined and MUST be ignored.

lcbPlcflvcNew (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated **listnum** field cache at offset **fcPlcflvcNew** in the Table Stream. SHOULD<119> be zero.

fcPlcflvcNewInline (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated **listnum** field cache begins at this offset. Information SHOULD NOT<120> be emitted at this offset and SHOULD<121> be ignored. If **lcbPlcflvcNewInline** is zero, **fcPlcflvcNewInline** is undefined and MUST be ignored.

lcbPlcflvcNewInline (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated **listnum** field cache at offset **fcPlcflvcNewInline** in the Table Stream. SHOULD<122> be zero.

fcPgdMother (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated document page layout cache begins at this offset. Information SHOULD NOT<123> be emitted at this offset and SHOULD<124> be ignored. If **lcbPgdMother** is zero, **fcPgdMother** is undefined and MUST be ignored.

lcbPgdMother (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated document page layout cache at offset **fcPgdMother** in the Table Stream.

fcBkdMother (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated document text flow break cache begins at this offset. Information SHOULD NOT<125> be emitted at this offset and SHOULD<126> be ignored. If **lcbBkdMother** is zero, then **fcBkdMother** is undefined and MUST be ignored.

lcbBkdMother (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated document text flow break cache at offset **fcBkdMother** in the Table Stream.

fcAfdMother (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated document author filter cache begins at this offset. Information SHOULD NOT<127> be emitted at this offset and SHOULD<128> be ignored. If **lcbAfdMother** is zero, then **fcAfdMother** is undefined and MUST be ignored.

lcbAfdMother (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated document author filter cache at offset **fcAfdMother** in the Table Stream.

fcPgdFtn (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated footnote layout cache begins at this offset. Information SHOULD NOT<129> be emitted at this offset and SHOULD<130> be ignored. If **lcbPgdFtn** is zero, then **fcPgdFtn** is undefined and MUST be ignored.

lcbPgdFtn (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated footnote layout cache at offset **fcPgdFtn** in the Table Stream.

fcBkdFtn (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated footnote text flow break cache begins at this offset. Information SHOULD NOT<131> be emitted at this offset and SHOULD<132> be ignored. If **lcbBkdFtn** is zero, **fcBkdFtn** is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

lcbBkdFtn (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated footnote text flow break cache at offset **fcBkdFtn** in the Table Stream.

fcAfdFtn (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated footnote author filter cache begins at this offset. Information SHOULD NOT<133> be emitted at this offset and SHOULD<134> be ignored. If **lcbAfdFtn** is zero, **fcAfdFtn** is undefined and MUST be ignored.

lcbAfdFtn (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated footnote author filter cache at offset **fcAfdFtn** in the Table Stream.

fcPgdEdn (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated endnote layout cache begins at this offset. Information SHOULD NOT<135> be emitted at this offset and SHOULD<136> be ignored. If **lcbPgdEdn** is zero, then **fcPgdEdn** is undefined and MUST be ignored.

lcbPgdEdn (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated endnote layout cache at offset **fcPgdEdn** in the Table Stream.

fcBkdEdn (4 bytes): An unsigned integer that specifies an offset in the Table Stream. The deprecated endnote text flow break cache begins at this offset. Information SHOULD NOT<137> be emitted at this offset and SHOULD<138> be ignored. If **lcbBkdEdn** is zero, **fcBkdEdn** is undefined and MUST be ignored.

lcbBkdEdn (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated endnote text flow break cache at offset **fcBkdEdn** in the Table Stream.

fcAfdEdn (4 bytes): An unsigned integer that specifies an offset in the Table Stream. Deprecated endnote author filter cache begins at this offset. Information SHOULD NOT<139> be emitted at this offset and SHOULD<140> be ignored. If **lcbAfdEdn** is zero, then **fcAfdEdn** is undefined and MUST be ignored.

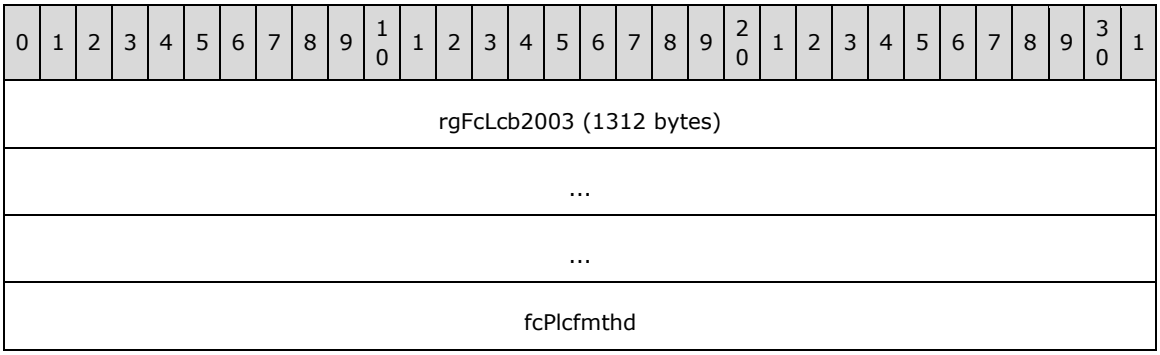
lcbAfdEdn (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated endnote author filter cache at offset **fcAfdEdn** in the Table Stream.

fcAfd (4 bytes): An unsigned integer that specifies an offset in the Table Stream. A deprecated **AFD** structure begins at this offset. Information SHOULD NOT<141> be emitted at this offset and SHOULD<142> be ignored. If **lcbAfd** is zero, **fcAfd** is undefined and MUST be ignored.

lcbAfd (4 bytes): An unsigned integer that specifies the size, in bytes, of the deprecated **AFD** structure at offset **fcAfd** in the Table Stream.

2.5.10 FibRgFcLcb2007

The **FibRgFcLcb2007** structure is a variable-sized portion of the Fib. It extends the FibRgFcLcb2003.



Copyright © 2025 Microsoft Corporation

lcbPlcfmthd
fcSttbfBkmkMoveFrom
lcbSttbfBkmkMoveFrom
fcPlcfBkfMoveFrom
lcbPlcfBkfMoveFrom
fcPlcfBklMoveFrom
lcbPlcfBklMoveFrom
fcSttbfBkmkMoveTo
lcbSttbfBkmkMoveTo
fcPlcfBkfMoveTo
lcbPlcfBkfMoveTo
fcPlcfBklMoveTo
lcbPlcfBklMoveTo
fcUnused1
lcbUnused1
fcUnused2
lcbUnused2
fcUnused3
lcbUnused3
fcSttbfBkmkArto
lcbSttbfBkmkArto
fcPlcfBkfArto
lcbPlcfBkfArto
fcPlcfBklArto
lcbPlcfBklArto

Copyright © 2025 Microsoft Corporation

fcArtoData
lcbArtoData
fcUnused4
lcbUnused4
fcUnused5
lcbUnused5
fcUnused6
lcbUnused6
fcOssTheme
lcbOssTheme
fcColorSchemeMapping
lcbColorSchemeMapping

rgFcLcb2003 (1312 bytes): The contained **FibRgFcLcb2003**.

fcPlcfmthd (4 bytes): This value is undefined and MUST be ignored.

lcbPlcfmthd (4 bytes): This value MUST be 0, and MUST be ignored.

fcSttbfBkmkMoveFrom (4 bytes): This value is undefined and MUST be ignored.

lcbSttbfBkmkMoveFrom (4 bytes): This value MUST be 0, and MUST be ignored.

fcPlcfBkfMoveFrom (4 bytes): This value is undefined and MUST be ignored

lcbPlcfBkfMoveFrom (4 bytes): This value MUST be 0, and MUST be ignored.

fcPlcfBklMoveFrom (4 bytes): This value is undefined and MUST be ignored.

lcbPlcfBklMoveFrom (4 bytes): This value MUST be 0, and MUST be ignored.

fcSttbfBkmkMoveTo (4 bytes): This value is undefined and MUST be ignored.

lcbSttbfBkmkMoveTo (4 bytes): This value MUST be 0, and MUST be ignored.

fcPlcfBkfMoveTo (4 bytes): This value is undefined and MUST be ignored.

lcbPlcfBkfMoveTo (4 bytes): This value MUST be 0, and MUST be ignored.

fcPlcfBklMoveTo (4 bytes): This value is undefined and MUST be ignored.

lcbPlcfBklMoveTo (4 bytes): This value MUST be 0, and MUST be ignored.

fcUnused1 (4 bytes): This value is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

lcbUnused1 (4 bytes): This value MUST be 0, and MUST be ignored.

fcUnused2 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused2 (4 bytes): This value MUST be 0, and MUST be ignored.

fcUnused3 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused3 (4 bytes): This value MUST be 0, and MUST be ignored.

fcSttbfBkmkArto (4 bytes): This value is undefined and MUST be ignored.

lcbSttbfBkmkArto (4 bytes): This value MUST be 0, and MUST be ignored.

fcPlcfBkfArto (4 bytes): This value is undefined and MUST be ignored.

lcbPlcfBkfArto (4 bytes): This value MUST be 0, and MUST be ignored

fcPlcfBklArto (4 bytes): Undefined and MUST be ignored.

lcbPlcfBklArto (4 bytes): MUST be zero, and MUST be ignored.

fcArtoData (4 bytes): This value is undefined and MUST be ignored.

lcbArtoData (4 bytes): This value MUST be 0, and MUST be ignored.

fcUnused4 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused4 (4 bytes): This value MUST be 0, and MUST be ignored.

fcUnused5 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused5 (4 bytes): This value MUST be 0, and MUST be ignored.

fcUnused6 (4 bytes): This value is undefined and MUST be ignored.

lcbUnused6 (4 bytes): This value MUST be 0, and MUST be ignored.

fcOssTheme (4 bytes): This value is undefined and MUST be ignored.

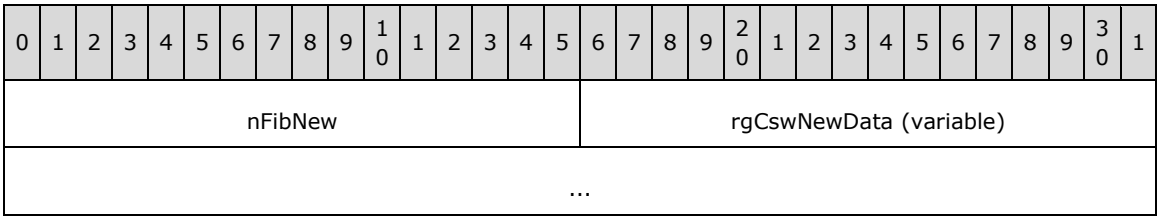
lcbOssTheme (4 bytes): This value SHOULD<143> be zero, and MUST be ignored.

fcColorSchemeMapping (4 bytes): This value is undefined and MUST be ignored.

lcbColorSchemeMapping (4 bytes): This value SHOULD<144> be zero, and MUST be ignored.

2.5.11 FibRgCswNew

The **FibRgCswNew** structure is an extension to the **Fib** structure that exists only if **Fib**.**cswNew** is nonzero.



nFibNew (2 bytes): An unsigned integer that specifies the version number of the file format that is used. This value MUST be one of the following.

Copyright © 2025 Microsoft Corporation

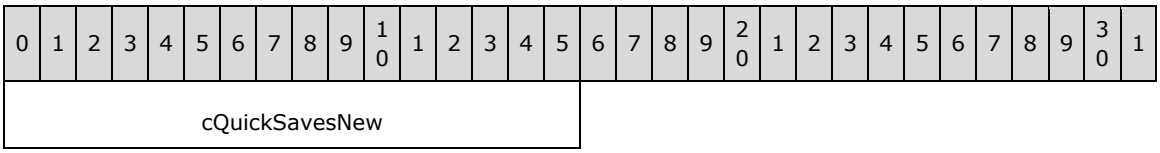
Value
0x00D9
0x0101
0x010C
0x0112

rgCswNewData (variable): Depending on the value of **nFibNew** this is one of the following.

Value of nFibNew	Meaning
0x00D9	fibRgCswNewData2000 (2 bytes)
0x0101	fibRgCswNewData2000 (2 bytes)
0x010C	fibRgCswNewData2000 (2 bytes)
0x0112	fibRgCswNewData2007 (8 bytes)

2.5.12 FibRgCswNewData2000

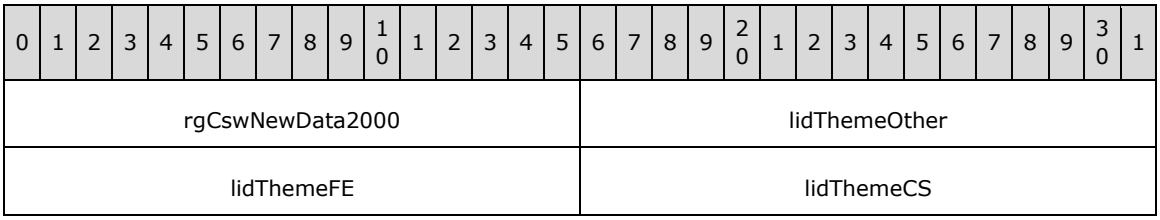
The **FibRgCswNewData2000** structure is a variable-sized portion of the Fib.



cQuickSavesNew (2 bytes): An unsigned integer that specifies the number of times that this document was incrementally saved since the last **full save**. This value MUST be between 0 and 0x000F, inclusively.

2.5.13 FibRgCswNewData2007

The **FibRgCswNewData2007** structure is a variable-sized portion of the Fib. It extends the FibRgCswNewData2000.



rgCswNewData2000 (2 bytes): The contained **FibRgCswNewData2000**.

lidThemeOther (2 bytes): This value is undefined and MUST be ignored.

lidThemeFE (2 bytes): This value is undefined and MUST be ignored.

lidThemeCS (2 bytes): This value is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

2.5.14 Determining the nFib

The **nFib** value specifies the version number of the file format that is used. The proper **nFib** value for the current document is determined in the following way:

1. Read the **FIB** from offset zero in the WordDocument Stream.
2. Check the value of FIB.cswNew.

- If the value is 0, **nFib** is specified by **FibBase.nFib.**
- Otherwise, the value is not 0 and **nFib** is specified by **FibRgCswNew.nFibNew.**

2.5.15 How to read the FIB

The **Fib** structure is located at offset 0 of the WordDocument Stream. Given the variable size of the **Fib**, the proper way to load it is the following:

1. Set all bytes of the in-memory version of the **Fib** being used to 0. It is recommended to use the largest version of the **Fib** structure as the in-memory version.
2. Read the entire **FibBase**, which MUST be present and has fixed size.
3. Read **Fib.csw**.
4. Read the minimum of **Fib.csw** * 2 bytes and the size, in bytes, of the in-memory version of **FibRgW97** into **FibRgW97**.
5. If the application expects fewer bytes than indicated by **Fib.csw**, advance by the difference thereby skipping the unknown portion of **FibRgW97**.
6. Read **Fib.cslw**.
7. Read the minimum of **Fib.cslw** * 4 bytes and the size, in bytes, of the in-memory version of **FibRgLw97** into **FibRgLw97**.
8. If the application expects fewer bytes than indicated by **Fib.cslw**, advance by the difference thereby skipping the unknown portion of **FibRgLw97**.
9. Read **Fib.cbRgFcLcb**.
10. Read the minimum of **Fib.cbRgFcLcb** * 8 bytes and the size, in bytes, of the in-memory version of **FibRgFcLcb** into **FibRgFcLcb**.
11. If the application expects fewer bytes than indicated by **Fib.cbRgFcLcb**, advance by the difference, thereby skipping the unknown portion of **FibRgFcLcb**.
12. Read **Fib.cswNew**.
13. Read the minimum of **Fib.cswNew** * 2 bytes and the size, in bytes, of the in-memory version of **FibRgCswNew** into **FibRgCswNew**.

2.6 Single Property Modifiers

The following sections specify the valid **Sprm** values.

For ease of implementation, the **Sprm**s are listed as 16-bit integers rather than structures. The following formulas specify the relationship between the 16-bit integer representation and the members of the **Sprm** structure. The single ampersand (&) represents the bitwise AND operation; all fractions are rounded down to the previous whole number.

Copyright © 2025 Microsoft Corporation

$$ispmd = sprm \ \& \ 0x01FF$$

$$f = \frac{sprm}{512} \ \& \ 0x0001$$

$$sgc = \frac{sprm}{1024} \ \& \ 0x0007$$

$$spra = \frac{sprm}{8192}$$

2.6.1 Character Properties

A **Prl** with a **sprm.sgc** of 2 modifies a character property.

The following table specifies the character property modifiers, including the valid **sprm** values, their function, and the corresponding **operand** type and meaning.

Sprm	ispmd	operand
sprmCFRMarkDel (0x0800)	0x00	A **ToggleOperand** that specifies whether the text is formatted as deleted revision mark text, which is text that was deleted while revision marking was on. By default, text is not formatted as deleted revision mark text.
sprmCFRMarkIns (0x0801)	0x01	A **ToggleOperand** that specifies whether the text is formatted as inserted revision mark text, which is text that was inserted while revision marking was on. By default, text is not formatted as inserted revision mark text.
sprmCFFldVanish (0x0802)	0x02	A **ToggleOperand** that specifies whether the field text is hidden. By default, field text is not hidden.
sprmCPicLocation (0x6A03)	0x03	A signed 32-bit integer that specifies either the position in the Data Stream of a picture or binary data or the name of an **OLE object** storage. Text with sprmCPicLocation applied MUST also have sprmCFSpec applied with a value of 1. The text range MUST contain only characters from the special characters specified in sprmCFSpec. The value of sprmCPicLocation is evaluated for each character in the text range. The value is evaluated differently depending on the character code, as shown following: If the character is U+0001: ▪ The operand of sprmCPicLocation is a position in the **DataStream**. If sprmCFData is also present and set to 1, the value specifies the position of a **NilPICFAndBinData** and describes binary data; otherwise the value specifies the position of a **PICFAndOfficeArtData** and describes a picture. If the character is U+0014: ▪ If sprmCFOle2 is also present and set to "true" and the associated field does not have grffldEnd.fZombieEmbed set, the operand of sprmCPicLocation specifies the location of an OLE object storage. If the file is not encrypted with Office Binary Document RC4 CryptoAPI Encryption (section 2.2.6.3), the value specifies the name of an OLE object storage in the **ObjectPool** of the document. Specifically, the decimal value is converted to a string, and prefixed with an underscore. The resultant string MUST be the name of a valid OLE storage in the **ObjectPool** of the

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		document. If the file is encrypted with Office Binary Document RC4 CryptoAPI Encryption, the value specifies an offset in the data stream which contains an **FOBJH** followed by an OLE object storage. When used in this fashion, the text range on which sprmCPicLocation is applied MUST contain exactly one character. If sprmCFOle2 is absent or set to "false" or the associated field has grffldEnd.fZombieEmbed set, sprmCPicLocation is unused and MUST be ignored. If there is another character, sprmCPicLocation MUST be ignored. sprmCPicLocation MUST be present for characters that indicate a picture, binary data, or OLE object storage.
sprmCIbstRMark (0x4804)	0x04	A signed 16-bit integer that specifies a zero-based index into **SttbfRMark**. This value MUST be greater than or equal to zero and MUST be less than **SttbfRMark.cData**. The string at this index is the name of the author who inserted the text. This is only recorded if revision marking is on at the time of the insertion. By default, this index is zero, which is the index of the "unknown" author.
sprmCDttmRMark (0x6805)	0x05	A **DTTM** that specifies the date and time at which the text was inserted. This is recorded only if revision marking is on at the time of the insertion. By default, all fields of this **DTTM** are zero.
sprmCFData (0x0806)	0x06	A **Bool8** that specifies whether the picture character in the text represents binary data. If set to **true**, the text range MUST contain exactly 1 character that is the picture character (U+0001) and sprmCPicLocation MUST be present to specify the location of the binary data. By default, a picture character specifies a picture and does not specify binary data.
sprmCIdslRMark (0x4807)	0x07	An unsigned 16-bit integer that specifies the reason value of the inserted or modified revision mark text. This is recorded only if revision marking is on at the time of the text insertion or modification. MUST be one of the values shown following. 0x0000 - Performed a normal edit 0x0001 - Applied a style 0x0002 - Adjusted alignment with a tab 0x0003 - Adjusted alignment with a tab 0x0004 - Removed extra paragraph mark 0x0005 - Replaced all caps with mixed caps 0x0006 - Replaced bullet character with bullet symbol 0x0007 - Replaced straight quote with smart quote 0x0008 - Replaced multiple-character symbol with single symbol 0x0009 - Replaced text with trademark symbol 0x000A - Replaced text with copyright symbol 0x000B - Replaced text with registered trademark symbol 0x000C - Adjusted spaces after period 0x000D - Replaced numbers with fraction symbol 0x000E - Applied a heading style 0x000F - Applied an outline style 0x0010 - Applied a list style 0x0011 - Applied a memo header style 0x0012 - Applied an address style

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		0x0013 - Applied a salutation style 0x0014 - Applied a closing phrase style 0x0015 - Applied a date style 0x0016 - Applied a distribution list style 0x0017 - Applied a bullet list style 0x0018 - Applied a column style 0x0019 - Applied a carbon copy style 0x001A - Replaced text with superscript 0x001B - Replaced whitespace galley with tabs 0x001C - Removed leading whitespace 0x001D - Removed manual numbering 0x001E - Replaced two hyphens with long (em) dash 0x001F - Adjusted spaces before: '!', '?', or ';' 0x0020 - Inserted paragraph mark 0x0021 - Replaced leading whitespace to first line indent 0x0022 - Removed a space from between a double byte character and a single byte character 0x0023 - Replaced to match to open parenthesis 0x0024 - Replaced double byte to single byte 0x0025 - Replaced single byte to double byte 0x0026 - Replaced manual emphasis 0x0027 - Replaced border characters with borders 0x0028 - Replaced e-mail history characters with indentation 0x0029 - Replaced URL or UNC with hyperlink 0x002A - Replaced Gateway-generated hex characters 0x002B - Applied outline level for document map By default, the reason value of text that is revision-marked is zero.
sprmCSymbol (0x6A09)	0x09	A **CSymbolOperand** structure that designates the character as a symbol and specifies the font and character code for the symbol. By default, characters are not symbols.
sprmCFOle2 (0x080A)	0x0A	A **Bool8** value that specifies whether the character is a placeholder for an OLE object. When **sprmCFOle2** is **true**, sprmCFObj MUST also be **true**, and sprmCPicLocation MUST also be set with the OLE storage name. The character representing the OLE object MUST be the field separator (U+00014) of an **EMBED** field (0x3A), **LINK** field (0x38), or **CONTROL** field (0x57). By default, characters are not placeholders for OLE objects.
sprmCHighlight (0x2A0C)	0x0C	An Ico value that specifies the highlighting color of the text. By default, text is not highlighted.
sprmCFWebHidden (0x0811)	0x11	A ToggleOperand value that specifies whether the text is hidden in **Web Layout view** of the document. By default, text is not hidden in Web Layout view.
sprmCRsidProp (0x6815)	0x15	An integer value that specifies a revision save ID, as specified in [ECMA-376] Part 1, Section 17.15.1.70 rsid (Single Session Revision Save ID), associated with character formatting. If not present, then no revision save ID is specified for this formatting.
sprmCRsidText (0x6816)	0x16	An integer that specifies a revision save ID, as specified in [ECMA-376] Part 1, Section 17.15.1.70 rsid (Single Session Revision Save ID), associated with insertion of text. If not present, then no revision save ID is specified for this text.
sprmCRsidRMDel (0x6817)	0x17	An integer that specifies a revision save ID, as specified in [ECMA-376] Part 1, Section 17.15.1.70 rsid (Single Session Revision Save ID), associated with tracked deletion of text. If not present, then no revision save ID is specified

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		for this deletion.
sprmCFSpecVanish (0x0818)	0x18	A **ToggleOperand** that specifies that this line break does not indicate a line break but serves as a style separator. A style separator allows one paragraph to consist of parts that have different paragraph styles. This **Sprm** MUST NOT be applied to any character other than a line break character (**Unicode** 0x000B). By default, line break characters specify regular line breaks, and are not used as style separators.
sprmCFMathPr (0xC81A)	0x1A	A **MathPrOperand** that specifies the justification of equations in the paragraph. This **Sprm** MUST only be applied to paragraph mark characters or line break characters (Unicode 0x000B). By default, equations are justified according to the **mthbpjc** member of the **DOPMTH**. MAY<145> be ignored.
sprmCIstd (0x4A30)	0x30	An unsigned integer that specifies the **istd** of a character style to apply. To apply the **istd**: 1. Reset the character properties of the text to match the results of the paragraph style (in other words, revert any formatting that is applied on top of the paragraph style). 2. Fetch the set of properties from the specified character style. (For instructions, see Applying Properties.) 3. Apply those properties to the current text. During steps 1 and 3, preserve the previous values of the following: ▪ Whether the text is formatted as deleted revision mark text (for example, by sprmCFRMarkDel). ▪ Whether the text is formatted with **right-to-left** layout (for example, by sprmCFBiDi). ▪ Whether the text is displayed right-to-left or is in a **South Asian language**. (for example, by sprmCFComplexScripts). ▪ Whether the field text is hidden (for example, by sprmCFFldVanish). ▪ Whether the text is formatted as inserted revision mark text (for example, by sprmCFRMarkIns). ▪ Whether the text has a special meaning and special display handling (for example, by sprmCFSpec). ▪ Whether the text has associated picture data (for example, by sprmCFData). ▪ Whether the character is a placeholder for an OLE object (for example, by sprmCFOle2). ▪ Whether the text is hidden in Web Layout view (for example, by sprmCFWebHidden). ▪ Whether the text is hidden and the image of a shape is displayed in its place (for example, by sprmCFObj). ▪ The position in the Data Stream of a picture, or the name of an **Object Linking and Embedding (OLE)** stream (for example, by sprmCPicLocation). ▪ Whether the text has an associated property revision mark, as well as its author and date/time (for example, by sprmCPropRMark). ▪ Paragraph properties that have been preserved for revision marking (for example, by sprmCWall). ▪ The reason value of the inserted or modified revision mark text (for example, by sprmCIdslRMark). ▪ Whether the text is a symbol and, if it is, the font and character code (for example, by sprmCSymbol). ▪ Any previous operand value of sprmCIdctHint. ▪ The highlighting color of the text (for example, from sprmCHighlight). ▪ Whether the text is hidden from display when hiding arbitrary XML delimiters (for example, from sprmCFSdtVanish). ▪ The type of font substitution that is needed for the associated text (for example, from sprmCNeedFontFixup). ▪ The revision save ID that is associated with the insertion of text (for example, from sprmCRsidText). ▪ The revision save ID that is associated with character formatting (for example, from sprmCRsidProp). ▪ The revision save ID that is associated with the tracked deletion of text

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		(for example, by sprmCRsidRMDel). ▪ The names of the authors who inserted the text, (for example, by sprmCIbstRMark). ▪ The dates and times at which the text was inserted (for example, by sprmCDttmRMark). ▪ The names of the authors who deleted the text (for example, by sprmCIbstRMarkDel). ▪ The dates and times at which the text was deleted (for example, by sprmCDttmRMarkDel). ▪ The justification of equations in the paragraph (for example, by sprmCFMathPr). By default, text has the character style specified by **istd** 0x000A.
sprmCIstdPermute (0xCA31)	0x31	An SPPOperand value that specifies a potential application of a different character style (**istd**). If the **istd** is not affected, this **Prl** MUST be ignored. If the **istd** is affected, the operation of this **sprm** specifies the new **istd** as equivalent to sprmCIstd. Note that the character properties of the text that are not specified by the current character style are reapplied after applying sprmCIstdPermute. By default, the character style of the text is unaffected.
sprmCPlain (0x2A33)	0x33	The operand is an unsigned integer that MUST be 0 and MUST be ignored. The presence of this **Sprm** specifies a reset of the character properties of the text to match that of the underlying **paragraph style** (taking **style** hierarchy into account), while preserving the previous values of properties in the following list. To determine the properties of the underlying paragraph style (taking style hierarchy into account), follow the algorithm in Determining Formatting Properties but stop before applying Direct Character Formatting. (In other words, the new values are determined by evaluating the properties of the text as if no character style or direct character formatting are applied; see Style Hierarchy in [ECMA-376] Part 1, Section 17.7.2 for further specification.) The following properties MUST NOT be affected by the application of sprmCPlain: ▪ Whether the text is formatted as deleted revision mark text (for example, by sprmCFRMarkDel). ▪ Whether the text is formatted with right-to-left layout (for example, by sprmCFBiDi). ▪ Whether the text is displayed right-to-left or is in a South Asian language. (for example, by sprmCFComplexScripts). ▪ Whether the field text is hidden (for example, by sprmCFFldVanish). ▪ Whether the text is formatted as inserted revision mark text (for example, by sprmCFRMarkIns). ▪ Whether the text has a special meaning and special display handling (for example, by sprmCFSpec). ▪ Whether the text has associated picture data (for example, by sprmCFData). ▪ Whether the character is a placeholder for an OLE object (for example, by sprmCFOle2). ▪ Whether the text is hidden in Web Layout view (for example, by sprmCFWebHidden). ▪ The names of the authors who inserted the text (for example, by sprmCIbstRMark). ▪ The dates and times at which the text was inserted (for example, by sprmCDttmRMark). ▪ The names of the authors who deleted the text (for example, by sprmCIbstRMarkDel). ▪ The dates and times at which the text was deleted (for example, by sprmCDttmRMarkDel).

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		▪ Whether the text has an associated property revision mark, as well as its author and date/time (for example, by sprmCPropRMark). ▪ Paragraph properties that have been preserved for revision marking (for example, by sprmCWall). ▪ The reason value of the inserted or modified revision mark text (for example, by sprmCIdslRMark). ▪ Whether the text is a symbol and, if it is, the font and character code (for example, by sprmCSymbol). ▪ The position in the Data Stream of a picture, or the name of an OLE stream (for example, by sprmCPicLocation). ▪ Any previous operand value of sprmCIdctHint. ▪ The highlighting color of the text (for example, by sprmCHighlight). ▪ The type of font substitution that is needed for the associated text (for example, by sprmCNeedFontFixup). ▪ The revision save ID that is associated with the insertion of text (for example, by sprmCRsidText). ▪ The revision save ID that is associated with character formatting (for example, by sprmCRsidProp). ▪ The revision save ID that is associated with the tracked deletion of text (for example, by sprmCRsidRMDel). ▪ The justification of equations in the paragraph (for example, by sprmCFMathPr). By default, the character properties of the text are not reset.
sprmCKcd (0x2A34)	0x34	A byte that specifies the kind of emphasis to apply to the text. The operand MUST be one of the following values. 0x00 - No emphasis 0x01 - Solid circle 0x02 - Comma above 0x03 - Circle above 0x04 - Solid circle below The operands map to Unicode characters as shown following. The **East Asian language** of the text is specified by sprmCRgLid1_80 and sprmCRgLid1. The default East Asian language is Japanese if sprmCRgLid1_80 or sprmCRgLid1 does not specify Japanese, Korean, Chinese (Taiwan), or Chinese (China). If the meaning of the operand is "solid circle", the following applies: ▪ In the Japanese language, the Unicode character of 0xFF0E is positioned above the text. ▪ In the Korean language, the Unicode character of 0x02D9 is positioned above the text. ▪ In the Chinese (Taiwan) language, the Unicode character of 0x2027 is positioned above the text. ▪ In the Chinese (China) language, the Unicode character of 0xFF0E is positioned below the text. If the meaning of the operand is "comma above", the following applies: ▪ In the Japanese language, the Unicode character of 0x3001 is positioned above the text. ▪ In the Korean language, the Unicode character of 0x02DA is positioned above the text.

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		▪ In the Chinese (Taiwan) language, the Unicode character of 0x3002 is positioned above the text. ▪ In the Chinese (China) language, the Unicode character of 0x3001 is positioned above the text. If the meaning of the operand is "circle above", the following applies: ▪ In the Japanese language, the Unicode character of 0x02DA is positioned above the text. ▪ In the Korean language, the Unicode character of 0x02DA is positioned above the text. ▪ In the Chinese (Taiwan) language, the Unicode character of 0x3002 is positioned above the text. ▪ In the Chinese (China) language, the Unicode character of 0x02DA is positioned above the text. If the meaning of the operand is "solid circle below", the following applies: ▪ In the Japanese language, the Unicode character of 0xFF0E is positioned below the text. ▪ In the Korean language, the Unicode character of 0xFF0E is positioned below the text. ▪ In the Chinese (Taiwan) language, the Unicode character of 0xFF0E is positioned below the text. ▪ In the Chinese (China) language, the Unicode character of 0xFF0E is positioned below the text. By default, text has no emphasis mark.
sprmCFBold (0x0835)	0x35	A **ToggleOperand** value that specifies whether the text is bold. By default, text is not bold.
sprmCFItalic (0x0836)	0x36	A **ToggleOperand** value that specifies whether the text is italicized. By default, text is not italicized.
sprmCFStrike (0x0837)	0x37	A **ToggleOperand** value that specifies whether the text is formatted with strikethrough. By default, text is not struck through.
sprmCFOutline (0x0838)	0x38	A **ToggleOperand** value that specifies whether only the outline contour of the characters in the text is rendered, with the inside of each character left empty. By default, text is rendered in normal solid characters. If **sprmCFEmboss**, or **sprmCFImprint** is **true**, then **sprmCFOutline** MUST be **false**.
sprmCFShadow (0x0839)	0x39	A **ToggleOperand** value that specifies whether the text is formatted with a shadow. By default, text has no shadow. If **sprmCFEmboss** or **sprmCFImprint** is **true**, then **sprmCFShadow** MUST be **false**.
sprmCFSmallCaps (0x083A)	0x3A	A **ToggleOperand** value that specifies whether the text characters are displayed as their capital letter equivalents, in a font size that is smaller than the actual font size that is specified for this text. It does not affect any nonalphabetic character. By default, the characters are displayed in their original character form.
sprmCFCaps (0x083B)	0x3B	A **ToggleOperand** value that specifies whether the text characters are displayed as their capital letter equivalents. It does not affect any nonalphabetic character. By default, the characters are displayed in their original character form.
sprmCFVanish (0x083C)	0x3C	A **ToggleOperand** value that specifies whether the text is formatted as hidden. By default, text is not hidden.
sprmCKul (0x2A3E)	0x3E	A **Kul** value that specifies the underlining style of the text. By default, text is not underlined.
sprmCDxaSpace (0x8840)	0x40	An **XAS** value that specifies the extra space, in **twips**, between a character and the one to its right. This does not vary with the directionality of the script or layout. Negative values indicate that space is removed, possibly producing character overlap. Negative space beyond the character width is ignored. By

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		default, the space to the right of a character is neither added nor removed.
sprmCIco (0x2A42)	0x42	An Ico value that specifies the color of the text. The default text color is **cvAuto**.
sprmCHps (0x4A43)	0x43	An unsigned 2-byte integer that specifies the size of the text, except for text that meets the qualifications for **sprmCHpsBi**. This value is specified in half-**points**. The specified value MUST be between 2 and 3276. By default, the font size is 20 half-points.
sprmCHpsPos (0x4845)	0x45	A signed integer value that specifies the vertical position, in half-points, of text relative to the normal position. The specified value MUST be between -3168 and 3168. By default, text is in its normal vertical position.
sprmCMajority (0xCA47)	0x47	A CMajorityOperand value that specifies which of the character properties of the text to reset to match the properties of the underlying paragraph style, taking the style hierarchy into account. If the character style index (the **istd**) of the text is not 10 (the default), this **Sprm** MUST be ignored. sprmCMajority can affect any of the character properties in the following list. If a character property is affected, that property on the text is then set to the value of that property in the underlying paragraph style, taking the style hierarchy into account. To determine if a given property (from the following list of potentially affected character properties) is affected, do the following: 1. Find the property value on the text. 2. Find the property value as specified in the **grpprl** member of CMajorityOperand. If the property value is not specified in the **grpprl** member, use the default value. 3. Compare the two values. 4. If the values match, the property is affected. After it is determined that a property is affected, see sprmCPlain for information about how to determine the properties of the underlying paragraph style. Note that two special cases occur in the determination of whether a property is affected: ▪ In the case of whether the text is excluded from the proofing analysis (for example, by sprmCFNoProof), if the value of the property on the text is 1 and the value of the property specified in the **grpprl** is 1, the property is not affected (it is left as 1 on the text.) Otherwise the preceding rules apply. ▪ In the case of the font index that is used only if the language for the text is an East Asian language (for example, by sprmCRgFtc1), if the preceding rules would lead to the application of a font index for this property that specifies the Times New Roman font, the property is not affected (it is left as before). The character properties (potentially) affected are: ▪ Whether the text is bold (for example, by sprmCFBold) ▪ Whether the text is italicized (for example, by sprmCFItalic) ▪ Whether the text is formatted in smaller capital forms (for example, by sprmCFSmallCaps) ▪ Whether the text is formatted as hidden (for example, by sprmCFVanish) ▪ Whether the text is bolded when displayed right-to-left (for example, by sprmCFBoldBi) ▪ Whether the text is italicized when the text is displayed right-to-left (for example, by sprmCFItalicBi) ▪ Whether the text is formatted with a strikethrough effect (for example, by sprmCFStrike) ▪ Whether the text is formatted in capital form (for example, by sprmCFCaps) ▪ Whether the text is formatted with a shadow effect (for example, by sprmCFShadow)

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		▪ Whether only the outline contour of the characters in the text is rendered, with the inside of each character left empty (for example, by sprmCFOutline) ▪ Whether the text is formatted with a double strikethrough effect (for example, by sprmCFDStrike) ▪ Whether the text is embossed (for example, by sprmCFEmboss) ▪ Whether the text is formatted with the imprint style (for example, by sprmCFImprint) ▪ Whether the text is excluded from the proofing analysis (for example, by sprmCFNoProof) ▪ The font index that is used to display the text only if the conditions for using these fonts do not apply: sprmCRgFtc1, sprmCRgFtc2 and sprmCFtcBi (for example, by sprmCRgFtc0) ▪ The font index that is used only if the language for the text is an East Asian language (for example, by sprmCRgFtc1) ▪ The font index that is used to display the text if the language for the text is one of those listed for sprmCRgFtc2 (for example, by sprmCRgFtc2) ▪ The font index that is used to display the text only if the text flow is right-to-left or if the language for the text is a South Asian language (for example, by sprmCFtcBi) ▪ The size of the text (for example, by sprmCHps) ▪ The size of the text, for text that is displayed right-to-left (for example, by sprmCHpsBi) ▪ The vertical position of the text relative to the normal position (for example, by sprmCHpsPos) ▪ The superscript or subscript for text (for example, by sprmCIss) ▪ The kind of emphasis to apply to the text (for example, by sprmCKcd) ▪ The underlining style of the text (for example, by sprmCKul) ▪ The extra space, in twips, between a character and the one to its right (for example, by sprmCDxaSpace) ▪ The color of the text (for example, by sprmCCv) ▪ The text effect of the text (for example, by sprmCSfxText) ▪ The language of the text, except for East Asian languages (for example, by sprmCRgLid0) ▪ The language of the text, if it is an East Asian language (for example, by sprmCRgLid1) ▪ The language of the text when the text is displayed right-to-left (for example, by sprmCLidBi) Any character property that is not in this list MUST NOT be affected by sprmCMajority.
sprmCIss (0x2A48)	0x48	An 8-bit unsigned integer that specifies superscript or subscript for text. By default, text is normal. The value MUST be one of those listed following. 0x00 - Normal text 0x01 - Superscript 0x02 - Subscript
sprmCHpsKern (0x484B)	0x4B	A signed integer that specifies a font size threshold, in half-points, at or above which kerning is applied to the text. If the operand is 0, no kerning is applied; otherwise, it MUST be a value between 1 and 3276. By default, kerning is not applied to any characters.
sprmCHresi (0x484E)	0x4E	An HresiOperand value that specifies the word-breaking behavior for the text. By default the text uses normal hyphenation.
sprmCRgFtc0 (0x4A4F)	0x4F	A 2-byte signed integer value that is an index into the font table (SttbfFfn). The font that is referenced by this index is used to display the text only if the conditions for using these fonts do not apply: sprmCRgFtc1, sprmCRgFtc2 and sprmCFtcBi. This value MUST be between 0 and a number that is one less than the count of entries in SttbfFfn unless there are 0 entries, in which case this value MUST be 0. By default, the font used under these conditions is **STSH.Stshi.Stshif.ftcAsci**.

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
sprmCRgFtc1 (0x4A50)	0x50	A 2-byte signed integer value that is an index into the font table (SttbfFfn). The font referenced by this index is used only if the language for the text is an East Asian language. This value MUST be between 0 and a number that is one less than the count of entries in SttbfFfn unless there are 0 entries, in which case this value MUST be 0. By default, the font that is used under these conditions is **STSH.Stshi.Stshif.ftcFE**.
sprmCRgFtc2 (0x4A51)	0x51	A 2-byte signed integer that is an index into the font table (SttbfFfn). The font that is referenced by this index is used to display text if the character falls outside the Unicode character range U+0020 to U+007F and the conditions for using these fonts do not apply: sprmCRgFtc1 and sprmCFtcBi. This value MUST be between 0 and a number that is one less than the count of entries in SttbfFfn unless there are 0 entries, in which case this value MUST be 0. By default, the font that is used under these conditions is **STSH.Stshi.Stshif.ftcOther**.
sprmCCharScale (0x4852)	0x52	A 2-byte unsigned integer that specifies the percentage by which to horizontally scale the text, thereby changing the shape of the characters. The value MUST be greater than or equal to 1, and less than or equal to 600. Values that are less than 100 represent the compressing of text. Values that are greater than 100 represent the expanding of text. By default, text is neither compressed nor expanded.
sprmCFDStrike (0x2A53)	0x53	A **ToggleOperand** value that specifies whether the text is formatted with the double strikethrough effect. By default, text is not struck through.
sprmCFImprint (0x0854)	0x54	A **ToggleOperand** value that specifies whether the text is formatted with the imprint effect. By default, text does not have this formatting applied. If sprmCFEmboss, sprmCFOutline or sprmCFShadow is "true", then sprmCFImprint MUST be "false".
sprmCFSpec (0x0855)	0x55	A **ToggleOperand** value that specifies whether the current text has a meaning that differs or displays differently than the underlying character to which it is applied. This value SHOULD<146> be applied only to the following characters. U+0001 - A picture location that is used in conjunction with sprmCPicLocation. U+0002 - An auto-numbered footnote reference. See plcffndRef. U+0003 - A short horizontal line. U+0004 - A long horizontal line that is the width of the content area of the page. U+0005 - An annotation reference character. See PlcfandRef. U+0008 - A drawn object. See plcfSpa. U+0013 - A field begin character. See Plcfld. U+0014 - A field separator character. See Plcfld. U+0015 - A field end character. See Plcfld. U+0028 - A symbol. See sprmCSymbol. U+003C - The start of a structured document tag bookmark range. See FibRgFcLcb2003.fcPlcfBkfSdt. U+003E - The end of a structured document tag bookmark range. See FibRgFcLcb2003.fcPlcfBklSdt. U+2002 - An en space. U+2003 - An em space. By default, characters have no special meaning beyond their underlying glyph.
sprmCFObj (0x0856)	0x56	A **Bool8** value that specifies whether the current text represents an embedded object. If sprmCFObj is "true", sprmCFOle2 MUST also be "true". By default, text is not an embedded object.

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
sprmCPropRMark90 (0xCA57)	0x57	A **PropRMarkOperand** value that specifies whether the character run has an associated **property revision mark**, as well as its author and date/time. By default, character runs have no property revision marks.
sprmCFEmboss (0x0858)	0x58	A **ToggleOperand** value that specifies whether the text is embossed. By default, text is not embossed. If sprmCFOutline, sprmCFShadow or sprmCFImprint is "true", sprmCFEmboss MUST be "false".
sprmCSfxText (0x2859)	0x59	A byte that specifies a text effect to apply to the text. By default, text does not have any text effects. The allowed values and their meanings are listed following. 0x0 - None. 0x1 - Las Vegas Lights. Text is bordered by marquee lights that blink between the colors red, yellow, green, and blue. 0x2 - Blinking background. Text has a black background that blinks on and off. 0x3 - Sparkle Text. Text is overlaid with multicolored stars that blink on and off at regular intervals. 0x4 - Marching Black Ants. Text is surrounded by a black dashed-line border. The border is animated so that the individual dashes appear to move clockwise around the text. 0x5 - Marching Red Ants. Text is surrounded by a red dashed-line border that is animated to appear to move clockwise around the text. 0x6 - Shimmer. Text is alternately blurred and unblurred at regular intervals, to give the appearance of shimmering.
sprmCFBiDi (0x085A)	0x5A	A **ToggleOperand** value that specifies whether the text is formatted with right-to-left layout. By default, text is displayed from right to left if the language for the text is a right-to-left language.
sprmCFBoldBi (0x085C)	0x5C	A **ToggleOperand** value that specifies whether the text is formatted bold when displayed right-to-left or determined to be complex script. By default, text is not bold.
sprmCFItalicBi (0x085D)	0x5D	A **ToggleOperand** value that specifies whether the text is italicized when displayed right-to-left or determined to be complex script. By default, text is not italicized.
sprmCFtcBi (0x4A5E)	0x5E	A 2-byte signed index into the font table (**SttbfFfn**). The font that is referenced by this index is used to display the text only if the text flow is right-to-left or if the text is a complex script. This value MUST be a number that is between 0 and one less than the count of entries in **SttbfFfn** unless there are 0 entries, in which case this value MUST be 0. By default, the font used under these conditions is **STSH.Stshi.ftcBi**.
sprmCLidBi (0x485F)	0x5F	A **LID** value that specifies the language of the text when the text is displayed right-to-left or if the text is complex script. By default, text language is undefined and text is not checked for spelling, grammar, or hyphenation.
sprmCIcoBi (0x4A60)	0x60	An **ICO** value that specifies the color of text when displayed right-to-left or determined to be complex script.<147>
sprmCHpsBi (0x4A61)	0x61	An unsigned 2-byte integer value that specifies the size of the text, for text that is displayed right-to-left or text that is a complex script. This value is specified in half-points. The specified value MUST be between 0 and 3276. By default, text of the following Unicode subranges uses the associated size, in half points, as specified in [MC-USB]. ▪ Thai, Mongolian, and Bangla use a font size of 28. ▪ Tibetan uses a font size of 32.

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		▪ Devanagari uses a font size of 20. ▪ Khmer uses a font size of 36. Text of other Unicode subranges uses a font size of 24 half points.
sprmCDispFldRMark (0xCA62)	0x62	A **DispFldRmOperand** value that indicates a revision within the result of a display field. This sprm MUST be applied only to a LISTNUM display field.
sprmCIbstRMarkDel (0x4863)	0x63	A signed 16-bit integer value that specifies a zero-based index into **SttbfRMark**. This value MUST be greater than or equal to zero and MUST be less than **SttbfRMark.cData**. The string at this index is the name of the author who deleted the text. This is recorded only if revision marking is on at the time of deletion. By default, this index is zero, which is the index of the "Unknown" author.
sprmCDttmRMarkDel (0x6864)	0x64	A **DTTM** value that specifies the date and time at which the text was deleted. This is recorded only if revision marking is on at the time of the deletion. By default, the date is 1/1/1900 and the time is 00:00:00.
sprmCBrc80 (0x6865)	0x65	A **Brc80** value that specifies the border of all four sides of the text. The **logical left** border is hidden if the previous character on the same line has the same border as this character. The **logical right** border is hidden if the next character on the same line has the same border as this character. By default, text has no border. **Brc80.dptSpace** MUST be ignored when applied to character borders.
sprmCShd80 (0x4866)	0x66	A **Shd80** structure that specifies the background shading for the text. By default, text is not shaded.
sprmCIdslRMarkDel (0x4867)	0x67	An unsigned 16-bit integer that specifies the reason why the text under revision was deleted. This is recorded only if revision marking is on at the time when the text is edited. This value MUST be one of the following. 0x0000 – Performed a normal edit. 0x0001 – Applied a style. 0x0002 – Adjusted alignment with a tab. 0x0003 – Adjusted alignment with a tab. 0x0004 – Removed extra paragraph mark. 0x0005 – Replaced all caps with mixed caps. 0x0006 – Replaced bullet character with bullet symbol. 0x0007 – Replaced straight quote with smart quote. 0x0008 – Replaced multiple-character symbol with single symbol. 0x0009 – Replaced text with trademark symbol. 0x000A – Replaced text with copyright symbol. 0x000B – Replaced text with registered trademark symbol. 0x000C – Adjusted spaces after period. 0x000D – Replaced numbers with fraction symbol. 0x000E – Applied a heading style. 0x000F – Applied an outline style. 0x0010 – Applied a list style.

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		0x0011 – Applied a memo header style. 0x0012 – Applied an address style. 0x0013 – Applied a salutation style. 0x0014 – Applied a closing phrase style. 0x0015 – Applied a date style. 0x0016 – Applied a distribution list style. 0x0017 – Applied a bullet list style. 0x0018 – Applied a column style. 0x0019 – Applied a carbon copy style. 0x001A – Replaced text with superscript. 0x001B – Replaced whitespace galley with tabs. 0x001C – Removed leading whitespace. 0x001D – Removed manual numbering. 0x001E – Replaced two hyphens with long (em) dash. 0x001F – Adjusted spaces before: '!', '?', or ';' 0x0020 – Inserted paragraph mark. 0x0021 – Replaced leading whitespace to first line indent. 0x0022 – Removed space between a double byte character and a single byte character. 0x0023 – Replaced to match to open parenthesis. 0x0024 – Replaced double byte to single byte. 0x0025 – Replaced single byte to double byte. 0x0026 – Replaced manual emphasis. 0x0027 – Replaced border characters with borders 0x0028 – Replaced e-mail history characters with indentation. 0x0029 – Replaced URL or UNC with hyperlink. 0x002A – Replaced Gateway-generated hex characters. 0x002B – Applied outline level for document map. By default, the reason for the revision is "Performed a normal edit."
sprmCFUsePgsuSettings (0x0868)	0x68	A **ToggleOperand** value that specifies whether the text is to be displayed according to the **document grid**. By default, text uses the document grid if one is defined. (See sprmSClm for more details about the document grid.)
sprmCRgLid0_80 (0x486D)	0x6D	A **LID** value that specifies the language of the text, except for East Asian languages. East Asian languages are specified by sprmCRgLid1_80. By default, the text language is undefined.
sprmCRgLid1_80	0x6E	A **LID** value that specifies the language of the text if it is an East Asian

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
(0x486E)		language. Other languages are specified by sprmCRgLid0_80. By default, the text language is undefined.
sprmCIdctHint (0x286F)	0x6F	An 8-bit unsigned integer value that specifies which of the language, font, size, bold, and italic properties is to be used for handling the text, in the case where this cannot be derived from the characters themselves. The valid values and their meanings are specified as follows. These meanings correspond to the values of the ST_Hint type specified in [ECMA-376] Part 1, Section 17.18.41. **0x00 - default** Use sprmCRgLid0 (or sprmCRgLid0_80) for language. Use sprmCRgFtc0 for font if the character is between 0x0020 and 0x007F, inclusive. Otherwise, use sprmCRgFtc2. Use sprmCHps for size, sprmCFBold for bold, and sprmCFItalic for italic. **0x01 - eastAsia** Use sprmCRgLid1 (or sprmCRgLid1_80) for language, sprmCRgFtc1 for font, sprmCHps for size, sprmCFBold for bold, and sprmCFItalic for italic. **0x02 - cs** Use sprmCLidBi for language, sprmCFtcBi for font, sprmCHpsBi for size, sprmCFBoldBi for bold, and sprmCFItalicBi for italic. **0xFF - No ST_Hint equivalent** Provides no guidance on how to treat ambiguous text.
sprmCCv (0x6870)	0x70	A **COLORREF** value that specifies the color of the text. The default text color is **cvAuto**.
sprmCShd (0xCA71)	0x71	A **SHDOperand** value that specifies the background shading for the text. By default, text is not shaded.
sprmCBrc (0xCA72)	0x72	A **BrcOperand** value that specifies the border on all four sides of the text. The logical left border is hidden if the previous character on the same line has the same border as this character. The logical right border is hidden if the next character on the same line has the same border as this character. By default, text has no border. **Brc.dptSpace** MUST be ignored when applied to character borders.
sprmCRgLid0 (0x4873)	0x73	A **LID** value that specifies the language of the text, except for East Asian languages. East Asian languages are specified by sprmCRgLid1. By default, the text language is undefined and text is not checked for spelling, grammar, or hyphenation.
sprmCRgLid1 (0x4874)	0x74	A **LID** value that specifies the language of the text if it is an East Asian language. Other languages are specified by the sprmCRgLid0. By default, the text language is undefined and text is not checked for spelling, grammar, or hyphenation.
sprmCFNoProof (0x0875)	0x75	A **ToggleOperand** value that specifies whether the text is excluded from the proofing analysis. By default, text is not excluded from the proofing analysis.
sprmCFitText (0xCA76)	0x76	A **CFitTextOperand** value that specifies a width, in twips, to which text is expanded or condensed to fit. By default, text is not modified to fit into a specific width.
sprmCCvUl (0x6877)	0x77	A **COLORREF** value that specifies the color of the text underline. The default underline color is **cvAuto**.
sprmCFELayout (0xCA78)	0x78	A **FarEastLayoutOperand** value that specifies text layout information for East Asian languages. By default, text layout is unchanged by the **sprmCFELayout** value.
sprmCLbcCRJ (0x2879)	0x79	An **LBCOperand** value that specifies that this character is a special character representing a line break of the given type. The presence of a line break character means that the line ends at this point and that the rest of the text

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		continues on another line even though it is part of the same paragraph. This **Sprm** MUST NOT be applied to any character other than a line break character (Unicode 0x000B). By default, text restarts at the beginning of the next line after a line break character.
sprmCFComplexScripts (0x0882)	0x82	A **ToggleOperand** value that specifies whether complex script formatting (for example, see sprmCFBoldBi) is applied to the text regardless of the Unicode characters themselves. By default, characters are evaluated to determine whether complex script formatting is applied.
sprmCWall (0x2A83)	0x83	A **Bool8** value that specifies whether the values of character properties are preserved for revision-marking purposes until the modifications are accepted or rejected by the user. A value of 1 specifies that the values of properties are preserved. All character **SPRM**s that are encountered before the **sprmCWall** in the text property evaluation specify the state of the character properties before revision marking is enabled, whereas all character **SPRM**s that follow the **sprmCWall** specify the character property modifications that occur after revision marking is enabled. A value of 0 specifies that no values have been preserved (overriding any previously encountered sprmCWall **SPRM**s that specify the contrary). Neither **SPRM**s encountered before the sprmCWall, nor subsequent **SPRM**s (until another sprmCWall, if any), are treated in any special way with regard to revision marking. By default, values of properties are not preserved.
sprmCCnf (0xCA85)	0x85	A **CNFOperand** that specifies conditional character formatting for a specific condition of a **table style**. The **grpprl** member of **CNFOperand** specifies the character formatting properties and MUST NOT contain any **Sprm**s that are disallowed in the **grpprlChpx** member of **UpxChpx**. This **sprm** MUST only be specified within the **grpprlChpx** member of a **UpxChpx** within a table style definition (**LPStd**). By default, a table style definition does not include conditional formatting.
sprmCNeedFontFixup (0x2A86)	0x86	An **FFM** that specifies the type of font substitution that is needed for the associated text. Font substitution is needed when certain language characters are not supported by the current font for the text, so it is necessary to pick a different font that supports the characters. By default, text is not marked as requiring font substitution.
sprmCPbiIBullet (0x6887)	0x87	A CP value in the Bullet Pictures document that specifies which picture is used as a bullet character when rendering the bullet. The CP value MUST be greater than or equal to zero. The Bullet Pictures document is stored within the main document and marked by a hidden bookmark (1) called "_PictureBullets." This **Sprm** MUST NOT be applied to any character other than a paragraph mark (Unicode 0x000D), a cell mark (Unicode 0x0007), or a section mark (Unicode 0x000C). If a picture bullet is used, sprmCPbiGrf MUST be present to specify the properties of the picture bullet. By default, pictures are not used for rendering bullets.
sprmCPbiGrf (0x4888)	0x88	A **PbiGrfOperand** value that specifies whether a picture is used as a bullet character when rendering the bullet. This value also specifies the properties of the picture bullet. This **Sprm** MUST NOT be applied to any character other than a paragraph mark (Unicode 0x000D). If a picture bullet is used, sprmCPbiIBullet MUST be present to specify the location of the picture that is used for the bullet. By default, pictures are not used to render bullets.
sprmCPropRMark (0xCA89)	0x89	A **PropRMarkOperand** value that specifies whether the text has an associated property revision mark, as well as its author and the date and time. By default, text has no property revision marks.
sprmCFSdtVanish (0x2A90)	0x90	A **Bool8** value that specifies whether the text is hidden from display when the option to hide arbitrary XML delimiters is enabled. This value MUST NOT

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	operand
		be applied to any characters other than '<' (U+003C) or '>' (U+003E) with sprmCFSpec set to "true". By default, text is not hidden when the option to hide XML delimiters is enabled.

2.6.2 Paragraph Properties

A **Prl** with a **sprm.sgc** of 1 modifies a paragraph property.

The following table specifies the paragraph property modifiers, including the valid **sprm** values, their function, and the corresponding **operand** type and meaning.

sprm	ispmd	Operand
sprmPIstd (0x4600)	0x00	An unsigned integer that specifies the **istd** of a paragraph style to apply. To apply the **istd**, fetch the complete set of paragraph and character properties from that style. (See Applying Properties for instructions.) Apply those properties to the current paragraph, while preserving the previous values of the following: ▪ Whether the paragraph is a Table Terminating Paragraph Mark (for example, by sprmPFTtp). (See Overview of Tables). ▪ Whether the paragraph is in a table (for example, by sprmPFInTable). ▪ The **table depth** of the paragraph (for example, by sprmPItap). ▪ Whether the paragraph is the final paragraph in a nested table cell (for example, by sprmPFInnerTableCell). ▪ The table style applied to the paragraph (for example, by sprmTIstd). ▪ The **ipgpSelf** value of the PGPInfo data that is applied to the paragraph (for example, by sprmPIpgp). ▪ Paragraph properties that have been preserved for revision marking (for example, by sprmPWall) See sprmPWall for the meaning of revision marking. ▪ The revision save ID that is associated with the paragraph (for example, by sprmPRsid), as specified in [ECMA-376] Part 1, Section 17.15.1.70. ▪ Whether the paragraph has an associated **property revision mark**, as well as its author and the date and time (for example, by sprmPPropRMark). ▪ The numbering revision mark for the paragraph (for example, by sprmPNumRM). ▪ Whether a numbered list was applied to the paragraph after the previous revision (for example, by sprmPFNumRMIns). An **istd** value in the range of 1 to 9, inclusive, also specifies the **outline level** of the paragraph (for example, by sprmPOutLvl), where the new outline level is equal to the value of the **istd** minus 1. If an **istd** value refers to an empty or nonexistent style, or to a style of a different type, a later Prl such as sprmPIstd or sprmPIstdPermute MUST change the **istd** to a valid value. Applying an **istd** that refers to an empty or nonexistent style, or to a style of a different type, is equivalent to applying the paragraph and character document default formatting (while preserving the same set of properties as when applying an **istd**.) By default, the paragraph style is unchanged.
sprmPIstdPermute (0xC601)	0x01	A SPPOperand value that specifies a potential change in the current paragraph style (**istd**).

[MS-DOC] - v20250218
Word (.doc) Binary File Format
Copyright © 2025 Microsoft Corporation
Release: February 18, 2025

sprm	ispmd	Operand
		If the **istd** is not affected, this Prl MUST be ignored. If the **istd** is affected, this **sprm** is equivalent to sprmPIstd with the operand being the new **istd**.
sprmPIncLvl (0x2602)	0x02	A signed 8-bit integer value. If the paragraph has an **istd** that is greater than or equal to 0x0001 and less than or equal to 0x0009, this value specifies an offset to the **istd** of the paragraph. If this value offsets the **istd** of the paragraph beyond one of the limits 0x0001 or 0x0009, then the **istd** of the paragraph is set to that limit. See Determining Formatting Properties for information about how to determine the **istd** of the paragraph. If the **istd** of the paragraph is not within the range that was specified earlier, this value specifies an offset to the outline level of the paragraph, unless the outline level of the paragraph is equal to 0x09, in which case this value MUST be ignored. If this offset adjusts the outline level beyond one of the limits 0x00 or 0x09, than the outline level of the paragraph is set to that limit. See sprmPOutLvl for the outline level of the paragraph.
sprmPJc80 (0x2403)	0x03	An unsigned 8-bit integer that specifies the physical justification of the paragraph. This MUST be one of the following values. 0 - Paragraph is physically left justified. 1 - Paragraph is centered. 2 - Paragraph is physically right justified. 3 - Paragraph is justified to both right and left with a low character compression ratio. 4 - Paragraph is justified to both right and left with a medium character compression ratio. 5 - Paragraph is justified to both right and left with a high character compression ratio. By default, paragraphs are physically left-justified.
sprmPFKeep (0x2405)	0x05	A Bool8 value that specifies whether an application SHOULD<148> keep this paragraph on a single page. By default, paragraphs are allowed to split across pages.
sprmPFKeepFollow (0x2406)	0x06	A Bool8 value that specifies whether an application SHOULD<149> keep the end of this paragraph on the same page as the beginning of the next paragraph. By default, adjacent paragraphs are allowed to be on separate pages.
sprmPFPageBreakBefore (0x2407)	0x07	A Bool8 value that specifies whether this paragraph has a page break before it. By default, paragraphs do not have page breaks before them.
sprmPIlvl (0x260A)	0x0A	An unsigned 8-bit integer that specifies the **list level** of the paragraph. This value MUST be ignored if this paragraph is not in a list (see sprmPIlfo). This value MUST be one of the following: **0x0 - 0x8** The value specifies the zero-based level of the list that contains this paragraph. For example, a value of 0x0 means that the paragraph is in the first level of the list. **0xC** The list skips this paragraph and does not include it in its numbering. By default, a paragraph is in the first level of the list.

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
sprmPIlfo (0x460B)	0x0B	A 16-bit signed integer value that is used to determine which list contains the paragraph. This value MUST be one of the following: **0x0000** This paragraph is not in a list, and any list formatting on the paragraph is removed. **0x0001 - 0x07FE** The value is a 1-based index into PlfLfo.rgLfo. The LFO at this index defines the list that this paragraph is in. **0xF801** This paragraph is not in a list. **0xF802 - 0xFFFF** The value is the negation of a 1-based index into PlfLfo.rgLfo. The LFO at this index defines the list that this paragraph is in. The logical left indentation (see sprmPDxaLeft) and the logical left first line indentation (see sprmPDxaLeft1) of the paragraph MUST be preserved despite any list formatting. By default, a paragraph is not in a list.
sprmPFNoLineNumb (0x240C)	0x0C	A Bool8 value that specifies whether this paragraph is ignored when the application counts or displays **line numbers**. By default, if line numbers are enabled, paragraphs have line numbers.
sprmPChgTabsPapx (0xC60D)	0x0D	A PChgTabsPapxOperand value that specifies custom tab stops to be added or ignored. By default, custom tab stops are neither added nor ignored.
sprmPDxaRight80 (0x840E)	0x0E	An XAS value that specifies the **physical right** indent of the paragraph, in **twips**. By default, there is no right indentation.
sprmPDxaLeft80 (0x840F)	0x0F	An XAS value that specifies the **physical left** indent of the paragraph, in twips. By default, there is no left indentation.
sprmPNest80 (0x4610)	0x10	An XAS value that is added to sprmPDxaLeft80 to specify the final indent of a paragraph. By default, there is no additional space added to sprmPDxaLeft80 to determine the final indent of a paragraph.
sprmPDxaLeft180 (0x8411)	0x11	An XAS value that specifies the logical left indent of the first line of the paragraph, in twips, relative to the rest of the paragraph. By default, the first line is not indented relative to the rest of the paragraph.
sprmPDyaLine (0x6412)	0x12	An LSPD value that specifies the spacing between lines in this paragraph. By default, paragraphs use single spacing.
sprmPDyaBefore (0xA413)	0x13	A two-byte unsigned integer value that specifies the size, in twips, of the spacing before this paragraph. The value MUST be a number between 0x0000 and 0x7BC0, inclusive. When auto-spacing is supported and the value of sprmPFDyaBeforeAuto is 1, this property is ignored. By default, the space before a paragraph is zero twips.
sprmPDyaAfter (0xA414)	0x14	A two-byte unsigned integer value that specifies the size, in twips, of the spacing after this paragraph. The value MUST be between 0x0000 and 0x7BC0, inclusive. When **auto spacing** is supported and the value of sprmPFDyaAfterAuto is 1, this property is ignored. By default, the space after a paragraph is zero twips.
sprmPChgTabs (0xC615)	0x15	A PChgTabsOperand value that specifies custom tab stops that are added or ignored. By default, custom tab stops are neither added nor ignored.
sprmPFInTable (0x2416)	0x16	A Bool8 value that specifies whether this paragraph is in a table. The value MUST be 1 any time the table depth is greater than zero. See section 2.4.3, Overview of Tables. By default, paragraphs are not in tables.
sprmPFTtp (0x2417)	0x17	A Bool8 that, when set to 1, specifies that the cell mark it is applied to is a Table Terminating Paragraph (TTP) mark. The TTP mark MUST be

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
		immediately preceded by a cell mark. See Overview of Tables. By default, a cell mark is not a Table Terminating Paragraph Mark.
sprmPDxaAbs (0x8418)	0x18	A XAS_plusOne that specifies the **logical left** horizontal position relative to the horizontal anchor of the **frame**. See sprmPPc for the frame anchor. If the value is any of the those that follow, the operand specifies a special descriptive, relative position. The meanings that are provided correspond to the values that are specified in [ECMA-376] Part 1, Section 22.9.2.18 ST_XAlign (Horizontal Alignment Location): 0x0000 - left 0xFFFC - center 0xFFF8 - right 0xFFF4 - inside 0xFFF0 - outside By default, the relative horizontal position is **Left**.
sprmPDyaAbs (0x8419)	0x19	A YAS_plusOne value that specifies downward vertical position relative to the vertical anchor of the frame. See sprmPPc for the frame anchor. If the value is any of those that follow, the operand specifies a special descriptive, relative position. The meanings that are provided correspond to the values that are specified in [ECMA-376] Part 1, Section 22.9.2.20 ST_YAlign (Vertical Alignment Location). 0x0000 - inline 0xFFFC - top 0xFFF8 - center 0xFFF4 - bottom 0xFFF0 - inside 0xFFEC - outside By default, the relative vertical position is 0x0000 (**Inline**).
sprmPDxaWidth (0x841A)	0x1A	A XAS_nonNeg value that specifies the width of the frame. If the operand value is 0, the width of the frame is automatically determined by the maximum line width of the content that is within the frame. By default, the width of the frame is automatically determined by the maximum line width of the content the frame contains.
sprmPPc (0x261B)	0x1B	A PositionCodeOperand that specifies the anchor from which the frame position is calculated.

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
sprmPWr (0x2423)	0x23	A 1-byte integer that specifies how text is wrapped around a frame. Its value MUST be one of those that follow, corresponding to the values of **ST_Wrap** that are specified in [ECMA-376] Part 1, Section 17.18.104 ST_Wrap (Text Wrapping around Text Frame type). **0x00** **ST_Wrap: auto** This value specifies automatic text wrapping. **0x01** **ST_Wrap: notBeside** This value specifies that there is no text wrapping to either side of the frame. **0x02** **ST_Wrap: around** This value specifies that text is wrapped around the frame. **0x03** **ST_Wrap: none** Text is not wrapped around the frame. **0x04** **ST_Wrap: tight** This value specifies that text is tightly wrapped around the frame. **0x05** **ST_Wrap: through** This value specifies that text is wrapped through the frame, to the contours of the contents of the frame. By default, text is automatically wrapped around a frame.
sprmPBrcTop80 (0x6424)	0x24	A Brc80 value that specifies the top border of the paragraph. This border is hidden if the previous paragraph is identical to this one in terms of its top, bottom, left, and right borders; its left and right indents; its table depth; and its sprmPIpgp value. By default, paragraphs have no top border.
sprmPBrcLeft80 (0x6425)	0x25	A Brc80 value that specifies the logical left border of the paragraph. By default, paragraphs have no logical left border.
sprmPBrcBottom80 (0x6426)	0x26	A Brc80 value that specifies the bottom border of the paragraph. This border is hidden if the next paragraph is identical to this one in terms of its top, bottom, left, and right borders; its left and right indents; its table depth; and its sprmPIpgp value. By default, paragraphs have no bottom border.
sprmPBrcRight80 (0x6427)	0x27	A Brc80 value that specifies the **logical right** border of the paragraph. By default, paragraphs have no logical right border.
sprmPBrcBetween80 (0x6428)	0x28	A Brc80 value that specifies the border between this paragraph and the next. This border is hidden unless the next paragraph is identical to this one in terms of its top, bottom, left, and right borders; its left and right indents; its table depth; and its sprmPIpgp value. By default, paragraphs have no borders between them.
sprmPBrcBar80 (0x6629)	0x29	A Brc80 value that has no effect.
sprmPFNoAutoHyph (0x242A)	0x2A	A Bool8 value that specifies whether this paragraph is **auto-hyphenated** when hyphenation is enabled for the document. A value of 1 specifies that this paragraph is not auto-hyphenated when hyphenation is enabled for the document. A value of 0 specifies that this paragraph is auto-hyphenated when hyphenation is enabled for the document. By default, paragraphs are auto-hyphenated when

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
		hyphenation is enabled for the document. Document hyphenation is enabled when the **fAutoHyphen** field of the DopBase structure is 1.
sprmPWHeightAbs (0x442B)	0x2B	A WHeightAbs value that specifies the height of the frame. By default, the height of a frame height is automatically determined based on the height of its contents.
sprmPDcs (0x442C)	0x2C	A DCS value that specifies the properties, if any, of the drop cap for this paragraph. By default, paragraphs do not have a drop cap.
sprmPShd80 (0x442D)	0x2D	A Shd80 structure that specifies the background shading for the paragraph. By default, paragraphs are not shaded.
sprmPDyaFromText (0x842E)	0x2E	A YAS_nonNeg value that specifies the minimum vertical distance between the edge of the frame and the edge of main document text that wraps around the frame. By default, the minimum vertical distance is 0 twips.
sprmPDxaFromText (0x842F)	0x2F	A XAS_nonNeg value that specifies the minimum horizontal distance between the edge of the frame and the edge of main document text that wraps around the frame. By default, the minimum horizontal distance is 0 twips.
sprmPFLocked (0x2430)	0x30	A Bool8 value that specifies whether the anchor of the frame which contains this paragraph is locked to its current location. By default, the frame anchor is not locked. This Sprm corresponds to the **anchorLock** attribute as specified in [ECMA-376] Part 1, Section 17.3.1.11 framePr (Text Frame Properties)
sprmPFWidowControl (0x2431)	0x31	A Bool8 value that specifies whether widow and orphan control is enabled for this paragraph. An orphaned line is the first line of a paragraph when it is displayed by itself at the bottom of a page. A widowed line is the last line of a paragraph when it is displayed by itself at the top of a page. When widow and orphan control is enabled, the application attempts to eliminate widowed and orphaned lines. By default, widow and orphan control is enabled.
sprmPFKinsoku (0x2433)	0x33	A Bool8 value that specifies whether this paragraph uses East Asian typography and line-breaking rules to determine the valid characters that are allowed to begin and end each line of East Asian text. These rules are specified in [ECMA-376] Part 1, Section 17.3.1.16 kinsoku paragraph property. By default, paragraphs use East Asian rules to determine the allowed values for the first and last characters of each line of text.
sprmPFWordWrap (0x2434)	0x34	A Bool8 value that, when equal to 0, specifies a preference to break Latin text that exceeds text line limits by breaking a word across two lines (breaking on the character level). If the language used is Korean, this property affects Korean text instead of Latin text. By default, the word is placed on the following line (breaking on the word level).
sprmPFOverflowPunct (0x2435)	0x35	A Bool8 value that, when equal to 0, specifies a preference against allowing a punctuation character that follows a word at the end of a line to appear beyond the extent of that line of text. By default, a single punctuation character that follows a word can appear beyond the extent of a line.
sprmPFTopLinePunct (0x2436)	0x36	A Bool8 value that specifies a preference to render punctuation characters at the beginning of a line so that they appear to be closer to both the beginning of the line and to the next character, regardless of the amount of whitespace in the glyph as defined by the font. By default, punctuation is rendered normally.
sprmPFAutoSpaceDE (0x2437)	0x37	A Bool8 value that specifies whether space is automatically inserted between East Asian and Latin text. By default, this option is enabled.
sprmPFAutoSpaceDN (0x2438)	0x38	A Bool8 value that specifies whether space is automatically inserted between East Asian text and numbers. By default, this option is enabled.

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
sprmPWAlignFont (0x4439)	0x39	A 16-bit unsigned integer that specifies vertical font alignment for East Asian languages. This Sprm corresponds to the **textAlignment** paragraph property that is specified in [ECMA-376] Part 1, Section 17.3.1.39. This value MUST be one of the following, corresponding to the values of ST_TextAlignment that are specified in [ECMA-376] Part 1, Section 17.18.91. **0x0000** **ST_TextAlignment: top** This value specifies that characters are aligned based on the top of each character. **0x0001** **ST_TextAlignment: center** This value specifies that characters are centered on the line. **0x0002** **ST_TextAlignment: baseline** This value specifies that characters are aligned based on their baseline. This is how standard Latin text is displayed. **0x0003** **ST_TextAlignment: bottom** This value specifies that characters are aligned based on the bottom of each character. **0x0004** **ST_TextAlignment: auto** This value specifies that alignment is automatically determined by the application. By default, font alignment is automatically determined by the application.
sprmPFrameTextFlow (0x443A)	0x3A	A FrameTextFlowOperand that specifies the direction of text flow in the frame. If this property is set, then at least one of the following paragraph properties MUST be set with a non-default value: ▪ sprmPDxaAbs ▪ sprmPDyaAbs ▪ sprmPDxaWidth ▪ sprmPPc ▪ sprmPWr ▪ sprmPWHeightAbs By default, paragraph text flows horizontally, without rotation.
sprmPOutLvl (0x2640)	0x40	An unsigned 8-bit integer value that specifies the outline level of the paragraph. This value MUST be one of the following. **0x0 - 0x8** The value is the zero-based outline level that this paragraph is in. **0x9** The paragraph at any outline level; instead, the paragraph is body text. This MUST be ignored if the paragraph has an **istd** that is greater than or equal to 0x1 and less than or equal to 0x9. By default, paragraphs are body text, and are therefore not in any outline level.
sprmPFBiDi (0x2441)	0x41	A Bool8 value that specifies whether the paragraph uses **right-to-left** layout. By default, a paragraph does not use right-to-left layout.
sprmPFNumRMIns	0x43	A Bool8 value that specifies whether a numbered list was applied to

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
(0x2443)		this paragraph after the previous revision. By default, paragraphs do not have numbered lists applied.
sprmPNumRM (0xC645)	0x45	A NumRMOperand value that specifies a numbering revision mark for this paragraph. By default, paragraphs do not have numbering revision marks.
sprmPHugePapx (0x6646)	0x46	A 4-byte unsigned integer that specifies a location in the Data Stream. A PrcData structure begins at this offset and specifies additional properties for the paragraph. The **cbGrpprl** member of the referenced PrcData structure MUST NOT be less than 10. If an application processes this PrcData, then it MUST NOT process any more Prl elements in the array that contained the sprmPHugePapx. If a Prl with a **sprm** of **sprmPHugePapx** is in an array of Prl elements and is not the first element of the array, then that Prl MUST be ignored. If a Prl with a **sprm** of **sprmPHugePapx** is contained in the **grpprl** array of a GrpPrlAndIstd structure, then it MUST be the only Prl in that array and the **istd** member of that GrpPrlAndIstd structure MUST be zero. The **sprmPHugePapx** and sprmPTableProps values can refer to PrcDatas containing each other, but the chain MUST eventually terminate in a PrcData structure does not contain a **sprmPHugePapx** value or a sprmPTableProps value.
sprmPFUsePgsuSettings (0x2447)	0x47	A Bool8 value that specifies whether the paragraph adheres to the vertical components of the **document grid**. By default, text uses the document grid if one is defined. (See sprmSClm for more details about the document grid.)
sprmPFAdjustRight (0x2448)	0x48	A Bool8 value that specifies whether this paragraph is set to automatically adjust the right indent when a document grid for **East Asian characters** is defined. This Sprm is the same as the **adjustRightInd** paragraph property specified in [ECMA-376] Part 1, Section 17.3.1.1. By default, this option is enabled.
sprmPItap (0x6649)	0x49	An integer value that specifies the table depth of this paragraph. See the Overview of Tables (section 2.4.3) for the rules that this value follows. This value, when present, MUST be a non-negative number. By default, paragraphs are not in tables.
sprmPDtap (0x664A)	0x4A	A signed integer that specifies an addition or subtraction to the existing table depth of this paragraph. It provides an alternate way of specifying table depth to sprmPItap or a way to increment or decrement any value that was already set by sprmPItap or **sprmPDtap**. The resultant table depth MUST be non-negative and MUST obey the rules described in Overview of Tables (section 2.4.3). By default, paragraphs are not in tables.
sprmPFInnerTableCell (0x244B)	0x4B	A Bool8 value that specifies whether this paragraph is the final paragraph in a nested table cell. When **true**, the nesting level of this paragraph MUST be greater than 1, indicating that this paragraph is in a table which is nested within another table. When **true**, this is the last paragraph of a nested table cell and its **paragraph mark** is treated as if it were an **end of cell mark**. By default, paragraphs are not the last paragraph of a nested table cell. See the Overview of Tables (section 2.4.3) for more information about nested tables.
sprmPFInnerTtp (0x244C)	0x4C	A Bool8 value that specifies whether this paragraph is the final paragraph in a nested table row. When 1, the table depth of this paragraph MUST be greater than 1, indicating that this paragraph is in a table that is nested within another table. When 1, this is the last paragraph of a nested table row and its paragraph mark is treated as if it were a TTP mark. By default, paragraphs are not the last paragraph of a nested table row. See the Overview of Tables for more information

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
		about nested tables.
sprmPShd (0xC64D)	0x4D	A SHDOperand value that specifies the background shading for the paragraph. By default, paragraphs are not shaded.
sprmPBrcTop (0xC64E)	0x4E	A BrcOperand value which specifies the top border of the paragraph. This border is hidden if the previous paragraph is identical to this one in terms of its top, bottom, left, and right borders; its left and right indents; its table depth; and its sprmPIpgp value. By default, paragraphs have no top border.
sprmPBrcLeft (0xC64F)	0x4F	A BrcOperand value that specifies the logical left border of the paragraph. By default, paragraphs have no logical left border.
sprmPBrcBottom (0xC650)	0x50	A BrcOperand value that specifies the bottom border of the paragraph This border is hidden if the next paragraph is identical to this one in terms of its top, bottom, left, and right borders; its left and right indents; its table depth; and its sprmPIpgp value. By default, paragraphs have no bottom border.
sprmPBrcRight (0xC651)	0x51	A BrcOperand value that specifies the logical right border of the paragraph. By default, paragraphs have no logical right border.
sprmPBrcBetween (0xC652)	0x52	A BrcOperand value that specifies the border between this paragraph and the next. This border is hidden unless the next paragraph is identical to this one in terms of its top, bottom, left, and right borders, left and right indents, table depth, and sprmPIpgp value. By default, paragraphs have no borders between them.
sprmPBrcBar (0xC653)	0x53	A BrcOperand value that has no effect.
sprmPDxcRight (0x4455)	0x55	A signed 16-bit integer value that specifies the logical right indent of the paragraph in hundredths of **character units**. By default, there is no right indentation.
sprmPDxcLeft (0x4456)	0x56	A signed 16-bit integer value that specifies the logical left indent of the paragraph in hundredths of character units. By default, there is no left indentation.
sprmPDxcLeft1 (0x4457)	0x57	A signed 16-bit integer value that specifies the logical left indent of the first line of the paragraph, in hundredths of character units, relative to the rest of the paragraph. By default, the first line is not indented relative to the rest of the paragraph.
sprmPDylBefore (0x4458)	0x58	A signed 16-bit integer value that specifies the spacing before the paragraph, in 1/100 **line units**. This value MUST be at least -20 and MUST NOT exceed 31680. By default, paragraphs do not have spacing before them.
sprmPDylAfter (0x4459)	0x59	A signed 16-bit integer that specifies the spacing after the paragraph, in 1/100 line units. MUST be at least -20 and MUST NOT exceed 31680. By default, paragraphs do not have spacing after them.
sprmPFOpenTch (0x245A)	0x5A	A Bool8 value that specifies whether this table cell mark was being displayed when this file was last saved, even though it immediately follows a nested table.
sprmPFDyaBeforeAuto (0x245B)	0x5B	A Bool8 value that specifies whether the space displayed before this paragraph uses auto spacing. A value of 1 specifies that the sprmPDyaBefore value MUST be ignored when the application supports auto spacing. By default, auto spacing is disabled for paragraphs.
sprmPFDyaAfterAuto (0x245C)	0x5C	A Bool8 value that specifies whether the space displayed after this paragraph uses auto spacing. A value of 1 specifies that sprmPDyaAfter MUST be ignored if the application supports auto spacing. By default, auto spacing is disabled for paragraphs.
sprmPDxaRight (0x845D)	0x5D	An XAS value that specifies the logical right indent of the paragraph, in twips. By default, there is no right indentation.
sprmPDxaLeft	0x5E	An XAS value that specifies the logical left indent of the paragraph, in

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
(0x845E)		twips. By default, there is no left indentation.
sprmPNest (0x465F)	0x5F	An XAS value that is added to the sprmPDxaLeft value to determine the final indent of a paragraph. By default, there is no additional space added to sprmPDxaLeft to determine the final indent of a paragraph. When present, this Sprm supersedes any value for sprmPNest80.
sprmPDxaLeft1 (0x8460)	0x60	An XAS value that specifies the logical left indent of the first line of the paragraph, in twips, relative to the rest of the paragraph. By default, the first line is not indented relative to the rest of the paragraph.
sprmPJc (0x2461)	0x61	An unsigned 8-bit integer value that specifies the logical justification of the paragraph. The value MUST be one of those listed following. Some of the values also correspond to the **ST_Jc** enumeration values that are specified in [ECMA-376] Part 1, Section 17.18.44 ST_Jc (Horizontal Alignment Type). **0** **St_Jc: left** Paragraph is logical left justified **1** **St_Jc: center** Paragraph is centered **2** **St_Jc: right** Paragraph is logical right justified **3** **St_Jc: both** Paragraph is justified to both right and left **4** **St_Jc:distribute** Paragraph characters are distributed to fill the entire width of the paragraph **5** **St_Jc: mediumKashida** If the language is Arabic, the paragraph uses medium-length Kashida. In other languages, text is justified with a medium character compression ratio. **6** Paragraph is indented **7** **St_Jc: highKashida** If the language is Arabic, the paragraph uses longer length Kashida. In other languages, text is justified with a high character compression ratio. **8** **St_Jc: lowKashida** If the language is Arabic, the paragraph uses small length Kashida. In other languages, text is justified with a high character compression ratio. **9** **St_Jc:thaiDistribute** If the language of the paragraph is Thai, the text is justified with Thai distributed justification. In other languages, text is justified with a low character compression ratio. The default is logical left justification.
sprmPFNoAllowOverlap (0x2462)	0x62	A Bool8 value that specifies whether the frame of this paragraph can overlap with other frames. A value of 1 specifies that frames MUST NOT overlap. By default, frames can overlap with other frames.
sprmPWall (0x2664)	0x64	A Bool8 value that specifies whether the values of paragraph properties are preserved for revision marking purposes until the modifications are accepted or rejected by the user.

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
		A value of 1 specifies that the property values were preserved. All SPRMs that are encountered before the **sprmPWall** in the property evaluation of the paragraph specify the state of properties before revision marking was enabled, whereas all SPRMs following the **sprmPWall** specify the property modifications that occur after revision marking was enabled. A value of 0 specifies that no values were preserved (overriding any previously encountered sprmPWall SPRMs that specify the contrary). Neither SPRMs that were encountered before the sprmPWall, nor subsequent SPRMs (until another sprmPWall, if any), are treated in any special way with regard to revision marking. By default, property values are not preserved.
sprmPIpgp (0x6465)	0x65	An unsigned integer value that specifies the PGPInfo.**ipgpSelf** value of the PGPInfo data that is applied to this paragraph. The table depth of the paragraph (see Overview of Tables) MUST match PGPInfo.**itap** unless the paragraph is a table terminating mark, in which case PGPInfo.**itap** MUST be 1 less than the paragraph table depth. PGPInfo.**ipgpSelf** values MUST NOT be applied in such a way as to break the hierarchy that is implied by the PGPInfo structures themselves. Given that the application of a particular PGPInfo.**ipgpSelf** value implies the application of all of the PGPInfo.**ipgpParent** values that are encountered by ascending the PGPInfo chain, ensuring that all occurrences of any PGPInfo.**ipgpSelf** are on adjacent paragraphs of the same table depth ensures that the hierarchy is not broken. By default, a paragraph has no associated PGPInfo.
sprmPCnf (0xC666)	0x66	A CNFOperand value that specifies conditional paragraph formatting for a specific condition of a **table style**. The **grpprl** member of the CNFOperand value specifies the paragraph formatting properties and MUST NOT contain any Sprms that are disallowed in the **grpprlPapx** member of UpxPapx. This sprm MUST only be specified within the **grpprlPapx** member of a UpxPapx within a table style definition (LPStd). By default, a table style definition does not include conditional formatting.
sprmPRsid (0x6467)	0x67	An integer value that specifies a revision save ID, as specified in [ECMA-376] Part 1, Section 17.15.1.70 rsid (Single Session Revision Save ID), associated with paragraph formatting. If not present, then no revision save ID is specified for this formatting.
sprmPIstdListPermute (0xC669)	0x69	An SPPOperand value that has no effect and MUST be ignored.
sprmPTableProps (0x646B)	0x6B	An unsigned integer value that specifies a location in the Data Stream. A PrcData structure begins at this offset and specifies additional properties for the paragraph. The **cbGrpprl** member of the referenced PrcData structure MUST NOT be less than 10. If an application processes this PrcData structure, then it MUST NOT process anymore Prl elements in the array that contained the **sprmPTableProps** value. SprmPHugePapx and **sprmPTableProps** values can refer to PrcData structures containing each other, but the chain MUST eventually terminate in a PrcData that contains neither sprmPHugePapx nor sprmPTableProps.
sprmPTIstdInfo (0xC66C)	0x6C	A PTIstdInfoOperand value that has no effect and MUST be ignored.
sprmPFContextualSpacing (0x246D)	0x6D	A Bool8 value that specifies whether contextual spacing is enabled for this paragraph. A value of 0x01 specifies that any space before this paragraph (sprmPDyaBefore) MUST be ignored if the preceding paragraph is of the same **paragraph style** and any space after this paragraph (sprmPDyaAfter) MUST be ignored if the following paragraph is of the same paragraph style. By default, paragraphs do not use contextual spacing.

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
sprmPPropRMark (0xC66F)	0x6F	A PropRMarkOperand value that specifies whether the paragraph has an associated property revision mark, as well as its author and the date and time. By default, paragraphs have no property revision marks.
sprmPFMirrorIndents (0x2470)	0x70	A Bool8 value that specifies whether the left and right indents that are set for this paragraph are interpreted as inside and outside margins for odd and even numbered pages. For specifications of the display behavior, see [ECMA-376] Part 1, Section 17.3.1.18 mirrorIndents (use Left/Right Indents as Inside/Outside Indents). By default, paragraph indents are not swapped.
sprmPTtwo (0x2471)	0x71	A 1-byte integer that specifies text wrapping options for a text box when tight wrapping is set for the text box. This option is the same as [ECMA-376] Part 1, Section 17.3.1.40 textboxTightWrap (Allow Surrounding Paragraphs to Tight Wrap to Text Box Contents) The value MUST be one of the following, which correspond to values specified in [ECMA-376] Part 1, Section 17.18.92 ST_TextboxTightWrap (Lines To Tight Wrap Within Text Box). **0x00** **ST_TextboxTightWrap: none** No lines of the paragraph allow the surrounding text to tightly wrap around their edges. **0x01** **ST_TextboxTightWrap: allLines** All lines of the paragraph allow the surrounding text to tightly wrap to their edges. **0x02** **ST_TextboxTightWrap: firstAndLastLine** Only the first and last lines of the paragraph allow the surrounding text to tightly wrap around their edges. **0x03** **ST_TextboxTightWrap: firstLineOnly** Only the first line of the paragraph allows the surrounding text to tightly wrap around its edges. **0x04** **ST_TextboxTightWrap: lastLineOnly** Only the last line of the paragraph allows the surrounding text to tightly wrap around its edges. By default, the surrounding text is not allowed to tightly wrap to the edges of the lines of a paragraph in a textbox.

2.6.3 Table Properties

A Prl with a **sprm.sgc** of 5 modifies a table property.

The following table specifies the table property modifiers, including the valid **sprm** values, their function, and the corresponding **operand** type and meaning.

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	Operand
sprmTJc90 (0x5400)	0x00	An unsigned 16-bit integer value that specifies the physical justification of the table. The valid values and their meanings are as follows. 0 - The table is **physical left** justified. 1 - The table is centered. 2 - The table is **physical right** justified. Tables do not have a default physical justification. Their default justification is **logical left**.
sprmTDxaLeft (0x9601)	0x01	An XAS value that, combined with sprmTDxaGapHalf, specifies the location of the horizontal origin of the table relative to the logical left margin. That is, the origin is the logical left margin, indented by this value minus the value of sprmTDxaGapHalf. The actual logical left edge of the table can be offset from the origin after also considering cell spacing, margins, and the line width of the border. The default logical left indent is 0.
sprmTDxaGapHalf (0x9602)	0x02	An XAS value that specifies the average width, in **twips**, between the left and right default cell margins for the first cell in the row. The actual cell margins are stored in sprmTCellPaddingDefault. This value is not used to layout cell contents within a cell. Rather, this value is used as an offset to the value in sprmTDxaLeft when positioning the logical left outer edge of the table. This value MUST be non-negative. By default, no offset is applied to sprmTDxaLeft when positioning the table.
sprmTFCantSplit90 (0x3403)	0x03	A Bool8 value. If this property is "true" (0x01) then table rows SHOULD NOT<150> be split across page breaks. By default, rows can be split across page breaks. Whenever cells are merged this property SHOULD<151> be set with a value of 0x01 for each row involved in the merge. This property SHOULD<152> be ignored and sprmTFCantSplit SHOULD<153> be used instead.
sprmTTableHeader (0x3404)	0x04	A Bool8 value that specifies that the current table row is a header row. If the value is 0x01 but sprmTTableHeader is not applied with a value of 0x01 for a previous row in the same table, then this property MUST be ignored. By default, a table row is not a header row.
sprmTTableBorders80 (0xD605)	0x05	A TableBordersOperand80 value that specifies border information for the cells in a table row. By default, table rows have no borders.
sprmTDyaRowHeight (0x9407)	0x07	A YAS value that specifies the height of the row. If this value is zero, the height of a row is derived from the height of the contents of the cells that the row contains. If this value is positive, then the value is treated as "at least", meaning the row is larger if the contents need more space. If this value is negative, then the absolute value is used, and the size is treated as "exact". The row does not grow to accommodate large contents. By default, table row heights are derived from the heights of the contents of the cells in the row.
sprmTDefTable (0xD608)	0x08	A TDefTableOperand value that specifies the number of columns in the table row, the width of each column, border attributes, and a variety of other settings. By default, a table row has zero columns. In order for a table to have columns, the file MUST provide a sprmTDefTable or a sprmTInsert for each table row.
sprmTDefTableShd80 (0xD609)	0x09	A DefTableShd80Operand value that specifies the default shading applied to each cell in a row. By default, no cells are shaded.

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	Operand
		If the nFib value is greater than 0x00D9 and the application can interpret **table styles**, then this Sprm MUST be ignored.
sprmTTlp (0x740A)	0x0A	A TLP structure that specifies the table style options for this table. By default, tables have no table style associated with them and all optional table styles are disabled.
sprmTFBiDi (0x560B)	0x0B	A Bool16 value that specifies whether this table is **right-to-left**. A table is right-to-left if either this Sprm or sprmTFBiDi90 is set to "true". By default, tables are **left-to-right**.
sprmTDefTableShd3rd (0xD60C)	0x0C	A DefTableShdOperand that specifies the default shading for cells 45 to 63 in the row, or all remaining cells in the row beginning with cell 45 if the row contains fewer than 63 cells. **cb** MUST NOT exceed 190 and **rgShd** MUST NOT exceed 19 elements. Non-shaded cells in **rgShd** are set to **ShdAuto**. By default, no cells are shaded. Cells 1 – 22 are shaded by sprmTDefTableShd, and cells 23 – 44 are shaded by sprmTDefTableShd2nd. If the nFib value is greater than 0x00D9 and the application can interpret table styles, then this **Sprm** MUST be ignored.
sprmTPc (0x360D)	0x0D	A PositionCodeOperand structure that specifies the origin that is used to calculate the table position when it is absolutely positioned. By default, tables are not absolutely positioned. By default, when a table is absolutely positioned, its position is relative to the top margin of the page, and to the left edge of the current column.
sprmTDxaAbs (0x940E)	0x0E	A XAS_plusOne value that specifies the horizontal position of the table relative to the horizontal anchor of the table. See sprmTPc for the table anchor. Except for the reserved values that are listed in the following table, the sprmTDxaAbs specifies the position of the physical left origin of the table. It MUST be less than or equal to 31681 (22 inches) and greater than or equal to -31679 (-22 inches). Furthermore, after accounting for the basis specified in sprmTPc, the absolute position MUST be greater than or equal to 0 inches. Several values of sprmTDxaAbs have special meanings as specified by [ECMA-376] Part 1, Section 22.9.2.18. These values are specified as follows. 0x0000 - Left aligned 0xFFFC - Centered 0xFFF8 - Right aligned 0xFFF4 - Inside 0xFFF0 - Outside By default, the relative horizontal position is left aligned.
sprmTDyaAbs (0x940F)	0x0F	A YAS_plusOne value that specifies downward vertical position relative to the vertical anchor of the table. See sprmTPc for the table anchor. If the value is any of the those that follow, the operand specifies a special descriptive relative position. The meanings that are provided correspond to values that are defined in [ECMA-376] Part 1, Section 22.9.2.20 ST_YAlign (Vertical Alignment Location). 0x0000 - inline 0xFFFC - top 0xFFF8 - center 0xFFF4 - bottom 0xFFF0 - inside 0xFFEC - outside By default, the relative vertical position is 0x0000 (inline).

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	Operand
sprmTDxaFromText (0x9410)	0x10	An XAS_nonNeg value that specifies the minimum horizontal distance between the physical left edge of the table and the physical right edge of the text that wraps around the table. By default, the minimum horizontal distance between a table and wrapping text is 0 twips.
sprmTDyaFromText (0x9411)	0x11	A YAS_nonNeg value that specifies the minimum vertical distance between the top edge of the table and the bottom edge of text that wraps around the table. By default, the minimum vertical distance between a table and wrapping text is 0 twips.
sprmTDefTableShd (0xD612)	0x12	A DefTableShdOperand value that specifies the default shading for cells 1 – 22 in the row, or all cells in the row if the row contains fewer than 22 cells. Non-shaded cells in **rgShd** are set to **ShdAuto**. By default, no cells are shaded. Cells 23 – 44 are shaded by sprmTDefTableShd2nd, and cells 45 – 63 are shaded by sprmTDefTableShd3rd. If nFib is greater than 0x00D9 and the application understands table styles, then this Sprm MUST be ignored.
sprmTTableBorders (0xD613)	0x13	A TableBordersOperand value that specifies the borders for this row unless modified by other Sprms applied to the cells. By default, table rows have no borders.
sprmTTableWidth (0xF614)	0x14	An FtsWWidth_Table structure that specifies the preferred total width of the table of which this row is a part. By default, tables have no preferred width.
sprmTFAutofit (0x3615)	0x15	A Bool8 value that specifies whether the table column widths are to be automatically resized to best fit the contents of the whole table. By default, table column widths are not automatically resized.
sprmTDefTableShd2nd (0xD616)	0x16	A DefTableShdOperand that specifies the default shading for cells 23 – 44 in the row, or all remaining cells in the row beginning with cell 23 if the row contains fewer than 44 cells. Non-shaded cells in **rgShd** are set to **ShdAuto**. By default, no cells are shaded. Cells 1 – 22 are shaded by sprmTDefTableShd, and cells 45 – 63 are shaded by sprmTDefTableShd3rd. If nFib is greater than 0x00D9 and the application understands table styles, then this Sprm MUST be ignored.
sprmTWidthBefore (0xF617)	0x17	An FtsWWidth_TablePart structure that specifies the preferred additional leading indent of the first cell of the row, relative to the leading edge of the table as a whole. By default, table rows have no preferred additional leading indent.
sprmTWidthAfter (0xF618)	0x18	An FtsWWidth_TablePart structure that specifies the preferred trailing indent following the last cell of the row. The indent is inward from the outer edge of the table as a whole. By default, table rows have no preferred additional trailing indent.
sprmTFKeepFollow (0x3619)	0x19	A Bool8 value that specifies whether page breaks are avoided between the rows of this table, if possible. By default, tables are allowed to have page breaks.
sprmTBrcTopCv (0xD61A)	0x1A	A BrcCvOperand value that specifies the color of the top border for each cell in a table row. By default, each color is **cvAuto**.
sprmTBrcLeftCv (0xD61B)	0x1B	A BrcCvOperand value that specifies the color of the logical left border for each cell in a table row. By default, each color is **cvAuto**.
sprmTBrcBottomCv (0xD61C)	0x1C	A BrcCvOperand value that specifies the color of the bottom border for each cell in a table row. By default, each color is **cvAuto**.
sprmTBrcRightCv (0xD61D)	0x1D	A BrcCvOperand value that specifies the color of the logical right border for each cell in a table row. By default, each color is **cvAuto**.
sprmTDxaFromTextRight (0x941E)	0x1E	An XAS_nonNeg value that specifies the minimum horizontal distance between the physical right edge of the table and the physical left edge of the text that wraps around the table. By default, the minimum horizontal distance between a table and wrapping text is 0 twips.
sprmTDyaFromTextBottom	0x1F	A YAS_nonNeg value that specifies the minimum vertical distance between

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	Operand
(0x941F)		the bottom edge of the table and the top edge of text that wraps around the table. By default, the minimum vertical distance between a table and wrapping text is 0 twips.
sprmTSetBrc80 (0xD620)	0x20	A TableBrc80Operand value that specifies the borders of a set of cells in the table row. By default, cells have no borders.
sprmTInsert (0x7621)	0x21	A TInsertOperand value that specifies a range of new table cell definitions to insert into the table row. The new cells have properties that are defined by the table style of the row. Each table row MUST specify at least one cell using sprmTInsert or sprmTDefTable, or a combination thereof.
sprmTDelete (0x5622)	0x22	An ItcFirstLim value that specifies a range of table cell definitions to delete from the table row. These cell definitions MUST have been inserted by a previous application of sprmTInsert or sprmTDefTable. The table row MUST have at least one cell remaining after the deletion.
sprmTDxaCol (0x7623)	0x23	A TDxaColOperand value that specifies the width of a range of cells in this table. By default, the column width is specified when the column is created in either sprmTInsert or sprmTDefTable.
sprmTMerge (0x5624)	0x24	An ItcFirstLim structure that specifies a set of cells in the current table row that are to be merged. The first cell in the range is considered the primary cell, and its contents and formatting flow into the layout region of the other cells. The contents and formatting of the other cells are not applied. By default, cells are not merged.
sprmTSplit (0x5625)	0x25	An ItcFirstLim structure that specifies a set of cells in the current table row that are not to be merged. All cells in the specified range render their own contents and formatting. Neighboring cells that are set to merge do not flow into these cells. The function of this Sprm is to undo the effects of sprmTMerge. When applied to cells that are not merged, nothing is changed. By default, cells are not merged.
sprmTTextFlow (0x7629)	0x29	A CellRangeTextFlow value that specifies a set of cells in the current table row and the text flow model for each cell. By default, the text flow of each cell in the row is **grpfTFlrtb**.
sprmTVertMerge (0xD62B)	0x2B	A VertMergeOperand value that specifies a cell in the current row, and whether that cell is vertically merged with the cell above or below it. By default, cells are not merged with other cells.
sprmTVertAlign (0xD62C)	0x2C	A CellRangeVertAlign value that specifies a set of cells in the current table row and the vertical alignment of cell contents in each cell. By default, cell contents are vertically aligned to the top of the cell.
sprmTSetShd (0xD62D)	0x2D	A TableShadeOperand value that specifies a set of cells in a table row and the background shading for each cell. If the nFib value is greater than 0x00D9 and the application can interpret table styles, this Sprm MUST be ignored. By default, the background shading of table cells is **ShdAuto**.
sprmTSetShdOdd (0xD62E)	0x2E	A TableShadeOperand value that specifies a set of cells in a table row and the background shading for odd numbered cells in that set. That is, if the set of cells is 0 through 5, then this sets the background shading for cells 0, 2 and 4. To set background shading for even numbered cells, specify a set of cells starting on the even numbered cell. If nFib is greater than 0x00D9 and the application can interpret table styles, then this Sprm MUST be ignored. By default, the background shading of table cells is **ShdAuto**.
sprmTSetBrc (0xD62F)	0x2F	A TableBrcOperand value that specifies the border type of a set of cells in a table row. By default, the border type is inherited from the table border properties.
sprmTCellPadding (0xD632)	0x32	A CSSAOperand value that specifies the **cell margin** for one or more cell sides. **cssa.ftsWidth** MUST be ftsNil (0x00) or ftsDxa (0x03). If

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	Operand
		cssa.ftsWidth is ftsDxa (0x03), then **cssa.wWidth** MUST be nonnegative and MUST NOT exceed 31680. By default, cell margins are specified by sprmTCellPaddingDefault.
sprmTCellSpacingDefault (0xD633)	0x33	A CSSAOperand that specifies the **cell spacing** for each cell in the entire row. **cssa.itc.itcFirst** MUST be 0, **cssa.itc.itcLim** MUST be 1, **cssa.grfbrc** MUST be fbrcSidesOnly (0x0F), **cssa.ftsWidth** MUST be ftsNil (0x00) or ftsDxa (0x03) or ftsDxaSys (0x13), and **cssa.wWidth** MUST be nonnegative and MUST NOT exceed 15840 (11"). By default, cells do not have cell spacing.
sprmTCellPaddingDefault (0xD634)	0x34	A CSSAOperand that specifies the cell margin for one or more cell sides for each cell in the entire row. **cssa.itc.itcFirst** MUST be 0, **cssa.itc.itcLim** MUST be 1, **cssa.ftsWidth** MUST be ftsNil (0x00) or ftsDxa (0x03), and **cssa.wWidth** MUST be nonnegative and MUST NOT exceed 31680. By default, rows use two sprmTCellPaddingDefault properties: the first to specify left and right cell margins, and the second to specify top and bottom cell margins. By default, left and right cell margins use the following CSSA. **itcFirst**: 0 **itcLim**: 1 **grfbrc**: fbrcLeft | fbrcRight (0x0A) **ftsWidth**: ftsDxa (0x03) **wWidth**: 108 By default, top and bottom cell margins use the following CSSA. **itcFirst**: 0 **itcLim**: 1 **grfbrc**: fbrcTop | fbrcBottom (0x05) **ftsWidth**: ftsDxa (0x03) **wWidth**: 0
sprmTCellWidth (0xD635)	0x35	A TableCellWidthOperand value that specifies the preferred width of one or more table cells. By default, table cells do not have a preferred width.
sprmTFitText (0xF636)	0x36	A CellRangeFitText value that specifies a set of cells in a table row and whether their contents are to be stretched or compressed to exactly fill their widths. By default the contents of table cells are not stretched or compressed.
sprmTFCellNoWrap (0xD639)	0x39	A CellRangeNoWrap value that specifies a set of cells in a table row and whether their contents wrap over multiple lines. By default, the contents of table cells wrap over multiple lines.
sprmTIstd (0x563A)	0x3A	An unsigned integer value that specifies the **istd** value of a table style to apply. To apply the **istd** value, fetch the complete set of table properties from that style (see Applying Properties for instructions.) Apply those properties to the current table, while preserving the previous values of the following: ▪ Whether the values of table properties have been preserved for revision marking purposes (for example, by sprmTWall). ▪ Whether the table row has an associated **property revision mark**, as well as its author and the date and time (for example, by sprmTPropRMark). ▪ Whether this table is right-to-left (for example, by sprmTFBiDi). ▪ The revision save ID that is associated with table formatting (sprmTRsid). ▪ The PositionCodeOperand structure that specifies the origin used to

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	Operand
		calculate the table position when it is absolutely positioned (for example, by sprmTPc). ▪ The horizontal position of the table relative to the horizontal anchor of the table (for example, by sprmTDxaAbs). ▪ The downward vertical position relative to the vertical anchor of the tables (for example, by sprmTDyaAbs). ▪ The minimum horizontal distance between the physical left edge of the table and the physical right edge of text that wraps around the table (for example, by sprmTDxaFromText). ▪ The minimum vertical distance between the top edge of the table and the bottom edge of text that wraps around the table (for example, by sprmTDyaFromText). ▪ The minimum horizontal distance between the physical right edge of the table and the physical left edge of text that wraps around the table (for example, by sprmTDxaFromTextRight). ▪ The minimum vertical distance between the bottom edge of the table and the top edge of text that wraps around the table (for example, by sprmTDyaFromTextBottom). ▪ The average width between the left and right default cell margins for the first cell in the row (for example, by sprmTDxaGapHalf). ▪ The height of the row (for example, by sprmTDyaRowHeight). ▪ The preferred total width of the table (for example, by sprmTTableWidth). ▪ Whether the table column widths are to be automatically resized to best fit the contents of the whole table (for example, by sprmTFAutofit). ▪ The **grfatl** member of the TLP structure that specifies the settings that are used when the current table row was last auto-formatted (for example, by sprmTTlp). This sprm also specifies that the current table has the table style that is specified by this **istd**. When computing paragraph or character properties inside the table, the current table style needs to be taken into account (see Applying Properties). When sprmTIstd is applied, the paragraph and character properties of the text within the table need to be recomputed. If the **istd** refers to an empty or non-existent style, or a style of a different type, a later Prl such as sprmTIstd MUST change the **istd** to a valid value. Applying an **istd** value that refers to an empty or nonexistent style, or a style of a different type, is equivalent to applying a sprmTIstd with an **istd** value of 0x000B (the default).
sprmTCellPaddingStyle (0xD63E)	0x3E	A CSSAOperand value that specifies the cell margin that is applied to one or more cell sides for each cell in the entire row defined by a Table style. **cssa.itc.itcFirst** MUST be 0, **cssa.itc.itcLim** MUST be 1, **cssa.ftsWidth** MUST be ftsDxa (0x03) and **cssa.wWidth** MUST be nonnegative and MUST NOT exceed 31680. By default, cell margins are set as specified by sprmTCellPaddingDefault.
sprmTCellFHideMark (0xD642)	0x42	A CellHideMarkOperand that specifies that table cell content is rendered with no height if all cells in the row are empty; however, cells have a visible height if they have nonzero cell borders, cell margins, or cell spacing. By default, cell heights are rendered based on the paragraph and character properties of the cell, regardless of whether they contain content.
sprmTSetShdTable (0xD660)	0x60	A SHDOperand value that specifies the background shading for the entire table. By default, tables are not shaded.
sprmTWidthIndent (0xF661)	0x61	An FtsWWidth_Indent structure that specifies the preferred leading indent of the table where the row resides. By default, tables have no preferred indent.
sprmTCellBrcType (0xD662)	0x62	A TCellBrcTypeOperand value that specifies the border type for the first several consecutive cells in a table row. By default, the border type is inherited from the table style of the whole table.

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	Operand
sprmTFBiDi90 (0x5664)	0x64	A Bool16 value that specifies whether this table is right-to-left. A table is right-to-left if either this Sprm or sprmTFBiDi is set to true. By default, tables are left-to-right.
sprmTFNoAllowOverlap (0x3465)	0x65	A Bool8 value that specifies whether the table is allowed to overlap other tables. A value of 0x01 specifies that the table is not allowed to overlap. By default, tables are allowed to overlap with other tables.
sprmTFCantSplit (0x3466)	0x66	A Bool8 value. If this property is "true" (1), table rows MUST NOT be split across page breaks. By default, rows can be split across page breaks.
sprmTPropRMark (0xD667)	0x67	A PropRMarkOperand that specifies whether the table row has an associated property revision mark, as well as its author and date/time. By default, table rows have no property revision marks.
sprmTWall (0x3668)	0x68	A Bool8 value that specifies whether the values of table properties are preserved for revision marking purposes until the modifications are accepted or rejected by the user. A value of 1 specifies that the values of properties are preserved. All SPRMs encountered before the sprmTWall in the property evaluation of the table row specify the state of properties before revision marking was enabled, whereas all SPRMs following the sprmTWall specify the property modifications that occurred afterwards. A value of 0 specifies that no values are preserved (overriding any previously encountered sprmTWall SPRMs that specify the contrary). Neither SPRMs encountered before the sprmTWall, nor subsequent SPRMs (until another sprmTWall, if any), are treated in any special way with regard to revision marking. By default, property values are not preserved.
sprmTIpgp (0x7469)	0x69	An unsigned integer value that specifies the PGPInfo.**ipgpSelf** value of the PGPInfo data to be applied to this table row. The table depth of the table row (see Overview of Tables) MUST be 1 greater than PGPInfo.**itap**. **ipgpSelf** values MUST NOT be applied in such a way as to break the hierarchy that is implied by the PGPInfo structures themselves. Given that the application of a particular PGPInfo.**ipgpSelf** value implies the application of all of the PGPInfo.**ipgpParent** values encountered ascending the PGPInfo chain, then ensuring that all occurrences of any PGPInfo.**ipgpSelf** are on adjacent rows of the same table depth or paragraphs of one table depth less than an adjacent row ensures that the hierarchy is not broken. There MUST be a corresponding sprmPIpgp with the same PGPInfo.**ipgpSelf** value applied to the table terminating mark of this row (See Overview of Tables). By default, a table row has no associated PGPInfo.
sprmTCnf (0xD66A)	0x6A	A CNFOperand that specifies conditional table formatting for a specific condition of a table style. The **grpprl** member of CNFOperand specifies the table/cell/row formatting properties and MUST NOT contain any Sprms that are disallowed in the **grpprlTapx** member of UpxTapx, with the exception of the following Sprms that are allowed: ▪ sprmTCellBrcTopStyle ▪ sprmTCellBrcBottomStyle ▪ sprmTCellBrcLeftStyle ▪ sprmTCellBrcRightStyle ▪ sprmTCellBrcInsideHStyle ▪ sprmTCellBrcInsideVStyle This sprm MUST only be specified within the **grpprlTapx** member of a UpxTapx within a table style definition (LPStd). By default, a table style definition does not include conditional formatting.
sprmTDefTableShdRaw (0xD670)	0x70	A DefTableShdOperand value that specifies the default shading for cells 1 to 22 in the row, or all cells in the row if the row contains fewer than 22 cells. If a cell is set to ShdAuto in **rgShd**, the cell is not shaded. If a cell is

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	Operand
		set to ShdNil in **rgShd**, the cell is shaded according to the table style. By default, cells are shaded according to the table style. Cells 23 to 44 are shaded by sprmTDefTableShdRaw2nd, and cells 45 to 63 are shaded by sprmTDefTableShdRaw3rd.
sprmTDefTableShdRaw2nd (0xD671)	0x71	A DefTableShdOperand value that specifies the default shading for cells 23 to 44 in the row, or all remaining cells in the row beginning with cell 23 if the row contains fewer than 44 cells. If a cell is set to ShdAuto in **rgShd**, the cell is not shaded. If a cell is set to ShdNil in **rgShd**, the cell is shaded according to the table style. By default, cells are shaded according to the table style. Cells 1 to 22 are shaded by sprmTDefTableShdRaw, and cells 45 to 63 are shaded by sprmTDefTableShdRaw3rd.
sprmTDefTableShdRaw3rd (0xD672)	0x72	A DefTableShdOperand that specifies the default shading for cells 45 to 63 in the row, or all remaining cells in the row beginning with cell 45 if the row contains fewer than 63 cells. **cb** MUST NOT exceed 190 and **rgShd** MUST NOT exceed 19 elements. If a cell is set to ShdAuto in **rgShd**, the cell is not shaded. If a cell is set to ShdNil in **rgShd**, the cell is shaded according to the table style. By default, cells are shaded according to the table style. Cells 1 to 22 are shaded by sprmTDefTableShdRaw, and cells 23 to 44 are shaded by sprmTDefTableShdRaw2nd.
sprmTRsid (0x7479)	0x79	An integer value that specifies a revision save ID, as specified in [ECMA-376] Part 1, Section 17.15.1.70 rsid (Single Session Revision Save ID), associated with table formatting. If not present, then no revision save ID is specified for this formatting.
sprmTCellVertAlignStyle (0x347C)	0x7C	A VerticalAlign value that specifies the vertical alignment of content within cells as defined by a Table style. By default, the value is vaTop.
sprmTCellNoWrapStyle (0x347D)	0x7D	A Bool8 value that specifies whether content within cells MAY<154> **word wrap**. This Sprm is used by table styles and MUST NOT appear outside of the **grpprlTapx** array of UpxTapx. If this property is "true" (1), content SHOULD NOT<155> word wrap. By default, content MAY<156> word wrap. This property is ignored if the cell has an absolute width set by using sprmTCellWidth with **ftsWidth** equal to ftsDxa (0x03)—cell content wraps if it cannot fit on a single line.
sprmTCellBrcTopStyle (0xD47F)	0x7F	A BrcOperand value that specifies the top border for cells that are affected by a CNFOperand value. This Sprm MUST NOT appear outside of the **grpprl** array of a CNFOperand value. By default, cells have no top border.
sprmTCellBrcBottomStyle (0xD680)	0x80	A BrcOperand value that specifies the bottom border for cells that are affected by a CNFOperand value. This Sprm MUST NOT appear outside of the **grpprl** array of a CNFOperand. By default, cells have no bottom border.
sprmTCellBrcLeftStyle (0xD681)	0x81	A BrcOperand value that specifies the logical left border for cells that are affected by a CNFOperand value. This Sprm MUST NOT appear outside of the **grpprl** array of a CNFOperand. By default, cells have no logical left border.
sprmTCellBrcRightStyle (0xD682)	0x82	A BrcOperand value that specifies the **logical right** border for cells that are affected by a CNFOperand value. This Sprm MUST NOT appear outside of the **grpprl** array of a CNFOperand. By default, cells have no logical right border.
sprmTCellBrcInsideHStyle (0xD683)	0x83	A BrcOperand value that specifies the border between a table row that is affected by a CNFOperand value and the following table row. This Sprm MUST NOT appear outside of the **grpprl** array of a CNFOperand. By default, table rows have no borders between them.
sprmTCellBrcInsideVStyle (0xD684)	0x84	A BrcOperand value that specifies the border between cells of a table row that are affected by a CNFOperand. This Sprm MUST NOT appear outside of the **grpprl** array of a CNFOperand. By default, cells have no border between them.
sprmTCellBrcTL2BRStyle (0xD685)	0x85	A BrcOperand value that specifies a diagonal border from the top, logical left corner to the bottom, logical right corner of each cell that is affected by a CNFOperand. This Sprm MUST NOT appear outside of the **grpprl** array of a CNFOperand. By default, cells have no diagonal border.

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	Operand
sprmTCellBrcTR2BLStyle (0xD686)	0x86	A BrcOperand value that specifies a diagonal border from the top, logical right corner to the bottom, logical left corner of each cell that is affected by a CNFOperand. This Sprm MUST NOT appear outside of the **grpprl** array of a CNFOperand. By default, cells have no diagonal border.
sprmTCellShdStyle (0xD687)	0x87	A SHDOperand value that specifies the background shading to be applied to an entire table defined by a Table style. By default, tables are not shaded.
sprmTCHorzBands (0x3488)	0x88	An unsigned 8-bit integer value that specifies the number of rows in a **horizontal band** that is used for conditional formatting as defined by a Table style. This value MUST be at least 1 and MUST NOT exceed 3. By default, tables are not shaded with horizontal bands.
sprmTCVertBands (0x3489)	0x89	An unsigned 8-bit integer value that specifies the number of columns in a **vertical band** that is used for conditional formatting as defined by a Table style. This value MUST be at least 1 and MUST NOT exceed 3. By default, tables are not shaded with vertical bands.
sprmTJc (0x548A)	0x8A	An unsigned 16-bit integer value that specifies the logical justification of the table. The following shows the valid values and their meanings. 0 - The table is logical left-justified 1 - The table is centered 2 - The table is logical right-justified By default, tables are logical left justified.

2.6.4 Section Properties

A Prl structure with a **sprm.sgc** of 4 modifies a **section** property.

The following table specifies the section property modifiers, including the valid **sprm** values, their function, and the corresponding **operand** type and meaning.

sprm	ispmd	Operand
sprmScnsPgn (0x3000)	0x00	A CNS indicating the number separator used between the chapter number and the page number for purpose of **chapter numbering** in page number fields (that is, when sprmSiHeadingPgn specifies a value other than 0). By default, the chapter number separator is a hyphen (see cnsHyphen).
sprmSiHeadingPgn (0x3001)	0x01	An unsigned 8-bit integer value that specifies which heading level starts new chapters for the purposes of chapter numbering in page number fields. The value MUST be in the interval [0, 9]. A value of 0 specifies that chapter numbers are not shown in page number fields, whereas values from 1 to 9 specify corresponding heading levels (1 specifies Heading 1, 2 specifies Heading 2, and so forth). By default, chapter numbers are not shown in page number fields. In the event that the style corresponding to the indicated heading level does not have associated numbering, chapter numbers are not shown in page number fields.
sprmSDxaColWidth (0xF203)	0x03	An SDxaColWidthOperand that specifies the width of a particular column, in case columns are not evenly spaced as specified by sprmSFEvenlySpaced.
sprmSDxaColSpacing (0xF204)	0x04	An SDxaColSpacingOperand that specifies the spacing between two columns in case columns are not evenly spaced (as instructed by sprmSFEvenlySpaced). The **iCol** field of the SDxaColSpacingOperand structure specifies the index of the first of the two columns. By default there is no spacing between columns.
sprmSFEvenlySpaced	0x05	A Bool8 value that specifies whether the space between page margins is

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
(0x3005)		distributed evenly between all columns (after subtracting the space between columns, as instructed by sprmSDxaColumns). A value of 1 specifies that space is distributed evenly; a value of 0 specifies that column widths and inter-column spacing MUST be specified by sprmSDxaColWidth and sprmSDxaColSpacing. By default, columns are evenly spaced.
sprmSFProtected (0x3006)	0x06	A Bool8 value that specifies whether the section is unprotected in case document editing is restricted to **form fields** only (see DopBase.**fProtEnabled**). A value of 1 indicates that the section is unprotected, whereas a value of 0 indicates that the section is protected. By default, the protection status of a section is specified by DopBase.**fProtEnabled**.
sprmSDmBinFirst (0x5007)	0x07	A SDmBinOperand that specifies the paper source used by the printer for the first page of the section. By default, no paper source is specified.
sprmSDmBinOther (0x5008)	0x08	An SDmBinOperand that specifies the paper source used by the printer for all pages in the section except the first. By default, no paper source is specified.
sprmSBkc (0x3009)	0x09	An SBkcOperand that specifies what kind of **section break** terminates the section. By default, section breaks are of type "Next Page" (see bkcNewPage).
sprmSFTitlePage (0x300A)	0x0A	A Bool8 value that specifies whether the section has a different first page (a "title page"). A value of 1 indicates that the first page is separate, having its own header and footer. A value of 0 indicates that there is no title page. By default, a section does not have a separate first page.
sprmSCcolumns (0x500B)	0x0B	An unsigned 16-bit integer whose value is one less than the number of columns in this section. MUST be less than or equal to 43. A value of zero specifies a section with a single column. By default, a section has a single column. If the value is larger than zero, and the columns are not evenly spaced (as instructed by sprmSFEvenlySpaced), then there MUST be the same number of sprmSDxaColWidth as the columns, each specifying the width of a different column. An end-of-column character (0xE) at a particular CP specifies a manual column break at that CP.
sprmSDxaColumns (0x900C)	0x0C	An XAS_nonNeg that specifies the space between columns, in case columns are evenly spaced (as instructed by sprmSFEvenlySpaced). By default, spacing between columns varies depending on implementation and system settings, so implementations SHOULD<157> write this **Sprm** out to ensure interoperability even if the value does not differ from the default. The default values are dependent on the installation language of the application. The installation LCID values and their corresponding defaults are shown following. LCID 1025: 720 twips LCID 1026: 708 twips LCID 1027: 708 twips LCID 1028: 720 twips LCID 1029: 708 twips LCID 1030: 708 twips LCID 1031: 720 twips LCID 1032: 720 twips LCID 1033: 720 twips LCID 1034: 720 twips

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
		LCID 1035: 708 twips LCID 1036: 720 twips LCID 1037: 720 twips LCID 1038: 708 twips LCID 1039: 708 twips LCID 1040: 720 twips LCID 1041: 720 twips LCID 1042: 720 twips LCID 1043: 708 twips LCID 1044: 708 twips LCID 1045: 708 twips LCID 1046: 720 twips LCID 1048: 708 twips LCID 1049: 720 twips LCID 1050: 720 twips LCID 1051: 708 twips LCID 1053: 720 twips LCID 1055: 708 twips LCID 1058: 720 twips LCID 1059: 720 twips LCID 1060: 708 twips LCID 1061: 708 twips LCID 1062: 720 twips LCID 1063: 1296 twips LCID 1067: 720 twips LCID 1068: 720 twips LCID 1069: 708 twips LCID 1078: 708 twips LCID 1079: 720 twips LCID 1086: 720 twips LCID 1087: 720 twips LCID 1088: 708 twips LCID 1089: 708 twips LCID 1092: 720 twips LCID 1104: 720 twips LCID 2052: 720 twips LCID 2070: 720 twips LCID 2074: 708 twips
sprmSNfcPgn (0x300E)	0x0E	An 8-bit MSONFC (as specified in [MS-OSHARED] section 2.2.1.3) that specifies the numbering format used for page numbers. An application MAY<158> fall back to a different MSONFC if the format specified by the value is not a counting number format—for example, if it is **msonfcBullet**. By default, page numbers use the **msonfcArabic** numbering format.
sprmSFPgnRestart (0x3011)	0x11	A Bool8 value that specifies whether the section starts with a new page number. A value of 1 indicates that the section starts with a new page number as specified by sprmSPgnStart97 or sprmSPgnStart. A value of 0 indicates that page numbers continue from the previous section (or begin at 1, if this is the first section).

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
		By default, page numbers continue from the previous section (or begin at 1, if this is the first section).
sprmSFEndnote (0x3012)	0x12	A Bool8 value that specifies whether **endnotes** are shown at the end of the section. This SPRM is only considered when endnotes are set to show at the ends of sections (see DOPBASE.**epc**). A value of 1 specifies that endnotes are shown at the end of the section. A value of 0 specifies that endnotes are suppressed for the current section, and they are shown at the end of the next section for which endnotes are not suppressed. If such a section does not exist, the endnotes are shown at the end of the last section of the document. By default, endnotes are not suppressed, and they show at the end of a section.
sprmSLnc (0x3013)	0x13	An SLncOperand that specifies the line numbering mode to use in case line numbers are enabled (see sprmSNLnnMod). By default, line numbers restart every page.
sprmSNLnnMod (0x5015)	0x15	An unsigned 16-bit integer that specifies the distance in the number of lines between line number labels. For example, a value of 1 indicates that every line displays a line number, whereas a value of 3 indicates that only every third line shows a line number. The value MUST be in the interval [0, 100]. A value of 0 specifies that line numbers are disabled. By default, line numbers are disabled.
sprmSDxaLnn (0x9016)	0x16	An XAS_nonNeg that specifies the distance between line numbers and the lines of text to which they apply. A value of 0 indicates that the application MUST automatically determine positioning. By default, the positioning of line numbers is automatically determined.
sprmSDyaHdrTop (0xB017)	0x17	A YAS_nonNeg that specifies the header distance, in **twips**, from the top edge of the page. Because the default distance is dependent on the implementation and system settings, implementations SHOULD<159> write this Sprm out even if the value does not differ from the default. The default values are dependent on the install language of the application. The installation LCID values and their corresponding defaults are shown following. LCID 1025: 720 twips LCID 1026: 708 twips LCID 1027: 708 twips LCID 1028: 720 twips LCID 1029: 708 twips LCID 1030: 708 twips LCID 1031: 720 twips LCID 1032: 720 twips LCID 1033: 720 twips LCID 1034: 720 twips LCID 1035: 708 twips LCID 1036: 720 twips LCID 1037: 720 twips LCID 1038: 708 twips LCID 1039: 708 twips LCID 1040: 720 twips LCID 1041: 720 twips LCID 1042: 720 twips LCID 1043: 708 twips

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
		LCID 1044: 708 twips LCID 1045: 708 twips LCID 1046: 720 twips LCID 1048: 708 twips LCID 1049: 720 twips LCID 1050: 720 twips LCID 1051: 708 twips LCID 1053: 720 twips LCID 1055: 708 twips LCID 1058: 708 twips LCID 1059: 708 twips LCID 1060: 708 twips LCID 1061: 708 twips LCID 1062: 720 twips LCID 1063: 567 twips LCID 1067: 708 twips LCID 1068: 708 twips LCID 1069: 708 twips LCID 1078: 708 twips LCID 1079: 708 twips LCID 1086: 720 twips LCID 1087: 708 twips LCID 1088: 708 twips LCID 1089: 708 twips LCID 1092: 708 twips LCID 1104: 720 twips LCID 2052: 720 twips LCID 2070: 720 twips LCID 2074: 708 twips
sprmSDyaHdrBottom (0xB018)	0x18	An YAS_nonNeg that specifies the footer distance, in twips, from the bottom edge of the page. Implementations SHOULD<160> write this Sprm out to ensure interoperability because the footer distance from the bottom is dependent on the implementation and system settings. The default values are the same as listed for sprmSDyaHdrTop.
sprmSLBetween (0x3019)	0x19	A Bool8 value that specifies whether lines are drawn between columns of text. By default, lines are not drawn between columns of text.
sprmSVjc (0x301A)	0x1A	A Vjc value that specifies the vertical justification of the section. By default, sections are top-aligned (vjcTop).
sprmSLnnMin (0x501B)	0x1B	An unsigned 16-bit integer whose value is one less than the starting value for line numbers. The value SHOULD<161> be less than or equal to 32766. By default, line numbers begin at 1.
sprmSPgnStart97 (0x501C)	0x1C	An unsigned 16-bit integer that specifies the starting value for page numbers when the section has page number restart enabled (as specified by sprmSFPgnRestart). This value MUST be ignored if the section does not have page number restart enabled. The value of the operand SHOULD<162> be less than or equal to 32766. By default, page numbers restart at 0.
sprmSBOrientation	0x1D	An SBOrientationOperand that specifies the page orientation of the section.

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
(0x301D)		By default, the page orientation is portrait.
sprmSXaPage (0xB01F)	0x1F	An unsigned 16-bit integer that specifies the page width of the section in twips. The value of the operand MUST be in the interval [144, 31680]. By default, the page width is 215.9 mm (8.5 inches, or 12240 twips).
sprmSYaPage (0xB020)	0x20	An unsigned 16-bit integer that specifies the page height of the section, in twips. The value of the operand MUST be in the interval [144, 31680]. By default, the page height is 279.4 mm (11 inches, or 15840 twips).
sprmSDxaLeft (0xB021)	0x21	An XAS_nonNeg that specifies the width, in twips, of the left margin. By default, the width of the left margin varies depending on the implementation and the system settings, so implementations MUST use this SPRM to specify the left margin of each section.
sprmSDxaRight (0xB022)	0x22	An XAS_nonNeg that specifies the width, in twips, of the right margin. By default, the width of the right margin varies depending on the implementation and the system settings, so implementations MUST use this SPRM to specify the right margin of each section.
sprmSDyaTop (0x9023)	0x23	A YAS that specifies the height of the top margin, in twips. A positive value indicates a minimum top margin; this margin MUST be grown to avoid overlapping the space that is occupied by **headers**. A negative value indicates a fixed margin; the top margin MUST be the absolute value of the value that is specified by this SPRM regardless of the space that is occupied by headers. Each section MUST specify a top margin. The top margin MUST be less than or equal to 31665 and greater than or equal to -31665.
sprmSDyaBottom (0x9024)	0x24	A YAS that specifies the height of the bottom margin, in twips. A positive value specifies a minimum bottom margin; this margin MUST be grown to avoid overlapping the space that is occupied by **footers** or **footnotes**. A negative value specifies a fixed margin; the bottom margin MUST be the absolute value of the value that is specified by this SPRM regardless of the space that is occupied by footers or footnotes. Each section MUST specify a bottom margin. The bottom margin MUST be less than or equal to 31665 and greater than or equal to -31665.
sprmSDzaGutter (0xB025)	0x25	An unsigned 16-bit integer that specifies the size of the **gutter** margin, in twips. By default, there is no **gutter margin**.
sprmSDmPaperReq (0x5026)	0x26	A 16-bit unsigned integer that specifies a tie-breaker value to be used when more than one available paper format ("Letter Matte", "Letter Gloss", "Letter w/ Letterhead", "Letter Pink", and so on) matches the page dimensions as specified by sprmSXaPage and sprmSYaPage. This tie-breaker value MAY<163> be ignored. The determination and interpretation of this value is implementation-specific. The determination of the paper sizes for an application is implementation-specific
sprmSFBiDi (0x3228)	0x28	A Bool8 value that specifies whether the section uses **right-to-left** layout; that is, line numbers are displayed on the right side of text and columns are populated from right to left. By default, sections do not use right-to-left layout.
sprmSFRTLGutter (0x322A)	0x2A	A Bool8 value that specifies whether the gutter margin requires right-to-left layout. A value of 1 indicates a right-to-left gutter margin. By default, gutter margins are not right-to-left.
sprmSBrcTop80 (0x702B)	0x2B	A Brc80 that specifies the top page border. By default, pages have no top border.
sprmSBrcLeft80 (0x702C)	0x2C	A Brc80 that specifies the left page border. By default, pages have no left border.
sprmSBrcBottom80 (0x702D)	0x2D	A Brc80 that specifies the bottom page border. By default, pages have no bottom border.
sprmSBrcRight80	0x2E	A Brc80 that specifies the right page border.

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
(0x702E)		By default, pages have no right border.
sprmSPgbProp (0x522F)	0x2F	An SPgbPropOperand that specifies page border properties. By default, page borders apply to all pages of the section (pgbAllPages), they are displayed in front of text and other content (pgbAtFront), and their distance is measured from text (pgbFromText).
sprmSDxtCharSpace (0x7030)	0x30	A signed 32-bit integer that specifies the difference between the desired **character pitch** for the **document grid**, if enabled (see sprmSClm), and the pitch of the font that is specified by the Normal style. The resolution of the operand is 4096/pt. That is, a 1-pt difference between the desired character pitch and the font size as specified by the Normal style would affect the operand by 4096. For example, if the Normal style specified a font size of 11 pt, an operand value of 6144 would specify a desired character pitch for document grid of 12.5 pt (because 6144 / 4096 = 1.5 pt, so 11 pt + 1.5 pt = 12.5 pt). By default, there is no difference between the desired character pitch for the document grid and the pitch of the font that is specified by the Normal style. This value MUST be greater than or equal to -670925 and MUST be less than or equal to 6488064.
sprmSDyaLinePitch (0x9031)	0x31	A YAS that specifies, in twips, the line height that is used for document grid, if enabled (see sprmSClm). This line height does not apply to lines within table cells in case the fDontAdjustLineHeightInTable flag is set in the document Dop2000. If the document grid is enabled (see sprmSClm), a section MUST specify the line height that is used for the document grid. This value MUST be greater than or equal to 1, and MUST be less than or equal to 31680.
sprmSClm (0x5032)	0x32	An SClmOperand that specifies the document grid mode that is in use for the section. By default, document grid is disabled (clmUseDefault).
sprmSTextFlow (0x5033)	0x33	A MSOTXFL that specifies the text flow of the section, as specified in [MS-ODRAW] section 2.4.5.
sprmSBrcTop (0xD234)	0x34	A BrcOperand that specifies the top page border. By default, pages have no top border.
sprmSBrcLeft (0xD235)	0x35	A BrcOperand that specifies the left page border. By default, pages have no left border.
sprmSBrcBottom (0xD236)	0x36	A BrcOperand that specifies the bottom page border. By default, pages have no bottom border.
sprmSBrcRight (0xD237)	0x37	A BrcOperand that specifies the right page border. By default, pages have no right border.
sprmSWall (0x3239)	0x39	A Bool8 value that specifies whether the values of section properties are preserved for revision marking purposes until the modifications are accepted or rejected by the user. A value of 1 specifies that the values of properties are preserved. All SPRMs that are encountered before the sprmSWall in the property evaluation of the section specify the state of properties before revision marking was enabled, whereas all SPRMs following the sprmSWall specify the property modifications that occurred afterwards. A value of 0 specifies that no values are preserved (overriding any previously encountered sprmSWall SPRMs that specify the contrary). Neither SPRMs encountered before the sprmSWall, nor subsequent SPRMs (until another sprmSWall, if any), are treated in any special way with regard to revision marking. By default, the values of properties are not preserved.
sprmSRsid (0x703A)	0x3A	An integer that specifies a revision save ID, as specified in [ECMA-376] Part 1, Section 17.15.1.70 rsid (Single Session Revision Save ID), associated with section formatting. If this value is not present, no revision save ID is specified

Copyright © 2025 Microsoft Corporation

sprm	ispmd	Operand
		for this formatting.
sprmSFpc (0x303B)	0x3B	An SFpcOperand that specifies the footnote positioning for the section. By default, footnotes are positioned at the bottom of the page (see fpcBottomPage).
sprmSRncFtn (0x303C)	0x3C	An Rnc that specifies whether and when footnote numbering is restarted. All possible values of the Rnc enumeration are allowed. By default, footnotes are numbered continuously (see rncCont).
sprmSRncEdn (0x303E)	0x3E	An Rnc value that specifies whether and when endnote numbering is restarted. The value MUST be either rncCont or rncRstSect, as rncRstPage does not apply to endnotes. By default, endnotes are numbered continuously (see rncCont).
sprmSNFtn (0x503F)	0x3F	An unsigned 16-bit integer that specifies an offset to add to footnote numbers in this section. If this section has continuous footnote numbering (as specified by sprmSRncFtn), then the value of the sprm minus one MUST be added to every footnote number. (For example, with an offset of 6, a footnote that would have been numbered 2 is now numbered 2+5=7.) The sprm value MUST be less than or equal to 16383. If this section does not have continuous footnote numbering, the value of this sprm MUST be ignored. By default, no offset is added to footnote numbers.
sprmSNfcFtnRef (0x5040)	0x40	A 16-bit MSONFC (as specified in [MS-OSHARED] section 2.2.1.3) that specifies the numbering format used for footnotes. By default, footnotes use the **msonfcArabic** numbering format.
sprmSNEdn (0x5041)	0x41	An unsigned 16-bit integer that specifies an offset to add to endnote numbers in this section. If this section has continuous endnote numbering (as specified by sprmSRncEdn), then every endnote number in this section is offset by the value of this operand minus one. (For example, with an offset of 6, a endnote that would have been numbered 2 is now numbered 2+5=7.) The operand value MUST be less than or equal to 16383. If this section does not have continuous endnote numbering, this operand MUST be ignored. By default, no offset is added to endnote numbers.
sprmSNfcEdnRef (0x5042)	0x42	A 16-bit MSONFC (as specified in [MS-OSHARED] section 2.2.1.3) that specifies the numbering format used for endnotes. By default, endnotes use the **msonfcLCRoman** numbering format.
sprmSPropRMark (0xD243)	0x43	A PropRMarkOperand that specifies whether the section has an associated **property revision mark**, as well as its author and date/time. By default, sections have no property revision marks.
sprmSPgnStart (0x7044)	0x44	An unsigned 32-bit integer that specifies the starting value for page numbers when the section has page number restart enabled (as specified by sprmSFPgnRestart). MUST be ignored if the section does not have page number restart enabled. The value of the operand MUST be less than or equal to 2147483646. By default, page numbers restart at 0.

2.6.5 Picture Properties

A Prl with a **sprm.sgc** of 3 modifies a picture property.

The following table specifies the picture property modifiers, including the valid **sprm** values, their function, and the corresponding **operand** type and meaning.

Copyright © 2025 Microsoft Corporation

Sprm	ispmd	Operand
sprmPicBrcTop80 (0x6C02)	0x02	A Brc80 that specifies the top border of the inline picture. The Brc80.**brcType** field MUST be less than or equal to 0x19. By default, inline pictures do not have borders.
sprmPicBrcLeft80 (0x6C03)	0x03	A Brc80 that specifies the left border of the inline picture. The Brc80.**brcType** field MUST be less than or equal to 0x19. By default, inline pictures do not have borders.
sprmPicBrcBottom80 (0x6C04)	0x04	A Brc80 that specifies the bottom border of the inline picture. The Brc80.**brcType** field MUST be less than or equal to 0x19. By default, inline pictures do not have borders.
sprmPicBrcRight80 (0x6C05)	0x05	A Brc80 that specifies the right border of the inline picture. The Brc80.**brcType** field MUST be less than or equal to 0x19. By default, inline pictures do not have borders.
sprmPicBrcTop (0xCE08)	0x08	A BrcOperand that specifies the top border of the inline picture. The BrcOperand.Brc.**brcType** field MUST be less than or equal to 0x1B. By default, inline pictures do not have borders.
sprmPicBrcLeft (0xCE09)	0x09	A BrcOperand that specifies the left border of the inline picture. The BrcOperand.Brc.**brcType** field MUST be less than or equal to 0x1B. By default, inline pictures do not have borders.
sprmPicBrcBottom (0xCE0A)	0x0A	A BrcOperand that specifies the bottom border of the inline picture. The BrcOperand.Brc.**brcType** field MUST be less than or equal to 0x1B. By default, inline pictures do not have borders.
sprmPicBrcRight (0xCE0B)	0x0B	A BrcOperand that specifies the right border of the inline picture. The BrcOperand.Brc.**brcType** field MUST be less than or equal to 0x1B. By default, inline pictures do not have borders.

2.7 Document Properties

2.7.1 Dop

The Dop structure contains the document and compatibility settings for the document.

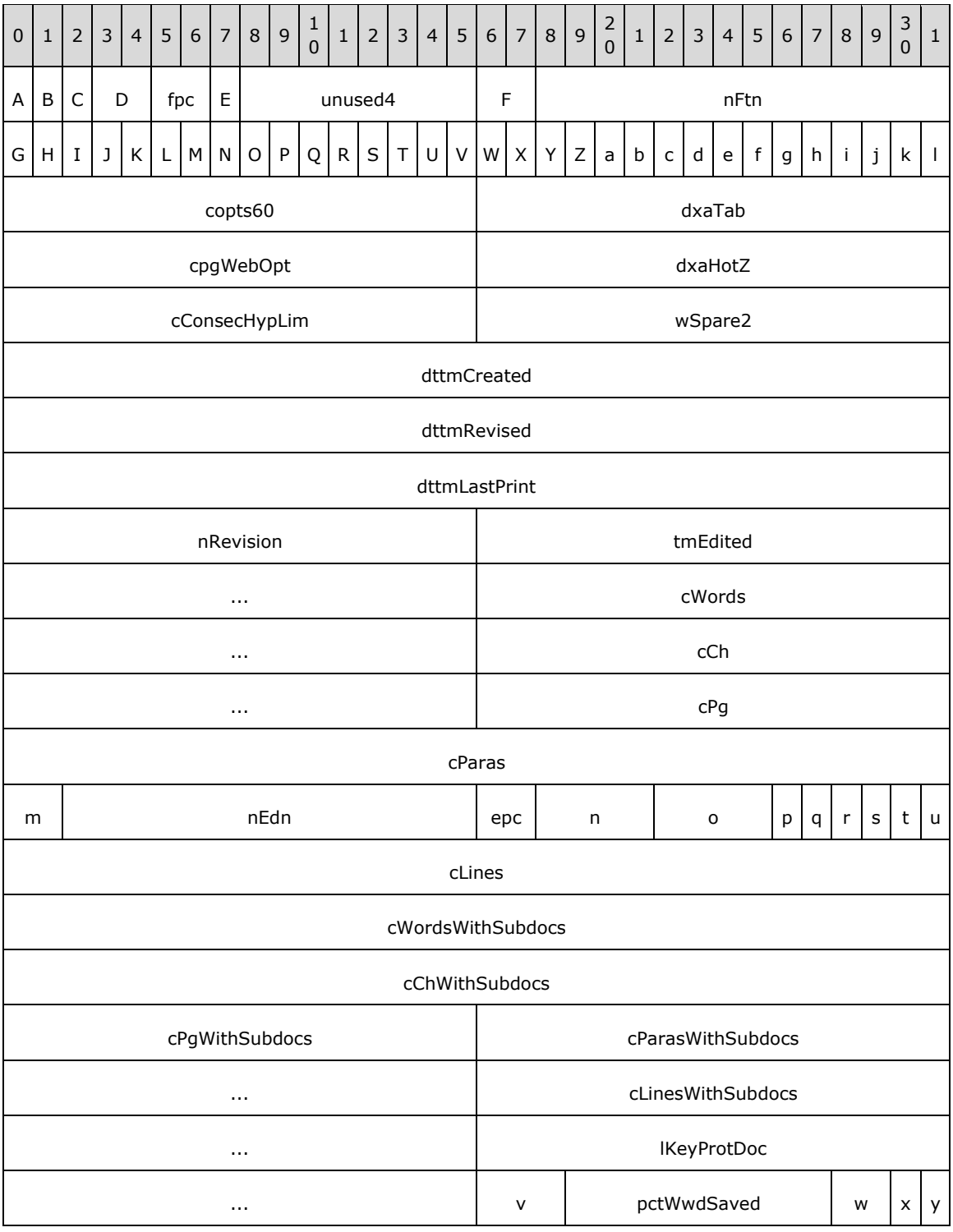
Based on the value of Fib.**cswNew**, the Dop is a structure from the following table.

Value	Meaning
0	Dop97
otherwise	Based on the value of FibRgCswNew.**nFibNew** the Dop is a structure from the following: ▪ 0x00D9 Dop2000 ▪ 0x0101 Dop2002 ▪ 0x010C Dop2003 ▪ 0x0112 if FibRgFcLcb97.**lcbDop** is 674, then the Dop is a Dop2007. If FibRgFcLcb97.**lcbDop** is 690, then the Dop is a Dop2010. If FibRgFcLcb97.**lcbDop** is 694, then the Dop is a section Dop2013. FibRgFcLcb97.**lcbDop** MUST be one of these three values.

2.7.2 DopBase

The **DopBase** structure contains document and compatibility settings that are common to all versions of the binary document. These settings influence the appearance and behavior of the current document and store document-level state.

Copyright © 2025 Microsoft Corporation



A - fFacingPages (1 bit): A bit that specifies whether even and odd pages have different headers and footers as specified in [ECMA-376] Part 1, Section 17.10.1 evenAndOddHeaders, where **titlePg** corresponds to the section property sprmSFTitlePage.

B - unused1 (1 bit): This value is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

C - fPMHMainDoc (1 bit): A bit that specifies whether this document is a **mail merge main document**.

D - unused2 (2 bits): This value is undefined and MUST be ignored.

fpc (2 bits): Specifies where footnotes are placed on the page when they are referenced by text in the current document for documents that have an **nFib** value that is less than or equal to 0x00D9. This MUST be one of the following values.

Value	Meaning
0	Specifies that all footnotes are placed at the end of the section in which they are referenced.
1	Specifies that footnotes are displayed at the bottom margin of the page on which the note reference mark appears.
2	Specifies that footnotes are displayed immediately following the last line of text on the page on which the note reference mark appears.

E - unused3 (1 bit): This value is undefined and MUST be ignored.

unused4 (8 bits): This value is undefined and MUST be ignored.

F - rncFtn (2 bits): Specifies when all automatic numbering for the footnote reference marks is restarted for documents that have an **nFib** value that is less than or equal to 0x00D9. For those documents that rely on **rncFtn**, when restarted, the next automatically numbered footnote in the document restarts to the specified **nFtn** value. This MUST be one of the following values.

Value	Meaning
0	Specifies that the numbering of footnotes continues from the previous section in the document.
1	Specifies that the numbering of footnotes is reset to the starting value for each unique section in the document.
2	Specifies that the numbering of footnotes is reset to the starting value for each unique page in the document.

nFtn (14 bits): For those documents that have an **nFib** value that is less than or equal to 0x00D9, this element specifies the starting number for the first automatically numbered footnotes in the document, and the first automatically numbered footnotes after each restart point that is specified by the **rncFtn** element.

G - unused5 (1 bit): This value is undefined and MUST be ignored.

H - unused6 (1 bit): This value is undefined and MUST be ignored.

I - unused7 (1 bit): This value is undefined and MUST be ignored.

J - unused8 (1 bit): This value is undefined and MUST be ignored.

K - unused9 (1 bit): This value is undefined and MUST be ignored.

L - unused10 (1 bit): This value is undefined and MUST be ignored.

M - fSplAllDone (1 bit): Specifies whether all content in this document was already checked by the spelling checker.

N - fSplAllClean (1 bit): Specifies whether all content in this document can be considered to be spelled correctly.

Copyright © 2025 Microsoft Corporation

O - fSplHideErrors (1 bit): Specifies whether visual cues are not displayed around content contained in a document which is flagged as a possible spelling error.

P - fGramHideErrors (1 bit): Specifies whether visual cues are not displayed around content that is contained in a document and flagged as a possible grammar error.

Q - fLabelDoc (1 bit): Specifies whether the document is a mail merge **labels document**.

When the value is 1, the document was created as a labels document.

R - fHyphCapitals (1 bit): Specifies whether words that are composed of all capital letters are hyphenated in a given document when **fAutoHyphen** is set to 1.

S - fAutoHyphen (1 bit): Specifies whether text is hyphenated automatically, as needed, when displayed as specified in [ECMA-376] Part 1, section 17.15.1.10 autoHyphenation.

T - fFormNoFields (1 bit): Specifies that there are no editable regions in a document that is currently protected for form field fill-in (**fProtEnabled** is 1). This value MUST be 0 if **fProtEnabled** is 0.

U - fLinkStyles (1 bit): Specifies whether the styles of the document are updated to match those of the attached template as specified in [ECMA-376] Part 1, Section 17.15.1.55 linkStyles, where the attachedTemplate value refers to entry 0x01 in SttbfAssoc.

V - fRevMarking (1 bit): Specifies whether edits are tracked as revisions. If the value of **fLockRev** is set to 1, the value of **fRevMarking** MUST also be set to 1, as specified in [ECMA-376] Part 1, Section 17.15.1.89 trackRevisions.

W - unused11 (1 bit): This value is undefined and MUST be ignored.

X - fExactCWords (1 bit): In conjunction with **fIncludeSubdocsInStats**, this bit specifies whether the values stored in **cCh**, **cChWS**, **cWords**, **cParas**, **cLines**, **cDBC**, **cChWithSubdocs**, **cChWSWithSubdocs**, **cWordsWithSubdocs**, **cParasWithSubdocs**, **cLinesWithSubdocs**, or **cDBCWithSubdocs** accurately reflect the current state of the document. When the value of **fExactCWords** is 0, none of the mentioned fields contain accurate values. When the value of **fExactCWords** is 1, the value of **fIncludeSubdocsInStats** determines which set of fields contains accurate values.

Y - fPagHidden (1 bit): This value is undefined and MUST be ignored.

Z - fPagResults (1 bit): This value is undefined and MUST be ignored.

a - fLockAtn (1 bit): Specifies whether protection for comments was applied to the document or, if Dop2003.**fTreatLockAtnAsReadOnly** has a value of 1, whether read-only protection was applied to the document. These restrictions are used to prevent unintentional changes to all or part of a document. Because this protection does not encrypt the document, malicious applications can circumvent its use. This protection is not intended as a security feature and can be ignored. When **fLockAtn** is 1, **fLockRev** MUST be 0 and **fProtEnabled** SHOULD<164> be 0. **fLockAtn** can be one of the following.

Value	Meaning
0	Specifies that the edits made to this document are restricted to the following: ▪ The insertion and deletion of comments within the document. ▪ The editing of the regions that are delimited by range permissions matching the editing rights of the user account that is being used to perform the editing.
1	Specifies that the edits made to this document are restricted to the following: ▪ The editing of the regions that are delimited by range permissions matching the editing rights of the user account that is being used to perform the editing.

Copyright © 2025 Microsoft Corporation

b - fMirrorMargins (1 bit): Specifies that the left and right margins that are defined in the section properties are swapped on facing pages.

c - fWord97Compat (1 bit): Specifies that this document was in **Word97 compatibility mode** when last saved.

d - unused12 (1 bit): This value is undefined and MUST be ignored.

e - unused13 (1 bit): This value is undefined and MUST be ignored.

f - fProtEnabled (1 bit): Specifies that the edits that are made to this document are restricted to the editing of form fields in sections that are protected (see sprmSFProtected). All other sections have no editing restrictions resulting from this setting. When **fProtEnabled** is 1, both **fLockAtn** and **fLockRev** SHOULD<165> be 0.

g - fDispFormFldSel (1 bit): If the document is currently protected for form field fill-in (**fProtEnabled** is 1), this bit specifies that the selection was within a display form field (check box or list box) the last time that the document was saved.

h - fRMView (1 bit): Specifies whether to show any revision markup that is present in this document.

i - fRMPrint (1 bit): Specifies whether to print any revision markup that is present in the document. SHOULD<166> be the same value as **fRMView**.

j - fLockVbaProj (1 bit): Specifies whether the Microsoft Visual Basic project is locked from editing and viewing.

k - fLockRev (1 bit): Specifies whether to track all edits made to this document as revisions. Additionally specifies that **fRevMarking** MUST be 1 for the duration that **fLockRev** is 1. When **fLockRev** is 1, **fLockAtn** MUST be 0 and **fProtEnabled** SHOULD<167> be 0.

l - fEmbedFonts (1 bit): Specifies that TrueType fonts are embedded in the document when the document is saved as specified in [ECMA-376] Part 1, Section 17.8.3.8 embedTrueTypeFonts.

copts60 (2 bytes): A copts60 that specifies compatibility options.

dxaTab (2 bytes): Specifies the default tab stop interval, in **twips**, to use when generating automatic tab stops as specified in [ECMA-376] Part 1, Section 17.15.1.25 defaultTabStop.

cpgWebOpt (2 bytes): Specifies the **code page** to use when saving to **HTML**.

dxaHotZ (2 bytes): Specifies the maximum amount of white space, in twips, allowed at the end of the line before attempting to hyphenate the next word as specified in [ECMA-376] Part 17, Section 17.15.1.53 hyphenationZone.

cConsecHypLim (2 bytes): Specifies the maximum number of consecutive lines that can end in a hyphenated word before ignoring automatic hyphenation rules for one line as specified in [ECMA-376] Part 1, Section 17.15.1.22 consecutiveHyphenLimit.

wSpare2 (2 bytes): This value MUST be 0, and MUST be ignored.

dttmCreated (4 bytes): A **DTTM** that MAY<168> specify the date and time at which the document was created.

dttmRevised (4 bytes): A **DTTM** that specifies the date and time at which the document was last saved.

Copyright © 2025 Microsoft Corporation

dttmLastPrint (4 bytes): A **DTTM** that MAY<169> specify the date and time at which the document was last printed.

nRevision (2 bytes): A signed integer that MAY<170> specify the number of times that this document was resaved. This MUST be a value between 0 and 0x7FFF.

tmEdited (4 bytes): A signed integer value that MAY<171> specify the time it took, in minutes, for the document to be opened for editing and then subsequently saved.

cWords (4 bytes): A signed integer value that specifies the last calculated or the estimated count of words in the main document, depending on **fExactCWords** and **fIncludeSubdocsInStats**.

cCh (4 bytes): A signed integer value that specifies the last calculated or estimated count of characters in the main document, depending on the values of **fExactCWords** and **fIncludeSubdocsInStats**. The character count excludes whitespace.

cPg (2 bytes): A signed integer value that specifies the last calculated or estimated count of pages in the main document, depending on the values of **fExactCWords** and **fIncludeSubdocsInStats**.

cParas (4 bytes): A signed integer value that specifies the last calculated or estimated count of paragraphs in the main document, depending on the values of **fExactCWords** and **fIncludeSubdocsInStats**.

m - rncEdn (2 bits): Specifies when automatic numbering for the endnote reference marks is reset to the beginning number for documents that have an nFib value that is less than or equal to 0x00D9. For those documents that rely on **rncEdn**, when restarted, the next automatically numbered endnote in the document is reset to the specified **nEdn** value. This value MUST be one of the following.

Value	Meaning
0	Specifies that the numbering of endnotes continues from the previous section in the document.
1	Specifies that the numbering of endnotes is reset to its starting value for each unique section in the document.
2	Specifies that the numbering of endnotes is reset to its starting value for each unique page in the document.

nEdn (14 bits): For those documents that have an nFib value that is less than or equal to 0x00D9, this element specifies the starting number for the first automatically numbered endnote in the document, and the first automatically numbered endnote after each restart point that is specified by the **rncEdn** element.

epc (2 bits): Specifies where endnotes are placed on the page when they are referenced by text in the current document. This value MUST be one of the following.

Value	Meaning
0	Specifies that endnotes are placed at the end of the section in which they are referenced.
3	Specifies that all endnotes are placed at the end of the current document, regardless of the section within which they are referenced.

n - unused14 (4 bits): This value is undefined and MUST be ignored.

o - unused15 (4 bits): This value is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

p - fPrintFormData (1 bit): Specifies whether to print only form field results, as specified in [ECMA-376] Part 1, Section 17.15.1.61 printFormsData.

q - fSaveFormData (1 bit): Specifies whether the application SHOULD<172> only save form field contents into a comma-delimited text file and ignore all other content in the document as specified in [ECMA-376] Part 1, Section 17.15.1.73 saveFormsData.

r - fShadeFormData (1 bit): Specifies whether to display visual cues around form fields as specified in [ECMA-376] Part 1, Section 17.15.1.39 doNotShadeFormData, where the meaning of the **doNotShadeFormData** element is the opposite of **fShadeFormData**.

s - fShadeMergeFields (1 bit): Specifies whether to display visual cues around mail merge fields.

t - reserved2 (1 bit): This value MUST be 0, and MUST be ignored.

u - fIncludeSubdocsInStats (1 bit): Specifies whether **cCh**, **cChWS**, **cWords**, **cParas**, **cLines**, **cDBC**, **cChWithSubdocs**, **cChWSWithSubdocs**, **cWordsWithSubdocs**, **cParasWithSubdocs**, **cLinesWithSubdocs**, or **cDBCWithSubdocs** are calculated and displayed, or estimated.

cLines (4 bytes): A signed integer that specifies the last calculated or estimated count of lines in the main document, depending on the values of **fExactCWords** and **fIncludeSubdocsInStats**.

cWordsWithSubdocs (4 bytes): A signed integer that specifies the last calculated or estimated count of words in the main document, footnotes, endnotes, and text boxes in the main document, depending on the values of **fExactCWords** and **fIncludeSubdocsInStats**.

cChWithSubdocs (4 bytes): A signed integer that specifies the last calculated or estimated count of characters, excluding whitespace, in the main document, footnotes, endnotes, and text boxes in the main document, depending on the values of **fExactCWords** and **fIncludeSubdocsInStats**.

cPgWithSubdocs (2 bytes): A signed integer that specifies the last calculated or estimated count of pages in the main document, footnotes, endnotes, and text boxes that are anchored in the main document, depending on the values of **fExactCWords** and **fIncludeSubdocsInStats**.

cParasWithSubdocs (4 bytes): A signed integer that specifies the last calculated or estimated count of paragraphs in the main document, footnotes, endnotes, and text boxes that are anchored in the main document, depending on the values of **fExactCWords** and **fIncludeSubdocsInStats**.

cLinesWithSubdocs (4 bytes): A signed integer that specifies the last calculated or estimated count of lines in the main document, footnotes, endnotes, and text boxes that are anchored in the main document, depending on the values of **fExactCWords** and **fIncludeSubdocsInStats**.

lKeyProtDoc (4 bytes): A signed integer that specifies the hash of the password that is used with document protection (**fLockRev**, **fProtEnabled**, **fLockAtn** and **fRevMarking**), as specified in [ECMA-376] Part 1, Section 17.15.1.29 documentProtection.

v - wvkoSaved (3 bits): Specifies the viewing mode that was in use when the document was last saved. If the viewing mode that was in use cannot be represented by a valid value, an alternate view mode is specified. See [ECMA-376] Part 1, section 17.15.1.92 view; the values are mapped as follows.

Copyright © 2025 Microsoft Corporation

wvkoSaved value	ECMA attribute value
0	none
1	print
2	outline
3	masterPages
4	normal
5	web

A value of 0 specifies the default view mode of the application.

pctWwdSaved (9 bits): Specifies the zoom percentage that was in use when the document was saved. A value of 0 specifies the default zoom percentage of the application. This value MUST be 0 or a value between 10 and 500.

w - zkSaved (2 bits): Specifies the zoom type that was in use when the document was saved. See [ECMA-376] Part 1, Section 17.18.105 ST_Zoom; the values are mapped as follows.

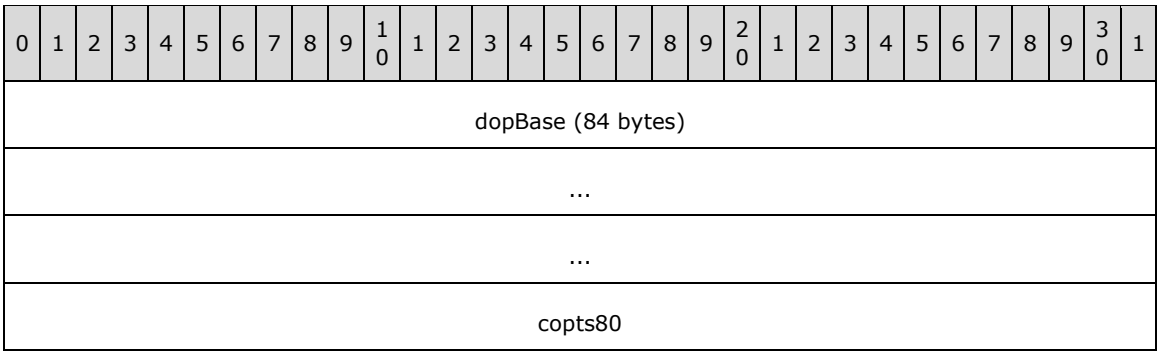
zkSaved value	ECMA attribute value
0	none
1	fullPage
2	bestFit
3	textFit

x - unused16 (1 bit): This value is undefined and MUST be ignored.

y - iGutterPos (1 bit): Specifies whether the document gutter shall be positioned at the top of the pages of the document when the document is displayed. See [ECMA-376] Part 1, Section 17.15.1.50 gutterAtTop, where **mirrorMargins** corresponds to **fMirrorMargins**, **bookFoldPrinting** corresponds to Dop2002.**fFolioPrint**, **bookFoldRevPrinting** corresponds to Dop2002.**fReverseFolio** and **printTwoOnOne** corresponds to DopTypography.**f2on1**.

2.7.3 Dop95

The **Dop95** structure contains document and compatibility settings. These settings influence the appearance and behavior of the current document and store document-level state.



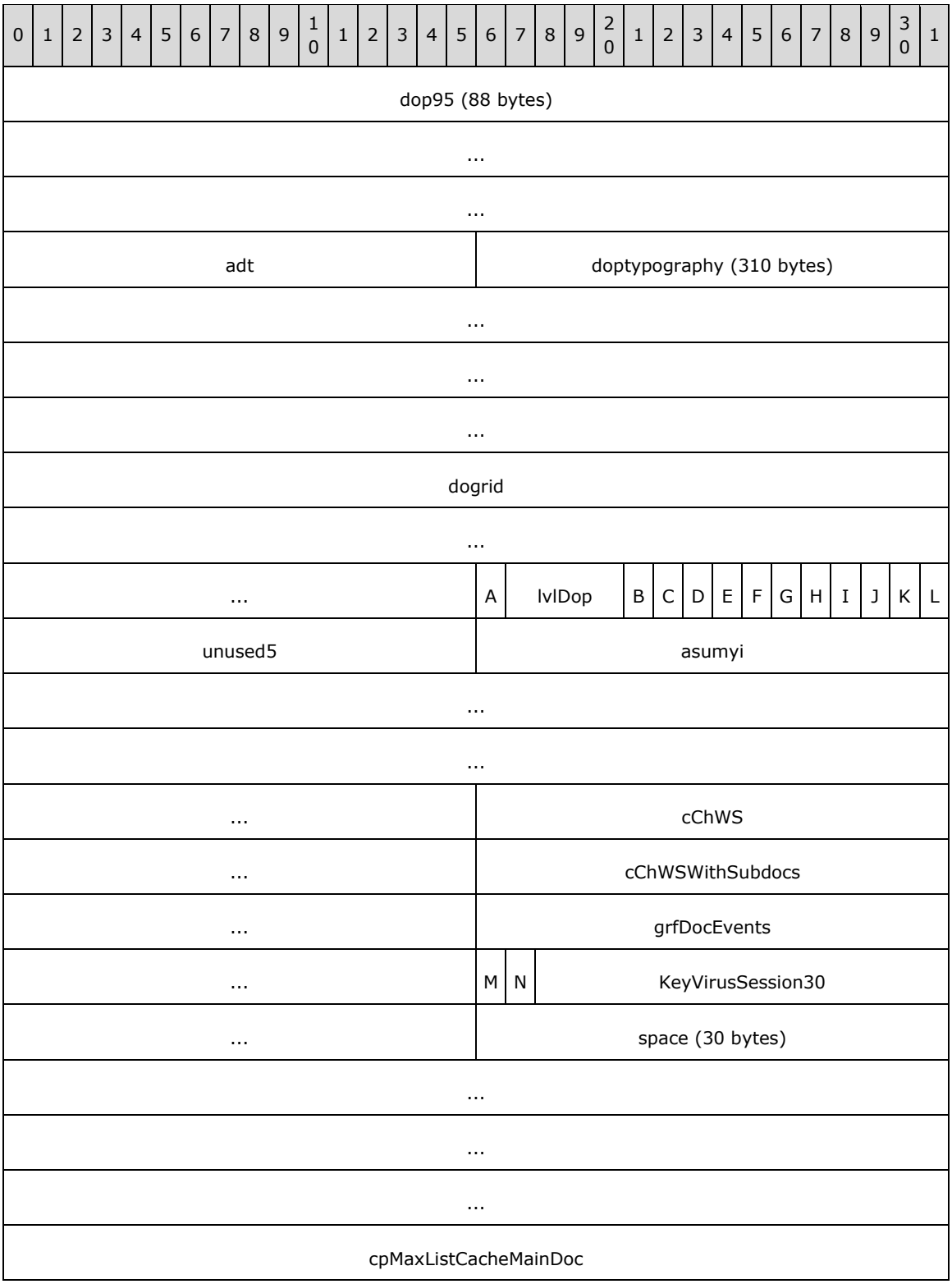
dopBase (84 bytes): A **DopBase** structure that specifies document and compatibility settings.

copts80 (4 bytes): A copts80 specifying compatibility options. **Copts80.copts60** components MUST be equal to **DopBase.copts60**.

Copyright © 2025 Microsoft Corporation

2.7.4 Dop97

The **Dop97** structure contains document and compatibility settings. These settings influence the appearance and behavior of the current document and store the document-level state.



Copyright © 2025 Microsoft Corporation

ilfoLastBulletMain	ilfoLastNumberMain
cDBC	
cDBCWithSubdocs	
reserved3a	
nfcFtnRef	nfcEdnRef
hpsZoomFontPag	dywDispPag

dop95 (88 bytes): A **Dop95** that specifies document and compatibility settings.

adt (2 bytes): Specifies the document classification as specified in [ECMA-376] Part 1, Section 17.15.1.30 documentType; the values are mapped as follows.

adt value	ECMA attribute value
0x0000	notSpecified
0x0001	letter
0x0002	eMail

doptypography (310 bytes): A DopTypography that specifies typography settings.

dogrid (10 bytes): A Dogrid that specifies the draw object grid settings.

A - unused1 (1 bit): This bit is undefined and MUST be ignored.

lvlDop (4 bits): This value SHOULD<173> specify which outline levels were showing in outline view at the time of the last save operation. This MUST be a value between 0 and 9, inclusive, or this value MUST be 15.

Value	Levels showing
0x0	Heading 1
0x1	Headings 1 and 2
0x2	Headings 1, 2 and 3
0x3	Headings 1, 2, 3 and 4
0x4	Headings 1, 2, 3, 4 and 5
0x5	Headings 1, 2, 3, 4, 5 and 6
0x6	Headings 1, 2, 3, 4, 5, 6 and 7
0x7	Headings 1, 2, 3, 4, 5, 6, 7 and 8
0x8	Headings 1, 2, 3, 4, 5 , 6, 7, 8 and 9
0x9	All levels
0xF	All levels

B - fGramAllDone (1 bit): Specifies whether the grammar of all content in this document was checked.

C - fGramAllClean (1 bit): Specifies whether all content in this document can be considered grammatically correct.

Copyright © 2025 Microsoft Corporation

D - fSubsetFonts (1 bit): Specifies whether to subset fonts when embedding as specified in [ECMA-376] Part 1, Section 17.8.3.15 saveSubsetFonts, where **embedTrueTypeFonts** refers to DopBase.**fEmbedFonts**.

E - unused2 (1 bit): This value is undefined and MUST be ignored.

F - fHtmlDoc (1 bit): This value SHOULD<174> be 0.

G - fDiskLvcInvalid (1 bit): This bit MAY<175> specify whether the saved **ListNum** field cache contains valid information. The **ListNum** field cache is specified by FibRgFcLcb97.**fcPlcfBteLvc**.

H - fSnapBorder (1 bit): Specifies whether to align paragraph and table borders with the page border, as specified in [ECMA-376] Part 1, Section 17.15.1.2 alignBordersAndEdges.

I - fIncludeHeader (1 bit): Specifies whether to draw the page border so that it includes the header area.

J - fIncludeFooter (1 bit): Specifies whether to draw the page border so that it includes the footer area.

K - unused3 (1 bit): This value is undefined and MUST be ignored.

L - unused4 (1 bit): This value is undefined and MUST be ignored.

unused5 (2 bytes): This value is undefined and MUST be ignored.

asumyi (12 bytes): An Asumyi that specifies the **AutoSummary** settings.

cChWS (4 bytes): Specifies the last calculated or estimated count of characters in the main document depending on the values of **fExactCWords** and **fIncludeSubdocsInStats**. The count of characters includes whitespace.

cChWSWithSubdocs (4 bytes): Specifies the last calculated or estimated count of characters in the main document, footnotes, endnotes, and text boxes that are anchored in the main document, depending on **fExactCWords** and **fIncludeSubdocsInStats**. The count of characters includes whitespace.

grfDocEvents (4 bytes): A bit field that specifies which document events are fired. The individual bits and their meanings are as follows.

Bit Mask	Event
0x00000001	New
0x00000002	Open
0x00000004	Close
0x00000008	Sync
0x00000010	XMLAfterInsert
0x00000020	XMLBeforeDelete
0x00000100	BBAfterInsert
0x00000200	BBBeforeDelete
0x00000400	BBOnExit
0x00000800	BBOnEnter
0x00001000	StoreUpdate
0x00002000	BBContentUpdate
0x00004000	LegoAfterInsert

All other bits MUST be set to 0.

Copyright © 2025 Microsoft Corporation

M - fVirusPrompted (1 bit): Specifies whether the macro security prompt is shown in this session for this document.

N - fVirusLoadSafe (1 bit): Specifies whether to disable macros for this session.

KeyVirusSession30 (30 bits): A random value to match against the current session key. If they match, this is the same session.

space (30 bytes): This value is undefined and MUST be ignored.

cpMaxListCacheMainDoc (4 bytes): This value MAY<176> specify the maximum CP value for which the **ListNum** field cache contains valid information. The **ListNum** field cache is specified by **FibRgFcLcb97.fcPlcfBteLvc**.

ilfoLastBulletMain (2 bytes): Specifies the index of the last **LFO** structure that was used for bullets in the document before the save operation. This value MUST be between 0 and a number that is one less than the number of entries in **FibRgFcLcb97.fcPlfLfo**, unless there are 0 entries, in which case this value MUST be 0.

ilfoLastNumberMain (2 bytes): Specifies the index of the last **LFO** structure that was used for list numbering in the document before the save operation. This value MUST be between 0 and a number that is one less than the number of entries in **FibRgFcLcb97.fcPlfLfo**, unless there are 0 entries, in which case this value MUST be 0.

cDBC (4 bytes): Specifies the last calculated or estimated count of double-byte characters in the main document, depending on the values of **DopBase.fExactCWords** and **DopBase.fIncludeSubdocsInStats**. The count of characters includes whitespace.

cDBCWithSubdocs (4 bytes): Specifies the last calculated or estimated count of double-byte characters in the main document, footnotes, endnotes, and text boxes anchored in the main document depending on **DopBase.fExactCWords** and **DopBase.fIncludeSubdocsInStats**. The character count includes whitespace.

reserved3a (4 bytes): This value is undefined and MUST be ignored.

nfcFtnRef (2 bytes): An **MSONFC** (as specified in [MS-OSHARED] section 2.2.1.3) that, for those documents that have an **nFib** which is less than or equal to 0x00D9, specifies the numbering format code to use for footnotes in the document.

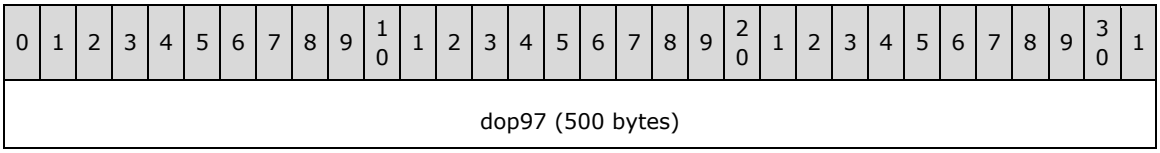
nfcEdnRef (2 bytes): An **MSONFC** (as specified in [MS-OSHARED] section 2.2.1.3) that, for those documents that have an **nFib** which is less than or equal to 0x00D9, specifies the numbering format code to use for endnotes in the document.

hpsZoomFontPag (2 bytes): Specifies the size, in half points, of the maximum font size to be enlarged in the view "online layout" at the time the document was last paginated. This value SHOULD<177> be ignored.

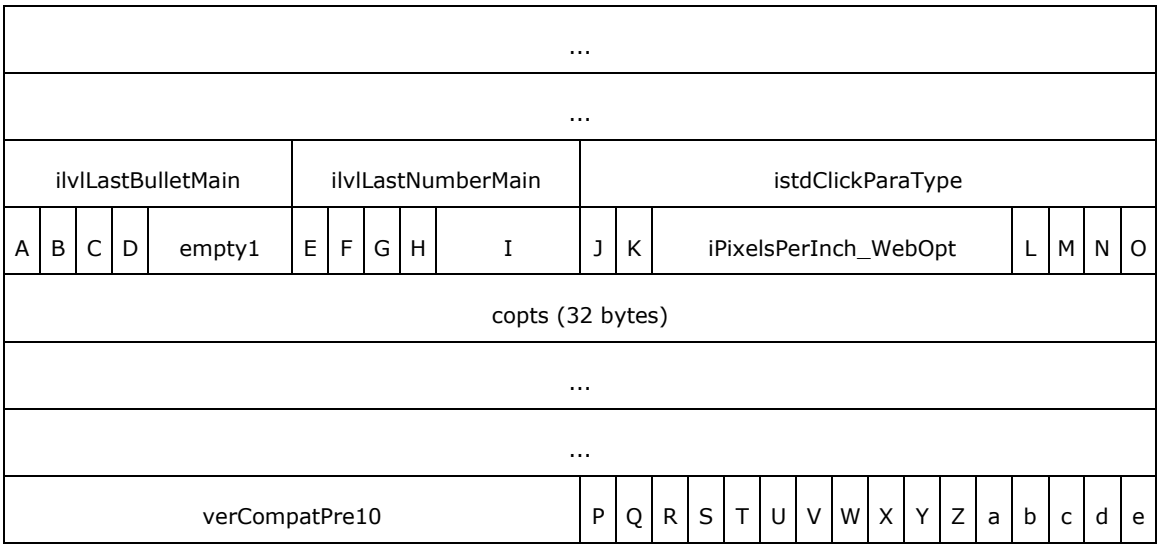
dywDispPag (2 bytes): Height of the screen, in pixels, at the time that the document was last paginated. This value SHOULD<178> be ignored.

2.7.5 Dop2000

A structure that contains document and compatibility settings. These settings influence the appearance and behavior of the current document and store document-level state.



Copyright © 2025 Microsoft Corporation



dop97 (500 bytes): A Dop97 that specifies document and compatibility settings.

ilvlLastBulletMain (1 byte): SHOULD<179> specify the last bullet level applied via the toolbar before saving. MUST be between 0 and 9. Default is 0.

ilvlLastNumberMain (1 byte): SHOULD<180> specify the last list numbering level applied via the toolbar before saving. MUST be between 0 and 9. Default is 0.

istdClickParaType (2 bytes): Specifies the ISTD of the paragraph style to use for paragraphs that are automatically created by the click and type feature to place the cursor where the user clicked. Default value is 0 (Normal paragraph style).

A - fLADAllDone (1 bit): Specifies whether **language auto-detection** has run to completion for the document. Default is 0.

B - fEnvelopeVis (1 bit): Specifies whether to show the E-Mail message header as specified in [ECMA-376] Part 1, Section 17.15.1.79 showEnvelope. Default is 0.

C - fMaybeTentativeListInDoc (1 bit): Specifies whether the document potentially contains tentative lists<181>. Default is 0. See LVLF.**fTentative.**

D - fMaybeFitText (1 bit): If this is 0, then there MUST NOT be any fit text (see sprmCFitText) in the document. Default is 0.

empty1 (4 bits): MUST be zero, and MUST be ignored.

E - fFCCAllDone (1 bit): Specifies whether the **format consistency checker** has run to completion for the document. Default is 0.

F - fRelyOnCSS_WebOpt (1 bit): Specifies whether to rely on **CSS** for font face formatting when saving as a Web page as specified in [ECMA-376] Part 1, Section 17.15.2.11 doNotRelyOnCSS, where the meaning is the opposite of **fRelyOnCSS_WebOpt**. The default is 1.

G - fRelyOnVML_WebOpt (1 bit): Specifies whether to use **VML** when saving as a Web page as specified in [ECMA-376] Part 4, Section 17.8.4.1 relyOnVML. The default is 0.

H - fAllowPNG_WebOpt (1 bit): Specifies whether to allow Portable Network Graphics (PNG) format as a graphic format when saving as a Web page as specified in [ECMA-376] Part 1, Section 17.15.2.1 allowPNG. Default value is 0.

Copyright © 2025 Microsoft Corporation

I - screenSize_WebOpt (4 bits): Specifies what the target screen size for the Web page is as specified in [ECMA-376] Part 1, Section 17.15.2.41 targetScreenSz, where **screenSize_WebOpt** value maps to **ST_TargetScreenSz** types as follows

Value	ST_TargetScreenSz string
0	544x376
1	640x480
2	720x512
3 (default)	800x600
4	1024x768
5	1152x882
6	1152x900
7	1280x1024
8	1600x1200
9	1800x1440
10	1920x1200

J - fOrganizeInFolder_WebOpt (1 bit): Specifies whether to place supporting files in a subdirectory when saving as a Web page as specified in [ECMA-376] Part 1, Section 17.15.2.10 doNotOrganizeInFolder, where the meaning is the opposite of **fOrganizeInFolder_WebOpt**. The default is 1.

K - fUseLongFileNames_WebOpt (1 bit): Specifies whether to use file names longer than 8.3 characters when saving as a Web page as specified in [ECMA-376] Part 1, Section 17.15.2.13 doNotUseLongFileNames, where the meaning is the opposite of **fUseLongFileNames_WebOpt**. The default is 1.

iPixelsPerInch_WebOpt (10 bits): Specifies the pixels per inch for graphics/images when saving as a Web page as specified in [ECMA-376] Part 1, Section 17.15.2.34 pixelsPerInch. If **fWebOptionsInit** is 1 then this MUST be between 19 and 480; otherwise, this is ignored. The default is 96.

L - fWebOptionsInit (1 bit): Specifies whether **fRelyOnCSS_WebOpt**, **fRelyOnVML_WebOpt**, **fAllowPNG_WebOpt**, **screenSize_WebOpt**, **fOrganizeInFolder_WebOpt**, **fUseLongFileNames_WebOpt** and **iPixelsPerInch_WebOpt** contain valid data. When **fWebOptionsInit** is set to 0, the value of all those fields MUST be ignored. The default is 0.

M - fMaybeFEL (1 bit): If this is 0, then there MUST NOT be any **Warichu**, **Tatenakayoko**, **Ruby**, **Kumimoji** or EncloseText in the document. Enclose Text is a layout feature that uses EQ fields ([ECMA-376] part 4, section 14.10.4.6) to enclose characters in circles or other characters. The default is 0.

N - fCharLineUnits (1 bit): If this is 0, then there MUST NOT be any character unit indents (sprmPDxcLeft, sprmPDxcLeft1, sprmPDxcRight) or line units (sprmPDylBefore, sprmPDylAfter) in use. The default is 0.

O - unused1 (1 bit): Undefined and MUST be ignored.

copts (32 bytes): A copts that specifies compatibility options. Components of **Copts.copts80** MUST be equal to components of **Dop97.copts80**.

verCompatPre10 (16 bits): A bit field that specifies the desired feature set to use for the document. This overrides DopBase.**fWord97Compat**. Values are composed from the following table:

Copyright © 2025 Microsoft Corporation

Bit Value	Meaning
0x0000 (default)	No Restrictions on feature use
0x0004	Use only features available in Microsoft Word for Windows 95.
0x0008	Use only features available in Microsoft Word 97.
0x0040	Use only features available in the East Asian release of Word for Windows 95.
0x0800	Use only features available in Microsoft Office Word 2003.

All other bits are undefined and MUST be ignored.

P - fNoMargPgvwSaved (1 bit): Specifies whether to suppress the display of the header and footer area when in print layout view so that the main text area of one page is displayed adjacent to the main text area of the next page as specified in [ECMA-376] Part 1, Section 17.15.1.35 doNotDisplayPageBoundaries. Default is 0.

Q - unused2 (1 bit): Undefined and MUST be ignored.

R - unused3 (1 bit): Undefined and MUST be ignored.

S - unused4 (1 bit): Undefined and MUST be ignored.

T - fBulletProofed (1 bit): Specifies that this document was produced by the Open and Repair feature. Default is 0.

U - empty2 (1 bit): MUST be zero, and MUST be ignored.

V - fSaveUim (1 bit): Specifies whether to save **UIM** data in the document. Default is 1.

W - fFilterPrivacy (1 bit): Specifies whether to remove personal information from the document properties on save as specified in [ECMA-376] Part 1, Section 17.15.1.68 removePersonalInformation. Default is 0.

X - empty3 (1 bit): MUST be zero, and MUST be ignored.

Y - fSeenRepairs (1 bit): Specifies whether the user has seen any repairs made by the Open and Repair feature. Default is 0.

Z - fHasXML (1 bit): Specifies whether the document has any form of **structured document tags** in it. Default is 0.

a - unused5 (1 bit): Undefined and MUST be ignored.

b - fValidateXML (1 bit): Specifies whether to validate custom XML markup against any attached schemas as specified in [ECMA-376] Part 1, Section 17.15.1.43 doNotValidateAgainstSchema, where the meaning is the opposite of **fValidateXML**. Default is 1

c - fSaveInvalidXML (1 bit): Specifies whether to allow saving the document as an XML file when the custom XML markup is invalid with respect to the attached schemas as specified in [ECMA-376] Part 1, Section 17.15.1.74 saveInvalidXml. Default is 0.

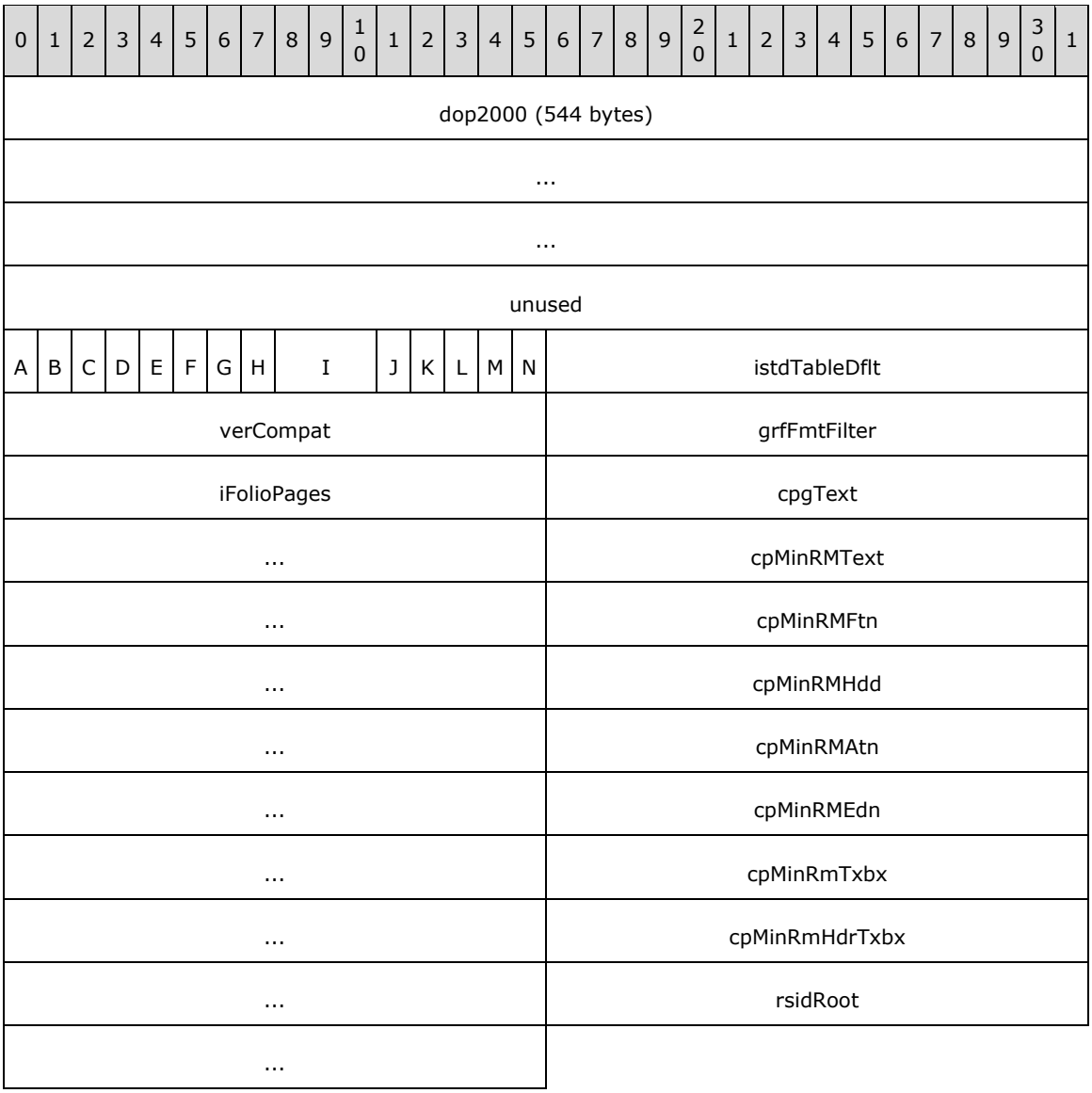
d - fShowXMLErrors (1 bit): Specifies whether to show a visual indicator for invalid custom XML markup as specified in [ECMA-376] Part 1, Section 17.15.1.34 doNotDemarcateInvalidXml, where the meaning is the opposite of **fShowXMLErrors**.

e - fAlwaysMergeEmptyNamespace (1 bit): Specifies whether to consider custom XML elements with no namespace as valid on open as specified in [ECMA-376] Part 1, Section 17.15.1.3 alwaysMergeEmptyNamespace. Default is 0.

Copyright © 2025 Microsoft Corporation

2.7.6 Dop2002

A structure that contains document and compatibility settings. These settings influence the appearance and behavior of the current document and store document-level state.



dop2000 (544 bytes): A Dop2000 that specifies document and compatibility settings.

unused (4 bytes): Undefined and MUST be ignored.

A - fDoNotEmbedSystemFont (1 bit): Specifies whether common system fonts are not to be embedded as specified in [ECMA-376] Part 1, Section 17.8.3.7 embedSystemFonts, where the meaning is the opposite of **fDoNotEmbedSystemFont** and the **embedTrueTypeFonts** element refers to DopBase.**fEmbedFonts**. Default is 1.

B - fWordCompat (1 bit): Specifies that features not compatible with the settings specified in **verCompat** will be disabled or removed when saving. Default is 0.

Copyright © 2025 Microsoft Corporation

C - fLiveRecover (1 bit): Specifies that this file is a recovered document from after a crash. Default is 0.

D - fEmbedFactoids (1 bit): Specifies whether smart tags are to remain in the document when saving. Smart tags are to be removed when **fEmbedFactoids** is set to 0. See [ECMA-376] Part 1, Section 17.15.1.36 doNotEmbedSmartTags, where the meaning is the opposite of **fEmbedFactoids**. Default is 1.

E - fFactoidXML (1 bit): Specifies whether to save smart tag data as an XML-based property bag at the head of the **HTML** page when saving as HTML as specified in [ECMA-376] Part 1, Section 17.15.2.36 saveSmartTagsAsXml. Default is 0.

F - fFactoidAllDone (1 bit): Specifies whether the document has been completely scanned for all possible smart tag creations. Default is 0.

G - fFolioPrint (1 bit): Specifies whether to use book fold printing as specified in [ECMA-376] Part 1, Section 17.15.1 11 bookFoldPrinting. Default is 0.

H - fReverseFolio (1 bit): Specifies whether to use reverse book fold printing as specified in [ECMA-376] Part 1, Section 17.15.1.13 bookFoldRevPrinting. If this is 1 then **fFolioPrint** MUST be 1. Default is 0.

I - iTextLineEnding (3 bits): Specifies what to end a line of text with when saving as a text file via automation. It MUST be one of the values in the following table:

Value	Meaning
0 (default)	Carriage return (0x0D) followed by line feed (0x0A).
1	Carriage return (0x0D).
2	Line feed (0x0A).
3	Line feed (0x0A) followed by carriage return (0x0D).
4	If the **code page** supports it, Line Separator (U+2028) or Paragraph Separator (U+2029) otherwise behave as follows: ▪ If the codepage is CP_JAPANEUC (51932), CP_CHINAEUC (51936), CP_KOREAEUC (51949) or CP_TAIWANEUC (51950) treat as if the value were 2. ▪ If the code page is greater than or equal to 10000 and less than 20000, then treat as if the value where 1. ▪ If neither of those apply, then treat as if the value were 0.

J - fHideFcc (1 bit): Specifies whether to refrain from showing a visual cue around ranges flagged by the **format consistency checker** as suspect. Default is 0.

K - fAcetateShowMarkup (1 bit): Specifies whether to visually indicate any additional nonprinting area used to display annotations when the annotations in this document are displayed. Default is 1.

L - fAcetateShowAtn (1 bit): Specifies if comments are included when the contents of this document are displayed. Default is 1.

M - fAcetateShowInsDel (1 bit): Specifies if revisions to content are included when the contents of this document are displayed. Default is 1.

N - fAcetateShowProps (1 bit): Specifies whether **property revision marks** are included when the contents of this document are displayed. Default is 1.

istdTableDflt (16 bits): An istd that specifies the default table style for newly inserted tables.

Copyright © 2025 Microsoft Corporation

verCompat (16 bits): A bit field that specifies the desired feature set to use for the document. This overrides DopBase.**fWord97Compat** and Dop2000.**verCompatPre10**. The bit values are as follows:

Value	Meaning
0x0000	No restrictions on feature use.
0x0001	Use features supported by Microsoft® Internet Explorer® 4.0.
0x0002	Use features supported by Microsoft® Internet Explorer® 5.0.
0x0004	Use features supported by Word for Windows 95.
0x0008	Use features supported by Word 97.
0x0010	Use features supported by the Word HTML format.
0x0020	Use features supported by the Word RTF format.
0x0040	Use features supported by East Asian versions of Word for Windows 95.
0x0080	Use features supported by plain text e-mail messages.
0x0100	Use features supported by Internet Explorer 6.0.
0x0200	Use features supported by the Word XML format.
0x0400	Use features supported by RTF e-mail messages.
0x0800	Do not use features introduced in Microsoft Office Word 2007.
0x1000	Use features supported by plain text.

Default is 0.

grfFmtFilter (2 bytes): Specifies the suggested filtering for the list of document styles as specified in [ECMA-376] Part 1, Section 17.15.1.85 stylePaneFormatFilter. Default is 0x5024.

iFolioPages (2 bytes): Specifies the number of pages per booklet as specified in [ECMA-376] Part 1, Section 17.15.1.12 bookFoldPrintingSheets, where bookFoldPrinting refers to **fFolioPrint** and bookFoldRevPrinting refers to **fReverseFolio**. Default is 0.

cpgText (4 bytes): Specifies the code page to use when saving as encoded text. Default is the current Windows ANSI code page for the system.

cpMinRMText (4 bytes): A CP in the main document before which there are no revisions. Default is 0.

cpMinRMFtn (4 bytes): A CP in the footnote document before which there are no revisions. Default is 0.

cpMinRMHdd (4 bytes): A CP in the header document before which there are no revisions. Default is 0.

cpMinRMAtn (4 bytes): A CP in the comment document before which there are no revisions. Default is 0.

cpMinRMEdn (4 bytes): A CP in the endnote document before which there are no revisions. Default is 0.

cpMinRmTxbx (4 bytes): A CP in the textbox document for the main document before which there are no revisions. Default is 0.

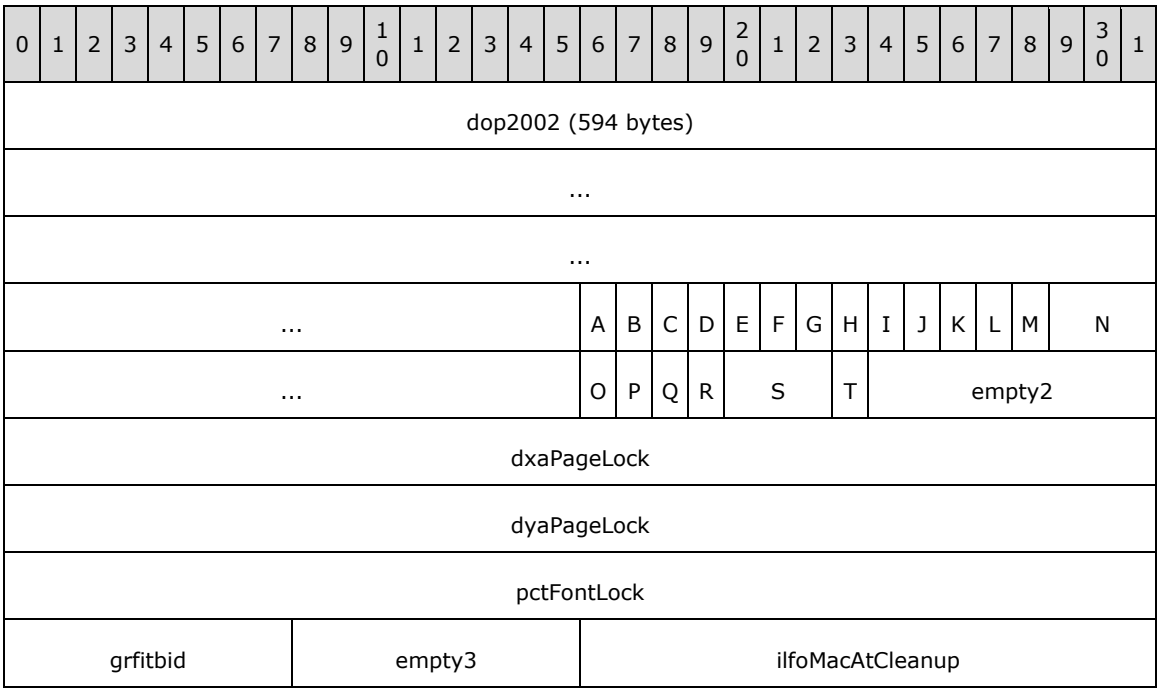
cpMinRmHdrTxbx (4 bytes): A CP in the header textbox document before which there are no revisions. Default is 0.

Copyright © 2025 Microsoft Corporation

rsidRoot (4 bytes): Specifies the original document revision save ID as specified in [ECMA-376] Part 1, Section 17.15.1.71 rsidRoot. By default the **rsidRoot** is not that of the currently running session.

2.7.7 Dop2003

The **Dop2003** structure contains document and compatibility settings. These settings influence the appearance and behavior of the current document and store document-level state.



dop2002 (594 bytes): A Dop2002 that specifies document and compatibility settings.

A - fTreatLockAtnAsReadOnly (1 bit): Specifies whether DopBase.**fLockAtn** means read-only protection instead of protect for comments. By default, this value is 0.

B - fStyleLock (1 bit): Specifies whether the styles available to use in the document are restricted to those styles with STD.Stdf.StdfBase.GRFSTD.**fLocked** set to 1 when style lock is enforced (**fStyleLockEnforced** is 1). By default, this value is 0.

C - fAutoFmtOverride (1 bit): Specifies whether to allow automatic formatting to override the **fStyleLock** setting as specified in [ECMA-376] Part 1, Section 17.15.1.9 autoFormatOverride. By default, this value is 0.

D - fRemoveWordML (1 bit): Specifies whether to save only custom XML markup when saving to XML as specified in [ECMA-376] Part 1, Section 17.15.1.77 saveXmlDataOnly. By default, this value is 0.

E - fApplyCustomXForm (1 bit): Specifies whether to save the document through the custom XML transform specified via FibRgFcLcb2003.**fcCustomXForm** and FibRgFcLcb2003.**lcbCustomXForm** when saving to XML as specified in [ECMA-376] Part 1, Section 17.15.1.91 useXSLTWhenSaving. By default, this value is 0.

F - fStyleLockEnforced (1 bit): Specifies whether to actively enforce the style restriction as specified by **fStyleLock**. If **fStyleLockEnforced** is 1, **fStyleLock** MUST be 1. By default, this value is 0.

Copyright © 2025 Microsoft Corporation

G - fFakeLockAtn (1 bit): Specifies that the DopBase.**fLockAtn** setting is to be honored only if the application does not support **fStyleLock**. By default, this value is 0.

H - fIgnoreMixedContent (1 bit): Specifies whether to ignore all text not in leaf nodes of the custom XML when validating custom XML markup as specified in [ECMA-376] Part 1, Section 17.15.1.54 ignoreMixedContent. By default, this value is 0.

I - fShowPlaceholderText (1 bit): Specifies whether to show some form of in-document placeholder text when custom XML markup contains no content and the custom XML tags are not being displayed as specified in [ECMA-376] Part 1, Section 17.15.1.4 alwaysShowPlaceholderText. By default, this value is 0.

J - unused (1 bit): This value is undefined and MUST be ignored.

K - fWord97Doc (1 bit): Specifies whether to disable UI for features incompatible with the Word Binary File Format. By default, this value is 0.

L - fStyleLockTheme (1 bit): Specifies whether to prevent modification of the document theme information as specified in [ECMA-376] Part 1, Section 17.15.1.84 styleLockTheme. By default, this value is 0.

M - fStyleLockQFSet (1 bit): Specifies whether to prevent the replacement of style sets as specified in [ECMA-376] Part 1, Section 17.15.1.83 styleLockQFSet. By default, this value is 0.

N - empty1 (19 bits): This value MUST be 0, and MUST be ignored.

O - fReadingModeInkLockDown (1 bit): Specifies whether to permanently set the layout to the specific set of page and text-sizing parameters specified by **dxaPageLock**, **dyaPageLock** and **pctFontLock** as specified in [ECMA-376] Part 1, Section 17.15.1.66 readModeInkLockDown. By default, this value is 0.

P - fAcetateShowInkAtn (1 bit): Specifies whether to include ink annotations when the contents of this document are displayed. By default, this value is 1.

Q - fFilterDttm (1 bit): Specifies whether to remove date and time information from annotations as specified in [ECMA-376] Part 1, Section 17.15.1.67 removeDateAndTime. By default, this value is 0.

R - fEnforceDocProt (1 bit): Specifies whether to enforce the document protection mode that is specified by **iDocProtCur**. By default, this value is 0.

S - iDocProtCur (3 bits): Specifies the document protection mode that is in effect when **fEnforceDocProt** is set to 1. This MUST be set to one of the following values.

Value	Meaning
0	Track all edits that are made to the document as revisions.
1	Comments are permitted to be inserted or deleted, and regions that are delimited by range permissions can be edited if they match the editing rights of the user account which is performing the editing. See PRTI.
2	Edits are restricted to the editing of form fields in sections where sprmSFProtected results in a value of "true". Edits are not restricted in sections where sprmSFProtected is not present or has a value of "false".
3 (Default)	Edits are restricted to regions delimited by range permissions which match the editing rights of the user account which is performing the editing. See PRTI.
7	There are no editing restrictions.

Copyright © 2025 Microsoft Corporation

T - fDispBkSpSaved (1 bit): Specifies whether to display background objects when displaying the document in print layout view as specified in [ECMA-376] Part 1, Section 17.15.1.26 displayBackgroundShape. By default, this value is 0.

empty2 (8 bits): This value MUST be 0, and MUST be ignored.

dxaPageLock (4 bytes): Specifies the width, in **twips**, of the virtual pages that are used in this document when **fReadingModeInkLockDown** is 1. By default, this value is 0.

dyaPageLock (4 bytes): Specifies the height, in twips, of the virtual pages that are used in this document when **fReadingModeInkLockDown** is 1. By default, this value is 0.

pctFontLock (4 bytes): Specifies the percentage to which text in the document is scaled before it is displayed on a virtual page when **fReadingModeInkLockDown** is 1. By default, this value is 0.

grfitbid (1 byte): A bit field that specifies what toolbars were shown because of document state rather than explicit user action at the moment of saving. This value MUST be composed of the following bit values.

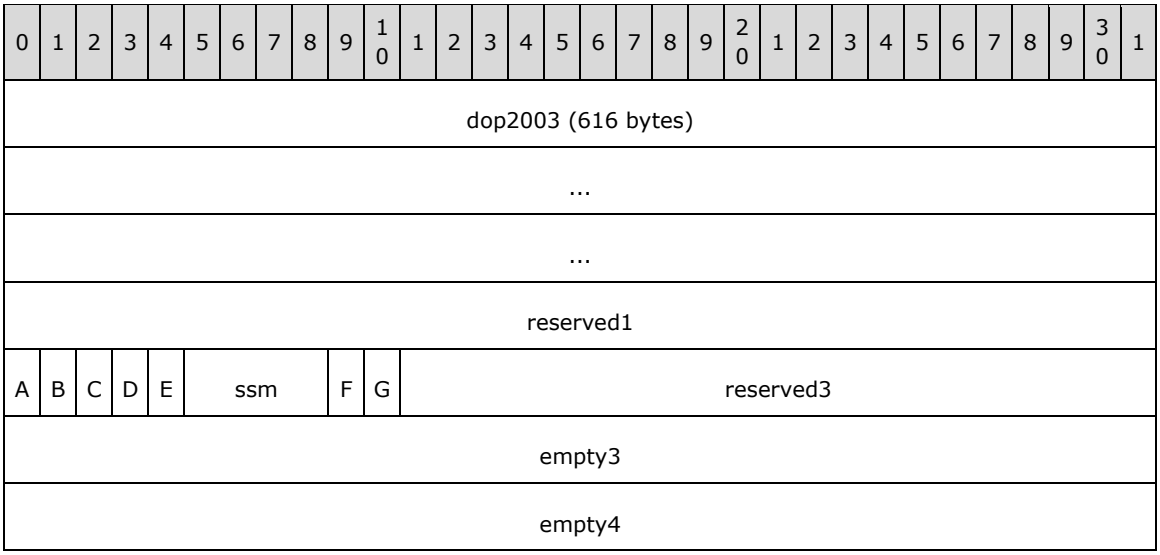
Value	Meaning
0x00 (default)	No toolbar was shown because of document state.
0x01	The reviewing toolbar was shown.
0x02	The Web toolbar was shown.
0x04	The mail merge toolbar was shown.

empty3 (1 byte): This value MUST be 0, and MUST be ignored.

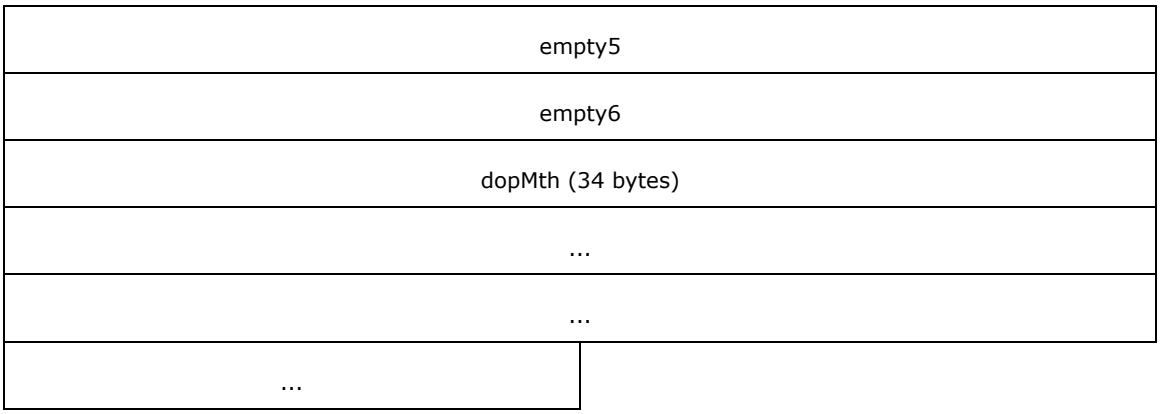
ilfoMacAtCleanup (2 bytes): Specifies the largest **ilfo** value (index into PlfLfo) such that all PlfLfo entries from 0 to **ilfoMacAtCleanup** are searched for unused values to be pruned as specified in [ECMA-376] Part 1, Section 17.9.19 numIdMacAtCleanup. By default, this value is 0.

2.7.8 Dop2007

The **Dop2007** structure contains document and compatibility settings. These settings influence the appearance and behavior of the current document and store document-level state.



Copyright © 2025 Microsoft Corporation



dop2003 (616 bytes): A Dop2003 that specifies document and compatibility settings.

reserved1 (4 bytes): This value is undefined, and MUST be ignored.

A - fRMTrackFormatting (1 bit): Specifies whether to track format changes when tracking for revisions (DopBase.**fRevMarking**). By default, this value is 1.

B - fRMTrackMoves (1 bit): Specifies whether to track moved text when tracking for revisions (DopBase.**fRevMarking**) instead of tracking for the deletions and insertions that are made. By default, this value is 1.

C - reserved2 (1 bit): This value MUST be 0, and MUST be ignored.

D - empty1 (1 bit): This value MUST be 0, and MUST be ignored.

E - empty2 (1 bit): This value MUST be 0, and MUST be ignored.

ssm (4 bits): An unsigned integer that specifies the sorting method to use when displaying document styles. This value MUST be one of the following.

Value	Meaning
0	Styles are sorted by name.
1 (default)	Styles are sorted by the default sorting method of the application.
2	Styles are sorted based on the font that they apply.
3	Styles are sorted by the style on which they are based.
4	Styles are sorted by their style types (character, linked, paragraph, and so on).

F - fReadingModeInkLockDownActualPage (1 bit): Specifies whether to render the document with actual pages or virtual pages as specified in [ECMA-376] Part 1, Section 17.15.1.66 readModeInkLockDown. By default, this value is 0.

G - fAutoCompressPictures (1 bit): Specifies whether pictures in the document are automatically compressed when the document is saved as specified in [ECMA-376] Part 1, Section 17.15.1.33 doNotAutoCompressPictures, where the meaning is the opposite of **fAutoCompressPictures**. By default, this value is 1.

reserved3 (21 bits): This value MUST be 0, and MUST be ignored.

empty3 (4 bytes): This value MUST be 0, and MUST be ignored.

Copyright © 2025 Microsoft Corporation

empty4 (4 bytes): This value MUST be 0, and MUST be ignored.

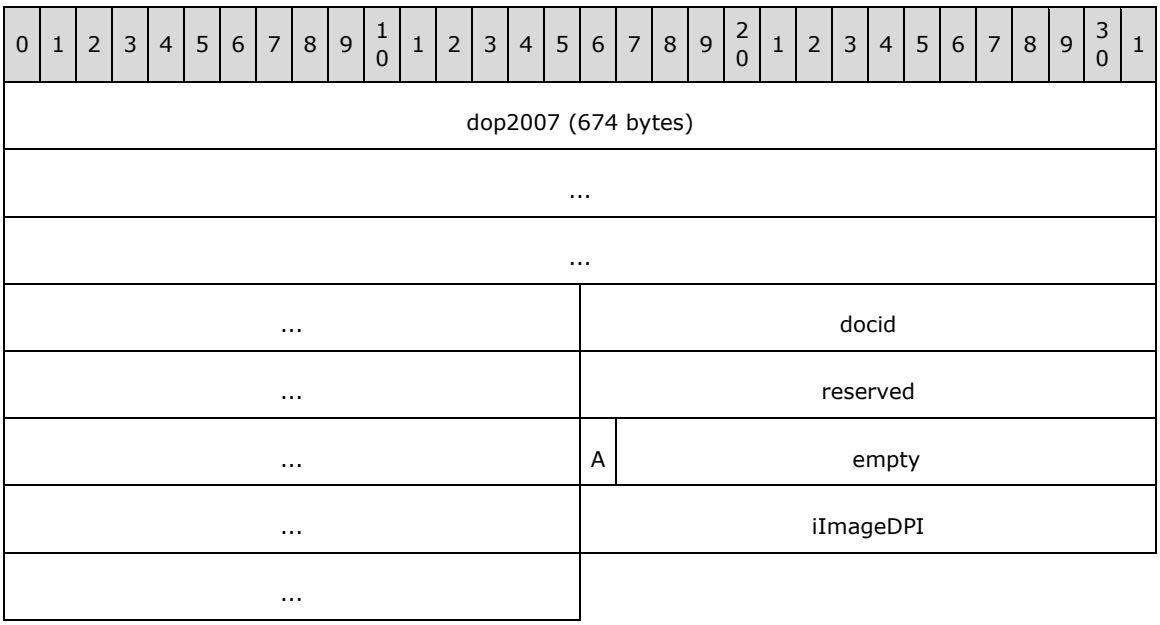
empty5 (4 bytes): This value MUST be 0, and MUST be ignored.

empty6 (4 bytes): This value MUST be 0, and MUST be ignored.

dopMth (34 bytes): A DopMth that specifies various math properties.

2.7.9 Dop2010

The **Dop2010** structure contains document and compatibility settings. These settings influence the appearance and behavior of the current document and store document-level state.



dop2007 (674 bytes): A **Dop2007** structure (section 2.7.8) that specifies document and compatibility settings.

docid (4 bytes): An unsigned integer that specifies an arbitrary identifier for the context of the paragraph identifiers in the document, as specified in [MS-DOCX] section 2.6.1.14 (**docId**). MUST be greater than 0 and less than 0x80000000

reserved (4 bytes): This value is undefined and MUST be ignored.

A - fDiscardImageData (1 bit): Specifies whether the cropped-out areas of images are to be discarded when the document is saved as specified in [MS-DOCX] section 2.6.1.13 (**discardImageEditingData**).

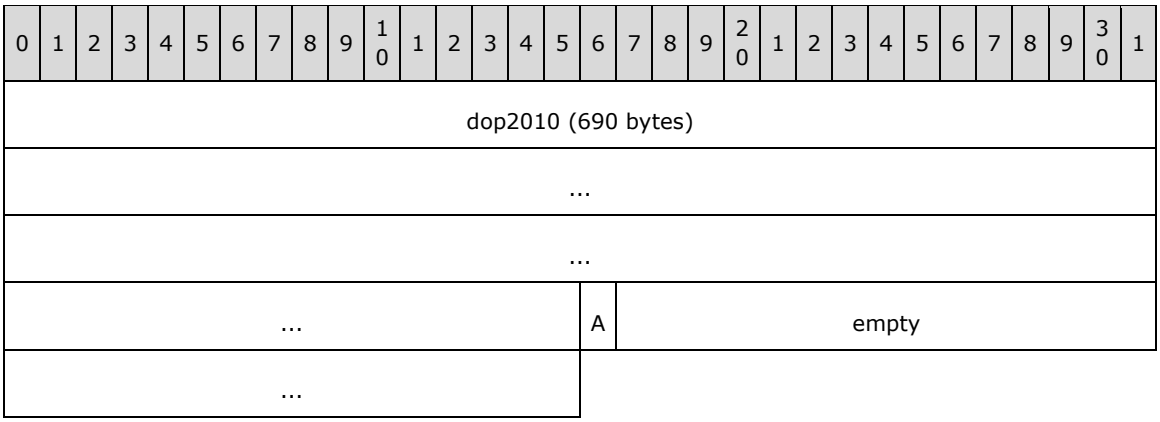
empty (31 bits): This value MUST be 0 and MUST be ignored.

iImageDPI (4 bytes): An unsigned integer that specifies the resolution at which to save images in the document, as specified in [MS-DOCX] section 2.6.1.12 (**defaultImageDpi**).

2.7.10 Dop2013

The **Dop2013** structure contains document and compatibility settings. These settings influence the appearance and behavior of the current document and store document-level state.

Copyright © 2025 Microsoft Corporation



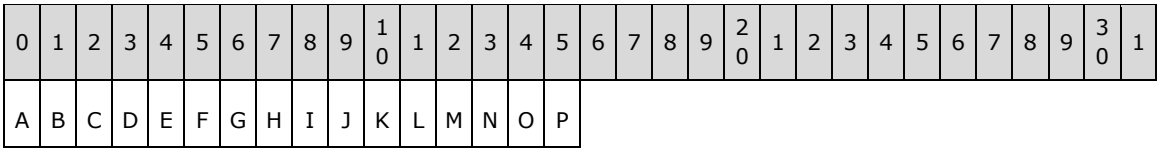
dop2010 (690 bytes): A **Dop2010** structure (section 2.7.9) that specifies document and compatibility settings.

A - fChartTrackingRefBased (1 bit): Specifies how the data point properties and data labels in all charts in this document behave, as specified in [MS-DOCX] section 2.5.1.2 (**chartTrackingRefBased**).

empty (31 bits): This value MUST be 0 and MUST be ignored.

2.7.11 Copts60

The **Copts60** structure specifies compatibility options.



A - fNoTabForInd (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.31 noTabHangInd.

B - fNoSpaceRaiseLower (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.30 noSpaceRaiseLower.

C - fSuppressSpBfAfterPgBrk (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.42 suppressSpBfAfterPgBrk.

D - fWrapTrailSpaces (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.58 wrapTrailSpaces.

E - fMapPrintTextColor (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.33 printColBlack.

F - fNoColumnBalance (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.27 noColumnBalance.

G - fConvMailMergeEsc (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.6 convMailMergeEsc.

H - fSuppressTopSpacing (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.43 suppressTopSpacing.

I - fOrigWordTableRules (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.53 useSingleBorderforContiguousCells.

Copyright © 2025 Microsoft Corporation

J - unused14 (1 bit): This value is undefined and MUST be ignored.

K - fShowBreaksInFrames (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.36 showBreaksInFrames.

L - fSwapBordersFacingPgs (1 bit): Specified in [ECMA-376] Part 4, Section 14.8.3.45 swapBordersFacingPages.

M - fLeaveBackslashAlone (1 bit): Specified in [ECMA-376] Part 1, Section 17.15.3.6 doNotLeaveBackslashAlone, where the meaning of the element is the opposite of **fLeaveBackslashAlone**

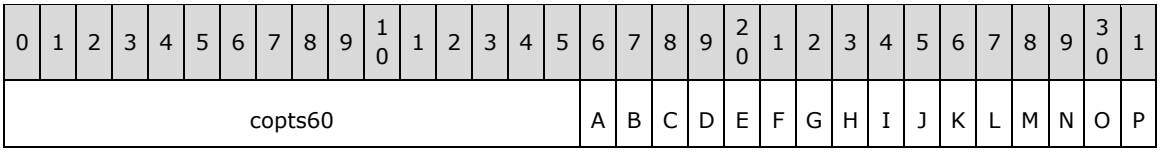
N - fExpShRtn (1 bit): Specified in [ECMA-376] Part 1, Section 17.15.3.5 doNotExpandShiftReturn, where the meaning is the opposite of **fExpShRtn**.

O - fDntULTrlSpc (1 bit): Specified in [ECMA-376] Part 1, Section 17.15.3.8 ulTrailSpace, where the meaning of the element is the opposite of **fDntULTrlSpc**.

P - fDntBlnSbDbWid (1 bit): Specified in [ECMA-376] Part 1, Section 17.15.3.3 balanceSingleByteDoubleByteWidth, where the meaning of the element is the opposite of **fDntBlnSbDbWid**.

2.7.12 Copts80

The **Copts80** structure specifies compatibility options.



copts60 (2 bytes): A Copts60 that specifies additional compatibility options.

A - fSuppressTopSpacingMac5 (1 bit): Specifies whether the minimum line height for the first line on the page is ignored as specified in [ECMA-376] Part 4, Section 14.8.3.41 suppressSpacingAtTopOfPage, where a **spacing** element with a **lineRule** attribute value of **atLeast** refers to sprmPDyaLine with a LSPD.**fMultLinespace** of 0 and **LSPD.dyaline** greater than 0.

B - fTruncDxaExpand (1 bit): Specifies whether text is expanded or condensed by whole points as specified in [ECMA-376] Part 4, Section 14.8.3.37 spacingInWholePoints, where spacing refers to sprmPDyaBefore and sprmPDyaAfter.

C - fPrintBodyBeforeHdr (1 bit): Specifies whether body text is printed before header and footer contents as specified in [ECMA-376] Part 4, Section 14.8.3.32 printBodyTextBeforeHeader.

D - fNoExtLeading (1 bit): Specifies whether leading is not added between lines of text as specified in [ECMA-376] Part 4, Section 14.8.3.29 noLeading.

E - fDontMakeSpaceForUL (1 bit): Specifies whether additional space is not added below the baseline for underlined **East Asian characters** as specified in [ECMA-376] Part 1, Section 17.15.3.7 spaceForUL, where u is sprmCKul and **textAlignment** with **val** of **baseline** is sprmPWAlignFont with a value of 2 and the overall meaning is the opposite of **fDontMakeSpaceForUL**.

F - fMWSmallCaps (1 bit): Specifies whether Word 5.x for the Macintosh small caps formatting is to be used as specified in [ECMA-376] Part 4, Section 14.8.3.26 mwSmallCaps.

Copyright © 2025 Microsoft Corporation

G - f2ptExtLeadingOnly (1 bit): Specifies whether line spacing emulates WordPerfect 5.x line spacing as specified in [ECMA-376] Part 4, Section 14.8.3.44 suppressTopSpacingWP.

H - fTruncFontHeight (1 bit): Specifies whether font height calculation emulates WordPerfect 6.x font height calculation as specified in [ECMA-376] Part 4, Section 14.8.3.46 truncateFontHeightsLikeWP6.

I - fSubOnSize (1 bit): Specifies whether the priority of font size is increased during font substitution as specified in [ECMA-376] Part 4, Section 14.8.3.39 subFontBySize.

J - fLineWrapLikeWord6 (1 bit): Specifies whether line wrapping emulates Microsoft® Word 6.0 line wrapping for East Asian characters as specified in [ECMA-376] Part 4, Section 14.8.3.25 lineWrapLikeWord6.

K - fWW6BorderRules (1 bit): Specifies whether the paragraph borders next to frames are not suppressed as specified in [ECMA-376] Part 4, Section 14.8.3.13 doNotSuppressParagraphBorders.

L - fExactOnTop (1 bit): Specifies whether content on lines with exact line height is not to be centered as specified in [ECMA-376] Part 4, Section 14.8.3.28 noExtraLineSpacing, where exact line height using the **spacing** element refers to sprmPDyaLine with LSPD.**fMultLinespace** of 0 and **LSPD.dyaline** is less than 0.

M - fExtraAfter (1 bit): Specifies whether the exact line height for the last line on a page is ignored as specified in [ECMA-376] Part 4, Section 14.8.3.40 suppressBottomSpacing, where exact line height has using the **spacing** element refers to sprmPDyaLine with LSPD.**fMultLinespace** of 0 and **LSPD.dyaline** is less than 0.

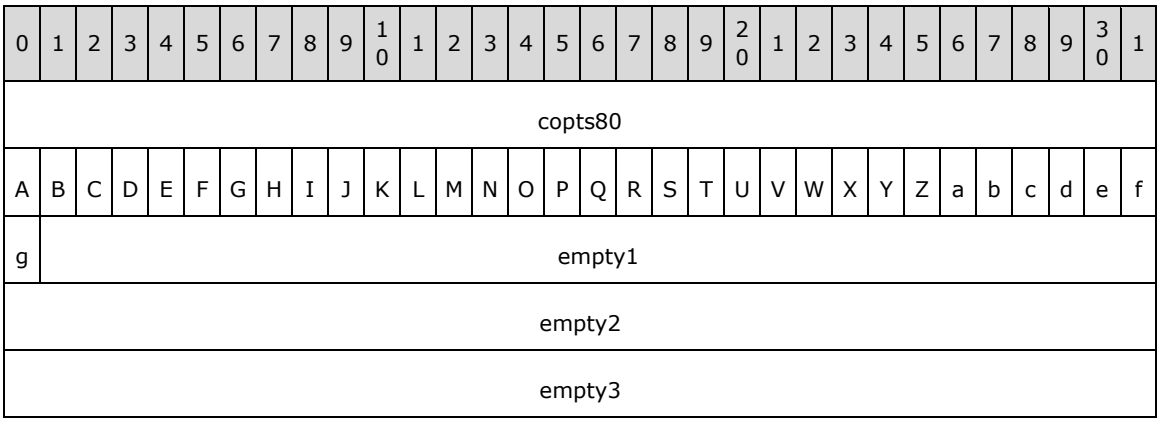
N - fWPSpace (1 bit): Specifies whether the width of a space emulates WordPerfect 5.x space width as specified in [ECMA-376] Part 4, Section 14.8.3.57 wpSpaceWidth.

O - fWPJust (1 bit): Specifies whether paragraph justification emulates WordPerfect 6.x paragraph justification as specified in [ECMA-376] Part 4, Section 14.8.3.56 wpJustification, where the **val** attribute value of **both** on the **jc** element refers to sprmPJc with a value of 3.

P - fPrintMet (1 bit): Specifies whether printer metrics are used to display documents as specified in [ECMA-376] Part 4, Section 14.8.3.52 usePrinterMetrics.

2.7.13 Copts

A structure that specifies compatibility options.



Copyright © 2025 Microsoft Corporation

empty4
empty5
empty6

copts80 (4 bytes): A Copts80 that specifies additional compatibility options.

A - fSpLayoutLikeWW8 (1 bit): Specifies whether to emulate Word 97 text wrapping around floating objects. Specified in [ECMA-376] part 4, 14.8.3.35 (shapeLayoutLikeWW8).

B - fFtnLayoutLikeWW8 (1 bit): Specifies whether to emulate Microsoft® Word 6.0, Word for Windows 95, or Word 97 footnote placement. Specified in [ECMA-376] Part 4, 14.8.3.20 (footnoteLayoutLikeWW8).

C - fDontUseHTMLParagraphAutoSpacing (1 bit): Specifies whether to use fixed paragraph spacing for paragraphs specifying auto spacing. Specified in [ECMA-376] Part 4, 14.8.3.15 (doNotUseHTMLParagraphAutoSpacing).

D - fDontAdjustLineHeightInTable (1 bit): Prevents lines within tables from having their heights adjusted to comply with the document grid. See sprmSDyaLinePitch and [ECMA-376] Part 1, 17.15.3.1 (adjustLineHeightInTable) where the meaning is the opposite of **fDontAdjustLineHeightInTable**.

E - fForgetLastTabAlign (1 bit): Specifies whether to ignore width of the last tab stop when aligning a paragraph if the tab stop is not left aligned. Specified in [ECMA-376] Part 4, 14.8.3.21 (forgetLastTabAlignment) where **jc** refers to sprmPJc and the **tab** element refers to either sprmPChgTabs or sprmPChgTabsPapx.

F - fUseAutospaceForFullWidthAlpha (1 bit): Specifies whether to emulate Word for Windows 95 full-width character spacing. Specified in [ECMA-376] Part 4, 14.8.3.4 (autoSpaceLikeWord for Windows 95).

G - fAlignTablesRowByRow (1 bit): Specifies whether to align table rows independently. Specified in [ECMA-376] Part 4, 14.8.3.1 (alignTablesRowByRow) where the **jc** element refers to sprmTJc or sprmTJc90.

H - fLayoutRawTableWidth (1 bit): Specifies whether to ignore space before tables when deciding if a table wraps a floating object. Specified in [ECMA-376] Part 4, 14.8.3.23 (layoutRawTableWidth).

I - fLayoutTableRowsApart (1 bit): Specifies whether to allow table rows to wrap inline objects independently. Specified in [ECMA-376] Part 4, 14.8.3.24 (layoutTableRowsApart).

J - fUseWord97LineBreakingRules (1 bit): Specifies whether to emulate Word 97 **East Asian line breaking rules**. Specified in [ECMA-376] Part 4, 14.8.3.55 (useWord97LineBreakRules).

K - fDontBreakWrappedTables (1 bit): Specifies whether to prevent floating tables from breaking across pages. Specified in [ECMA-376] Part 4, 14.8.3.10 (doNotBreakWrappedTables) where the **tblpPr** element refers to any of sprmTDxaAbs, sprmTDyaAbs, sprmTPc, sprmTDyaFromTextBottom, sprmTDyaFromText, sprmTDxaFromTextRight, or sprmTDxaFromText with a nondefault value specified.

L - fDontSnapToGridInCell (1 bit): Specifies whether to not snap to the document grid in table cells with objects. Specified in [ECMA-376] Part 4, 14.8.3.11 (doNotSnapToGridInCell) where the **docGrid** element refers to any of sprmSClm, sprmSDyaLinePitch or sprmSDxtCharSpace with a nondefault value specified.

Copyright © 2025 Microsoft Corporation

M - fDontAllowFieldEndSelect (1 bit): Specifies whether to select an entire field when the first or last character of the field is selected. Specified in [ECMA-376] Part 4, 14.8.3.34 (selectFldWithFirstOrLastChar).

N - fApplyBreakingRules (1 bit): Specifies whether to use legacy Ethiopic and Amharic line breaking rules. Specified in [ECMA-376] Part 1, 17.15.3.2 (applyBreakingRules).

O - fDontWrapTextWithPunct (1 bit): Specifies whether to prevent hanging punctuation with the character grid. Specified in [ECMA-376] Part 4, 14.8.3.19 (doNotWrapTextWithPunct) where the **docGrid** element refers to any of sprmSClm, sprmSDyaLinePitch or sprmSDxtCharSpace with a nondefault value specified and the **overflowPunct** element refers to sprmPFOverflowPunct.

P - fDontUseAsianBreakRules (1 bit): Specifies whether to disallow the compressing of compressible characters when using the document grid. Specified in [ECMA-376] Part 4, 14.8.3.14 (doNotUseEastAsianBreakRules) where the **docGrid** element refers to any of sprmSClm, sprmSDyaLinePitch, or sprmSDxtCharSpace with a nondefault value specified

Q - fUseWord2002TableStyleRules (1 bit): Specifies whether to emulate Microsoft Word 2002 table style rules. Specified in [ECMA-376] Part 4, 14.8.3.54 (useWord2002TableStyleRules).

R - fGrowAutoFit (1 bit): Specifies whether to allow tables to autofit into the page margins. Specified in [ECMA-376] Part 4, 14.8.3.22 (growAutofit).

S - fUseNormalStyleForList (1 bit): Specifies whether to not automatically apply the list paragraph style to bulleted or numbered text. Specified in [ECMA-376] Part 4, 14.8.3.51 (useNormalStyleForList). MAY<182> be ignored.

T - fDontUseIndentAsNumberingTabStop (1 bit): Specifies whether to ignore the hanging indent when creating a tab stop after numbering. Specified in [ECMA-376] Part 4, 14.8.3.16 (doNotUseIndentAsNumberingTabStop). MAY<183> be ignored.

U - fFELineBreak11 (1 bit): Specifies whether to use an alternate set of East Asian line breaking rules. Specified in [ECMA-376] Part 4, 14.8.3.48 (useAltKinsokuLineBreakRules). MAY<184> be ignored.

V - fAllowSpaceOfSameStyleInTable (1 bit): Specifies whether to allow contextual spacing of paragraphs in tables. Specified in [ECMA-376] Part 4, 14.8.3.2 (allowSpaceOfSameStyleInTable) where the **contextualSpacing** element refers to sprmPFContextualSpacing. MAY<185> be ignored.

W - fWW11IndentRules (1 bit): Specifies whether to not ignore floating objects when calculating paragraph indentation. Specified in [ECMA-376] Part 4, 14.8.3.12 (doNotSuppressIndentation). MAY<186> be ignored.

X - fDontAutofitConstrainedTables (1 bit): Specifies whether to not autofit tables such that they fit next to wrapped objects. Specified in [ECMA-376] Part 4, 14.8.3.8 (doNotAutofitConstrainedTables). MAY<187> be ignored.

Y - fAutofitLikeWW11 (1 bit): Specifies whether to allow table columns to exceed the preferred widths of the constituent cells. Specified in [ECMA-376] Part 4, 14.8.3.3 (autofitToFirstFixedWidthCell). MAY<188> be ignored.

Z - fUnderlineTabInNumList (1 bit): Specifies whether to underline the tab following numbering when both the numbering and the first character of the numbered paragraph are underlined. Specified in [ECMA-376] Part 4, 14.8.3.47 (underlineTabInNumList). MAY<189> be ignored.

a - fHangulWidthLikeWW11 (1 bit): Specifies whether to use fixed width for Hangul characters. Specified in [ECMA-376] Part 4, 14.8.3.7 (displayHangulFixedWidth). MAY<190> be ignored.

Copyright © 2025 Microsoft Corporation

b - fSplitPgBreakAndParaMark (1 bit): Specifies whether to move paragraph marks to the page after a page break. Specified in [ECMA-376] Part 4, 14.8.3.38 (splitPgBreakAndParaMark). MAY<191> be ignored.

c - fDontVertAlignCellWithSp (1 bit): Specifies whether to not vertically align cells containing floating objects. Specified in [ECMA-376] Part 4, 14.8.3.17 (doNotVertAlignCellWithSp). MAY<192> be ignored.

d - fDontBreakConstrainedForcedTables (1 bit): Specifies whether to not break table rows around floating tables. Specified in [ECMA-376] Part 4, 14.8.3.9 (doNotBreakConstrainedForcedTable) where **cantSplit** element refers to either sprmTFCantSplit or sprmTFCantSplit90 and **tblpPr** element refers to any of sprmTDxaAbs, sprmTDyaAbs, sprmTPc, sprmTDyaFromTextBottom, sprmTDyaFromText, sprmTDxaFromTextRight, or sprmTDxaFromText with a nondefault value specified. MAY<193> be ignored.

e - fDontVertAlignInTxbx (1 bit): Specifies whether to ignore vertical alignment in text boxes. Specified in [ECMA-376] Part 4, 14.8.3.18 (doNotVertAlignInTxbx). MAY<194> be ignored.

f - fWord11KerningPairs (1 bit): Specifies whether to use ANSI kerning pairs from fonts instead of the **Unicode** kerning pair info. Specified in [ECMA-376] Part 4, 14.8.3.49 (useAnsiKerningPairs). MAY<195> be ignored.

g - fCachedColBalance (1 bit): Specifies whether to use cached paragraph information for column balancing. Specified in [ECMA-376] Part 4, 14.8.3.5 (cachedColBalance). MAY<196> be ignored.

empty1 (31 bits): Undefined, and MUST be ignored.

empty2 (4 bytes): Undefined, and MUST be ignored.

empty3 (4 bytes): Undefined, and MUST be ignored.

empty4 (4 bytes): Undefined, and MUST be ignored.

empty5 (4 bytes): Undefined, and MUST be ignored.

empty6 (4 bytes): Undefined, and MUST be ignored.

2.7.14 Asumyi

The **Asumyi** structure specifies **AutoSummary** state information

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>20</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>30</td><td>1</td></tr>
<tr><td>A</td><td>B</td><td colspan="2">C</td><td>D</td><td colspan="11">reserved</td><td colspan="16">wDlgLevel</td></tr>
<tr><td colspan="32">lHighestLevel</td></tr>
<tr><td colspan="32">lCurrentLevel</td></tr>
</table>

A - fValid (1 bit): Specifies whether the rest of the information in the Asumyi is currently valid.

B - fView (1 bit): Specifies whether the AutoSummary view is currently active.

C - iViewBy (2 bits): Specifies the type of AutoSummary to use. This value MUST be one of the following.

Copyright © 2025 Microsoft Corporation

Value	Meaning
0	Highlight the text that is to be included in the summary.
1	Hide all text that is not part of the summary
2	Insert the summary at the top of the document.
3	Create a new document that contains the summary.

D - fUpdateProps (1 bit): Specifies whether to update the document summary information to reflect the AutoSummary results after the next summarization.

reserved (11 bits): This value MUST be 0, and MUST be ignored.

wDlgLevel (2 bytes): Specifies the desired size of the summary. This value SHOULD<197> either be between 0 and 100, expressing the percentage of the original document size, or be one of the following values.

Value	Meaning
0xFFFE	10 sentences.
0xFFFD	20 sentences.
0xFFFC	100 words.
0xFFFB	500 words.
0xFFFA	10 percent of the original document size.
0xFFF9	25 percent of the original document size.
0xFFF8	50 percent of the original document size.
0xFFF7	75 percent of the original document size.

lHighestLevel (4 bytes): If **fValid** is set to 1, this value MUST be greater than or equal to the highest value of ASUMY.**lLevel**.

lCurrentLevel (4 bytes): If **fValid** is set to 1, this value MUST be equal to the following.

$$\frac{wDlgLevel \times lHighestLevel + 50}{100}$$

If **wDlgLevel** is between 0xFFF7 and 0xFFFE, the value to use for **wDlgLevel** is the equivalent percentage to maintain the meaning of **wDlgLevel**. This value is compared to ASUMY.**lLevel** to see if is to be part of the summary. If ASUMY.**lLevel** is less than or equal to **lCurrentLevel**, it is to be part of the summary.

2.7.15 Dogrid

The **Dogrid** structure specifies parameters for the drawn object properties of the document.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
xaGrid																yaGrid															
dxaGrid																dyaGrid															

Copyright © 2025 Microsoft Corporation

dyGridDisplay	A	dxGridDisplay	B

xaGrid (2 bytes): An XAS_nonNeg that specifies horizontal origin point of the drawing grid. See [ECMA-376] Part 1, Section 17.15.1.44 (drawingGridHorizontalOrigin), where **doNotUseMarginsForDrawingGridOrigin** has the opposite meaning of **fFollowMargins**. The default value is 1701.

yaGrid (2 bytes): A YAS_nonNeg that specifies the vertical origin point of the drawing grid. See [ECMA-376] Part 1, Section 17.15.1.46 (drawingGridVerticalOrigin), where **doNotUseMarginsForDrawingGridOrigin** has the opposite meaning of **fFollowMargins**. The default value is 1984.

dxaGrid (2 bytes): An XAS_nonNeg that specifies the horizontal grid unit size of the drawing grid. See [ECMA-376] Part 1, Section 17.15.1.45 (drawingGridHorizontalSpacing). The default value is 180.

dyaGrid (2 bytes): A YAS_nonNeg that specifies the vertical grid unit size of the drawing grid. See [ECMA-376] Part 1, Section 17.15.1.47 (drawingGridVerticalSpacing). The default value is 180.

dyGridDisplay (7 bits): A positive value, in units specified by **dyaGrid**, that specifies the distance between vertical gridlines. See [ECMA-376] Part 1, Section 17.15.1.28 (displayVerticalDrawingGridEvery) where drawingGridVerticalSpacing refers to **dyaGrid**. The default value is 1.

A - unused (1 bit): This value is undefined, and MUST be ignored.

dxGridDisplay (7 bits): A positive value, in units specified by **dxaGrid**, that specifies the distance between horizontal gridlines. See [ECMA-376] Part 1, Section 17.15.1.27. (displayHorizontalDrawingGridEvery) where drawingGridHorizontalSpacing refers to **dxaGrid**. The default value is 1.

B - fFollowMargins (1 bit): A value that specifies whether to use margins for drawing grid origin. See [ECMA-376] Part 1, Section 17.15.1.42 (doNotUseMarginsForDrawingGridOrigin), where the meaning is the opposite of **fFollowMargins**. The default is 1.

2.7.16 DopTypography

The **DopTypography** structure contains **East Asian language** typography settings.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>20</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>30</td><td>1</td></tr>
<tr><td>A</td><td colspan="2">B</td><td colspan="2">C</td><td>D</td><td>E</td><td colspan="3">F</td><td>G</td><td colspan="6">reserved</td><td colspan="15">cchFollowingPunct</td></tr>
<tr><td colspan="16">cchLeadingPunct</td><td colspan="16">rgxchFPunct (202 bytes)</td></tr>
<tr><td colspan="32">...</td></tr>
<tr><td colspan="32">...</td></tr>
<tr><td colspan="32">...</td></tr>
<tr><td colspan="32">rgxchLPunct (102 bytes)</td></tr>
<tr><td colspan="32">...</td></tr>
</table>

Copyright © 2025 Microsoft Corporation

…
…

A - fKerningPunct (1 bit): Specifies whether to kern punctuation characters as specified in [ECMA-376] Part 1, Section 17.15.1.60 noPunctuationKerning, where the meaning of noPunctuationKerning is the opposite of **fKerningPunct**.

B - iJustification (2 bits): Specifies the character-level whitespace compression as specified in [ECMA-376] Part 1, Section 17.15.1.18 characterSpacingControl. This value MUST be one of the following.

Value	Meaning
0 (default)	doNotCompress
1	compressPunctuation
2	compressPunctuationAndJapaneseKana

C - iLevelOfKinsoku (2 bits): This value MAY<198> specify which set of line breaking rules to use for **East Asian characters**. This value MUST be one of the following.

Value	Meaning
0 (default)	Chinese (Simplified) 1. Cannot start a line: !%),.:;>?]}¢¨°·ˇ¯—‖ ‘’‛“”„…‰′″›℃∶ 、。〃〉》」』】〕〗〞 ‿⁀﹏﹐﹑﹒﹔﹕﹖﹘﹚﹜﹞！＂％＇）．：；？］｀｜｝～￠ 2. Cannot end a line:$([{£¥·‘“〈《「『【〔〖〝﹙﹛＄（．［｛￡￥ Chinese (Traditional) 1. Cannot start a line: !),.:;?]}¢·–— ’”•‧…・′‾ 、。〉》」』】〕〞 ∶｜｜｝ ‿⁀‿‿﹏﹐﹑﹒﹔﹕﹖﹘﹚﹜﹞！）．：； ？］｜｝、 2. Cannot end a line: ([{£¥‘“〈《「『【〔〝﹙﹛（｛ Japanese 1. Cannot start a line: !%),.:;?]}¢°’”‰′″℃、。々〉》」』】〕゛゜ヽヾ・ヽヾ！%），．：；？］｝。」、・ﾞﾟ ¢ 2. Cannot end a line: $([\{£¥‘“〈《「『【〔$（［｛「￡¥ Korean 1. Cannot start a line: !%),.:;?]}¢°’”′″℃〉》」』】〕！%），．：；？］｝ ¢ 2. Cannot end a line: $([\{£¥‘“〈《「『【〔$（［｛￡¥₩
1	Identical to 0 for all but Japanese where the following is used: Cannot start a line: !%),.:;?]}¢°’”‰′″℃、。々〉》」』】〕ぁぃぅぇぉっゃゅょゎ゛゜ヽヾァィゥェォッャュョヮヵケ ・ーヽヾ！%），．：；？］｝。」、・ｧｨｩｪｫｯｬｭｮｯｰﾞﾟ ¢ Cannot end a line: $([\{£¥‘“〈《「『【〔$（［｛「￡¥
2	The characters that are forbidden to be used for starting or ending a line are specified by **rgxchFPunct** and **rgxchLPunct**.

D - f2on1 (1 bit): Specifies whether to print two pages per sheet, as specified in [ECMA-376] Part 1, Section 17.15.1.64 printTwoOnOne.

Copyright © 2025 Microsoft Corporation

E - unused (1 bit): This value is undefined and MUST be ignored.

F - iCustomKsu (3 bits): This value specifies for what language the characters in **rgxchFPunct** are **kinsoku** overrides<199>. All other languages act according to the description of **iLevelOfKinsoku** with a value of 0. This MUST be one of the following values.

Value	Language identifier
0 (default)	No language
1	Japanese
2	Chinese (Simplified)
3	Korean
4	Chinese (Traditional)

G - fJapaneseUseLevel2 (1 bit): This value specifies that line breaking rules for Japanese acts according to the description of **iLevelOfKinsoku** with a value of 1<200>. The default value is 0.

reserved (5 bits): This value MUST be 0, and MUST be ignored.

cchFollowingPunct (2 bytes): A signed integer that specifies the number of characters in **rgxchFPunct**. This MUST be a value between 0x0000 and 0x0064 inclusive. By default, this value is 0x0000.

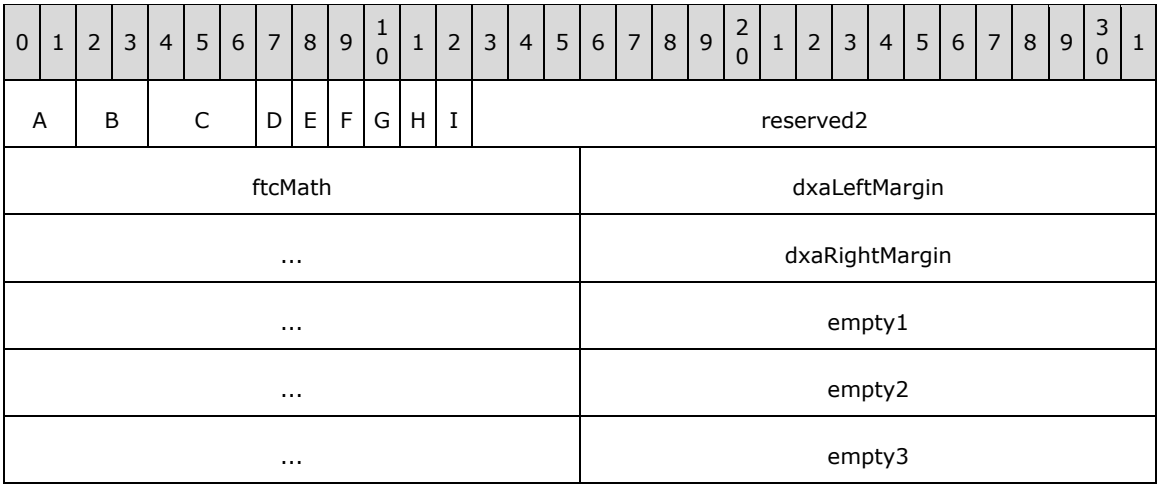
cchLeadingPunct (2 bytes): A signed integer that specifies the number of characters in **rgxchLPunct**. This MUST be a value between 0x0000 and 0x0032, inclusive. By default, this value is 0x0000.

rgxchFPunct (202 bytes): An array of **cchFollowingPunct** **Unicode** characters that cannot start a line if the language of the text matches the language specified in **iCustomKsu**. If **iCustomKsu** has a value of 0, this array has no effect on the document.

rgxchLPunct (102 bytes): An array of **cchLeadingPunct** Unicode characters that cannot end a line if the language of the text matches the language specified in **iCustomKsu**. If **iCustomKsu** has a value of 0, this array has no effect on the document.

2.7.17 DopMth

The **DopMth** structure specifies document-wide math settings.



[MS-DOC] - v20250218
Word (.doc) Binary File Format
Copyright © 2025 Microsoft Corporation
Release: February 18, 2025

...	empty4
...	dxaIndentWrapped
...	

A - mthbrk (2 bits): Specifies how to break on binary operators as specified in [ECMA-376] Part 1, Section 22.1.2.16 brkBin. This MUST be one of the following values.

Value	Meaning
0 (default)	Before. In line wrapping, breaks occur on binary operators, so the binary operator appears before the break.
1	After. In line wrapping, breaks occur on binary operators, so the binary operator appears after the break.
2	Repeat. In line wrapping, breaks occur on binary operators, so the binary operator appears on both sides of the break.

B - mthbrkSub (2 bits): Specifies how to break on binary subtraction when **mthbrk** is 2 as specified in [ECMA-376] Part 1, Section 22.1.2.17 brkBinSub. This value MUST be one of the following.

Value	Meaning
0 (default)	Minus Minus. Repetition of a subtraction sign after a line-wrapping break is minus on the first and second lines.
1	Plus Minus. Repetition of a subtraction sign after a line-wrapping break is plus on the first line and minus on the second line.
2	Minus Plus. Repetition of a subtraction sign after a line-wrapping break is minus on the first line and plus on the second line.

C - mthbpjc (3 bits): Specifies the default justification of math as specified in [ECMA-376] Part 1, Section 22.1.2.25 defJc. This MUST be one of the following values.

Value	Meaning
1 (default)	Centered as Group. Justifies equations with respect to each other and centers the group of equations with respect to the page.
2	Center. Centers each equation individually with respect to margins.
3	Left. Left justification of the paragraph containing only math.
4	Right. Right justification of the paragraph containing only math.

D - reserved1 (1 bit): This value is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

E - fMathSmallFrac (1 bit): Specifies whether to use a reduced fraction size when displaying math that contains fractions as specified in [ECMA-376] Part 1, Section 22.1.2.98 smallFrac. By default, this value is 0.

F - fMathIntLimUndOvr (1 bit): Specifies that the default placement of integral limits when converting from a linear format is directly above and below the base as opposed to on the side of the base as specified in [ECMA-376] Part 1, Section 22.1.2.49 intLim. By default, this value is 0.

G - fMathNaryLimUndOvr (1 bit): Specifies that the default placement of n-ary limits other than integrals is directly above and below the base, as opposed to on the side of the base, as specified in [ECMA-376] Part 1, Section 22.1.2.71 naryLim. By default, this value is 0.

H - fMathWrapAlignLeft (1 bit): Specifies the left justification of the wrapped line of an equation as opposed to right justification of the wrapped line of an equation as specified in [ECMA-376] Part 1, Section 22.1.2.121 wrapRight where the meaning is the opposite of **fMathWrapAlignLeft**. By default, this value is 1.

I - fMathUseDispDefaults (1 bit): Specifies whether to use display math defaults as specified in [ECMA-376] Part 1, Section 22.1.2.30 dispDef. By default, this value is 1.

reserved2 (19 bits): This value MUST be 0, and MUST be ignored.

ftcMath (2 bytes): An index into an **SttbfFfn** structure that specifies the font to use for new equations in the document. The default font is Cambria Math.

dxaLeftMargin (4 bytes): A signed integer, in **twips**, that specifies the left margin for math. MUST be greater than or equal to 0 and less than or equal to 31680 as specified in [ECMA-376] Part 1, Section 22.1.2.59 lMargin. By default, this value is 0.

dxaRightMargin (4 bytes): A signed integer in twips that specifies the right margin for math. This value MUST be greater than or equal to 0 and less than or equal to 31680, as specified in [ECMA-376] Part 1, Section 22.1.2.90 rMargin. By default, this value is 0.

empty1 (4 bytes): This value MUST be 120, and MUST be ignored.

empty2 (4 bytes): This value MUST be 120, and MUST be ignored.

empty3 (4 bytes): This value MUST be 0, and MUST be ignored.

empty4 (4 bytes): This value MUST be 0, and MUST be ignored.

dxaIndentWrapped (4 bytes): A signed integer, in twips, that specifies the indentation of the wrapped line of an equation. This value MUST be greater than or equal to 0 and less than or equal to 31680 as specified in [ECMA-376] Part 1, Section 22.1.2.120 wrapIndent. By default, this value is 1440.

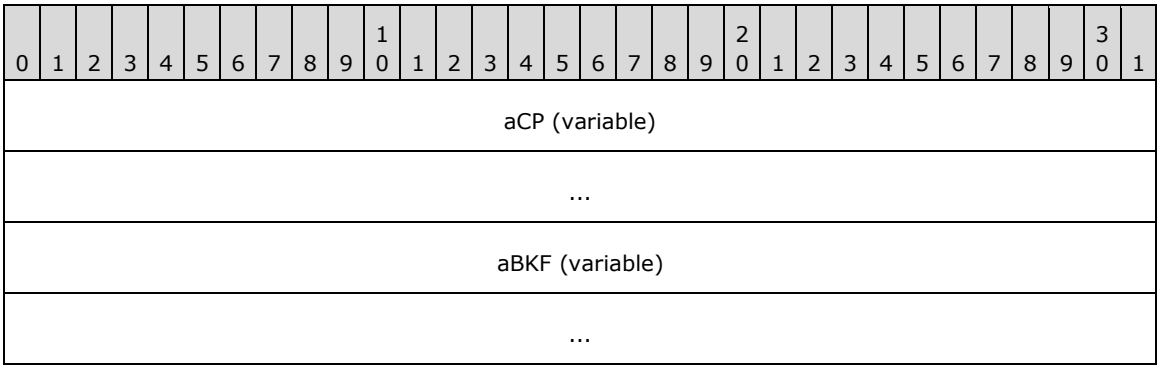
2.8 PLCs

2.8.1 Plcbkf

The **Plcbkf** structure is a PLC whose data elements are BKF structures (6 bytes each). Each CP in the Plcbkf, with the exception of the last CP, represents the character position of the start of a **bookmark** in a Document Part. For every Plcbkf, there is a corresponding Plcbkl. Each data element in the Plcbkf is associated in a one-to-one correlation with a data element in that Plcbkl, whose corresponding CP represents the character position of the end of the same bookmark. Constraints on the CPs inside a Plcbkf as they relate to the CPs in its corresponding Plcbkl can be found in the description of Plcfbkf, which shares the same constraints in relation to its corresponding Plcfbkl.

Copyright © 2025 Microsoft Corporation

The only type of bookmark found in a Plcbkf is a **range-level protection bookmark**. The largest valid value for a CP marking the start or end of a range-level protection bookmark is the CP representing the end of all document parts.



aCP (variable): An array of CPs. Each CP in the array specifies the start of a bookmark in the document.

aBKF (variable): An array of BKFs (6 bytes each), each of which specifies additional information about the bookmark starting at the corresponding CP in **aCP**.

2.8.2 Plcbkfd

The **Plcbkfd** structure is a PLC whose data elements are BKFD structures (10 bytes each). Each CP in the Plcbkfd that is not the last CP represents the character position of the start of a **bookmark** in a Document Part. For every Plcbkfd, there is a corresponding Plcbkld. Each data element in the Plcbkfd is associated in a one-to-one correlation with a data element in the corresponding Plcbkld. The CP corresponding to the data element in the Plcbkld represents the character position of the end of the same bookmark. Constraints upon the CPs inside a Plcbkfd as they relate to the CPs in its corresponding Plcbkld can be found with the description of Plcfbkf, which shares the same constraints in relation to its corresponding Plcfbkl.

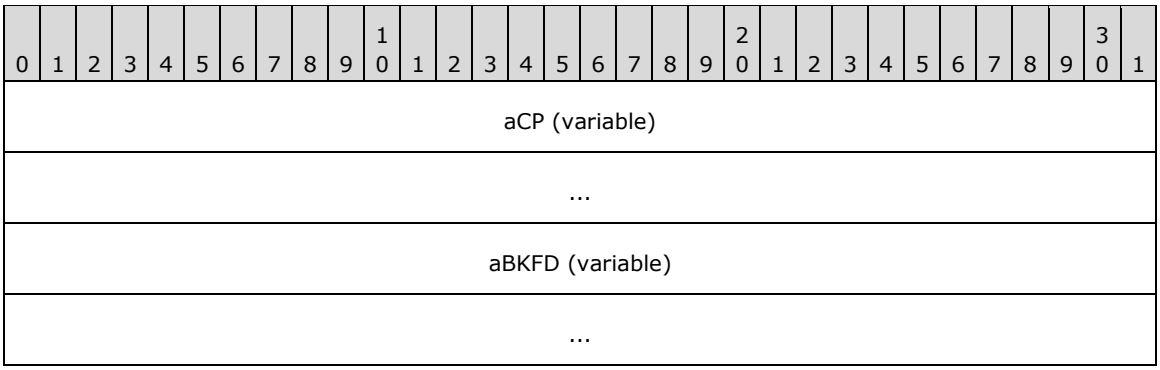
The only type of bookmark found in a Plcbkfd is a **structured document tag bookmark**. When a structured document tag bookmark is created, a character demarcating the start of an arbitrary XML range (see sprmCFSpec) is inserted into the CP stream at the start of the bookmark range. The CP defining the start of a structured document tag bookmark MUST be the offset of that character. As a result, the start CPs of structured document tag bookmarks MUST be unique within their containing PLC.

When a structured document tag bookmark is created, a character demarcating the end of an arbitrary XML range (see sprmCFSpec) is inserted into the CP stream at the end of the bookmark range. The CP defining the limit of a structured document tag bookmark MUST be 1 greater than the CP of that character. As a result, the limit CPs of structured document tag bookmarks MUST be unique within their containing PLC, and the CP specifying the start of a structured document tag bookmark MUST be less than the CP specifying the end of the bookmark by at least 2.

If the range of text spanned by a structured document tag bookmark's CPs contains the CP defining the start or end of another structured document tag bookmark, then it MUST contain the entire range of text spanned by that other bookmark. If the range of text spanned by a structured document tag bookmark's CPs contains content from inside a table and content from outside that table, then it MUST contain the entire table, with possible omission of the table's final **end of cell mark** and TTP mark. In such case, the final end of cell and TTP mark MUST be omitted if and only if the structured document tag bookmark's range does not include text following the table's final TTP mark.

The largest value that a CP marking the start or end of a structured document tag bookmark is allowed to have is the CP representing the end of all document parts.

Copyright © 2025 Microsoft Corporation

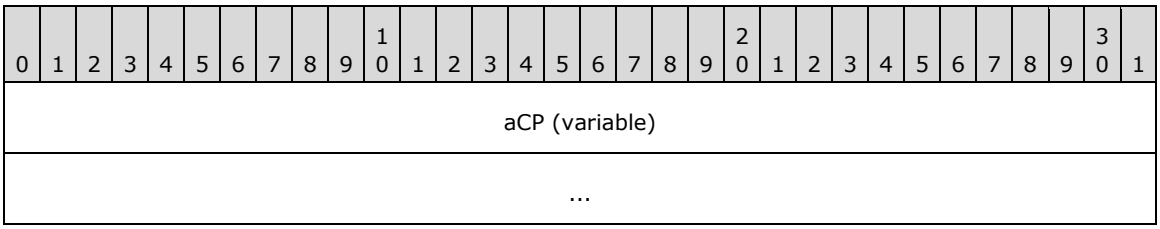


aCP (variable): An array of CPs, each indicating the start of a bookmark in the document.

aBKFD (variable): An array of BKFDs (10 bytes each), each of which specifies additional information about the bookmark starting at the corresponding CP in **aCP**.

2.8.3 Plcbkl

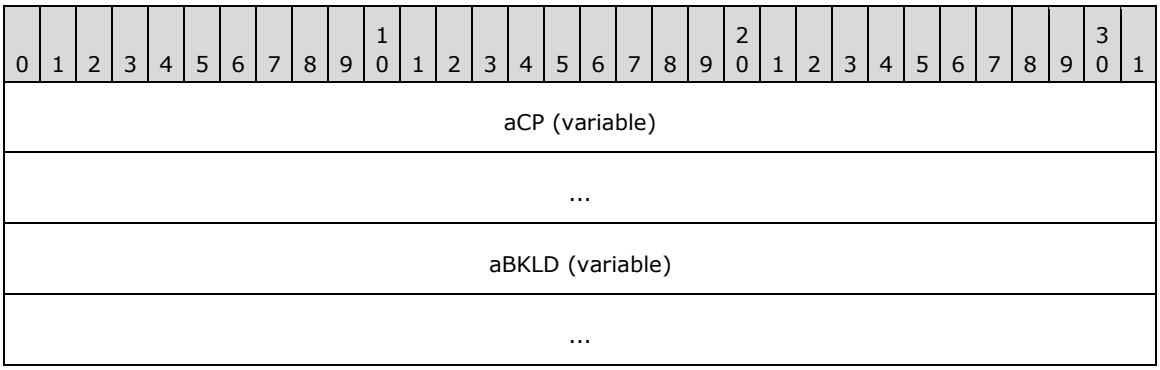
A **Plcbkl** is a PLC that contains only CPs and no additional data. It is thus equivalent to a Plcfbkl. Each CP in the Plcbkl that is not the last CP represents the character position marking the first character beyond the end of a **bookmark** in a Document Part. Additional constraints upon the CPs inside a Plcbkl can be found in the specification of Plcbkf.



aCP (variable): An array of CPs, each indicating the first character beyond the end of a bookmark in the document.

2.8.4 Plcbkld

A **Plcbkld** is a PLC whose data elements are BKLD structures (8 bytes each). Each CP in the Plcbkld, with the exception of the last CP, represents the character position of the first character following the end of a **bookmark** in a Document Part. Additional constraints on the CPs inside a Plcbkld can be found in the description of Plcbkfd.



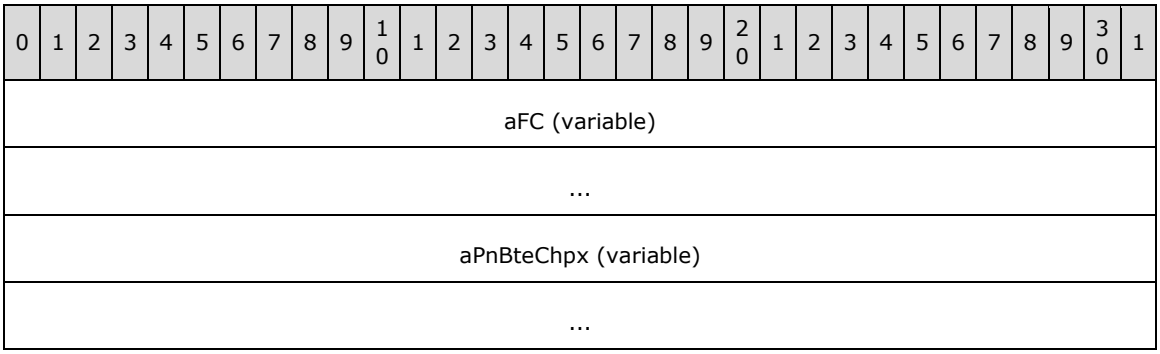
Copyright © 2025 Microsoft Corporation

aCP (variable): An array of CPs. Each CP in the array indicates the first character following the end of a bookmark in the document.

aBKLD (variable): An array of BKLDs (8 bytes each), each of which specifies additional information about the bookmark ending at the corresponding CP in **aCP**.

2.8.5 PlcBteChpx

The **PlcBteChpx** structure is a PLC that maps the offsets of text in the WordDocument stream to the character properties of that text. Where most PLCs map CPs to data, the **PlcBteChpx** maps stream offsets to data instead. A **PlcBteChpx** MUST NOT contain duplicate stream offsets.



aFC (variable): An array of unsigned integers. Each element in this array specifies an offset in the WordDocument stream where text begins. The end of each range is the beginning of the next range. As with all PLCs, the elements of **aFC** MUST be sorted in ascending order.

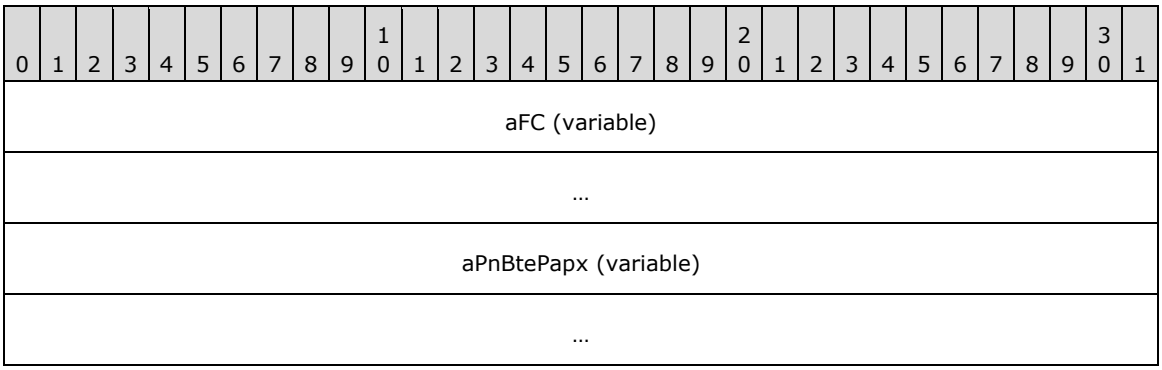
aPnBteChpx (variable): An array of PnFkpChpx (4 bytes each). Each element of this array specifies the location in the WordDocument stream of a ChpxFkp. That ChpxFkp contains the character properties for the text at the corresponding offset in **aFC**.

2.8.6 PlcBtePapx

The **PlcBtePapx** structure is a PLC that specifies paragraph, table row, or table cell properties as described later. Where most PLCs map CPs to data, the **PlcBtePapx** maps stream offsets to data instead. The offsets in **aFC** partition a portion of the WordDocument stream into adjacent ranges.

Consider the collection of paragraphs, table rows, and table cells whose last character occurs at an offset in the WordDocument stream larger than or equal to **aFC**[*i*] but smaller than **aFC**[*i*+1]. Then, **aPnBtePapx**[*i*] specifies the properties of these paragraphs, table rows, or table cells.

A **PlcBtePapx** MUST NOT contain duplicate stream offsets. Each data element of **PlcBtePapx** is 4 bytes long.



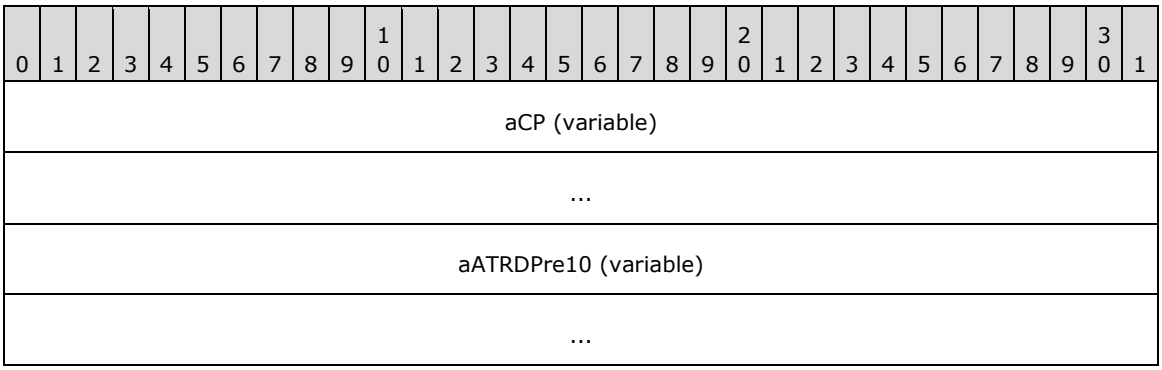
Copyright © 2025 Microsoft Corporation

aFC (variable): An array of unsigned integers. Each element in this array specifies an offset in the WordDocument stream. The elements of **aFC** MUST be sorted in ascending order, and there MUST NOT be any duplicate entries.

aPnBtePapx (variable): An array of PnFkpPapx. The *i*th entry in **aPnBtePapx** is a PnFkpPapx that specifies the properties of all paragraphs, table rows, and table cells whose last character occurs at an offset in the WordDocument stream larger than or equal to **aFC**[*i*] but smaller than **aFC**[*i*+1]; **aPnBtePapx** MUST contain one less entry than **aFC**.

2.8.7 PlcfandRef

The **PlcfandRef** structure is a PLC whose data elements are ATRDPre10 structures (30 bytes each).

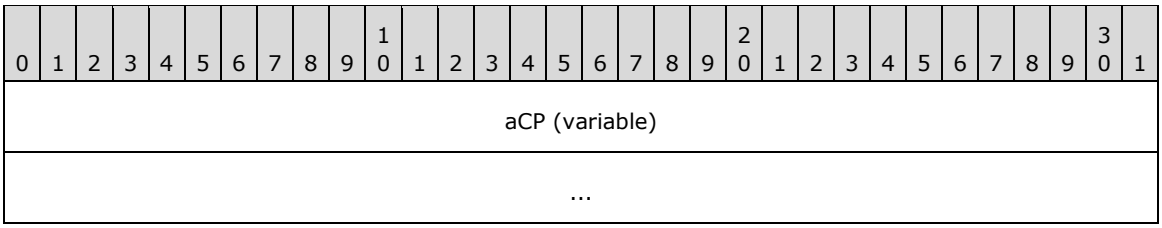


aCP (variable): An array of CPs, all but the last of which specify the location of comment references in the main document. All but the last CP MUST be greater than or equal to zero and less than **FibRgLw97.ccpText**. Each position in the main document specified by one of these CPs MUST be character 0x05 and have sprmCFSpec applied with a value of 1. The last CP MUST be ignored. A **PlcfandRef** MUST NOT contain duplicate CPs.

aATRDPre10 (variable): An array of ATRDPre10 structures (30 bytes each) that associate data with a comment located at the corresponding CP. Each ATRDPre10 structure contains the initials of the user who made the comment, an index into a string table of authors, and a **bookmark** index. See ATRDPre10 and ATRDPost10 for more information about data associated with comments.

2.8.8 PlcfandTxt

The **PlcfandTxt** structure is a PLC that contains only CPs and no additional data. This means that the size of the data is 0 bytes.



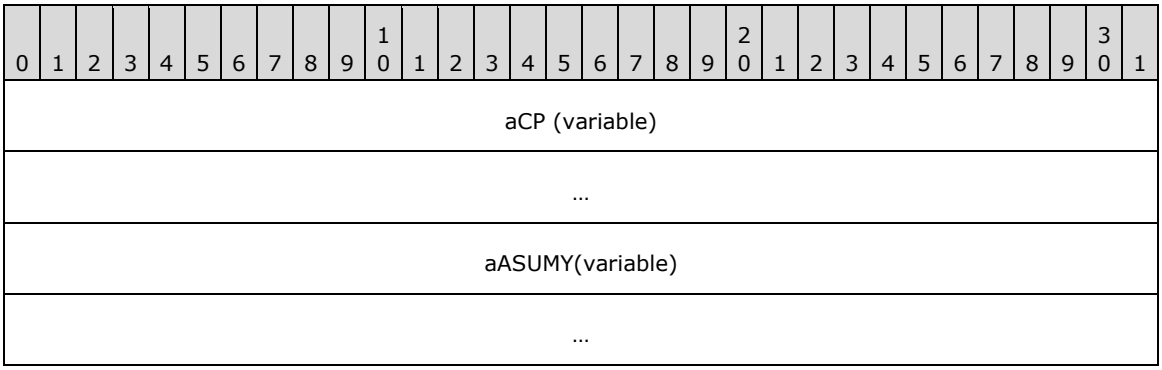
aCP (variable): An array of CPs that specifies positions in the comment document. Each CP except the last two specifies the beginning of a range of text to appear in a comment indicated by the corresponding PlcfandRef CPs. The range of text MUST begin with character 0x0005 with sprmCFSpec applied with a value of 1, and MUST end with a paragraph mark (**Unicode** 0x000D) at table depth zero immediately before the next CP. Each range MUST be a valid selection. Except for the last CPs, each CP MUST be greater than or equal to zero and less than **FibRgLw97.ccpAtn**. The second-to-last CP only ends the last text range and MUST be equal to

Copyright © 2025 Microsoft Corporation

FibRgLw97.ccpAtn decremented by 1. The last CP is undefined and MUST be ignored. A PlcfandTxt MUST NOT contain duplicate CPs.

2.8.9 PlcfAsumy

The **PlcfAsumy** structure is a PLC whose data elements are ASUMY (4 bytes each).



aCP (variable): An array of CPs. CPs are positions in the set of all document parts. CPs are relative to the start of the main document, but can extend into any of the document parts.

Each CP specifies the beginning of a range of text to which the corresponding ASUMY structure applies. The range of text ends immediately prior to the next CP. A PlcfAsumy MUST NOT contain duplicate CPs.

The last CP does not begin a new text range; it only terminates the previous one.

aASUMY (variable): An array of ASUMY that indicates the priority of the corresponding text range for purposes of **AutoSummary**.

2.8.10 Plcfbkf

A **Plcfbkf** is a PLC whose data elements are FBKF structures (4 bytes each). Each CP in the Plcfbkf that is not the last CP represents the character position of the start of a **bookmark** in a document part. For every Plcfbkf, there is a corresponding Plcfbkl. Each data element in the Plcfbkf is associated in a one-to-one correlation with a data element in that Plcfbkl, whose corresponding CP represents the character position of the end of the same bookmark.

The following constraints apply to CPs in all bookmark PLCs.

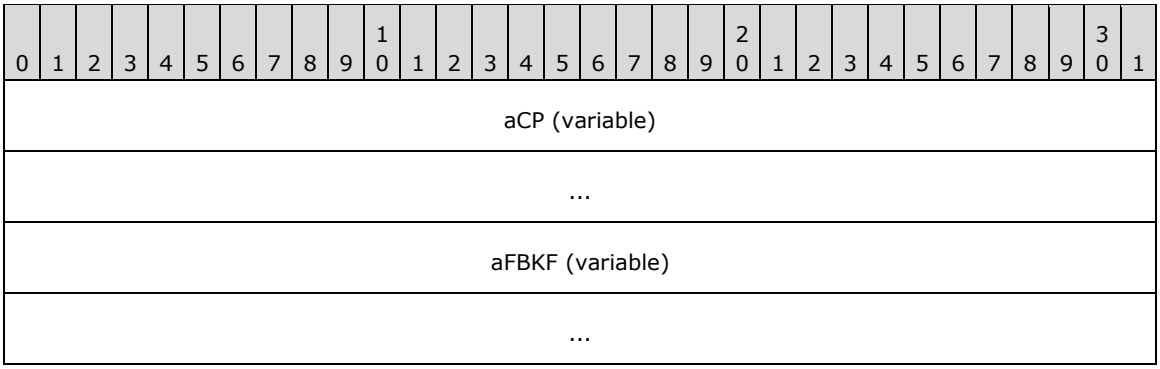
The last CP in a bookmark PLC MUST have a value that is one greater than the largest CP that a bookmark of the type associated with the PLC is allowed to have and MUST be ignored. Unless otherwise specified by a particular type of bookmark, bookmark PLCs can contain duplicate CPs because bookmarks can overlap. The CP defining the start of a bookmark MUST be less than or equal in value to the CP defining the limit of the bookmark. The range of text spanned by a bookmark's (1) CPs MUST obey all constraints, excluding those concerning tables, upon valid selections defined in section 2.2.3. The following constraints reference entities defined in section 2.4.3 Overview of Tables. For bookmark types whose **BKC.fCol** MUST be 0, the following rule 1 MUST apply. Otherwise, the following rule 2 MUST apply:

1. If the range of text spanned by a bookmark's (1) CPs contains a table cell mark, then its start CP MUST be less than or equal to the CP of the beginning of the cell in question and its limit CP MUST either be one less than the CP of a cell mark in that table, one greater than the CP of a TTP mark in that table, or outside the table. If the range of text spanned by a bookmark's (1) CPs contains a TTP mark in a table, then its start CP MUST be outside the table, or the first character of a row in the table. If the range of text spanned by a bookmark's (1) CPs contains a TTP mark in a table,

Copyright © 2025 Microsoft Corporation

then its limit CP MUST be outside the table, or two less than the CP of a TTP mark in the table, or one greater than the CP of a TTP mark in the table.

2. If the range of text spanned by a bookmark's (1) CPs contains content from a cell in a table and content from outside that table, then it MUST contain only whole rows of the table containing that cell. If the range of text spanned by a bookmark's (1) CPs contains a table cell mark or TTP mark, then it MUST NOT span partial rows of the table containing that cell or TTP.



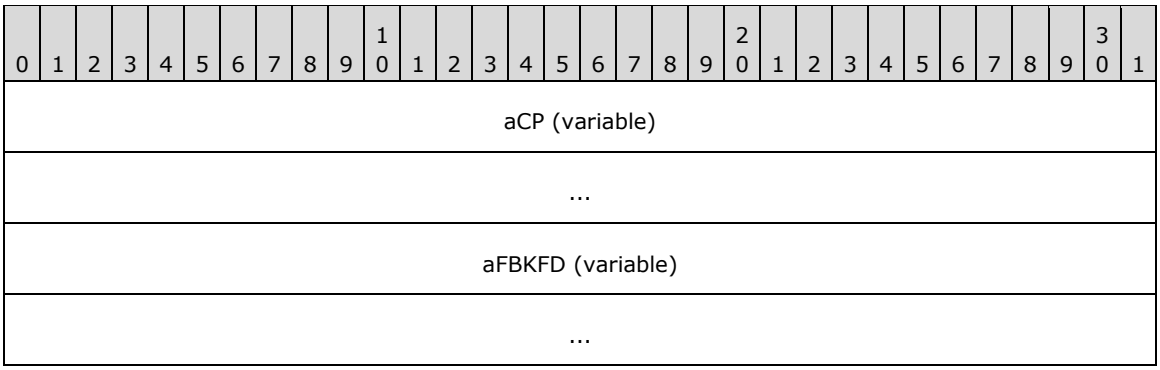
aCP (variable): An array of CPs, each indicating the start of a bookmark (1) in the document.

aFBKF (variable): An array of FBKFs (4 bytes each), each of which specifies additional information about the bookmark starting at the corresponding CP in **aCP**.

2.8.11 Plcfbkfd

The **Plcfbkfd** structure is a PLC whose data elements are FBKFD structures (6 bytes each). Each CP in the Plcfbkfd, with the exception of the last CP, represents the character position of the start of a **bookmark** in a document part. For every Plcfbkfd, there is a corresponding Plcfbkld. Each data element in the Plcfbkfd is associated in a one-to-one correlation with a data element in that Plcfbkld, whose corresponding CP represents the character position of the end of the same bookmark. Constraints on the CPs inside a Plcfbkfd as they relate to the CPs in its corresponding Plcfbkld can be found in the description of Plcfbkf, which shares the same constraints in relation to its corresponding Plcfbkl.

The only types of bookmark found in a Plcfbkfd are **format consistency-checker bookmarks** and **smart tag bookmarks**. The largest value that a CP marking the start or end of a format consistency-checker bookmark or a smart tag bookmark is allowed to have is the CP representing the end of all document parts.



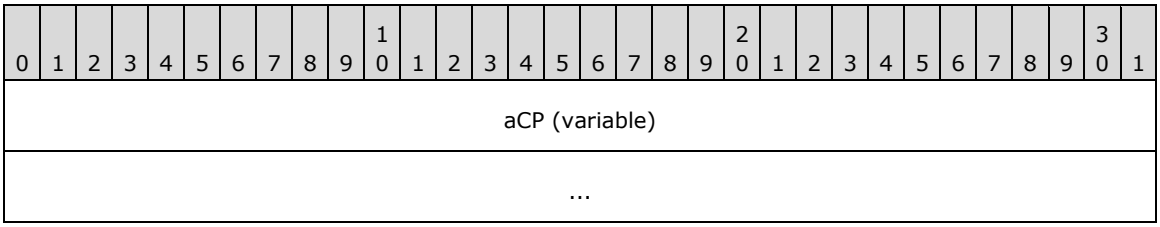
aCP (variable): An array of CPs. Each CP in the array indicates the start of a bookmark in the document.

Copyright © 2025 Microsoft Corporation

aFBKFD (variable): An array of FBKFDs (6 bytes each), each of which specifies additional information about the bookmark starting at the corresponding CP in **aCP**.

2.8.12 Plcfbkl

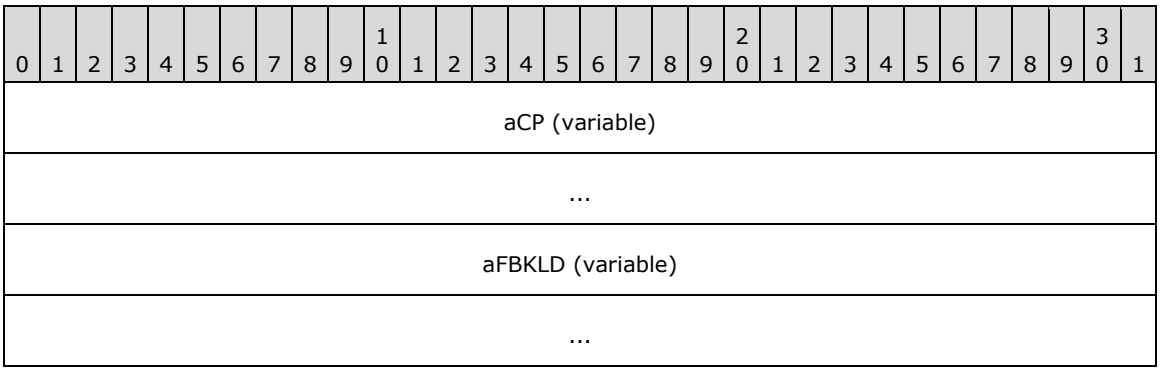
The **Plcfbkl** structure is a PLC that contains only CPs and no additional data. Thus, a **Plcfbkl** is equivalent to a Plcbkl. Each CP in the Plcfbkl, with the exception of the last CP, represents the character position marking the first character following the end of a **bookmark** in a document part. Further constraints on the CPs inside a Plcfbkl can be found in the description of Plcfbkf.



aCP (variable): An array of CPs. Each CP in the array indicates the first character following the end of a bookmark in the document.

2.8.13 Plcfbkld

The **Plcfbkld** structure is a PLC whose data elements are FBKLD structures (4 bytes each). Each CP in the Plcfbkld that is not the last CP represents the character position of the first character following the end of a **bookmark** in a document part. Further constraints on the CPs inside a Plcfbkld can be found in the description of Plcfbkfd.

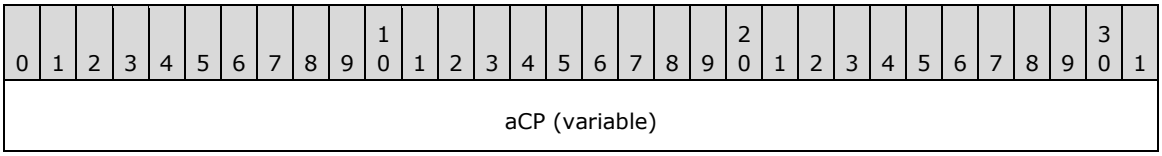


aCP (variable): An array of CPs. Each CP in the array indicates the first character following the end of a bookmark in the document.

aFBKLD (variable): An array of FBKLDs (4 bytes each), each of which specifies additional information about the bookmark ending at the corresponding CP in **aCP**.

2.8.14 Plcfcookie

The **Plcfcookie** structure is a PLC whose data elements are FCKS structures (10 bytes).



Copyright © 2025 Microsoft Corporation

…
aFCKS (variable)
…

aCP (variable): An array of CPs specifying the starting points of text ranges associated with **grammar checker cookie** data. The last CP in the array MUST be ignored. CPs are positions in the set of all document parts. CPs are relative to the start of the main document, but can extend into any of the document parts. A **PlcfcookieOld** MAY contain duplicate CP values if the corresponding **grammar checker** chose to store more than one grammar checker cookie at the same CP.

aFCKS (variable): An array of FCKS structures (10 bytes each). Each **FCKS** specifies information about a grammar checker cookie which applies to text starting at the corresponding CP value.

2.8.15 PlcfcookieOld

The **PlcfcookieOld** structure is a **PLC** whose data elements are **FCKSOLD** structures (16 bytes).

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
aCP (variable)																															
…																															
aFCKSOLD (variable)																															
…																															

aCP (variable): An array of CPs specifying the starting points of text ranges associated with **grammar checker cookie** data. The last CP in the array MUST be ignored. CPs are positions in the set of all document parts. CPs are relative to the start of the main document, but can extend into any of the document parts. A **PlcfcookieOld** MAY contain duplicate CP values if the corresponding **grammar checker** chose to store more than one grammar checker cookie at the same CP.

aFCKSOLD (variable): An array of FCKSOLD structures (16 bytes each). Each FCKSOLD specifies information about a grammar checker cookie which applies to text starting at the corresponding CP value.

2.8.16 PlcfendRef

The **PlcfendRef** is a PLC whose data elements are integers of 2 bytes each.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
aCP (variable)																															
…																															

Copyright © 2025 Microsoft Corporation

aEndIdx (variable)
...

aCP (variable): An array of CPs, all but the last of which specify the location of endnote references in the main document. All but the last CP MUST be greater than or equal to zero and less than **FibRgLw97.ccpText**. The last CP MUST be ignored. A **PlcfendRef** MUST NOT contain duplicate CPs.

aEndIdx (variable): An array of 2-byte integers that specifies whether each endnote is automatically numbered or uses a custom symbol. If equal to zero, the endnote reference uses a custom symbol; otherwise, it is automatically numbered. If the endnote reference is automatically numbered, the character in the main document at the position specified by the corresponding CP MUST equal 0x02 and have sprmCFSpec applied with a value of 1. See sprmCSymbol for more information about custom symbols and sprmSRncEdn, sprmSNEdn, and sprmSNfcEdnRef for more information about automatically numbered endnotes.

2.8.17 PlcfendTxt

The **PlcfendTxt** structure is a **PLC** that contains only CPs and no additional data. The data thus has a size of zero bytes.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
aCP (variable)																															
...																															

aCP (variable): An array of CPs that specifies offsets into the endnote document. Each CP except the last two specifies the beginning of a range of text to appear in an endnote. The range of text MUST end in character 0x0D immediately before the next CP. Except for the last CP, each CP MUST be greater than or equal to zero and less than **FibRgLw97.ccpEdn**. The second-to-last CP only ends the last text range and MUST be equal to **FibRgLw97.ccpEdn** – 1. The last CP is undefined and MUST be ignored. A **PlcfendTxt** MUST NOT contain duplicate CPs.

2.8.18 Plcffactoid

The **Plcffactoid** structure is a **PLC** structure where the data elements are FactoidSpls structures of 2 bytes each.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
aCP (variable)																															
...																															
aFactoidSpls (variable)																															
...																															

Copyright © 2025 Microsoft Corporation

aCP (variable): An array of CPs. CPs are positions in the set of all document parts. CPs are relative to the start of the main document, but can extend into any of the document parts.

Each CP specifies the beginning of a range of text where the state in the corresponding FactoidSpls structure applies. The range of text ends immediately prior to the next CP.

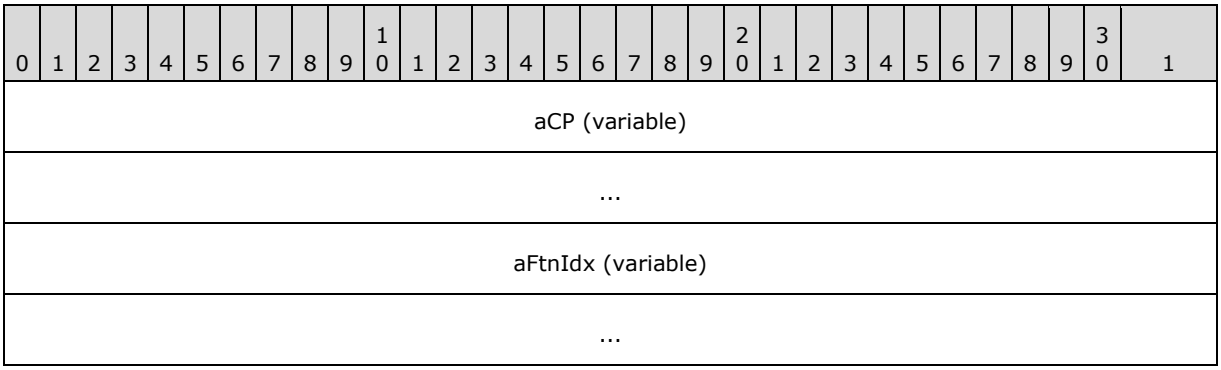
A **Plcffactoid** can contain duplicate CPs. Duplicate CPs specify an **insertion point** or a **deletion point** at that CP and the corresponding FactoidSpls state applies to that point.

The last CP does not begin a new text range; it only terminates the previous one.

aFactoidSpls (variable): An array of 2-byte FactoidSpls structures. Each FactoidSpls structure contains the state of the **smart tag recognizer** for the corresponding text range.

2.8.19 PlcffndRef

The **PlcffndRef** structure is a PLC whose data elements are integers of 2 bytes each.

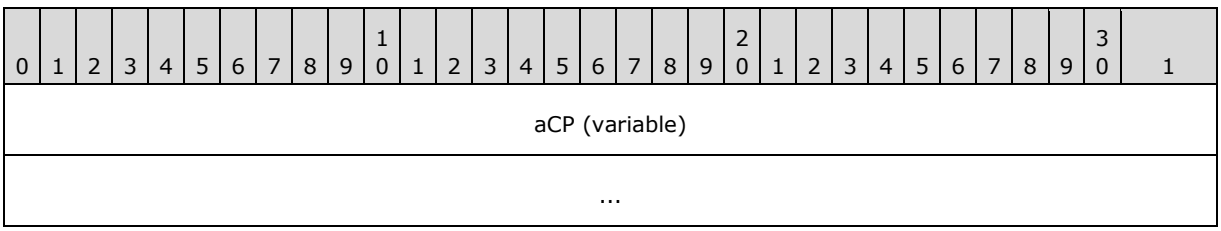


aCP (variable): An array of CPs, all but the last of which specify the location of footnote references in the main document. All but the last CP MUST be greater than or equal to zero and less than **FibRgLw97.ccpText**. The last CP MUST be ignored. A **PlcffndRef** MUST NOT contain duplicate CPs.

aFtnIdx (variable): An array of 2-byte integers that specifies whether each footnote is automatically numbered or uses a custom symbol. If equal to zero, the footnote reference uses a custom symbol; otherwise, it is automatically numbered. If the footnote reference is automatically numbered, the character in the main document at the position specified by the corresponding CP MUST equal 0x02 and have sprmCFSpec applied with a value of 1. See sprmCSymbol for more information about custom symbols and sprmSRncFtn, sprmSNFtn, and sprmSNfcFtnRef for more information about automatically numbered footnotes.

2.8.20 PlcffndTxt

The **PlcffndTxt** structure is a PLC that contains only CPs and no additional data. The data thus has a size of 0 bytes.

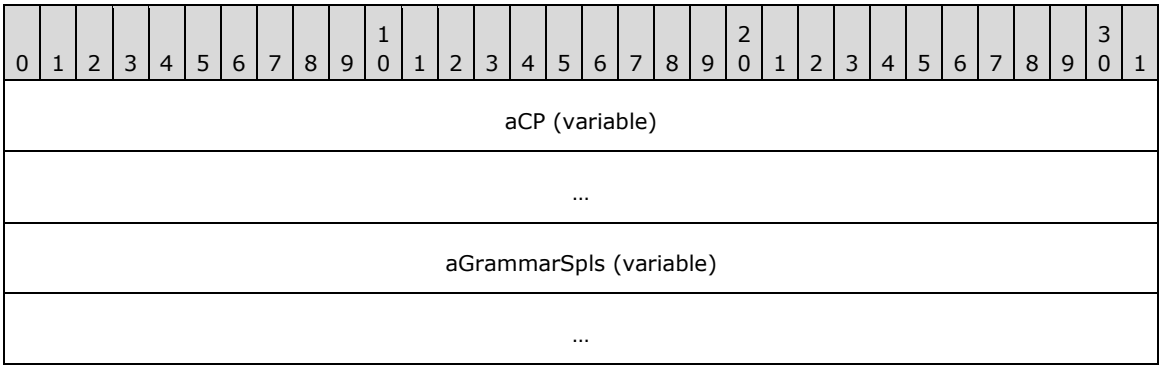


Copyright © 2025 Microsoft Corporation

aCP (variable): An array of CPs that specifies offsets into the footnote document. Each CP except the last two specifies the beginning of a range of text to appear in a footnote. The range of text MUST end in character 0x0D immediately before the next CP. Except for the last CP, each CP MUST be greater than or equal to zero and less than **FibRgLw97.ccpFtn**. The second-to-last CP only ends the last text range and MUST be equal to **FibRgLw97.ccpFtn** – 1. The last CP is undefined and MUST be ignored. A **PlcffndTxt** MUST NOT contain duplicate CPs.

2.8.21 Plcfgram

The **Plcfgram** structure is a PLC structure where the data elements are GrammarSpls structures (2 bytes each).



aCP (variable): An array of CPs. CPs are positions in the set of all document parts. CPs are relative to the start of the main document but can extend into any of the document parts.

Each CP specifies the beginning of a range of text where the state in the corresponding GrammarSpls structure applies. The range of text ends immediately prior to the next CP.

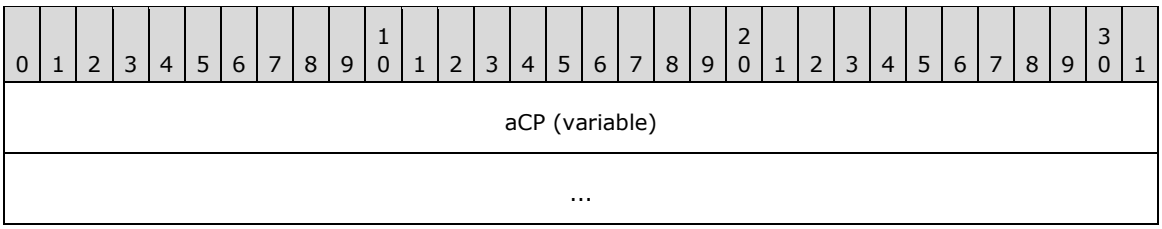
A **Plcfgram** can contain duplicate CPs. Duplicate CPs specify an **insertion point** or a **deletion point** at that CP and the corresponding GrammarSpls state applies to that point.

The last CP does not begin a new text range; it only terminates the previous one.

aGrammarSpls (variable): An array of 2-byte GrammarSpls structures. Each GrammarSpls structure contains the state of the grammar checker for the corresponding text range.

2.8.22 Plcfhdd

The **Plcfhdd** structure is a PLC that contains only CPs and no additional data. It specifies where header document stories begin and end.



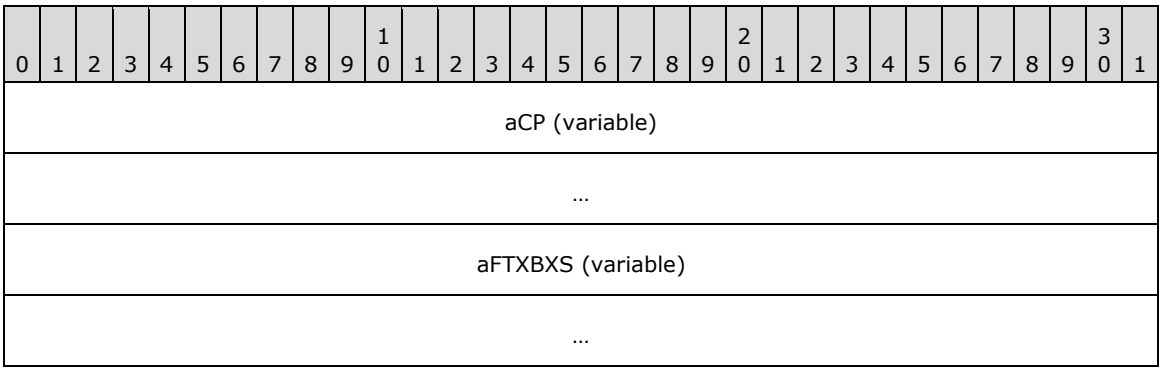
aCP (variable): An array of CPs. Each CP except the last two specifies the beginning of a story in the header document. Each story ends immediately prior to the next CP. If the next CP in **Plcfhdd** has the same value as a CP specifying the beginning of a story, then the story is considered empty.

Copyright © 2025 Microsoft Corporation

Except for the last CP, each CP of **Plcfhdd** MUST be greater than or equal to 0 and less than FibRgLw97.**ccpHdd**. The second-to-last CP only ends the last story and MUST be equal to FibRgLw97.**ccpHdd** minus 1. The last CP is undefined and MUST be ignored.

2.8.23 PlcfHdrtxbxTxt

The **PlcfHdrtxbxTxt** structure is a PLC structure in which the data elements are FTXBXS structures (22 bytes each).



aCP (variable): An array of CPs. CPs are positions in the header textboxes document.

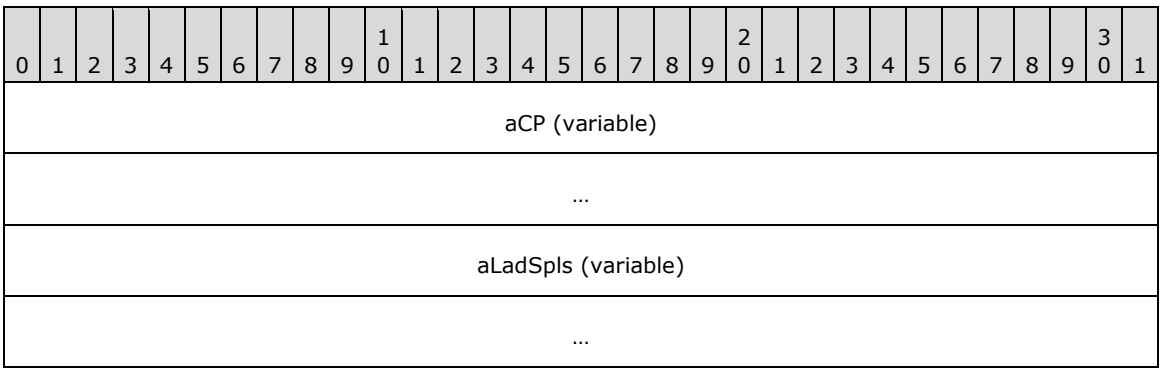
Each CP specifies the beginning of a range of text to appear in a text box indicated by the corresponding FTXBXS structure. The range of text ends immediately prior to the next CP. The last CP does not begin a new text range; it only terminates the previous one.

A **PlcfHdrtxbxTxt** MUST NOT contain duplicate CPs. The text ranges for each FTXBXS structure are separated by 0x0D characters that MUST be the last character in each range. The last text range is an exception. The text in the last range is ignored, and the 0x0D character is not required.

aFTXBXS (variable): An array of FTXBXS (22 bytes each) structures that associate the text ranges with shape objects.

2.8.24 Plcflad

The **Plcflad** structure is a **PLC** structure where the data elements are **LadSpls** structures (2 bytes each).



aCP (variable): An array of CPs. CPs are positions in the set of all document parts. CPs are relative to the start of the main document, but can extend into any of the document parts.

Copyright © 2025 Microsoft Corporation

Each CP specifies the beginning of a range of text where the state in the corresponding LadSpls structure applies. The range of text ends immediately prior to the next CP.

A **Plcflad** can contain duplicate CPs. Duplicate CPs specify an **insertion point** or a **deletion point** at that CP and the corresponding LadSpls state applies to that point.

The last CP does not begin a new text range; it only terminates the previous one.

aLadSpls (variable): An array of 2-byte LadSpls structures. Each LadSpls structure contains the state of **language auto-detection** for the corresponding text range.

2.8.25 Plcfld

The **Plcfld** structure is a PLC whose data elements are Flds (2 bytes each). It specifies the location of **field**s in the document.

A field consists of two parts: field instructions and, optionally, a result. All fields MUST begin with **Unicode** character 0x0013 with sprmCFSpec applied with a value of 1. This is the *field begin character*. All fields MUST end with a Unicode character 0x0015 with sprmCFSpec applied with a value of 1. This is the *field end character*. If the field has a result, then there MUST be a Unicode character 0x0014 with sprmCFSpec applied with a value of 1 somewhere between the field begin character and the field end character. This is the *field separator*. The *field result* is the content between the field separator and the field end character. The *field instructions* are the content between the field begin character and the field separator, if one is present, or between the field begin character and the field end character if no separator is present. The field begin character, field end character, and field separator are collectively referred to as *field characters*.

The field instructions and field result MUST each be a valid selection.

The CPs of a **Plcfld** specify the location of the field characters. A **Plcfld** MUST NOT contain duplicate CPs. Each document part has its own **Plcfld**, with CPs relative to the start of that document part.

The last CP in **aCP** does not specify the location of a field character. Because a **Plcfld** is a PLC, **aCP** MUST be sorted. Because **aCP** MUST NOT contain duplicate CPs, the last CP MUST be the largest in **aCP**. Other than those constraints, the last CP in **aCP** is undefined and MUST be ignored.

The Flds MUST be arranged such that the sequence of Fld.fldch.ch is a valid **FieldList** according to the following **Augmented Backus-Naur Form (ABNF)** rulelist. ABNF is specified in [RFC4234].

Begin	**=**	**0x13**
Sep	=	0x14
End	=	0x15
Field	=	<Begin> *<Field> [Sep] *<Field> <End>
FieldList	=	*<Field>

Additionally, the field characters of the following five field types MUST NOT appear in **aFld**.

1. XE, as specified in [ECMA-376] Part 1, Section 17.16.5.72
2. TC, as specified in [ECMA-376] Part 1, Section 17.16.5.63
3. RD, as specified in [ECMA-376] Part 1, Section 17.16.5.50
4. TA, as specified in [ECMA-376] Part 1, Section 17.16.5.62
5. PRIVATE, as specified in [ECMA-376] Part 1, Section 17.16.5.48

Copyright © 2025 Microsoft Corporation

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
aCP (variable)																															
...																															
aFld (variable)																															
...																															

aCP (variable): An array of CPs. Specifies the positions of field characters in the document.

aFld (variable): An array of **Fld**. Specifies properties for the field character at the corresponding CP. Fldch.ch of each **Fld** MUST be equal to the character at the corresponding CP.

2.8.26 PlcfSed

The **PlcfSed** structure is a **PLC** structure where the data elements are **Sed** structures (12 bytes each).

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
aCP (variable)																															
...																															
aSed (variable)																															
...																															

aCP (variable): An array of CPs. Each CP specifies the beginning of a range of text in the main document that constitutes a **section**. The range of text ends immediately prior to the next CP. A **PlcfSed** MUST NOT contain duplicate CPs. There MUST also be an end-of-section character (0x0C) as the final character in the text range of all but the last section. An end-of-section character (0x0C) which occurs at a CP and which is not the last character in a section specifies a manual page break.

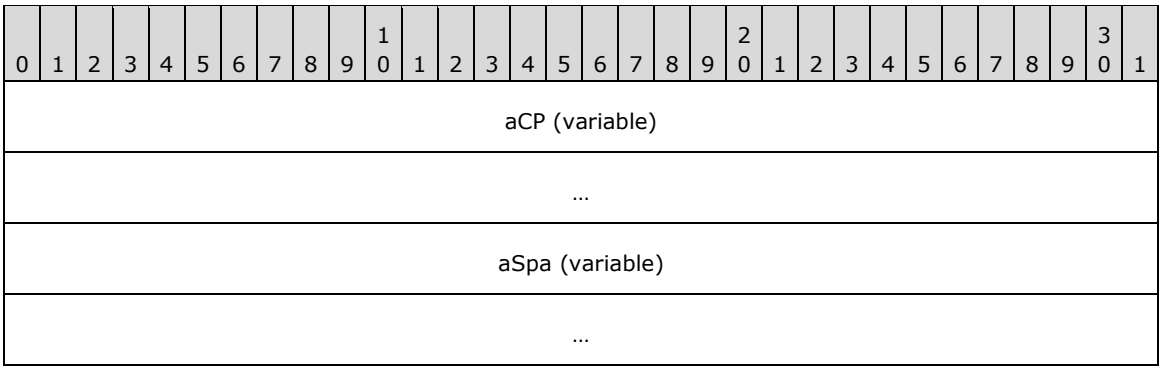
The last CP does not begin a new section. It MUST be at or beyond the end of the main document. Sections only contain text from the main document, so even when the last CP comes after text in other document parts, that text is not part of the last section.

aSed (variable): An array of 12-byte Sed structures. Each Sed structure contains the location of properties pertaining to the section that begins at the corresponding CP.

2.8.27 PlcfSpa

The **PlcfSpa** structure is a **PLC** structure in which the data elements are **SPA** structures (26 bytes each).

Copyright © 2025 Microsoft Corporation

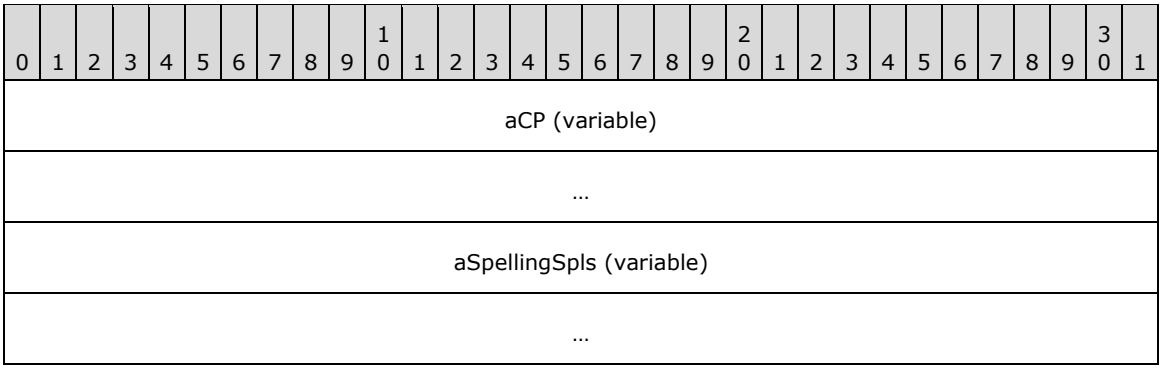


aCP (variable): An array of CPs. Each CP specifies the position in the document part of the anchor for a shape. This array MUST NOT contain duplicate CPs. The characters at all but the last CP MUST be 0x08 and MUST have sprmCFSpec applied with a value of 1. See sprmCFSpec for more information.

aSpa (variable): An array of SPAs (26 bytes each) that specify properties for the shape at the corresponding CP.

2.8.28 Plcfspl

The **Plcfspl** structure is a Plc structure whose data elements are SpellingSpls structures (2 bytes each).



aCP (variable): An array of CPs. CPs are positions in the set of all document parts. CPs are relative to the start of the main document but can extend into any of the document parts.

Each CP specifies the beginning of a range of text where the state in the corresponding SpellingSpls structure applies. The range of text ends immediately prior to the next CP.

A **Plcfspl** can contain duplicate CPs. Duplicate CPs specify an **insertion point** or a **deletion point** at that CP and the corresponding SpellingSpls state applies to that point.

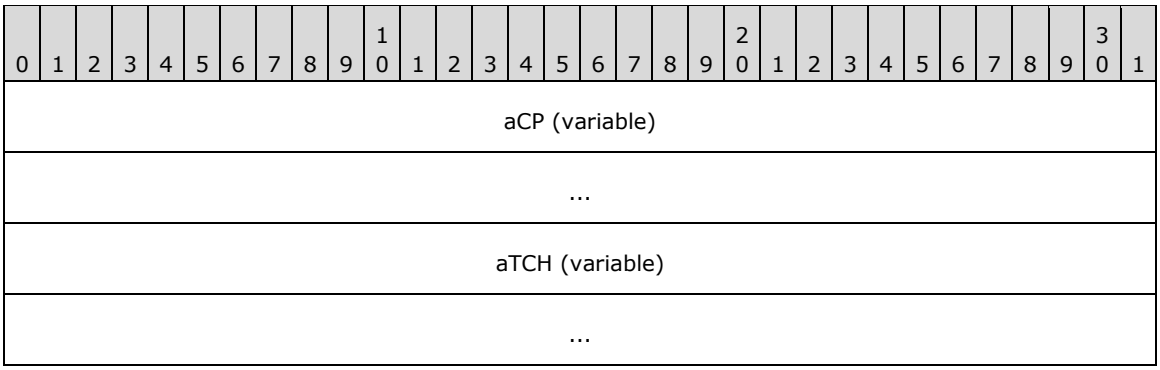
The last CP does not begin a new text range; it only terminates the previous one.

aSpellingSpls (variable): An array of 2-byte SpellingSpls structures. Each SpellingSpls structure contains the state of the spelling checker for the corresponding text range.

2.8.29 PlcfTch

The **PlcfTch** structure is a **PLC** whose data elements are **Tch** structures (4 bytes each). The count of CPs MUST be equal to one more than the count of **Tch**. Each pair of CPs represents a range of text in the main document described by the corresponding **Tch**.

Copyright © 2025 Microsoft Corporation



This information is a deprecated cache of table characters that SHOULD<201> be ignored. The following three CPs and the following two **Tch** structures SHOULD<202> be written to specify that this cache is undefined.

CP
0
FibRgLw97**.ccpText**
FibRgLw97**.ccpText** + 2

The following specifies the values for the fields of the first **Tch** structure.

Field	Value
fUnk	0
fUnused	0

The following specifies the values for the fields of the second **Tch** structure.

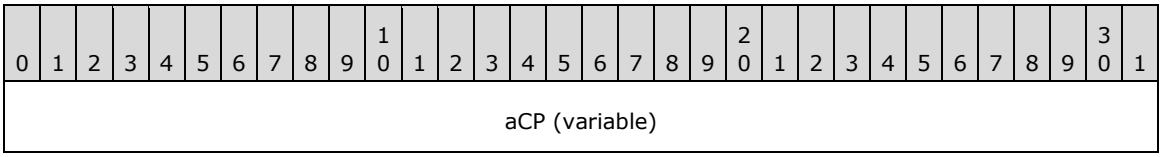
Field	Value
fUnk	1
fUnused	0

aCP (variable): An array of CPs. Each CP specifies the beginning of a range of text where a table character cache is stored. The last CP denotes the end of the last range of text. The range of text ends immediately prior to the next CP. MUST NOT contain duplicate CPs.

aTCH (variable): An array of Tch structures (4 bytes each) that each specifies a table character cache at the corresponding CP in **aCP**.

2.8.30 PlcfTxbxBkd

The **PlcfTxbxBkd** structure is a **PLC** structure where the data elements are **Tbkd** structures (6 bytes each).



Copyright © 2025 Microsoft Corporation

…
aTbkd (variable)
…

aCP (variable): An array of CPs. CPs are positions in the textboxes document.

Each CP specifies the beginning of a range of text to appear in a textbox specified in the corresponding **Tbkd** structure. The range of text ends immediately prior to the next CP. The last CP does not begin a new text range; it only terminates the previous one.

A **PlcfTxbxBkd** MUST NOT contain duplicate CPs.

aTbkd (variable): An array of 6-byte **Tbkd** structures that associate the text ranges with **FTXBXS** objects from **PlcftxbxTxt**.

2.8.31 PlcfTxbxHdrBkd

The **PlcfTxbxHdrBkd** structure is a PLC structure where the data elements are Tbkd structures (6 bytes each).

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
aCP (variable)																															
…																															
aTbkd (variable)																															
…																															

aCP (variable): An array of CPs. CPs are positions in the header textboxes document.

Each CP specifies the beginning of a range of text to appear in a textbox specified in the corresponding Tbkd structure. The range of text ends immediately prior to the next CP. The last CP does not begin a new text range; it only terminates the previous one.

A **PlcfTxbxHdrBkd** MUST NOT contain duplicate CPs.

aTbkd (variable): An array of 6-byte Tbkd structures that associates the text ranges with FTXBXS objects from PlcfHdrtxbxTxt.

2.8.32 PlcftxbxTxt

The **PlcftxbxTxt** structure is a **PLC** structure where the data elements are **FTXBXS** structures (22 bytes each).

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
aCP (variable)																															

Copyright © 2025 Microsoft Corporation

…
aFTXBXS (variable)
…

aCP (variable): An array of CPs. CPs are positions in the textboxes document.

Each CP specifies the beginning of a range of text to appear in a textbox indicated by the corresponding FTXBXS structure. The range of text ends immediately prior to the next CP. The last CP does not begin a new text range. It only terminates the previous one.

A **PlcftxbxTxt** MUST NOT contain duplicate CPs. The text ranges for each FTXBXS structure are separated by 0x0D characters that MUST be the last character in each range. The last text range is an exception. The text in the last range is ignored, and the 0x0D character is not required.

aFTXBXS (variable): An array of FTXBXS structures (22-bytes each) that associates the text ranges with shape objects.

2.8.33 Plcfuim

A **Plcfuim** structure is a **PLC** whose data elements are **UIM**s (20 bytes each), with the exception that the elements are not sorted according to their CPs.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
aCP (variable)																															
…																															
aUIM (variable)																															
…																															

aCP (variable): An array of CPs. CPs are positions in the set of all document parts. CPs are relative to the start of the main document but can extend into any of the document parts. Each CP in the **Plcfuim**, except the last one, represents the starting position of a range of text specified in the corresponding UIM. The last CP is undefined and MUST be ignored. Duplicate CPs are valid in a **Plcfuim**.

aUIM (variable): An array of UIMs.

2.8.34 PlcfWKB

The **PlcfWKB** is a PLC whose data elements are WKB structures (12 bytes each). Each **subdocument** is assigned one **WKB** structure.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
aCP (variable)																															

Copyright © 2025 Microsoft Corporation

...
aWKB (variable)
...

aCP (variable): An array of CPs. CPs are relative to the start of the main document. Each CP in the **PlcfWKB**, except the last, specifies the location in the main document where a subdocument begins. The CPs, except for the last, MUST be unique, greater than or equal to zero, and less than **FibBase**.**ccpText**. The last CP MUST be **FibBase**.**ccpText** incremented by 2.

aWKB (variable): An array of WKBs. Each WKB contains information about a subdocument.

2.8.35 PlcPcd

The **PlcPcd** structure is a PLC whose data elements are Pcds (8 bytes each). A **PlcPcd** MUST NOT contain duplicate CPs.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
aCP (variable)																															
...																															
aPcd (variable)																															
...																															

aCP (variable): An array of CPs that specifies the starting points of text ranges. The end of each range is the beginning of the next range. All CPs MUST be greater than or equal to zero. If any of the fields **ccpFtn**, **ccpHdd**, **ccpAtn**, **ccpEdn**, **ccpTxbx**, or **ccpHdrTxbx** from FibRgLw97 are nonzero, then the last CP MUST be equal to the sum of those fields plus **ccpText**+1. Otherwise, the last CP MUST be equal to **ccpText**.

aPcd (variable): An array of Pcds (8 bytes each) that specify the location of text in the WordDocument stream and any additional properties of the text. If **aPcd[*i*].fc.fCompressed** is 1, then the byte offset of the last character of the text referenced by **aPcd[*i*]** is given by the following.

$$\frac{\text{aPcd}[i].\text{fc.fc}}{2} + \mathbf{aCP}[i+1] - \mathbf{aCP}[i] - 1$$

Otherwise, the byte offset of the last character of the text referenced by **aPcd[*i*]** is given by the following.

$$\text{aPcd}[i].\text{fc.fc} + 2(\mathbf{aCP}[i+1] - \mathbf{aCP}[i] - 1)$$

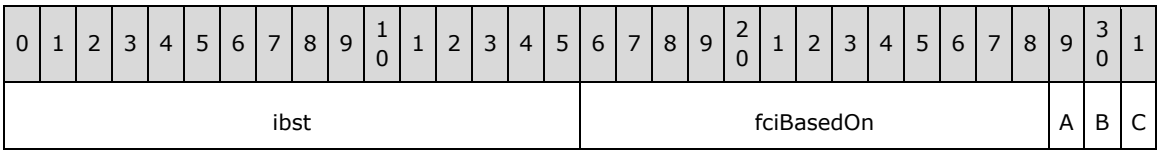
Because **aCP** MUST be sorted in ascending order and MUST NOT contain duplicate CPs, (**aCP**[*i*+1]-**aCP**[*i*])>0, for all valid indexes *i* of **aPcd**. Because a PLC MUST contain one more CP than a data element, *i*+1 is a valid index of **aCP** if *i* is a valid index of **aPcd**.

Copyright © 2025 Microsoft Corporation

2.9 Basic Types

2.9.1 Acd

The **Acd** structure specifies an **allocated command**.



ibst (2 bytes): Index in the Command String Table (TcgSttbf.**sttbf**) where a string representation of the argument to the allocated command is specified.

fciBasedOn (13 bits): An Fci that identifies the allocated command. MUST be one of the following Fci values. Each item specifies what the value of the argument as specified by **ibst** is.

- ApplyStyleName. The argument specifies the style to apply. The argument MUST be at least 2 characters long. The 16-bit value of the first character MUST be either 0x0001 or 0x0002.
- If the 16-bit value of the first character is 0x0001, then the argument MUST be exactly 3 characters long. The second and third characters specify the **sti** of the style to apply (see StdfBase.**sti**). The **sti** is given by (c_2 & 0x00FF) * 256 + (c_3 & 0x00FF) where c_2 and c_3 represent the character codes of the second and third characters. The **sti** value MUST be less than 267.
- If the 16-bit value of the first character is 0x0002, then the remaining characters in the argument specify the name of the style to apply.
- ApplyFontName. The argument is the name of the font to apply when this command is executed.
- ApplyAutoTextName. The argument is the name of the AutoText entry to insert when this command is executed.
- Columns. The argument specifies the number of columns to apply. The number of columns is the character code of the first character in the string.
- Condensed. The argument specifies the amount to condense by. The amount is specified in **twips** and is given by (c_1 & 0x00FF) * 256 + (c_2 & 0x00FF) where c_1 and c_2 represent the character codes of the first and second characters in the argument string.
- Expanded. The argument specifies the amount to expand by. The amount is specified in twips and is given by (c_1 & 0x00FF) * 256 + (c_2 & 0x00FF) where c_1 and c_2 represent the character codes of the first and second characters in the argument string.
- FontSize. The argument specifies the font size. The amount is specified in half points and is given by (c_1 & 0x00FF) * 256 + (c_2 & 0x00FF) where c_1 and c_2 represent the 16-bit values of the first and second characters in the argument string.
- Lowered. The argument specifies the amount to lower the text by. The amount is specified in half points and is given by (c_1 & 0x00FF) * 256 + (c_2 & 0x00FF) where c_1 and c_2 represent the 16-bit values of the first and second characters in the argument string.
- Raised. The argument specifies the amount to raise the text by. The amount is specified in half points and is given by (c_1 & 0x00FF) * 256 + (c_2 & 0x00FF) where c_1 and c_2 represent the 16-bit values codes of the first and second characters in the argument string.
- FileOpenFile. The argument specifies the file name to open.

Copyright © 2025 Microsoft Corporation

- Shading. The argument specifies which shading pattern to apply. The 16-bit value of the first character of the argument is an IPat.
- Borders. The argument specifies which border to apply. The 16-bit value of the first character of the argument MUST be one of the following values, and specifies which border to apply.

Value	Meaning
0	Clear all borders.
1	Apply top border.
2	Apply bottom border.
3	Apply left border.
4	Apply right border.
5	Apply inside borders.
6	Apply box borders.
7	Apply grid borders.

  The weight and style of the border applied is that of the last border applied by the user during the editing session, or a single, black border if no border has been applied in this session.
- Color. The argument specifies the color to apply. The 16-bit value of the first character of the argument is an Ico.
- Symbol. The argument specifies the symbol character and font to insert. The first character of the argument is the symbol character to insert. If there are more characters in the argument, they form the name of the font to apply to the newly inserted character. If the **character set** of the font to use is the SYMBOL_CHARSET then the symbol character to insert is given by (c_1 & 0x00FF).

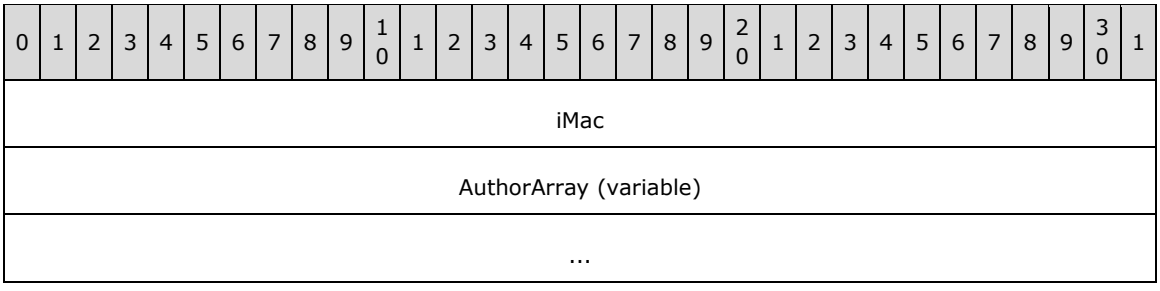
A - reserved (1 bit): This value MUST be 1.

B - fFree (1 bit): Specifies whether the current Acd is an unused slot in PlfAcd.**rgacd**. A value of 1 specifies that the current Acd is unused. A value of 0 specifies that the current Acd is valid and used.

C - fRef (1 bit): Specifies whether the current Acd is being referenced by a command. If **fFree** is 1, **fRef** MUST be 0; if **fFree** is 0, **fRef** MUST be 1.

2.9.2 Afd

The **AFD** structure is an array of indices into the author list that specifies whose revisions and comments were being hidden when this document was last saved.



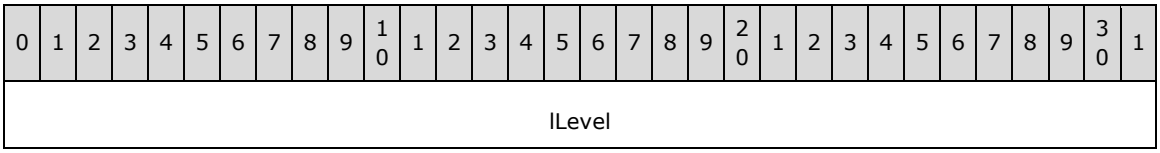
iMac (4 bytes): A signed integer that specifies the number of elements in **AuthorArray**. This value MUST be a non-negative number.

Copyright © 2025 Microsoft Corporation

AuthorArray (variable): An array of 16-bit integers that specifies the indexes in **SttbfRMark** of authors whose revisions and comments were being hidden from view when this document was last saved.

2.9.3 ASUMY

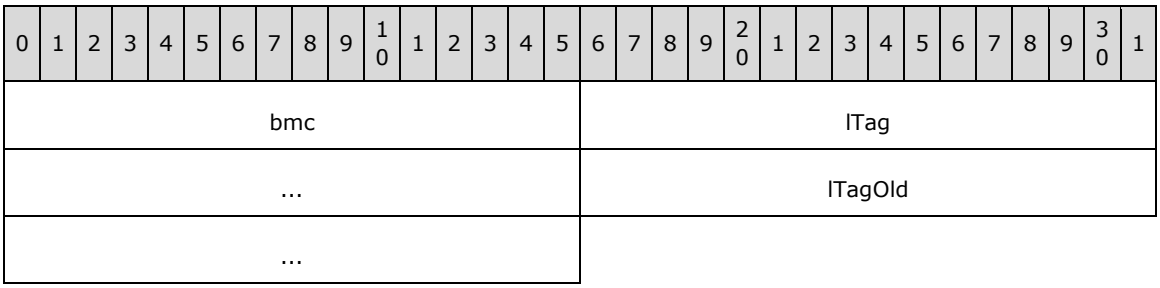
The **ASUMY** structure indicates the priority of a text range for **AutoSummary**.



lLevel (4 bytes): An integer that specifies the priority of the corresponding text range for AutoSummary. A smaller number implies greater importance of a text range to the summary. lLevel MUST be greater than 0, and MUST be less than or equal to the **asumyi.lHighestLevel** field of the Dop97.

2.9.4 ATNBE

The **ATNBE** structure contains information about an **annotation bookmark** in the document.



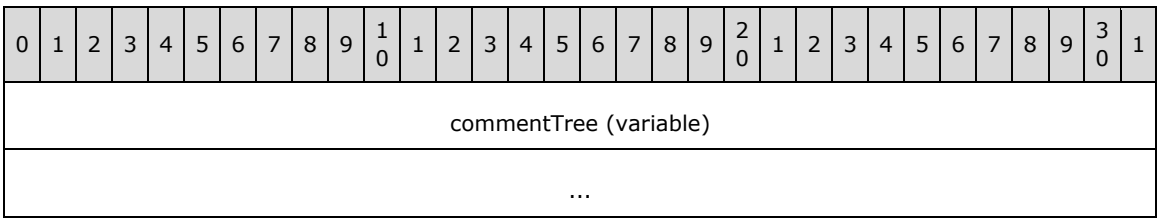
bmc (2 bytes): An unsigned integer specifying the bookmark (1) class that MUST be 0x0100, for annotation.

lTag (4 bytes): An unsigned integer that specifies a unique value used by the **lTagBkmk** member of ATRDPre10 structures inside the PlcfandRef at offset **fcPlcfandRef** in **lTag**'s nearest parent FibRgFcLcb97 to reference the annotation associated with this ATNBE. This MUST be unique for all ATNBEs inside a given SttbfAtnBkmk.

lTagOld (4 bytes): Unused. This value MUST be -1, and MUST be ignored.

2.9.5 AtrdExtra

The **AtrdExtra** structure is an array of information about comments that are kept parallel to the array of ATRDPre10s in the PlcfandRef specified by **fcPlcfandRef** in FibRgFcLcb97.

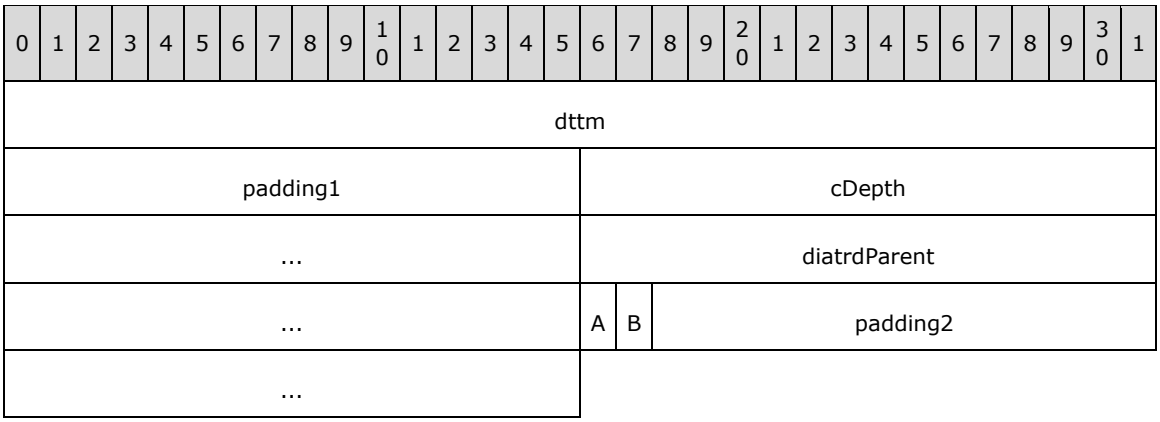


Copyright © 2025 Microsoft Corporation

commentTree (variable): An array of ATRDPost10s. The number of elements in this array MUST be equal to the number of ATRDPre10s in the PlcfandRef referenced by the **fcPlcfandRef** member of FibRgFcLcb97. This array is a tree that contains information about the comments in the document. The order of the comments in this array is determined by a pre-order traversal of the comment tree. A comment is considered a parent of a second comment if the second is a comment on the first. The depth of the comment in the tree is specified by **cDepth** in ATRDPost10. The location of the parent comment is specified by **diatrdParent** in ATRDPost10.

2.9.6 ATRDPost10

The **ATRDPost10** structure represents information about a comment that includes a date and time stamp, information about whether the comment was inked, and the tree structure of the comments. See the description of AtrdExtra for more about the tree layout. The location of the comment about which an **ATRDPost10** contains information is specified by the CP corresponding to the ATRDPre10 in the PlcfandRef specified by **fcPlcfandRef** in FibRgFcLcb97 with the same index as the **ATRDPost10**.



dttm (4 bytes): A DTTM specifying the date and time on which this comment was last created or modified.

padding1 (16 bits): This value MUST be 0, and MUST be ignored.

cDepth (4 bytes): The depth of this comment in the tree. If **cDepth** is 0, this comment has no parent and **diatrdParent** MUST be equal to zero. If this comment has a parent then **cDepth** MUST be equal to the **cDepth** value of the parent incremented by 1.

diatrdParent (4 bytes): The offset in the Table Stream of the parent of this comment in the tree. The parent is located 18***diatrdParent** bytes from the position of this comment. If **diatrdParent** is negative, the parent is located earlier in the stream; if **diatrdParent** is positive, the parent is located later in the stream. If **diatrdParent** is 0, this comment has no parent and **cDepth** MUST be equal to zero.

A - fOWSDiscussionItem (1 bit): This value MUST be 0, and MUST be ignored.

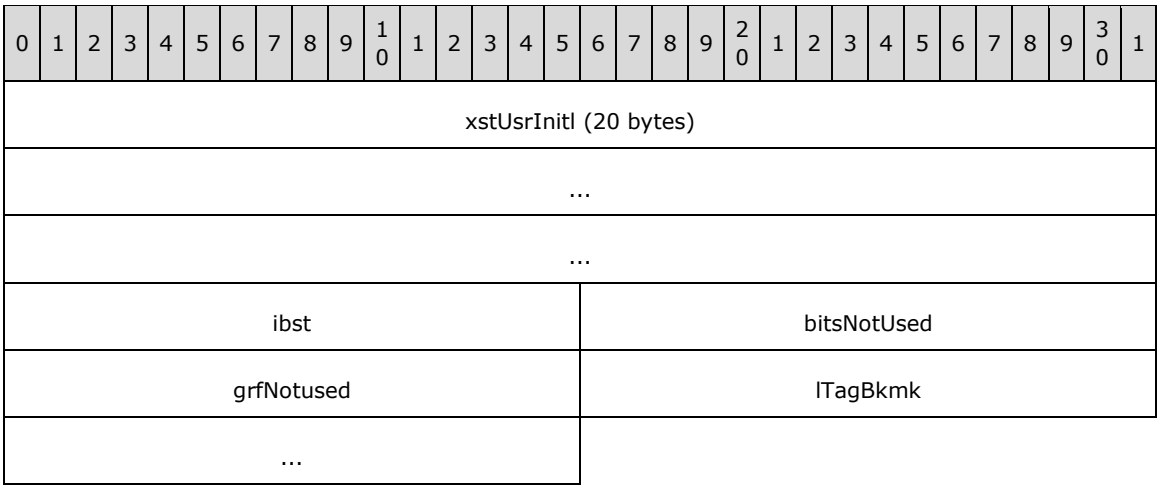
B - fInkAtn (1 bit): Denotes whether this comment is an ink annotation comment.

padding2 (30 bits): This value MUST be 0, and MUST be ignored.

2.9.7 ATRDPre10

The **ATRDPre10** structure contains information about a comment in the document including the initials of the author, an index to a string table with the name of the author, and a bookmark (1) identifier. More information about the comment can be specified in a corresponding **ATRDPost10** in the **AtrdExtra** at position **fcAtrdExtra**.

Copyright © 2025 Microsoft Corporation



xstUsrInitl (20 bytes): An LPXCharBuffer9 containing the initials of the user who left the annotation.

ibst (2 bytes): An index into the string table of comment author names. MUST be greater than or equal to zero, and MUST be less than the number of XSTs at position **fcGrpXstAtnOwners**.

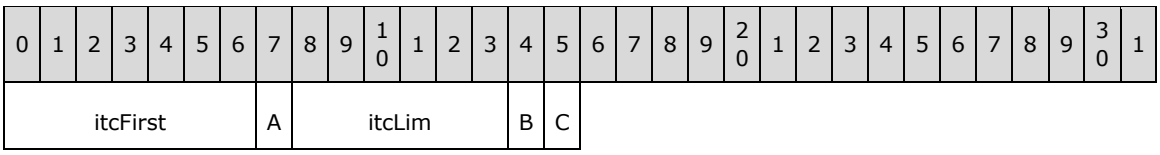
bitsNotUsed (2 bytes): This value MUST be 0, and MUST be ignored.

grfNotused (2 bytes): This value MUST be 0, and MUST be ignored.

lTagBkmk (4 bytes): A 4-byte value that identifies a bookmark (1) identifier. This value MUST be equal to -1 if and only if this comment is on a length zero text range in the Main Document. Otherwise MUST be equal to the **lTag** of one of the ATNBE structures in the SttbfAtnBkmk structure at position **fcSttbfAtnBkmk**.

2.9.8 BKC

The **BKC** structure contains information about how a **bookmark** interacts with tables.



itcFirst (7 bits): If **fCol** is zero, this value MUST be ignored. Otherwise, this value is an unsigned integer specifying the zero-based index of the table column that is the start of the table column range associated with the bookmark described by this BKC. See **itcLim** for additional constraints on the value of **itcFirst**.

A - fPub (1 bit): This value MUST be 0, and MUST be ignored.

itcLim (6 bits): If **fCol** is zero, this value MUST be ignored. Otherwise, this value is an unsigned integer specifying the zero-based index of the first column beyond the end of the table column range associated with the bookmark described by this BKC.

For all bookmark types, **itcFirst** MUST be less than **itcLim** if **fCol** is not zero.

For **range-level protection bookmarks**, itcLim MUST be exactly 1 greater than itcFirst if fCol is not zero.

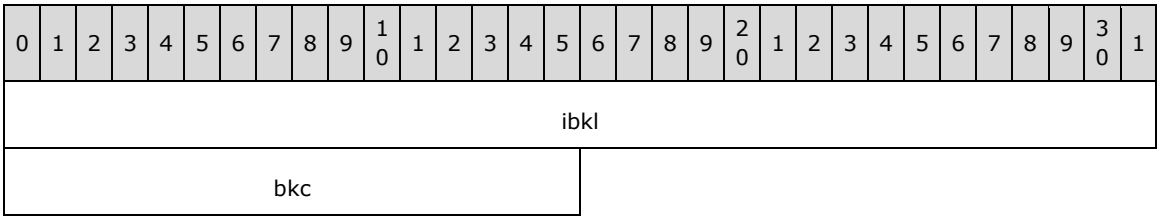
Copyright © 2025 Microsoft Corporation

B - fNative (1 bit): A bit flag that specifies whether an application is expected to include the bookmark described by this BKC when saving its file as **Rich Text Format (RTF)**, **HTML**, or **XML**. If **fNative** is zero, the bookmark is no longer needed and is a disposable item that was generated by the application to act as a temporary placeholder at run time. The bookmark is not expected to be included if the file is saved as RTF, HTML, or XML.

C - fCol (1 bit): For **structured document tag bookmarks** and **annotation bookmarks**, **fCol** MUST be zero. Otherwise, if the lowest table depth within the span of text defined by the CPs of a bookmark is greater than zero, and the span of text defined by the CPs of that bookmark contains a table cell mark from that table and nothing outside that table, then the **fCol** member of the bookmark's (1) BKC MUST be 1. Otherwise, it MUST be zero. If the **fCol** member of the BKC of a range-level protection bookmark is set to 1, the span of text that is defined by the CPs of that bookmark MUST NOT include more than one table terminating paragraph mark. Further constraints upon the span of text defined by the CPs of a bookmark can be found in section Plcfbkf.

2.9.9 BKF

The **BKF** structure contains information about a **bookmark**.

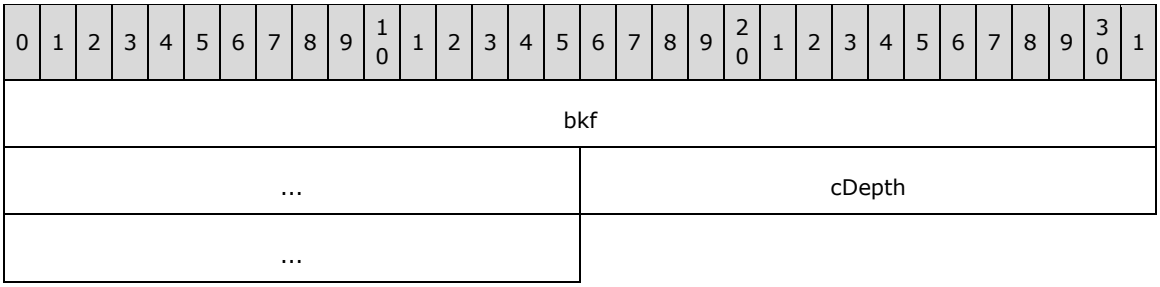


ibkl (4 bytes): An unsigned integer that specifies a zero-based index into the Plcbkl or Plcbkld that is paired with the Plcbkf or Plcbkfd containing this BKF. The entry found at that index specifies the location of the end of the bookmark that is associated with this BKF. **Ibkl** MUST be unique for all BKFs in a given Plcbkf or Plcbkfd.

bkc (2 bytes): A BKC that specifies further information about the bookmark.

2.9.10 BKFD

The **BKFD** structure is a BKF with additional information used for structured document tag **bookmarks**.



bkf (6 bytes): A BKF specifying further information about the bookmark.

cDepth (4 bytes): An integer specifying the number of bookmarks in the document of the same type as the bookmark associated with this BKFD, the ranges of which overlap the beginning of the range of this bookmark. To increment the count, a bookmark MUST meet the following constraints:

1. The BKFD of the bookmark occupies the Plcbkfd containing this BKFD

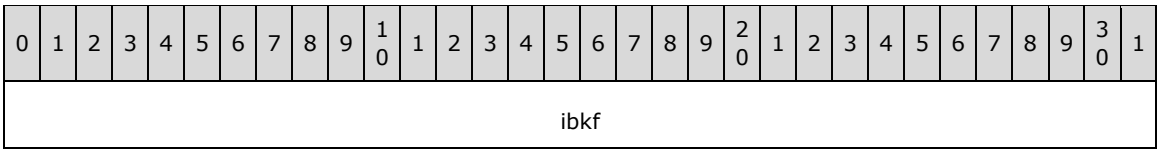
Copyright © 2025 Microsoft Corporation

2. The start CP (cpS) and limit CP (cpL) of the bookmark, as defined in the prose for that Plcbkfd and the Plcbkld it is paired with, satisfy the following in relation to the CP (cpCur) marking the beginning of the bookmark of this BKFD: cpS == cpCur == cpL || cpS <= cpCur < cpL

   Because BKFD is associated only with **structured document tag bookmarks**, **cDepth** can be rephrased more simply as the one-based count of other structured document tag bookmarks in the file that contain the bookmark associated with this BKFD.

2.9.11 BKL

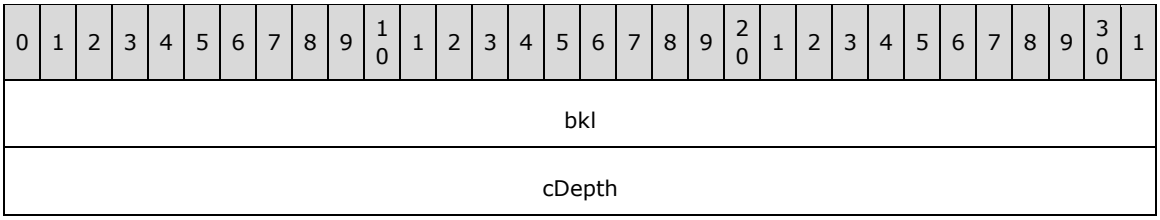
The **BKL** structure links the end of a **bookmark** to the beginning of the same bookmark.



ibkf (4 bytes): An unsigned integer that specifies a zero-based index into the Plcbkfd that is paired with the Plcbkld containing this BKL. The entry found at this index specifies the location of the beginning of the bookmark associated with this BKL. **Ibkf** MUST be unique for all BKLs in a given Plcbkld.

2.9.12 BKLD

The **BKLD** structure is a BKL with additional information used for structured document tag **bookmarks**.



bkl (4 bytes): A BKL specifying further information about the bookmark.

cDepth (4 bytes): An integer specifying the number of bookmarks in the document of the same type as the bookmark associated with this **BKLD**, the ranges of which overlap the limit of this bookmark range. To increment the count, a bookmark MUST meet the following constraints:

1. The bookmark **BKLD** occupies the **Plcbkld** containing this **BKLD**

2. The bookmark limit CP (cpL) and start CP (cpS), as defined in the specification of that Plcbkld and the **Plcbkfd** it is paired with, satisfy the following in relation to the CP (cpCur) marking the limit of the bookmark of this **BKLD**

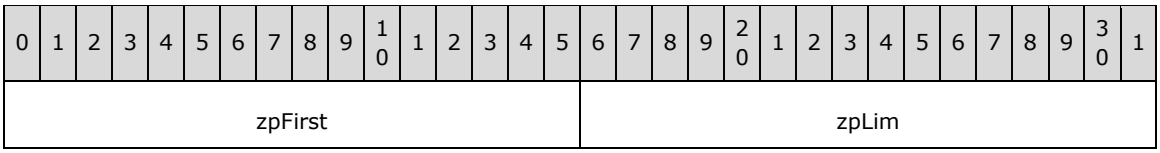
$$cpS \neq cpL$$
$$cpS \leq cpCur < cpL$$

   Because **BKLD** is only associated with structured document tag bookmarks, cDepth can be rephrased more simply as the zero-based count of other structured document tag bookmarks in the file that contain the bookmark associated with this **BKLD**.

Copyright © 2025 Microsoft Corporation

2.9.13 BlockSel

The **BlockSel** structure is used by Selsf to specify the left and right boundaries of a text block selection. The values are pixels at the zoom level in which the selection was made.



zpFirst (2 bytes): A signed integer that specifies the **physical left** boundary of the selection, in pixels. The physical left page margin is at pixel zero.

zpLim (2 bytes): A signed integer that specifies the **physical right** boundary of the selection, in pixels. **zpLim** MUST be greater than or equal to **zpFirst**.

2.9.14 Bool16

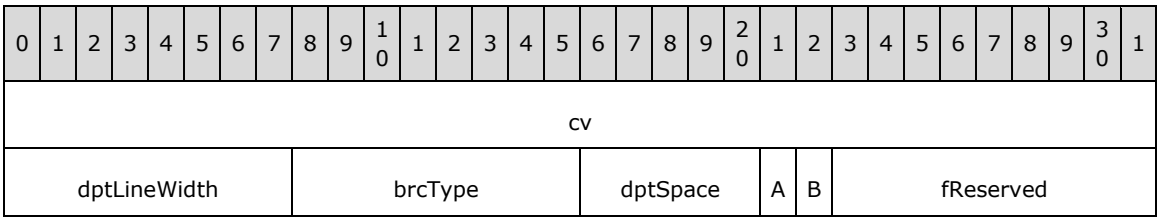
The **Bool16** structure is a 16-bit unsigned integer. This value MUST be either 0x0000 ("false") or 0x0001 ("true").

2.9.15 Bool8

The **Bool8** structure is an 8-bit unsigned integer. This value MUST be either 0x00 ("false") or 0x01 ("true").

2.9.16 Brc

The **Brc** structure specifies a border.



cv (4 bytes): A COLORREF that specifies the color of this border.

dptLineWidth (8 bits): Specifies the width of the border. Different meanings based on brcType.

brcType	Meaning
brcType < 0x40	An unsigned integer that specifies the width of the border in 1/8-point increments. Values of less than 2 are considered to be equivalent to 2.
brcType >= 0x40	An unsigned integer that specifies the width of the border in 1-point increments. This value MUST be less than 32.

brcType (1 byte): A BrcType that specifies the type of this border.

Copyright © 2025 Microsoft Corporation

dptSpace (5 bits): An unsigned integer that specifies the distance from the text to the border, in points. For page borders, sprmSPgbProp can specify that this value shall specify the distance from the edge of the page to the border.

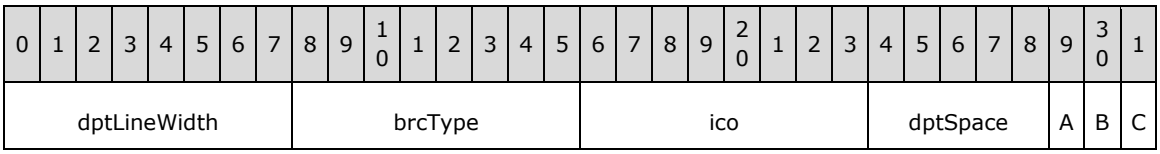
A - fShadow (1 bit): If this bit is set, the border has an additional shadow effect. For top, **logical left**, and between borders, this has no visual effect.

B - fFrame (1 bit): If this bit is set, then the border has a three-dimensional effect. For top, logical left, and between borders, this has no visual effect. For visually symmetric border types, this has no visual effect.

fReserved (9 bits): This value is unused and MUST be ignored.

2.9.17 Brc80

The **Brc80** structure describes a border.



dptLineWidth (8 bits): An unsigned integer that specifies the width of the border in 1/8-**point** increments. Values of less than 2 are considered to be equivalent to 2.

brcType (1 byte): A BrcType that specifies the type of this border. This value MUST NOT be 0x1A or 0x1B.

ico (1 byte): An Ico that specifies the color of this border.

dptSpace (5 bits): An unsigned integer that specifies the distance from the text to the border, in points.

A - fShadow (1 bit): If this bit is set, the border has an additional shadow effect. For top and **logical left** borders, this bit has no visual effect.

B - fFrame (1 bit): Specifies whether the specified border is modified to create a frame effect by reversing the appearance of the border from the edge nearest the text to the edge furthest from the text. The frame effect shall only be applied to right and bottom borders.

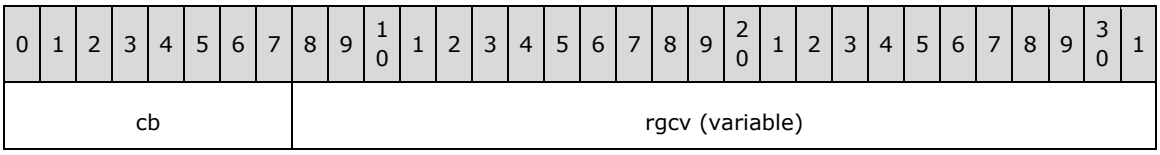
C - reserved (1 bit): This bit MUST be zero, and MUST be ignored.

2.9.18 Brc80MayBeNil

The **Brc80MayBeNil** structure is a **Brc80** structure. When all bits are set (0xFFFFFFFF when interpreted as a 4-byte unsigned integer), this structure specifies that the region in question has no border.

2.9.19 BrcCvOperand

The **BrcCvOperand** structure specifies border colors.



Copyright © 2025 Microsoft Corporation

...

cb (1 byte): An unsigned integer value that specifies the size, in bytes, of **rgcv.** This value MUST be 4*_n_, where _n_ is the number of cells in the table row.

rgcv (variable): An array of COLORREF. Each COLORREF specifies the color of the border for the corresponding cell in the table row, starting from the logical, left-most cell. If any of the COLORREFs in this array have the following value, it specifies that there is no corresponding border.

Member	Value
Red	0xFF
Green	0xFF
Blue	0xFF
fAuto	0xFF

2.9.20 BrcMayBeNil

The **BrcMayBeNil** structure is either a **NilBrc** or **Brc** structure, depending on the value of the last four bytes of the structure.

If the last four bytes are 0xFFFFFFFF, the **BrcMayBeNil** is a **NilBrc** that specifies that the table cells in question have no border. Otherwise, it is a **Brc** structure that specifies the border type of table cells.

2.9.21 BrcOperand

The **BrcOperand** structure is the operand to several **SPRM**s that control borders.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
cb								brc																							
...																															
...																															

cb (1 byte): An unsigned integer value that specifies the size of this BrcOperand, not including this byte. The cb MUST be 8.

brc (8 bytes): A **BRC** that specifies the border to be applied.

2.9.22 BrcType

brcType (8 bits): An unsigned integer that specifies the type of border. Values that are larger than 0x1B are not valid unless they describe a page border, in which case they can be a value in the range of 0x40 to 0xE3, inclusive.

Values MUST be from the following table. The reference column specifies for each **brcType** value the ST_Border enumeration value in [ECMA-376] part 1, section 17.18.2, that further specifies the meaning of the border type.

Copyright © 2025 Microsoft Corporation

Value	Meaning	Reference
0x00	No border.	none
0x01	A single line.	single
0x03	A double line.	double
0x05	A thin single solid line.	
0x06	A dotted border.	dotted
0x07	A dashed border with large gaps between the dashes.	dashed
0x08	A border of alternating dots and dashes.	dotDash
0x09	A border of alternating sets of two dots and one dash.	dotDotDash
0x0A	A triple line border.	triple
0x0B	A thin outer border and a thick inner border with a small gap between them.	thinThickSmallGap
0x0C	A thin outer border and thick inner border with a small gap between them.	thickThinSmallGap
0x0D	A thin outer border, a thick middle border, and a thin inner border with a small gap between them.	thinThickThinSmallGap
0x0E	A thin outer border and a thick inner border with a medium gap between them.	thinThickMediumGap
0x0F	A thin outer border and a thick inner border and a medium gap between them.	thickThinMediumGap
0x10	A thin outer border, a thick middle border, and a thin inner border with a medium gaps between them.	thinThickThinMediumGap
0x11	A thick outer border and a thin inner border with a large gap between them.	thinThickLargeGap
0x12	A thin outer border and a thick inner border with a large gap between them.	thickThinLargeGap
0x13	A thin outer border, a thick middle border, and a thin inner border with large gaps between them.	thinThickThinLargeGap
0x14	A single wavy line.	wave
0x15	A double wavy line.	doubleWave
0x16	A dashed border with small gaps between the dashes.	dashSmallGap
0x17	A border consisting of alternating groups of 5 and 1 thin diagonal lines.	dashDotStroked
0x18	A thin light gray outer border, a thick medium gray middle border, and a thin black inner border with no gaps between them.	threeDEmboss
0x19	A thin black outer border, a thick medium gray middle border, and a thin light gray inner border with no gaps between them.	threeDEngrave
0x1A	A thin light gray outer border and a thin medium gray inner border with a large gap between them.	outset

Copyright © 2025 Microsoft Corporation

Value	Meaning	Reference
0x1B	A thin medium gray outer border and a thin light gray inner border with a large gap between them.	inset
0x40	An image border.	apples
0x41	An image border.	archedScallops
0x42	An image border.	babyPacifier
0x43	An image border.	babyRattle
0x44	An image border.	balloons3Colors
0x45	An image border.	balloonsHotAir
0x46	An image border.	basicBlackDashes
0x47	An image border.	basicBlackDots
0x48	An image border.	basicBlackSquares
0x49	An image border.	basicThinLines
0x4A	An image border.	basicWhiteDashes
0x4B	An image border.	basicWhiteDots
0x4C	An image border.	basicWhiteSquares
0x4D	An image border.	basicWideInline
0x4E	An image border.	basicWideMidline
0x4F	An image border.	basicWideOutline
0x50	An image border.	bats
0x51	An image border.	birds
0x52	An image border.	birdsFlight
0x53	An image border.	cabins
0x54	An image border.	cakeSlice
0x55	An image border.	candyCorn
0x56	An image border.	celticKnotwork
0x57	An image border.	certificateBanner
0x58	An image border.	chainLink
0x59	An image border.	champagneBottle
0x5A	An image border.	checkedBarBlack
0x5B	An image border.	checkedBarColor
0x5C	An image border.	checkered
0x5D	An image border.	christmasTree

Copyright © 2025 Microsoft Corporation

Value	Meaning	Reference
0x5E	An image border.	circlesLines
0x5F	An image border.	circlesRectangles
0x60	An image border.	classicalWave
0x61	An image border.	clocks
0x62	An image border.	compass
0x63	An image border.	confetti
0x64	An image border.	confettiGrays
0x65	An image border.	confettiOutline
0x66	An image border.	confettiStreamers
0x67	An image border.	confettiWhite
0x68	An image border.	cornerTriangles
0x69	An image border.	couponCutoutDashes
0x6A	An image border.	couponCutoutDots
0x6B	An image border.	crazyMaze
0x6C	An image border.	creaturesButterfly
0x6D	An image border.	creaturesFish
0x6E	An image border.	creaturesInsects
0x6F	An image border.	creaturesLadyBug
0x70	An image border.	crossStitch
0x71	An image border.	cup
0x72	An image border.	decoArch
0x73	An image border.	decoArchColor
0x74	An image border.	decoBlocks
0x75	An image border.	diamondsGray
0x76	An image border.	doubleD
0x77	An image border.	doubleDiamonds
0x78	An image border.	earth1
0x79	An image border.	earth2
0x7A	An image border.	eclipsingSquares1
0x7B	An image border.	eclipsingSquares2
0x7C	An image border.	eggsBlack
0x7D	An image border.	fans

Copyright © 2025 Microsoft Corporation

Value	Meaning	Reference
0x7E	An image border.	film
0x7F	An image border.	firecrackers
0x80	An image border.	flowersBlockPrint
0x81	An image border.	flowersDaisies
0x82	An image border.	flowersModern1
0x83	An image border.	flowersModern2
0x84	An image border.	flowersPansy
0x85	An image border.	flowersRedRose
0x86	An image border.	flowersRoses
0x87	An image border.	flowersTeacup
0x88	An image border.	flowersTiny
0x89	An image border.	gems
0x8A	An image border.	gingerbreadMan
0x8B	An image border.	gradient
0x8C	An image border.	handmade1
0x8D	An image border.	handmade2
0x8E	An image border.	heartBalloon
0x8F	An image border.	heartGray
0x90	An image border.	hearts
0x91	An image border.	heebieJeebies
0x92	An image border.	holly
0x93	An image border.	houseFunky
0x94	An image border.	hypnotic
0x95	An image border.	iceCreamCones
0x96	An image border.	lightBulb
0x97	An image border.	lightning1
0x98	An image border.	lightning2
0x99	An image border.	mapPins
0x9A	An image border.	mapleLeaf
0x9B	An image border.	mapleMuffins
0x9C	An image border.	marquee
0x9D	An image border.	marqueeToothed

Copyright © 2025 Microsoft Corporation

Value	Meaning	Reference
0x9E	An image border.	moons
0x9F	An image border.	mosaic
0xA0	An image border.	musicNotes
0xA1	An image border.	northwest
0xA2	An image border.	ovals
0xA3	An image border.	packages
0xA4	An image border.	palmsBlack
0xA5	An image border.	palmsColor
0xA6	An image border.	paperClips
0xA7	An image border.	papyrus
0xA8	An image border.	partyFavor
0xA9	An image border.	partyGlass
0xAA	An image border.	pencils
0xAB	An image border.	people
0xAC	An image border.	peopleWaving
0xAD	An image border.	peopleHats
0xAE	An image border.	poinsettias
0xAF	An image border.	postageStamp
0xB0	An image border.	pumpkin1
0xB1	An image border.	pushPinNote2
0xB2	An image border.	pushPinNote1
0xB3	An image border.	pyramids
0xB4	An image border.	pyramidsAbove
0xB5	An image border.	quadrants
0xB6	An image border.	rings
0xB7	An image border.	safari
0xB8	An image border.	sawtooth
0xB9	An image border.	sawtoothGray
0xBA	An image border.	scaredCat
0xBB	An image border.	seattle
0xBC	An image border.	shadowedSquares
0xBD	An image border.	sharksTeeth

Copyright © 2025 Microsoft Corporation

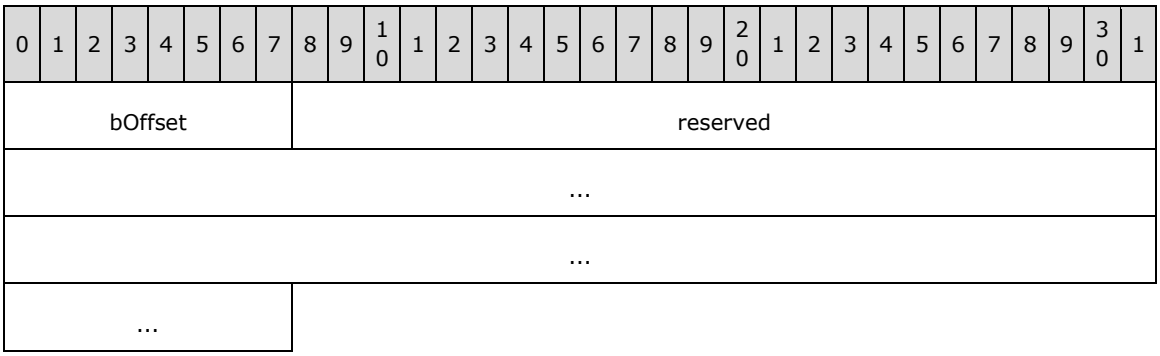
Value	Meaning	Reference
0xBE	An image border.	shorebirdTracks
0xBF	An image border.	skyrocket
0xC0	An image border.	snowflakeFancy
0xC1	An image border.	snowflakes
0xC2	An image border.	sombrero
0xC3	An image border.	southwest
0xC4	An image border.	stars
0xC5	An image border.	starsTop
0xC6	An image border.	stars3d
0xC7	An image border.	starsBlack
0xC8	An image border.	starsShadowed
0xC9	An image border.	sun
0xCA	An image border.	swirligig
0xCB	An image border.	tornPaper
0xCC	An image border.	tornPaperBlack
0xCD	An image border.	trees
0xCE	An image border.	triangleParty
0xCF	An image border.	triangles
0xD0	An image border.	tribal1
0xD1	An image border.	tribal2
0xD2	An image border.	tribal3
0xD3	An image border.	tribal4
0xD4	An image border.	tribal5
0xD5	An image border.	tribal6
0xD6	An image border.	twistedLines1
0xD7	An image border.	twistedLines2
0xD8	An image border.	vine
0xD9	An image border.	waveline
0xDA	An image border.	weavingAngles
0xDB	An image border.	weavingBraid
0xDC	An image border.	weavingRibbon
0xDD	An image border.	weavingStrips

Copyright © 2025 Microsoft Corporation

Value	Meaning	Reference
0xDE	An image border.	whiteFlowers
0xDF	An image border.	woodwork
0xE0	An image border.	xIllusions
0xE1	An image border.	zanyTriangles
0xE2	An image border.	zigZag
0xE3	An image border.	zigZagStitch
0xFF	This MUST be ignored.	

2.9.23 BxPap

The **BxPap** structure specifies the offset of a **PapxInFkp** in **PapxFkp**.

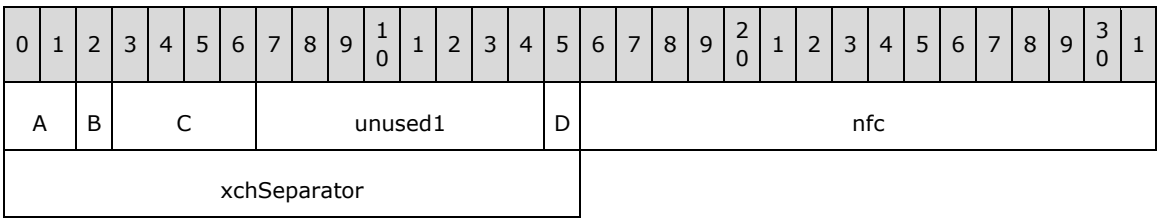


bOffset (1 byte): An unsigned integer that specifies the offset of a **PapxInFkp** in a **PapxFkp**. The offset of the **PapxInFkp** is **bOffset***2. If **bOffset** is 0 then there is no **PapxInFkp** for this paragraph and this paragraph has the default properties as specified in section 2.6.2.

reserved (12 bytes): Specifies version-specific paragraph height information. This value SHOULD<203> be 0 and SHOULD<204> be ignored.

2.9.24 CAPI

The **CAPI** structure contains information about a **caption**.



A - iLocation (2 bits): An unsigned integer that specifies the insert location for the caption. This MUST be one of the following values.

Value	Meaning
0x0	Insert the caption below the selected item.

Copyright © 2025 Microsoft Corporation

Value	Meaning
0x1	Insert the caption above the selected item.

B - fChapNum (1 bit): A bit that specifies whether or not to include a chapter number in the caption.

C - iHeading (4 bits): An unsigned integer that specifies which **heading style** marks the beginning of a new chapter for the purpose of **chapter numbering** in this caption. This value MUST be one of the following.

Value	Meaning
0x1	Heading 1 marks the beginning of a new chapter.
0x2	Heading 2 marks the beginning of a new chapter.
0x3	Heading 3 marks the beginning of a new chapter.
0x4	Heading 4 marks the beginning of a new chapter.
0x5	Heading 5 marks the beginning of a new chapter.
0x6	Heading 6 marks the beginning of a new chapter.
0x7	Heading 7 marks the beginning of a new chapter.
0x8	Heading 8 marks the beginning of a new chapter.
0x9	Heading 9 marks the beginning of a new chapter.

If **fChapNum** is zero, this field MUST be ignored.

unused1 (8 bits): This field is undefined and MUST be ignored.

D - fNoLabel (1 bit): A bit that specifies whether or not to include the label in the caption. This bit MAY<205> be ignored.

nfc (2 bytes): An MSONFC, as specified in [MS-OSHARED] section 2.2.1.3, that specifies the formatting of the caption number.

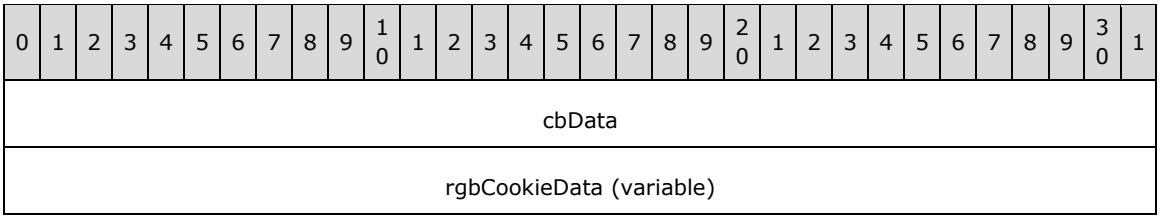
xchSeparator (2 bytes): A **Unicode** character that specifies the character that separates the chapter number and caption number of the caption. This value MUST be one of the following.

Value	Meaning
0x001E	A hyphen (-) separates the chapter number and caption number.
0x002E	A period (.) separates the chapter number and the caption number.
0x003A	A colon (:) separates the chapter number and the caption number.
0x2013	An en-dash (–) separates the chapter number and the caption number.
0x2014	An em-dash (—) separates the chapter number and the caption number.

If **fChapNum** is zero, this value MUST be ignored.

2.9.25 CDB

The **CDB** structure contains implementation-specific binary data that represents a **grammar checker cookie** that is stored by the given **grammar checker**.



Copyright © 2025 Microsoft Corporation

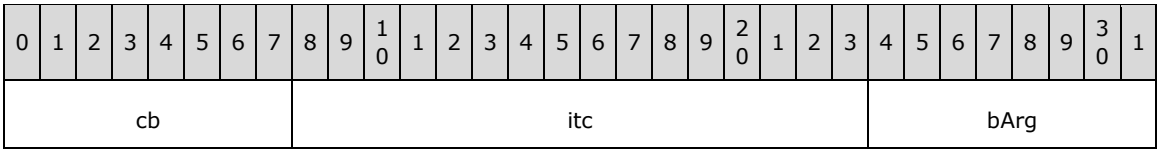
...

cbData (4 bytes): An unsigned integer value that specifies the length of **rgbCookieData**, in bytes.

rgbCookieData (variable): An array of BYTE. The grammar checker cookie data.

2.9.26 CellHideMarkOperand

The **CellHideMarkOperand** structure is an operand that is used by sprmTCellFHideMark. This operand specifies which cells are rendered with no height when cells are empty.



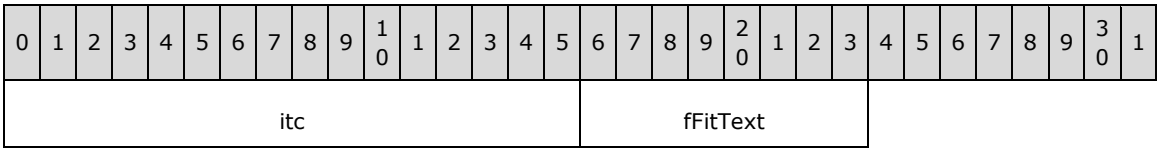
cb (1 byte): An unsigned integer that specifies the size of this operand in bytes, not including **cb**. **cb** MUST be 3.

itc (2 bytes): An **ItcFirstLim** that specifies which cells this **CellHideMarkOperand** applies to.

bArg (1 byte): A **Bool8** that specifies whether cells **itc.itcFirst** through **itc.itcLim**, decremented by 1, are rendered with no height if all cells in the row are empty.

2.9.27 CellRangeFitText

The **CellRangeFitText** structure is an operand that is used by sprmTFitText. This operand specifies a set of cells in a table row, and whether their contents stretch or compress to fill their widths.

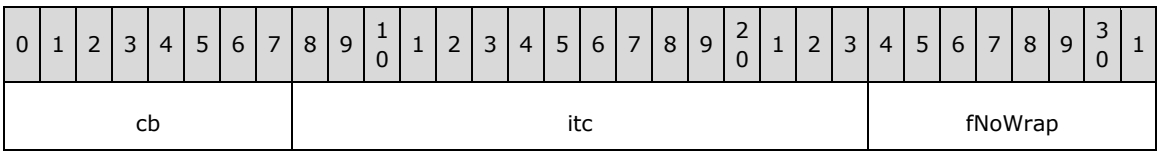


itc (2 bytes): A ItcFirstLim structure that specifies a cell range in the table row.

fFitText (1 byte): A Bool8. When set, the contents of each table cell only line wrap at the end of a paragraph, or at a line break character. Furthermore, the application SHOULD apply other properties as necessary to cause the contents of the first line in each cell to stretch or compress such that they exactly fill the width of the table cell.

2.9.28 CellRangeNoWrap

The **CellRangeNoWrap** structure is an operand that is used by sprmTFCellNoWrap. This operand specifies a set of cells in a table row and the preferred line wrapping layout of each.



cb (1 byte): An unsigned integer that specifies the size in bytes of the remainder of this structure. MUST be 3.

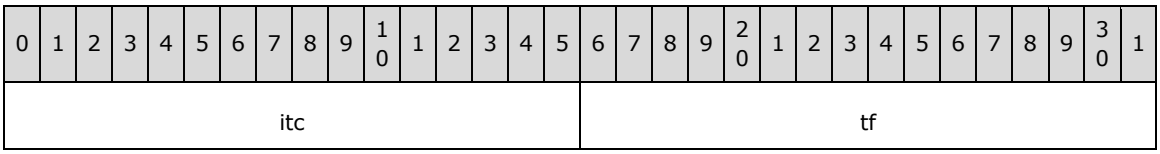
itc (2 bytes): A ItcFirstLim structure that specifies a cell range to which **fNoWrap** applies.

Copyright © 2025 Microsoft Corporation

fNoWrap (1 byte): A Bool8. When set, the preferred layout of the contents of each cell is a single line. This preference is ignored when the preferred width of the cell is set to ftsDxa.

2.9.29 CellRangeTextFlow

The **CellRangeTextFlow** structure specifies a range of cells in a table row, and the text flow model of the cell contents.

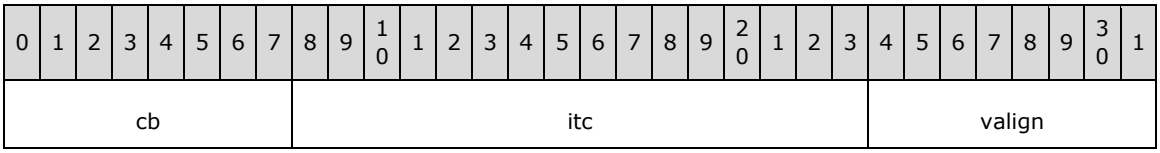


itc (2 bytes): An ItcFirstLim that specifies a cell range in the table row.

tf (2 bytes): A TextFlow that specifies how contents in each cell flow, and how text is rotated.

2.9.30 CellRangeVertAlign

The **CellRangeVertAlign** structure specifies a range of cells in a table row, and the vertical alignment of the cell contents.



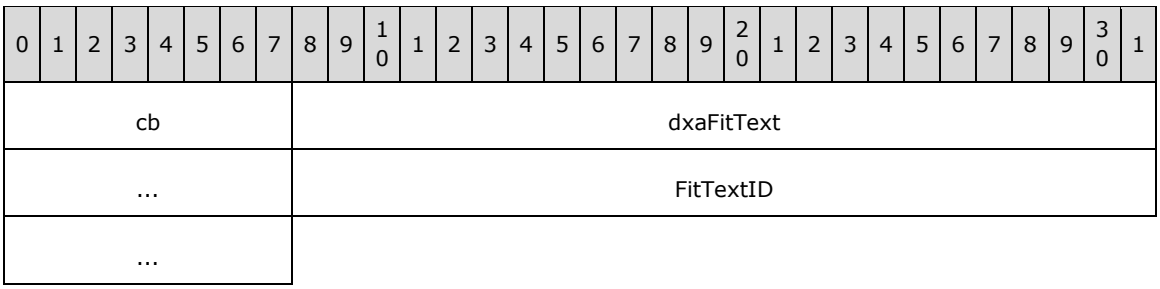
cb (1 byte): Specifies the byte count of the remainder of this structure. The value MUST be 3.

itc (2 bytes): An ItcFirstLim that specifies a cell range in the table row.

valign (1 byte): A VerticalAlign that specifies how contents inside each cell in the range are aligned.

2.9.31 CFitTextOperand

The **CFitTextOperand** structure is an operand that is used by sprmCFitText to specify how text runs are formatted to fit a particular width.



cb (1 byte): The number of bytes that this operand occupies. This value MUST be 0x08.

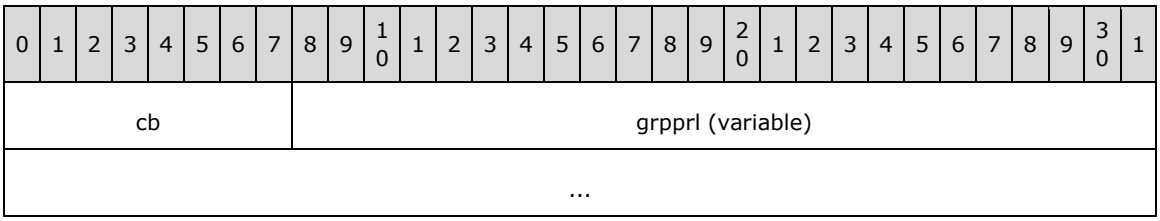
dxaFitText (4 bytes): A 32-bit signed integer value that specifies, in **twips**, the size of the space in which to fit the text. Text that would occupy a smaller width than specified has space added between characters. Text that would occupy a greater width than specified is compressed proportionally. A value of zero specifies that the Sprm is ignored. A value representing a width that is too large for the text run is also ignored. A negative value or a value representing a width that is too small for the text run specifies the minimum width.

Copyright © 2025 Microsoft Corporation

FitTextID (4 bytes): A 32-bit signed integer that uniquely identifies a fit text region across multiple character runs and instances of sprmCFitText. Contiguous character runs that share a common **FitTextID** are part of the same fit text region. If the runs are not contiguous, the **FitTextID** is ignored and they are not linked.

2.9.32 Chpx

The **Chpx** structure specifies a set of properties for text.

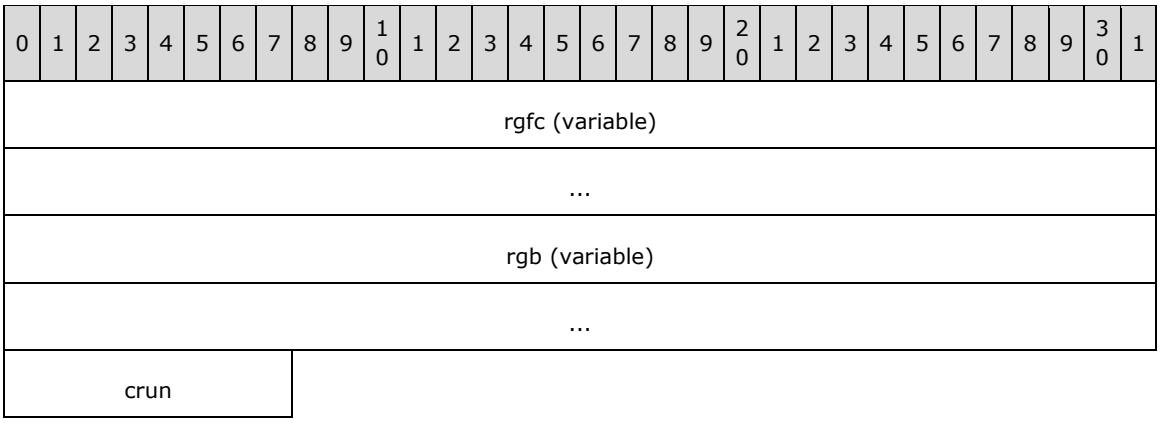


cb (1 byte): An unsigned integer that specifies the size of **grpprl**, in bytes.

grpprl (variable): An array of Prl. Specifies the properties. This array MUST contain a whole number of Prls.

2.9.33 ChpxFkp

The **ChpxFkp** structure maps text to its character properties. A **ChpxFkp** structure is 512 bytes in size, with **crun** in the last byte. The elements of **rgb** point to Chpxs that start at offsets between **crun** and the end of **rgb**.



rgfc (variable): An array of 4-byte unsigned integers. Each element of this array specifies an offset in the WordDocument Stream where a run of text begins. This array MUST be sorted in ascending order and MUST NOT contain duplicates. Each run ends at the beginning of the next run. This array contains **crun**+1 elements, where the last element specifies the end of the last run.

rgb (variable): An array of 1-byte unsigned integers, followed by an array of Chpx structures. The elements of this array, which has **crun** elements and parallels **rgfc**, each specify the offset of one of the Chpxs within this ChpxFkp. The offset is computed by multiplying the value of the byte by 2.

For each *i* from 0 to **crun**, rgb[i]×2 MUST either specify an offset, in bytes, between the beginning of the ChpxFkp and **crun**, or be equal to zero, which specifies that there is no Chpx associated with this element of **rgb**.

Copyright © 2025 Microsoft Corporation

Each Chpx specifies the character properties for the run of text that is indicated by the corresponding element of **rgfc**.

crun (1 byte): An unsigned integer that specifies the number of runs of text this ChpxFkp describes. **Crun** is the last byte of the ChpxFkp. **Crun** MUST be at least 0x01, and MUST NOT exceed 0x65, as that would cause **rgfc** and **rgb** to grow too large for the ChpxFkp to be 512 bytes.

2.9.34 Cid

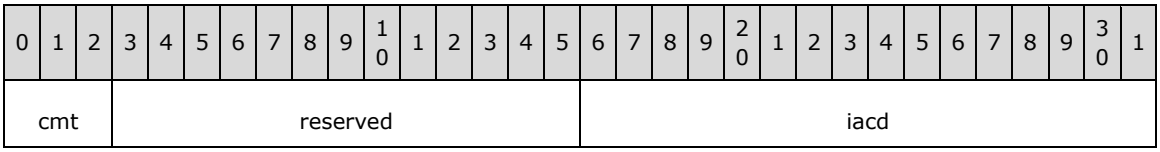
The **Cid** structure is a command identifier—a 4-byte structure that specifies a command. This element is used in other structures to identify a particular command to be executed.

The 3 least significant bits of the first byte of the structure together form a **Cmt** value which specifies the command type; the whole structure MUST be interpreted according to this command type, as follows.

Value	Meaning
cmtFci	This structure is a CidFci.
cmtMacro	This structure is a CidMacro.
cmtAllocated	This structure is a CidAllocated.
cmtNil	Specifies that the command identifier is empty and does not specify a command. If the first 3 bits of this command identifier are cmtNil, the value of the entire command identifier MUST be 0xFFFFFFFF.

2.9.35 CidAllocated

The **CidAllocated** structure specifies an **allocated command**.



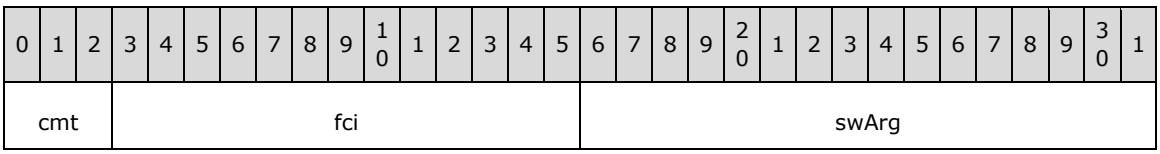
cmt (3 bits): A Cmt value that specifies the command type. This value MUST be cmtAllocated.

reserved (13 bits): This value MUST be ignored.

iacd (2 bytes): An unsigned integer that is an index of the Acd structure in PlfAcd.**rgacd** and that specifies the allocated command to be executed.

2.9.36 CidFci

The **CidFci** structure is a command identifier that specifies a built-in command.



cmt (3 bits): A Cmt value that specifies the command type. MUST be cmtFci.

fci (13 bits): An unsigned integer that specifies the command. The integer MUST be either a valid Fci value, or 0x0193. The value also MUST be one of the following:

Copyright © 2025 Microsoft Corporation

- Less than 0x049D
- Greater than or equal to 0x0FA0, and less than 0x1011
- Greater than 0x1388

  When emitting, the following special rules apply.

- If the intended command is OfficeDrawingCommand and the argument to the OfficeDrawingCommand (the value of **swArg**) is not in the intervals:
  - Greater than or equal to 0x0002, and less than 0x012C.
  - Greater than or equal to 0x1001, and less than 0x10CB.
  - Greater than or equal to 0x2001, and less than 0x20CB.
  - Greater than or equal to 0x3000, and less than 0x3011.
- Then **fci** MUST be FileAOCEAddMailer; otherwise, OfficeDrawingCommand MUST be emitted.
- If the intended command is any of the following, **fci** MUST be 0x0193 AND the intended command MUST be in swArg:
  - ToolsWordCountList
  - OutlineLevel
  - ShowLevel
- If the intended command is ToolsFixHHC then **fci** MUST be MenuFormatBackground AND swArg MUST be ToolsFixHHC.
- If the intended command is any of the following, **fci** MUST be ToolsTranslateChinese AND the intended command MUST be in swArg.
  - FileNewContext
  - LineSpacing
  - AcceptChangesSelected
  - RejectChangesSelected
  - InsertNewComment
- If the intended command is not one of the following:
  - ToolsWordCountList
  - OutlineLevel
  - ShowLevel
  - OfficeDrawingCommand
  - FileNewContext
  - LineSpacing
  - AcceptChangesSelected
  - RejectChangesSelected

Copyright © 2025 Microsoft Corporation

- InsertNewComment
- ToolsFixHHC

AND the intended command is a valid Fci value AND it is NOT one of the following:

- Less than 0x049D.
- Greater than or equal to 0x0FA0 and less than 0x1011.
- Greater than 0x1388.

  Then, fci MUST be Bold.

  The following special meaning applies:

- If the value of fci is FileAOCEAddMailer and the value of swArg is not 0, the CidFci SHOULD<206> have the same meaning as if fci were OfficeDrawingCommand.
- If the value of **fci** is either 0x0193, MenuFormatBackground, ToolsTranslateChinese, or Bold, and the value of **swArg** is a valid Fci value that is not allowed in **fci**, the CidFci SHOULD<207> have the same meaning as if **fci** was the Fci specified in **swArg** and the value of **swArg** is 0.

swArg (2 bytes): Depends on the value of **fci** as follows:

- If the value of **fci** is OfficeDrawingCommand (or FileAOCEAddMailer instead of OfficeDrawingCommand, as specified in the special rules for **fci**), then **swArg** is a MSODGCID, as specified in [MS-ODRAW] section 2.4.2, that specifies a drawing command.
- If the value of **fci** is 0x0193, then **swArg** is an Fci value that specifies the command. It MUST be either ToolsWordCountList, OutlineLevel, or ShowLevel.
- If the value of **fci** is MenuFormatBackground, ToolsTranslateChinese, or Bold, then **swArg** MUST be either an Fci value that is allowed as specified in the special rules for **fci**, or 0, which specifies that the special rules do not apply and the command is actually what **fci** indicates.
- If the value of **fci** is FormatDrawingObject, then **swArg** is an unsigned integer that specifies which tab of the Format Object dialog is selected by default. The value of **swArg** MUST be one of the following:
  - 0x0000 – no preference.
  - 0x0046 – the tab which contains line width options.
  - 0x0047 – the tab which contains arrow options.
  - 0x0245 – the tab which contains color and line options.
  - 0x0249 – the tab which contains size options.
- If the value of **fci** is FontColor, ShadingColor, Highlight, BorderLineColor, UnderlineColor, or UnderlineStyle, then **swArg** is an unsigned integer that specifies whether a whole or partial control is needed. If valid, **swArg** MUST be one of the following:
  - 0x0000 – whole control.
  - 0x03E8 (not valid for UnderlineStyle) – only the portion that contains "Automatic" or "No Color" / "No Fill".
  - 0x03E9 (not valid for UnderlineStyle) – only the portion that contains a grid of pre-defined colors.

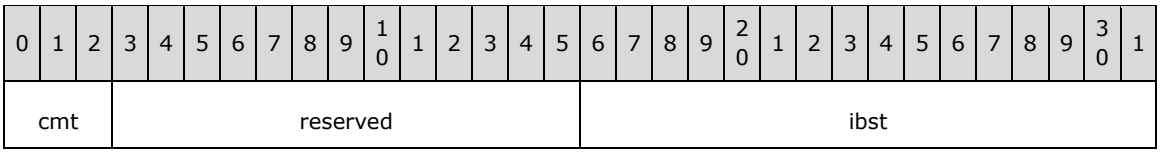
Copyright © 2025 Microsoft Corporation

- 0x03EA (not valid for Highlight) – only the portion that contains "More Colors" or "More Underlines".

- If the value of **fci** is either FixSpellingChange or SpellingAndAutoCorrect, then **swArg** is a signed integer that specifies the 0-based index of the spelling suggestion being chosen by the command. Negative values MUST be ignored.
- If the value of **fci** is FileMru, then **swArg** is an unsigned integer that specifies the 0-based index in the "Most Recently Used" list of the file to be open.
- If the value of **fci** is ToolsAutoManager, then **swArg** is an unsigned integer that specifies which variant of the Auto options dialog is needed. It MUST be one of the following:
  - 0x0000 – generic Auto options dialog (AutoCorrect, AutoFormat, and so on).
  - 0x017A – dialog geared towards editing AutoCorrect options.
  - 0x03D9 – dialog geared towards editing AutoText entries.
- If the value of **fci** is FormatObjectCore, then **swArg** is an unsigned integer that specifies whether the intention of the command is formatting the borders of the object. It MUST be either of the following:
  - 0x0000 – formatting the object.
  - 0x00BD – formatting the borders.
  - If the value of **fci** is RunToggle, then **swArg** is a signed integer that MUST be either of the following:
  - 0x0000 – toggles between right-to-left and left-to-right input.
  - Greater than 0 – specifies a 1-based index of a keyboard layout to switch to. The availability of keyboard layouts is implementation-specific.
- If the value of **fci** is FixSynonymMenu, then **swArg** MUST be ignored.
- If the value of **fci** is ToolbarLabel, then **swArg** specifies the **toolbar control identifier (TCID)** of the label. A list of possible values can be found in [MS-CTDOC] section 2.2.
- For all other values of **fci**, the value of **swArg** MUST be 0.

2.9.37 CidMacro

The **CidMacro** structure is a command identifier that specifies a command based on a macro.



cmt (3 bits): This value MUST be cmtMacro.

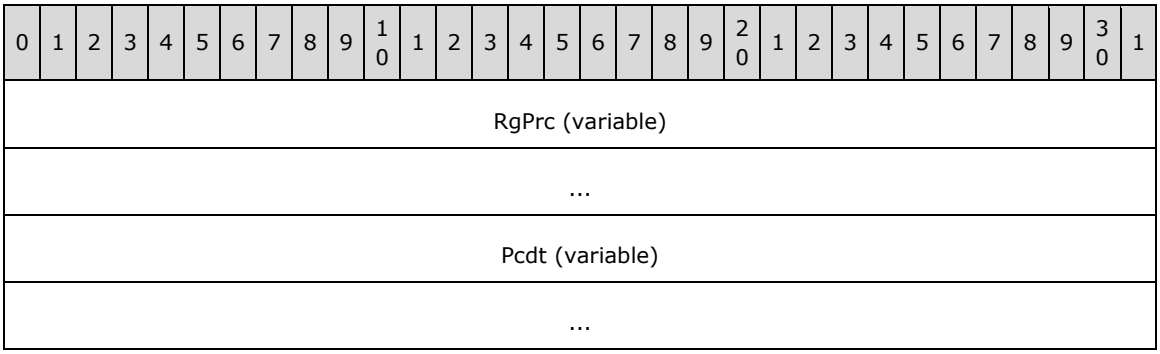
reserved (13 bits): This field MUST be ignored.

ibst (2 bytes): An unsigned integer that specifies the name of the macro to be executed. The macro name is specified by MacroName.**xstz** of the MacroName entry in the MacroNames such that MacroName.**ibst** equals **ibst**. MacroNames MUST contain such an entry.

Copyright © 2025 Microsoft Corporation

2.9.38 Clx

The **Clx** structure is an array of zero, 1, or more Prcs followed by a Pcdt.

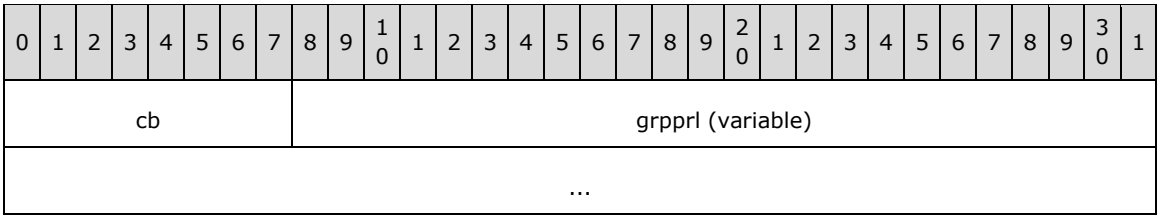


RgPrc (variable): An array of Prc. If this array is empty, the first byte of the Clx MUST be 0x02. 0x02 is invalid as the first byte of a Prc, but required for the Pcdt.

Pcdt (variable): A Pcdt.

2.9.39 CMajorityOperand

The **CMajorityOperand** structure is used by sprmCMajority to specify which character properties of the text to reset to match that of the underlying paragraph style.



cb (1 byte): An unsigned 8-bit integer that specifies the size, in bytes, of **grpprl**.

grpprl (variable): An array of Prl. Specifies character property Sprms which, when combined with default values for non-specified properties, give a set of character properties to compare against. For a specific set of properties, if the properties of the current text match those of the combined set, the value for the property is set to that of the current paragraph style (taking style hierarchy into account.) Details and exceptions are specified in sprmCMajority.

2.9.40 Cmt

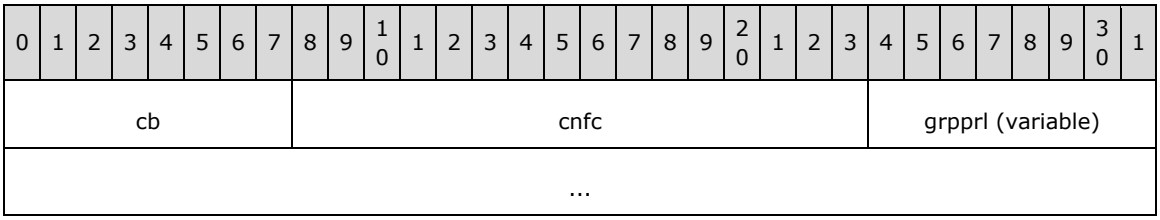
The **Cmt** enumeration provides an unsigned 3-bit integer that specifies the type of a command; see Cid for more details. The valid values are as follows.

Name	Value	Meaning
cmtFci	0x1	Command based on a built-in command. See CidFci.
cmtMacro	0x2	Macro command. See CidMacro.
cmtAllocated	0x3	**Allocated command**. See CidAllocated.
cmtNil	0x7	No command. See Cid.

Copyright © 2025 Microsoft Corporation

2.9.41 CNFOperand

The **CNFOperand** structure provides conditional formatting for a table style.



cb (1 byte): An unsigned integer that specifies the size, in bytes, of this CNFOperand, excluding the **cb** member.

cnfc (2 bytes): A signed integer that specifies the condition for which the formatting in **grpprl** applies.

Value	Meaning
0x0001	Header row.
0x0002	Footer row.
0x0004	First column.
0x0008	Last column.
0x0010	Banded columns.
0x0020	Even column banding.
0x0040	Banded rows.
0x0080	Even row banding.
0x0100	Top right cell.
0x0200	Top left cell.
0x0400	Bottom right cell.
0x0800	Bottom left cell.

The value of **cnfc** MUST be one of these values.

grpprl (variable): An array of **Prl**. Specifies the formatting to apply (on top of the non-conditional formatting specified in the table style) when the condition is satisfied (see section 2.4.6 Applying Properties).

2.9.42 CNS

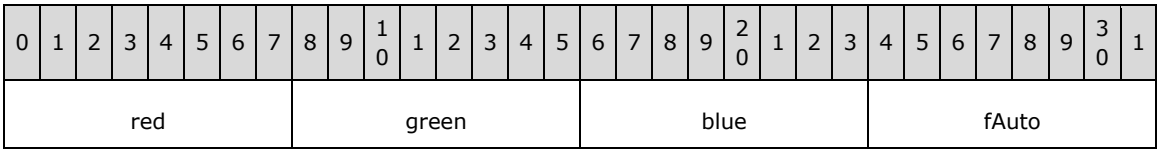
The **CNS** enumeration provides an unsigned 8-bit integer that specifies the separator character to be used between the chapter number and the page number when **chapter numbering** is enabled in page number fields.

Name	Value	Meaning
cnsHyphen	0x00	Specifies that the separator character is a hyphen ("-").
cnsPeriod	0x01	Specifies that the separator character is a period (".").
cnsColon	0x02	Specifies that the separator character is a colon (":").
cnsEmDash	0x03	Specifies that the separator character is an em dash ("—").
cnsEnDash	0x04	Specifies that the separator character is an en dash ("–").

Copyright © 2025 Microsoft Corporation

2.9.43 COLORREF

The **COLORREF** structure specifies a color in terms of its red, green, and blue components.



red (1 byte): An unsigned integer that specifies the intensity of the color red. A value of zero specifies that there is no red. Larger numbers specify a more intense red than smaller numbers.

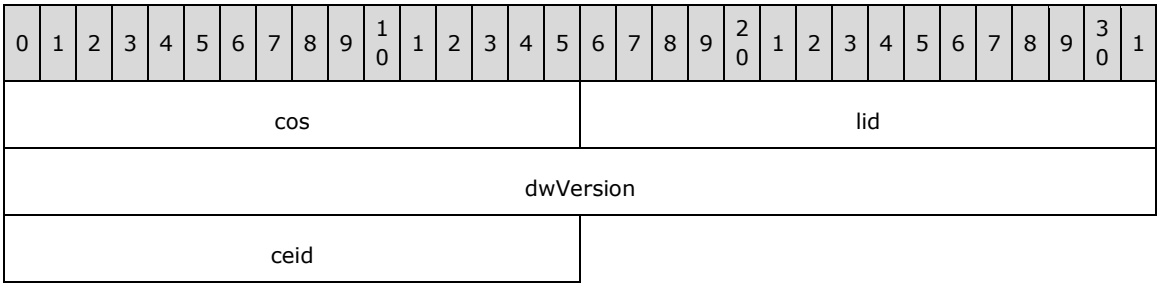
green (1 byte): An unsigned integer that specifies the intensity of the color green. A value of zero specifies that there is no green. Larger numbers specify a more intense green than smaller numbers.

blue (1 byte): An unsigned integer that specifies the intensity of the color blue. A value of zero specifies that there is no blue. Larger numbers specify a more intense blue than smaller numbers.

fAuto (1 byte): An unsigned integer whose value MUST be either 0xFF or 0x00. If the value is 0xFF, the values of red, green, and blue in this COLORREF SHOULD<208> all be 0x00. If fAuto is 0xFF, this COLORREF designates the default color for the application. An application MAY<209> use different default colors based on context. This documentation refers to the COLORREF with fAuto set to 0xFF as cvAuto.

2.9.44 COSL

The **COSL** structure specifies the option set to use for a grammar checker implementing the **NLCheck** interface, as well as information to identify the corresponding grammar checker.



cos (2 bytes): An unsigned integer that specifies a NLCheck option set, which is implementation-specific to the grammar checker that is identified by **lid**, **dwVersion**, and **ceid**.

The **cos** values for English, Spanish, French, German and Japanese MUST be one of the following values.

Language	Value	Meaning
English	0x0000	Grammar & Style
English	0x0001	Grammar
Spanish	0x0000	Grammar & Style
Spanish	0x0001	Grammar
French	0x0000	Grammar & Style
French	0x0001	Grammar
German	0x0000	User-defined
German	0x0001	Grammar
Japanese	0x0000	Casual Style

Copyright © 2025 Microsoft Corporation

Language	Value	Meaning
Japanese	0x0001	Normal Style
Japanese	0x0002	Normal Style (editorial)
Japanese	0x0003	Official Style (editorial)
Japanese	0x0004	User-defined 1
Japanese	0x0005	User-defined 2
Japanese	0x0006	User-defined 3

By default, the value is 0x0001.

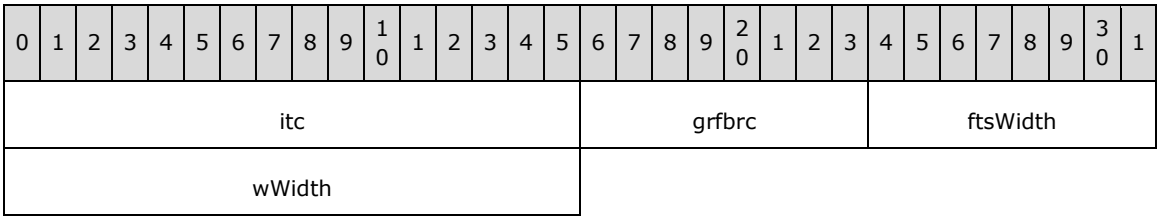
lid (2 bytes): A LID that specifies the language of the associated grammar checker.

dwVersion (4 bytes): An unsigned integer value that is the version number of the associated grammar checker, as specified through NLCheck.

ceid (2 bytes): An unsigned integer value that is the company identifier of the associated grammar checker, as specified through NLCheck.

2.9.45 CSSA

The **CSSA** structure specifies a cell spacing SPRM argument used by many Table SPRMs to define table **cell** margins and **cell spacing**.



itc (2 bytes): An ItcFirstLim that specifies which cells this CSSA structure applies to.

grfbrc (1 byte): A bit field that specifies which cell sides this cell margin or cell spacing applies to. The bit values and their meanings are as follows.

Name	Bit Mask	Meaning
fbrcTop	0x01	Specifies the top side.
fbrcLeft	0x02	Specifies the left side.
fbrcBottom	0x04	Specifies the bottom side.
fbrcRight	0x08	Specifies the right side.

Setting all four side bits results in fBrcSidesOnly (0x0F). All other bits MUST be 0.

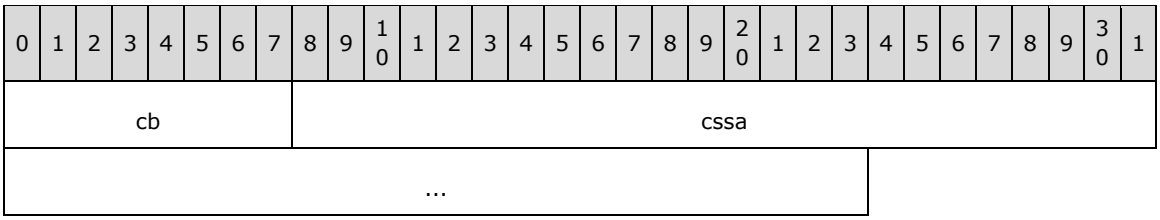
ftsWidth (1 byte): An Fts that specifies how **wWidth** is defined.

wWidth (2 bytes): An unsigned integer value that specifies the cell margin or cell spacing that is applied to cells **itc.itcFirst** through **itc.itcLim** – 1. The interpretation of this value depends on the value of **ftsWidth**. If **ftsWidth** is ftsNil (0x00), then **wWidth** MUST be zero.

2.9.46 CSSAOperand

The **CSSAOperand** structure is an operand that is used by several Table SPRMs to specify a table **cell margin** or **cell spacing**.

Copyright © 2025 Microsoft Corporation

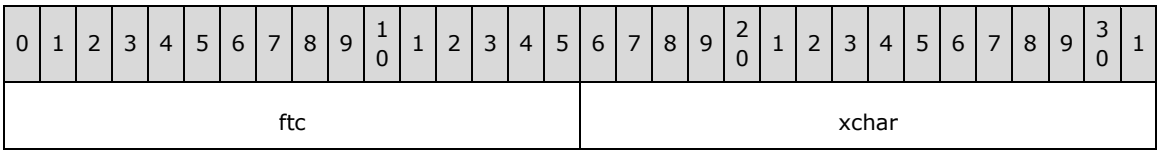


cb (1 byte): An unsigned integer value that specifies the size of this operand in bytes, not including **cb**. The **cb** MUST be 6.

cssa (6 bytes): A CSSA that specifies the cell margin or cell spacing to apply.

2.9.47 CSymbolOperand

The **CSymbolOperand** structure specifies the properties of a symbol character.

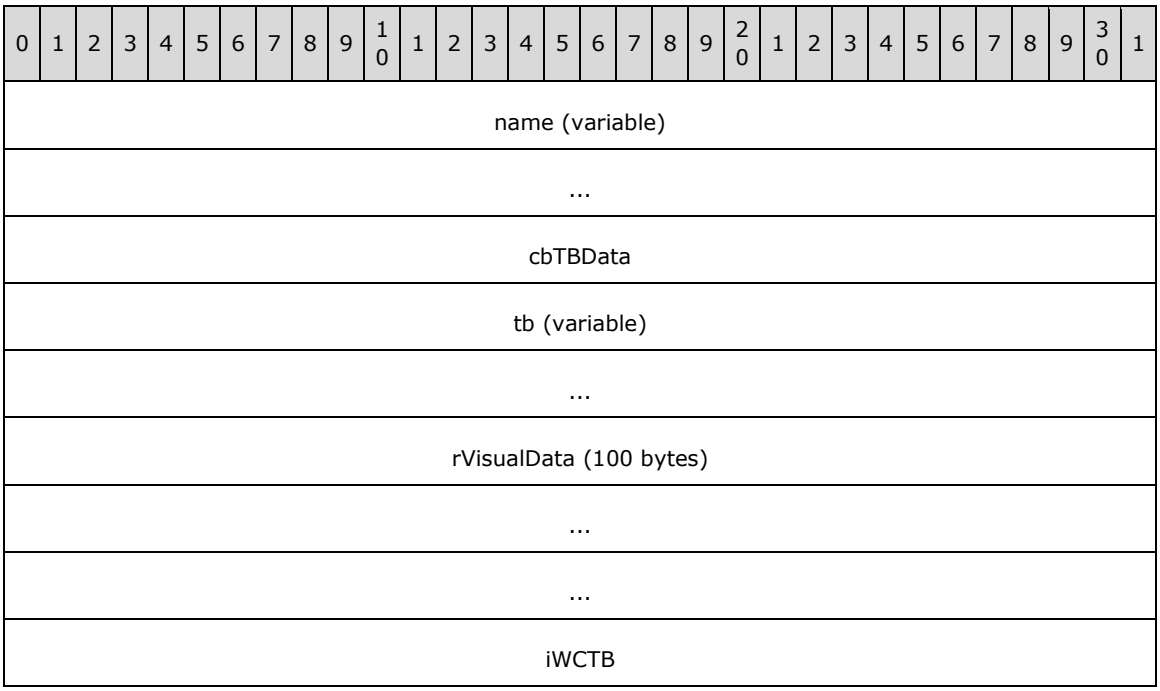


ftc (2 bytes): A 16-bit unsigned integer that is an index into the font table SttbfFfn and that specifies the font for this symbol.

xchar (2 bytes): A 16-bit unsigned integer that specifies the **Unicode** character code of the specified font.

2.9.48 CTB

The **CTB** structure specifies a **custom toolbar**.



Copyright © 2025 Microsoft Corporation

reserved	unused
cCtls	
rTBC (variable)	
...	

name (variable): A structure of type **Xst** that specifies the name of this custom toolbar.

cbTBData (4 bytes): A signed integer value that specifies the size, in bytes, of this structure excluding the **name**, **cCtls**, and **rTBC** fields. The value is given by the following formula.

cbTBData = sizeof(**tb**) + sizeof(**rVisualData**) + 12

tb (variable): A structure of type **TB**, as specified in [MS-OSHARED]. This structure contains toolbar data.

rVisualData (100 bytes): A zero-based index array of **TBVisualData**, as specified in [MS-OSHARED] structures. The number of elements in this array MUST be 5. The index of each structure in the array corresponds to a Word view number. Refer to the following table for the meaning of each TBVisualData, as defined in [MS-OSHARED] structures, according to its position in this array.

Array index of structure	Meaning of TBVisualData
0	Contains the visual information for this toolbar to be used when the application is in **Normal view**.
1	Contains the visual information for this toolbar to be used when the application is in **Print Preview view**.
2	Contains the visual information for this toolbar to be used when the application is in **full screen view**.
3	Contains the visual information for this toolbar to be used when the application is in both Print Preview view and full screen view.
4	Contains the visual information for this toolbar to be used when the application is in **Hyperlink view**<210>.

iWCTB (4 bytes): A signed integer that specifies the zero-based index of the Customization structure that contains this structure in the **rCustomizations** array that contains the Customization structure that contains this structure. The value MUST be greater or equal to 0x00000000 and MUST be less than the value of the **cCust** field of the CTBWRAPPER structure that contains the **rCustomizations** array that contains the Customization structure that contains this structure.

reserved (2 bytes): This MUST be 0x0000 and MUST be ignored.

unused (2 bytes): This is undefined and MUST be ignored.

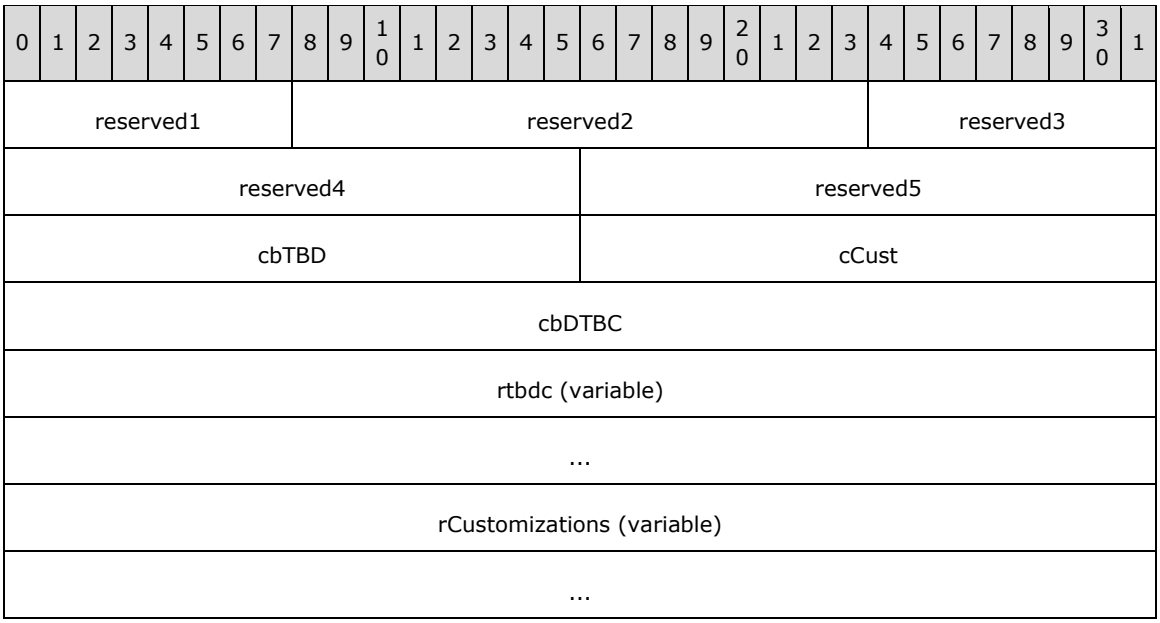
cCtls (4 bytes): A signed integer that specifies the number of **toolbar controls** in this toolbar.

rTBC (variable): A zero-based index array of TBC structures. The number of elements in this array MUST equal **cCtls**.

Copyright © 2025 Microsoft Corporation

2.9.49 CTBWRAPPER

The **CTBWRAPPER** structure is a **custom toolbar** wrapper. This structure contains the custom toolbars and **toolbar deltas** that are saved to the file.



reserved1 (1 byte): This value MUST be 0x12.

reserved2 (2 bytes): This value MUST be 0x0000.

reserved3 (1 byte): This value MUST be 0x07.

reserved4 (2 bytes): This value MUST be 0x0006.

reserved5 (2 bytes): This value MUST be 0x000C.

cbTBD (2 bytes): A signed integer that specifies the size, in bytes, of a TBDelta structure. This value MUST be 0x0012.

cCust (2 bytes): A signed integer that specifies the number of elements in the **rCustomizations** array. This value MUST be greater than 0x0000.

cbDTBC (4 bytes): A signed integer that specifies the size, in bytes, of the **rtbdc** array. This value MUST be greater or equal to 0x00000000.

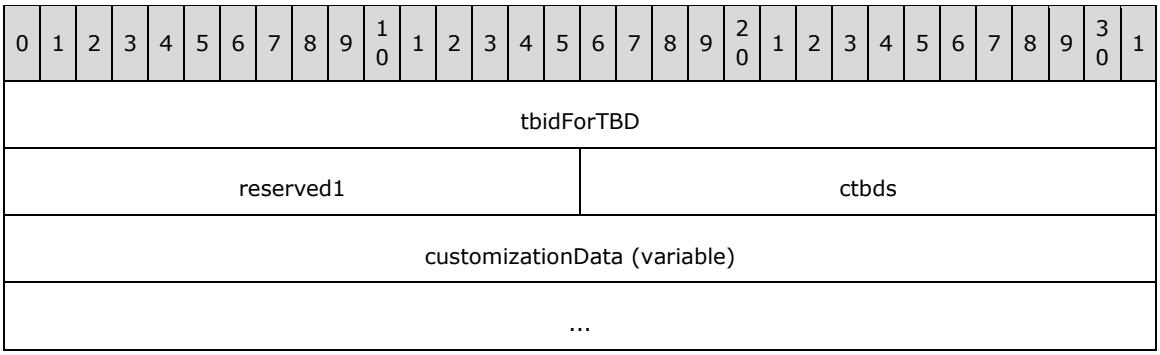
rtbdc (variable): An array of TBC structures. The total size of this array, in bytes, MUST be equal to the value of **cbDTBC**. The TBC structures in this array specify **toolbar controls** that are associated with TBDelta structures.

rCustomizations (variable): A zero-based index array of Customization structures. The number of elements MUST be equal to **cCust**.

2.9.50 Customization

The **Customization** structure specifies either a **custom toolbar** or **toolbar delta** values.

Copyright © 2025 Microsoft Corporation



tbidForTBD (4 bytes): A signed integer that specifies if **customizationData** contains a CTB structure or an array of TBDelta structures. This value MUST be greater than or equal to 0x00000000. If this value equals 0x00000000, **customizationData** MUST contain a CTB structure. If this value does not equal 0x00000000, **customizationData** MUST contain an array of TBDelta structures and the value of this field specifies the **toolbar** identifier of the toolbar affected by the TBDelta structures contained in the array.

reserved1 (2 bytes): This MUST be 0x0000 and MUST be ignored.

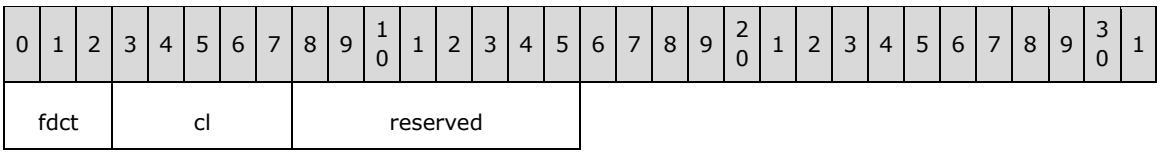
ctbds (2 bytes): A signed integer that specifies, if **tbidForTBD** is not equal to 0x00000000, the number of TBDelta structures that are contained in the **customizationData** array. This MUST be 0x0000 if **tbidForTBD** equals 0x00000000.

customizationData (variable): The type of this structure depends on the value of **tbidForTBD**. The types of this structure are shown following.

Value of tbidForTBD	Type of customizationData
0x00000000	CTB
not 0x00000000	A zero-based index array of TBDelta structures. The number of elements in the array MUST be equal to **ctbds**.

2.9.51 DCS

The **DCS** structure specifies the drop cap properties for a paragraph.



fdct (3 bits): An integer that specifies the drop cap type. This MUST be one of the following values.

Value	Meaning
1	Regular drop cap, which is a single letter beginning at the leading edge of the paragraph.
2	A drop cap which is in the margin of the page, outside of the paragraph.

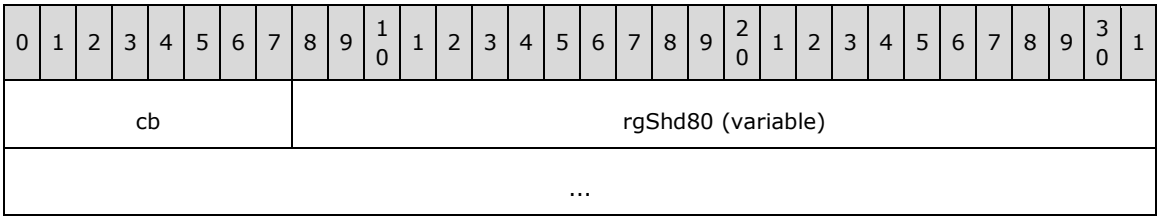
cl (5 bits): An unsigned integer that specifies the number of lines to drop. This determines the size of the drop cap letter. The value MUST be between 1 and 10, inclusive.

Copyright © 2025 Microsoft Corporation

reserved (8 bits): Undefined and MUST be ignored.

2.9.52 DefTableShd80Operand

The **DefTableShd80Operand** structure is an operand that is used by several Table Sprms to specify each style of background shading that is applied to each of the cells in a single row.

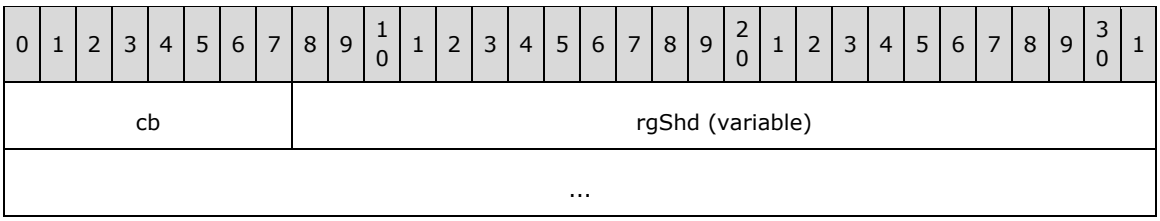


cb (1 byte): An unsigned integer that specifies the size in bytes of this operand, not including **cb**. **cb** MUST be a multiple of 2 (the size of Shd80).

rgShd80 (variable): An array of Shd80. The number of elements is equal to **cb** divided by 2 and MUST NOT exceed the number of cells in the row. Each Shd80 structure is applied sequentially to each cell in the row, beginning with the first cell.

2.9.53 DefTableShdOperand

The **DefTableShdOperand** structure is an operand that is used by several Table Sprms to specify each style of background shading that is applied to each of the cells in a single row.

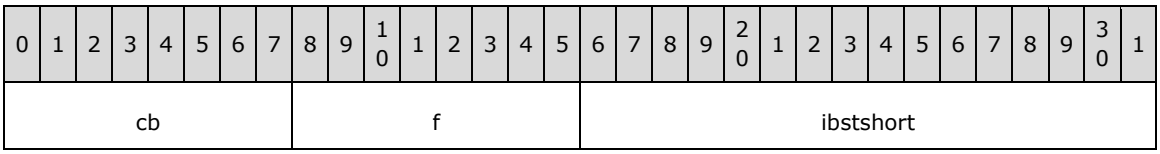


cb (1 byte): An unsigned integer that specifies the size in bytes of this operand, not including **cb**. The **cb** value MUST be a multiple of 10, the size of Shd, and MUST NOT exceed 220.

rgShd (variable): An array of Shd. The number of elements is equal to **cb** / 10 and MUST NOT exceed 22. Each Shd structure is applied sequentially to each cell in the row. The first cell **rgShd** applies to is either 1, 23, or 45, depending on which Table Sprm is applying this operand. **rgShd** only contains elements necessary to define all shaded cells in the row. Non-shaded cells that follow the last shaded cell in the row are omitted from the array. Non-shaded cells that precede the last shaded cell in the row are set to ShdAuto or ShdNil, depending on which Table Sprm is applying this operand.

2.9.54 DispFldRmOperand

The **DispFldRmOperand** structure is an operand that is used by **sprmCDispFldRMark** and specifies whether the result of a LISTNUM display field contains a revision.



Copyright © 2025 Microsoft Corporation

dttm
xst (32 bytes)
...
...

cb (1 byte): An unsigned integer that specifies the size, in bytes, of the remainder of this structure. This value MUST be 39.

f (1 byte): An unsigned integer that specifies whether there is a revision in the result of this LISTNUM display field. Any nonzero value specifies that there is a revision. A value of zero specifies that there are no revisions in the result of this field.

ibstshort (2 bytes): An unsigned integer that specifies the index into SttbfRMark. The value in the string table at index **ibstshort** specifies the author who made this revision.

dttm (4 bytes): A DTTM that specifies the time of the revision.

xst (32 bytes): A 15-character XST that specifies the previous result of this LISTNUM display field.

2.9.55 Dofr

The **Dofr** structure is a type that wraps a different data type for each type of record specified by **Dofrh.dofrt**. When **Dofrh.dofrt** specifies dofrtFs, this type is not applicable, and MUST be left out.

Value	Meaning
dofrtFsn	Contains a **DofrFsn**.
dofrtFsnp	Contains a **DofrFsnp**.
dofrtFsnName	Contains a **DofrFsnName**.
dofrtFsnFnm	Contains a **DofrFsnFnm**.
dofrtFsnSpbd	Contains a **DofrFsnSpbd**.
dofrtRglstsf	Contains a **DofrRglstsf**.

2.9.56 DofrFsn

The **DofrFsn** structure specifies the properties of a frame. There can be multiple **DofrFsn** records for a particular frame. If **fsnk** is **fsnkFrame**, this record introduces a new frame. Otherwise this record applies to the frame that is associated with the previous **DofrFsn** with **fsnk** equal to **fsnkFrame**, unless it appears before the first **DofrFsn** with **fsnk** equal to **fsnkFrame**. In that case, this record applies to the outermost frame.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fssd																															
...																															

Copyright © 2025 Microsoft Corporation

tCols		
fsnk		
dxMargin		
dyMargin		
iidsScroll		
A	B	fUnused1
fUnused2		

fssd (8 bytes): An **Fssd** that specifies the position of the divider. If **fsnk** is not **fsnkFrame**, this value MUST be ignored.

tCols (4 bytes): A signed integer value that specifies whether the child frames are displayed horizontally or vertically This field MUST contain one of the following values.

Value	Meaning
0xFFFFFFFF	No child frames
0x00000000	Arrange child frames into rows
0x00000001	Arrange child frames into columns

fsnk (4 bytes): A **Fsnk** that specifies the type of **DofrFsn** that contains this field.

dxMargin (4 bytes): A signed integer that specifies the left and right margins, in pixels, for this frame.

dyMargin (4 bytes): A signed integer that specifies the top and bottom margins, in pixels, for this frame.

iidsScroll (4 bytes): An **IScrollType** that specifies the scroll bar behavior for this frame.

A - fLinked (1 bit): Specifies whether the frame is linked to an external file.

B - fNoResize (1 bit): Specifies whether the size of the frame is locked and cannot be changed.

fUnused1 (30 bits): This value is undefined and MUST be ignored.

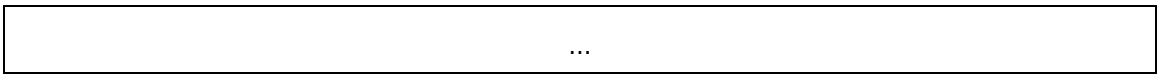
fUnused2 (32 bits): This value is undefined and MUST be ignored.

2.9.57 DofrFsnFnm

The **DofrFsnFnm** structure is an **Xstz** that specifies the file name of the file that is loaded into the frame. **DofrFsnFnm** applies to the frame that is associated with the most recently read **DofrFsn** record.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
xstzFilename (variable)																															

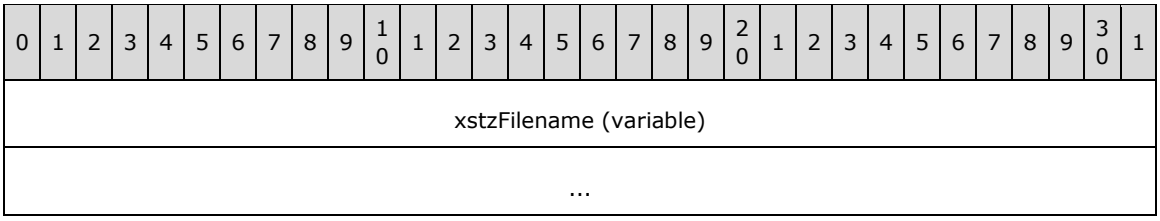
Copyright © 2025 Microsoft Corporation



xstzFilename (variable): An **Xstz** that specifies the file name and path of the frame. The string MUST be between 0 and 258 characters in length.

2.9.58 DofrFsnName

The **DofrFsnName** structure is a type that specifies the name of the frame. **DofrFsnName** applies to the frame that is associated with the most recently read **DofrFsn** record.



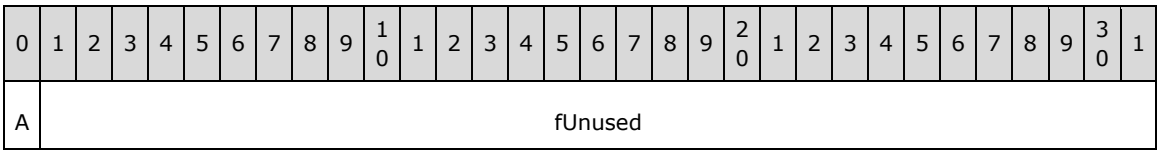
xstzFilename (variable): An **Xstz** that specifies the name of the frame. The name MUST be between 0 and 255 characters in length.

2.9.59 DofrFsnp

The **DofrFsnp** structure marks the beginning or end of a group of child frames. In the first marker, **fPush** is set to "true"; in the ending marker, **fPush** is set to "false". The enclosed child frames belong to the frame associated with the record that appears immediately before the **DofrFsnp**, with **fPush** set to "true".

DofrFsnp records can be nested. While loading the child nodes of frame A there appears another **DofrFsnp** with **fPush** set to "true". This means that the most recently loaded child record B does have child nodes. All the nodes between that **DofrFsnp** and the corresponding **DofrFsnp** with **fPush** set to "false" are the child nodes of frame B. This is how frame records support an arbitrary level of nesting within the frame set.

DofrFsnp records MUST be equally matched. There MUST be as many records with **fPush** set to "false" as there are records with **fPush** set to "true".



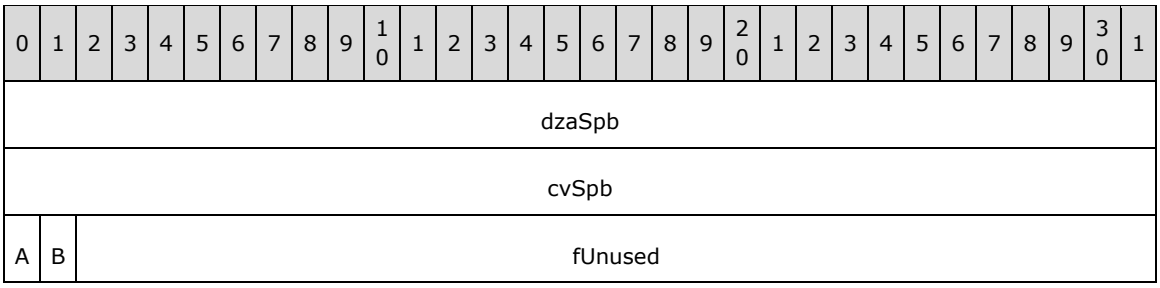
A - fPush (1 bit): Specifies if this marker indicates the beginning or end of a group of frames. A value of 1 specifies the beginning of a set of child frames. A value of 0 specifies the end of the child frames.

fUnused (31 bits): This value is unused and MUST be ignored.

2.9.60 DofrFsnSpbd

The **DofrFsnSpbd** structure specifies borders and divider (splitter bar) properties for the entire frame set.

Copyright © 2025 Microsoft Corporation



dzaSpb (4 bytes): A signed integer that specifies the width, in twips, of the borders and dividers. This value MUST be between 0 and 31,680. If this value is 0, the default border size is used.

cvSpb (4 bytes): A **COLORREF** that specifies the color of the borders and dividers.

A - fNoBorder (1 bit): Specifies whether the frame set has visible borders. If this value is zero, it displays borders. If this value is 1, it does not.

B - f3DBorder (1 bit): Specifies whether the frame set border uses a raised style.

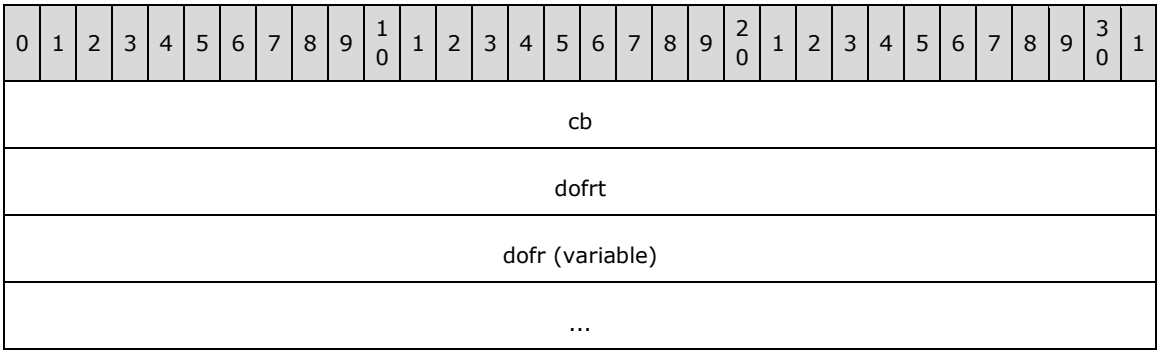
fUnused (30 bits): This value MUST be 0 and MUST be ignored.

2.9.61 Dofrh

The **Dofrh** structure is the general record header that wraps each record type specified in the section **Dofr**. Every record begins with this header.

Records that specify a frame set MUST begin with a record containing a **dofrt** equal to **dofrtFs**, followed by any number of records of other types, according to the rules defined in the section for each record type. Each frame MUST have one or more records that specify the attributes of the frame.

Similarly, an array of list specifications MUST begin with a record containing a **dofrt** equal to **dofrtRglstsf**, followed by any number of list records.



cb (4 bytes): An unsigned integer that specifies the size of the **Dofrh**, including all contained variable or optional data such as the **dofr**.

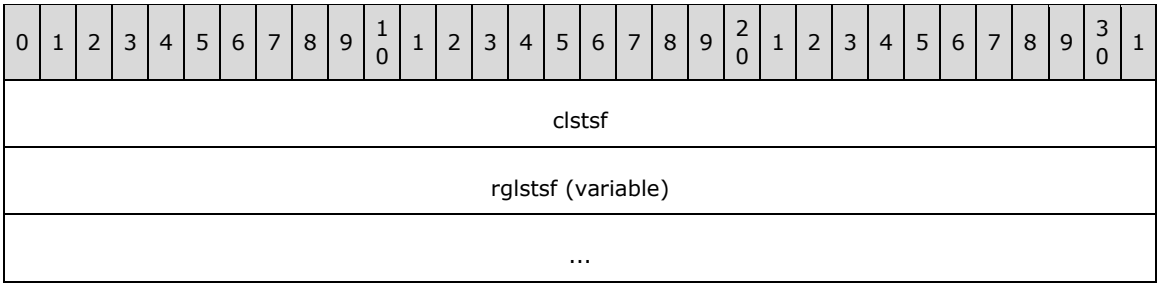
dofrt (4 bytes): A **Dofrt** that specifies the type of data contained in **dofr**.

dofr (variable): A **Dofr** that contains data for each record type. If **dofrt** is **dofrtFs**, this field MUST NOT exist. For all other records, this field MUST exist.

2.9.62 DofrRglstsf

The **DofrRglstsf** structure specifies the list styles that are used in the document.

Copyright © 2025 Microsoft Corporation



clstsf (4 bytes): A signed integer that specifies the count of the items in **rglstsf**.

rglstsf (variable): An array of **Lstsf** that specifies the list styles used in the document.

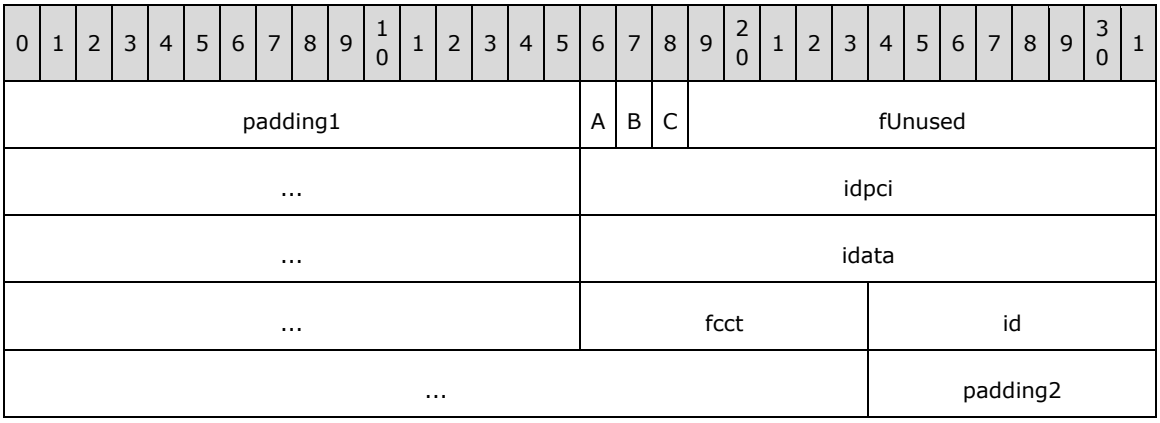
2.9.63 Dofrt

The **Dofrt** enumeration provides a 32-bit unsigned integer that specifies the type of record contained in a **Dofrh**. A field of this type MUST contain one of the following values.

Name	Value	Meaning
dofrtFs	0x00000000	Frame set root record.
dofrtFsn	0x00000001	Frame record.
dofrtFsnp	0x00000002	Frame child marker.
dofrtFsnName	0x00000003	Frame name.
dofrtFsnFnm	0x00000004	Frame file path.
dofrtFsnSpbd	0x00000005	Frame border attributes.
dofrtRglstsf	0x00000006	An array of list styles used in the document.

2.9.64 DPCID

The **DPCID** structure contains information about a **format consistency-checker bookmark** in the document.



padding1 (2 bytes): Two bytes that are used for padding. This MUST be ignored.

Copyright © 2025 Microsoft Corporation

A - fSquiggle (1 bit): A bit flag that specifies whether an application is expected to display a squiggle under the region of text denoted by the **bookmark** associated with this DPCID. If the region of text is inside the Main Document Part, **fSquiggle** MUST be 0.

B - fIgnored (1 bit): A bit flag that specifies whether the user requested that the flagging of the region of text by the **format consistency checker** that is denoted by the bookmark associated with this DPCID be ignored. If the region of text is inside the Main Document Part, **fIgnored** MUST be 1.

C - fSquiggleChanged (1 bit): A bit flag that specifies whether the squiggle under the region of text denoted by the bookmark associated with this DPCID has recently been changed. If the region of text is inside the Main Document Part, **fSquiggleChanged** MUST be 1.

fUnused (29 bits): This value MUST be 0 and MUST be ignored.

idpci (4 bytes): An IDPCI that specifies the kind of formatting that the format consistency checker flagged, within the range of text that is covered by the format consistency-checker bookmark associated with this DPCID. If the range of text is inside the Main Document Part, **idpci** MUST be idpciFmt, idpciPapc, or idpciLvl.

idata (4 bytes): This value is undefined and MUST be ignored.

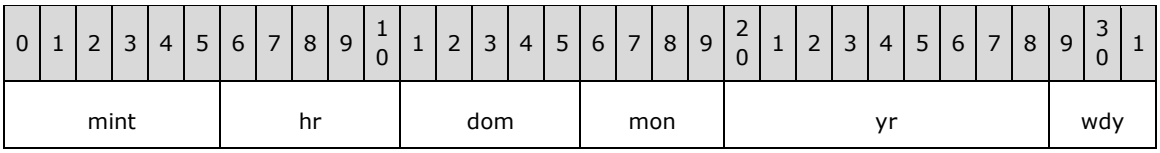
fcct (1 byte): An FCCT that contains further information about the format consistency-checker bookmark associated with this DPCID.

id (4 bytes): An unsigned integer that specifies a unique value used to reference the format consistency-checker bookmark associated with this DPCID. This value MUST be unique for all DPCIDs inside a given SttbfBkmkFcc.

padding2 (1 byte): This value is undefined and MUST be ignored.

2.9.65 DTTM

The **DTTM** structure specifies date and time.



mint (6 bits): An unsigned integer that specifies the minute. This value MUST be less than or equal to 0x3B.

hr (5 bits): An unsigned integer that specifies the hour. This value MUST be less than or equal to 0x17.

dom (5 bits): An unsigned integer that specifies the day of the month. This value MUST be less than or equal to 0x1F. If this value is equal to zero, this DTTM MUST be ignored.

mon (4 bits): An unsigned integer that specifies the month. The values 0x1 through 0xC specify the months January through December, respectively. This value MUST be less than or equal to 0xC. If this value is equal to zero, this DTTM MUST be ignored.

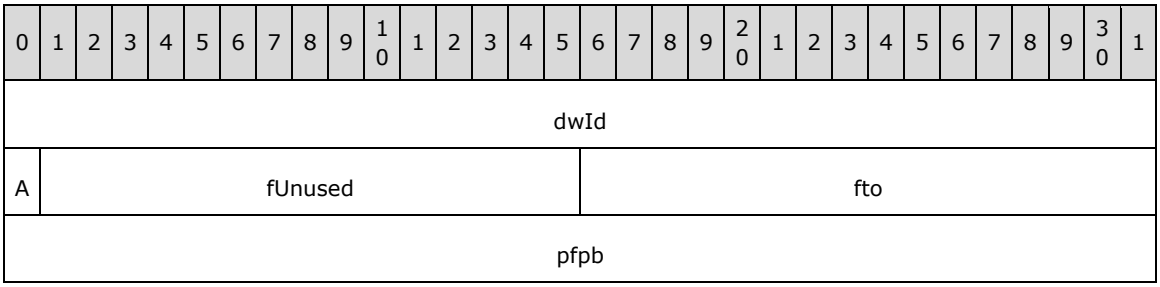
yr (9 bits): An unsigned integer that specifies the year, offset from 1900.

wdy (3 bits): An unsigned integer that specifies the day of the week, starting from Sunday (0x0). This value MUST be less than or equal to 0x6.

Copyright © 2025 Microsoft Corporation

2.9.66 FACTOIDINFO

The **FACTOIDINFO** structure contains information about a **smart tag bookmark** in the document.



dwId (4 bytes): An unsigned integer that specifies a unique value this is used to reference the smart tag bookmark associated with this **FACTOIDINFO**. This MUST be unique for all **FACTOIDINFO** structures in all Document Parts.

A - fSubEntity (1 bit): 1 specifies it is a sub-entity of a larger smart tag from the grammar checker.

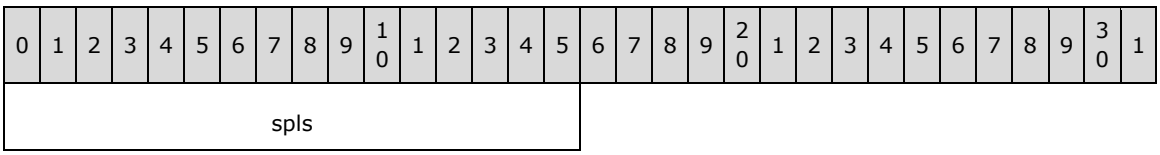
fUnused (15 bits): This field MUST be ignored.

fto (2 bytes): An **FTO** specifying further information about the smart tag bookmark that is associated with this **FACTOIDINFO**.

pfpb (4 bytes): This field MUST be ignored.

2.9.67 FactoidSpls

The **FactoidSpls** structure is an **SPLS** structure that specifies the state of the **smart tag recognizer** over a range of text. Some states that are possible in a generic **SPLS** are not allowed in a FactoidSpls structure.



spls (2 bytes): An SPLS structure.

The **spls.fError**, **spls.fExtend**, and **spls.fTypo** fields are not used and MUST be zero.

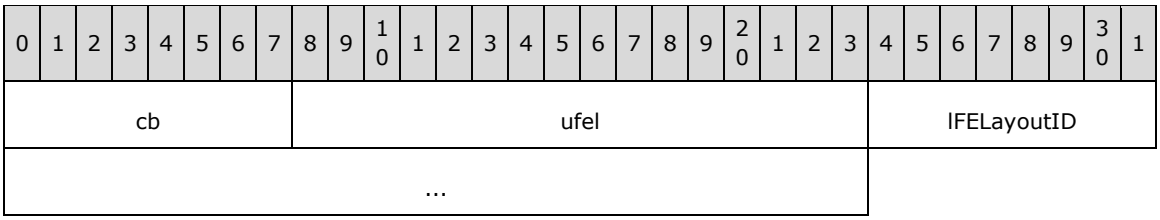
The **spls.splf** field MUST be one of the following:

- splfPending
- splfMaybeDirty
- splfDirty
- splfEdit
- splfClean

2.9.68 FarEastLayoutOperand

The **FarEastLayoutOperand** structure specifies layout information for text in **East Asian languages**, as well as the text that is considered part of the same layout unit.

Copyright © 2025 Microsoft Corporation



cb (1 byte): The size of this structure, in bytes, not including this byte. **cb** MUST be 0x06.

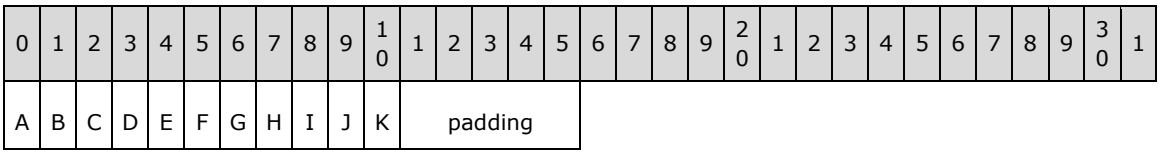
ufel (2 bytes): A UFEL that specifies the layout information.

lFELayoutID (4 bytes): An integer that specifies whether the corresponding text is in the same layout unit as other text. If two adjacent text runs have the same **lFELayoutID** value applied to them, they are laid out together.

2.9.69 FatI

The **FatI** structure is a bit field that SHOULD<211> specify which optional formats from a **table style** or table auto-format are enabled.

Not all formatting categories are available for every table style or table auto-format.



A - fatIBorders (1 bit): This bit MAY<212> specify that the border formats of a table auto-format were applied by the last table auto-format.

B - fatIShading (1 bit): This bit MAY<213> specify that the background shading formats of a table auto-format were applied by the last table auto-format.

C - fatIFont (1 bit): This bit MAY<214> specify that the text font formats of a table auto-format were applied by the last table auto-format.

D - fatIColor (1 bit): This bit MAY<215> specify that a color variant of a table auto-format was applied by the last table auto-format. When this bit is not set, the monochrome variant was applied.

E - fatIBestFit (1 bit): This bit MAY<216> specify that the columns of the table were resized to best fit their contents during the last table auto-format.

F - fatIHdrRows (1 bit): This bit SHOULD<217> specify that the top row of the table receives special formatting.

G - fatILastRow (1 bit): This bit SHOULD<218> specify that the bottom row of the table receives special formatting.

H - fatIHdrCols (1 bit): This bit SHOULD<219> specify that the logically leftmost column receives special formatting.

I - fatILastCol (1 bit): This bit SHOULD<220> specify that the logically rightmost column receives special formatting.

J - fatINoRowBands (1 bit): This bit SHOULD<221> specify that odd numbered rows do not receive different formatting than even numbered rows.

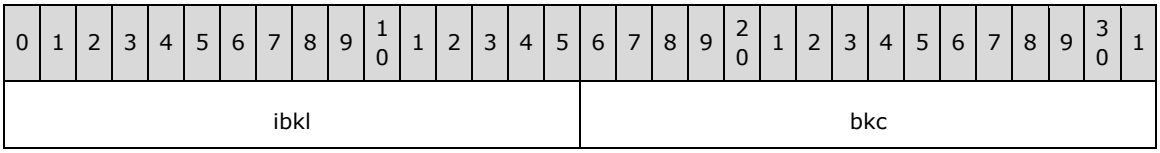
Copyright © 2025 Microsoft Corporation

K - fatlNoColBands (1 bit): This bit SHOULD<222> specify that odd numbered columns do not receive different formatting than even numbered columns.

padding (5 bits): This MUST be zero and MUST be ignored.

2.9.70 FBKF

The **FBKF** structure contains information about a **bookmark**.

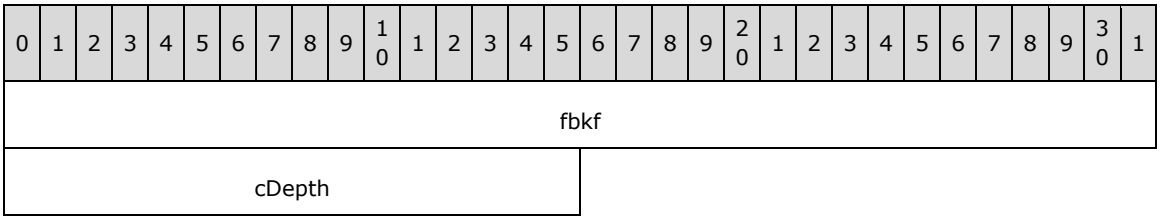


ibkl (2 bytes): An unsigned integer that specifies a zero-based index into the Plcfbkl or Plcfbkld that is paired with the Plcfbkf or Plcfbkfd containing this FBKF. The entry that is found at such an index specifies the location of the end of the bookmark associated with this FBKF. **Ibkl** MUST be unique for all FBKFs inside a given Plcfbkf or Plcfbkfd.

bkc (2 bytes): A BKC that specifies further information about the bookmark associated with this FBKF.

2.9.71 FBKFD

The **FBKFD** structure contains information about a **bookmark**.



fbkf (4 bytes): An **FBKF** specifying further information about the bookmark.

cDepth (2 bytes): An integer value that specifies the number of bookmarks in the document of the same type as the bookmark associated with this **FBKFD**, the ranges of which overlap the beginning of the range of this bookmark. To increment the count, a bookmark MUST meet the following constraints:

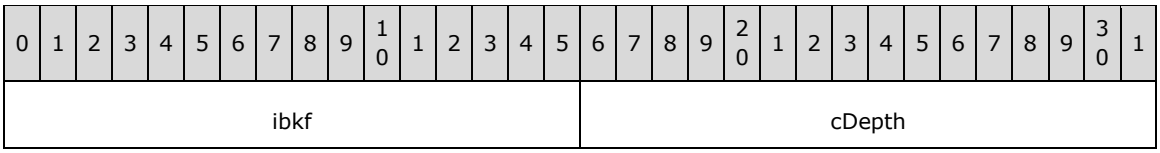
- The **FBKFD** of the bookmark occupies the Plcfbkfd containing this **FBKLD**.
- The starting CP (cpS) and limit CP (cpL) of the bookmark, as defined in the specification of that Plcfbkfd and the Plcfbkld it is paired with, satisfy the following in relation to the CP (cpCur) that marks the beginning of the bookmark of this **FBKFD**.

```
cpS == cpCur == cpL || cpS <= cpCur < cpL
```

2.9.72 FBKLD

The **FBKLD** structure contains information about a **bookmark**.

Copyright © 2025 Microsoft Corporation



ibkf (2 bytes): An unsigned integer that specifies a zero-based index into the Plcfbkfd that is paired with the Plcfbkld containing this FBKLD. The entry that is found at the index specifies the location of the start of the bookmark. **Ibkf** MUST be unique for all FBKLDs in a given Plcfbkld.

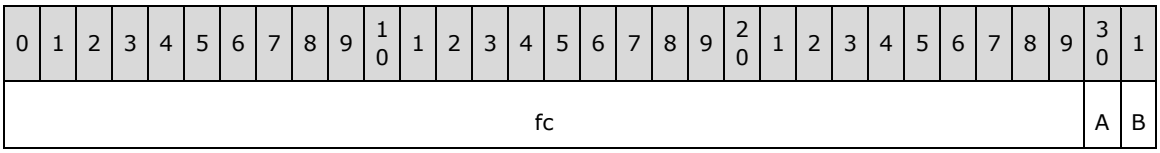
cDepth (2 bytes): An integer that specifies the number of bookmarks in the document of the same type as the bookmark associated with this FBKLD, the ranges of which overlap the limit of the range of this bookmark. To increment the count, a bookmark MUST meet the following constraints:

- The FBKLD of the bookmark occupies the Plcfbkld containing this FBKLD.
- The limit CP (cpL) and the start CP (cpS) of the bookmark, as specified in the Plcfbkld and the Plcfbkfd it is paired with, satisfy the following in relation to the CP (cpCur) that marks the limit of the bookmark of this FBKLD.

$$cpS \neq cpL$$
$$cpS \leq cpCur < cpL$$

2.9.73 FcCompressed

The **FcCompressed** structure specifies the location of text in the WordDocument Stream.



fc (30 bits): An unsigned integer that specifies an offset in the WordDocument Stream where the text starts. If **fCompressed** is zero, the text is an array of 16-bit **Unicode** characters starting at offset **fc**. If **fCompressed** is 1, the text starts at offset **fc**/2 and is an array of 8-bit Unicode characters, except for the values which are mapped to Unicode characters as follows.

Copyright © 2025 Microsoft Corporation

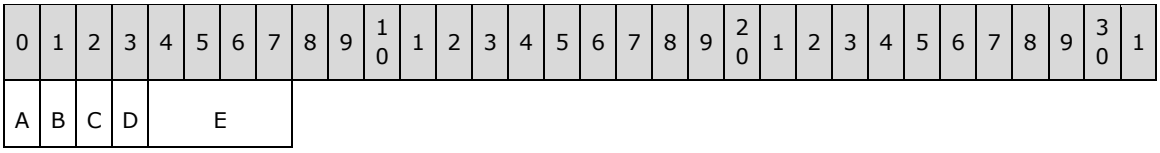
Byte	Unicode Character
0x82	0x201A
0x83	0x0192
0x84	0x201E
0x85	0x2026
0x86	0x2020
0x87	0x2021
0x88	0x02C6
0x89	0x2030
0x8A	0x0160
0x8B	0x2039
0x8C	0x0152
0x91	0x2018
0x92	0x2019
0x93	0x201C
0x94	0x201D
0x95	0x2022
0x96	0x2013
0x97	0x2014
0x98	0x02DC
0x99	0x2122
0x9A	0x0161
0x9B	0x203A
0x9C	0x0153
0x9F	0x0178

A - fCompressed (1 bit): A bit that specifies whether the text is compressed.

B - r1 (1 bit): This bit MUST be zero, and MUST be ignored.

2.9.74 FCCT

The **FCCT** structure specifies information about a **format consistency-checker bookmark**.



A - fcctChp (1 bit): A bit field specifying that the character properties associated with the region of text were flagged as inconsistent with those in other regions of text in the file.

B - fcctPap (1 bit): A bit field specifying that paragraph properties associated with the region of text were flagged as inconsistent with those in other regions of text in the file. This bit field MUST be 0.

C - fcctTap (1 bit): A bit field specifying that table properties associated with the region of text were flagged as inconsistent with those in other regions of text in the file.

D - fcctSep (1 bit): A bit field specifying that line-separation properties associated with the region of text were flagged as inconsistent with those in other regions of text in the file.

Copyright © 2025 Microsoft Corporation

E - fcctUnused (4 bits): This MUST be zero and MUST be ignored.

2.9.75 Fci

The **Fci** enumeration provides a 13-bit unsigned integer that specifies a built-in command.

Name	Value	Meaning
Help	0x0001	Help for the current task or command.
HelpTool	0x0002	Displays Help documentation about a command or screen region or displays a detailed breakdown of the properties of text at a location on the screen.
HelpUsingHelp	0x0003	Displays instructions about how to use the Help documentation.
HelpActiveWindow	0x0004	Displays information about the active pane or document view.
HelpKeyboard	0x0005	Lists the keys and their actions.
HelpIndex	0x0006	Displays the Help index.
HelpQuickPreview	0x0007	Has no effect.
HelpExamplesAndDemos	0x0008	Has no effect.
HelpAbout	0x0009	Displays the application information, version number and the copyright.
HelpWordPerfectHelp	0x000A	Has no effect.
GrowFont	0x000B	Increases the font size of the selection.
ShrinkFont	0x000C	Decreases the font size of the selection.
Overtype	0x000D	Toggles the typing mode between replacing and inserting.
ExtendSelection	0x000E	Turns on extend selection mode and then expands the selection with the direction keys.
Spike	0x000F	Deletes the selection and adds it to the special AutoText entry.
InsertSpike	0x0010	Empties the spike AutoText entry and inserts all of its contents into the document.
ChangeCase	0x0011	Changes the case of the letters in the selection.
MoveText	0x0012	Moves the selection to a specified location.
CopyText	0x0013	Makes a copy of the selection at a specified location.
InsertAutoText	0x0014	Replaces the name of the AutoText entry with its contents.
OtherPane	0x0015	Switches to the other window pane.
NextWindow	0x0016	Switches to the next document window.
PrevWindow	0x0017	Switches back to the previous document window.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
RepeatFind	0x0018	Repeats Go To or Find to find the next occurrence.
NextField	0x0019	Moves to the next field.
PrevField	0x001A	Moves to the previous field.
ColumnSelect	0x001B	Selects a columnar block of text.
DeleteWord	0x001C	Deletes the next word without putting it on the Clipboard.
DeleteBackWord	0x001D	Deletes the previous word without putting it on the Clipboard.
EditClear	0x001E	Performs a forward delete or removes the selection without putting it on the Clipboard.
InsertFieldChars	0x001F	Inserts a field with the enclosing field characters.
UpdateFields	0x0020	Updates and displays the results of the selected fields.
UnlinkFields	0x0021	Permanently replaces the field codes with the results.
ToggleFieldDisplay	0x0022	Shows the field codes or the results for the selection (toggle).
LockFields	0x0023	Locks the selected fields to prevent updating.
UnlockFields	0x0024	Unlocks the selected fields for updating.
UpdateSource	0x0025	Copies the modified text of a linked file back to its source.
Indent	0x0026	Moves the .**logical left**. indent to the next tab stop.
UnIndent	0x0027	Moves the .logical left. indent to the previous tab stop.
HangingIndent	0x0028	Increases the hanging indent.
UnHang	0x0029	Decreases the hanging indent.
Font	0x002A	Changes the font of the selection.
FontSizeSelect	0x002B	Changes the font size of the selection.
WW2_RulerMode	0x002C	Makes the ruler active.
Bold	0x002D	Makes the selection bold (toggle).
Italic	0x002E	Makes the selection italic (toggle).
SmallCaps	0x002F	Makes the selection small capitals (toggle).
AllCaps	0x0030	Makes the selection all capitals (toggle).
Strikethrough	0x0031	Makes the selection strikethrough (toggle).
Hidden	0x0032	Makes the selection hidden text (toggle).
Underline	0x0033	Formats the selection with a continuous underline (toggle).

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
DoubleUnderline	0x0034	Double underlines the selection (toggle).
WordUnderline	0x0035	Underlines the words but not the spaces in the selection (toggle).
Superscript	0x0036	Makes the selection superscript (toggle).
Subscript	0x0037	Makes the selection subscript (toggle).
ResetChar	0x0038	Makes the selection the default character format of the applied style.
CharColor	0x0039	Changes the color of the selected text.
LeftPara	0x003A	Aligns the paragraph at the .logical left. indent.
CenterPara	0x003B	Centers the paragraph between the indents.
RightPara	0x003C	Aligns the paragraph at the .**logical right**. indent.
JustifyPara	0x003D	Aligns the paragraph at both the .logical left. and the .logical right. indent.
SpacePara1	0x003E	Sets the line spacing to single space.
SpacePara15	0x003F	Sets the line spacing to one-and-one-half space.
SpacePara2	0x0040	Sets the line spacing to double space.
CloseUpPara	0x0041	Removes extra spacing above the selected paragraph.
OpenUpPara	0x0042	Sets extra spacing above the selected paragraph.
ResetPara	0x0043	Makes the selection the default paragraph format of the applied style.
EditRepeat	0x0044	Repeats the last action.
GoBack	0x0045	Returns to the previous insertion point.
SaveTemplate	0x0046	Saves the **document template** of the active document.
OK	0x0047	Confirms a location for copying or moving the selection.
Cancel	0x0048	Terminates an action.
CopyFormat	0x0049	Copies the formatting of the selection to a specified location.
PrevPage	0x004A	Moves to the previous page.
NextPage	0x004B	Moves to the next page.
NextObject	0x004C	Moves to the next object on the page.
PrevObject	0x004D	Moves to the previous object on the page.
DocumentStatistics	0x004E	Displays the statistics of the active document.
FileNew	0x004F	Opens New Document taskpane.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
FileOpen	0x0050	Opens an existing document or template.
MailMergeOpenDataSource	0x0051	Opens a data source for mail merge or insert database.
MailMergeOpenHeaderSource	0x0052	Opens a header source for mail merge.
FileSave	0x0053	Saves the active document or template.
FileSaveAs	0x0054	Saves a copy of the document in a separate file.
FileSaveAll	0x0055	Saves all open files, macros, and building blocks and prompts for each one separately.
FileSummaryInfo	0x0056	Shows the summary information about the active document.
FileTemplates	0x0057	Changes the active template and the template options.
FilePrint	0x0058	Prints the active document.
FilePrintPreview	0x0059	Displays full pages as they will be printed.
WW2_PrintMerge	0x005A	Performs mail merge using header and data files.
WW2_PrintMergeCheck	0x005B	Performs a check on a mail merge that uses header and data files.
WW2_PrintMergeToDoc	0x005C	Performs a mail merge using header and data files and places the result into the document.
WW2_PrintMergeToPrinter	0x005D	Performs a mail merge using header and data files and sends the result to the printer.
WW2_PrintMergeSelection	0x005E	Sets mail merge options for mail merges using header and data files.
WW2_PrintMergeHelper	0x005F	Has no effect.
MailMergeReset	0x0060	Resets a mail merge main document to a normal document.
FilePrintSetup	0x0061	Changes the printer and the printing options.
FileExit	0x0062	Quits the application and prompts to save the documents.
FileFind	0x0063	Locates the documents in any directory, drive, or folder.
FileMru	0x0064	Opens a file from the list of most-recently used files.
ApplyStyleName	0x0065	Applies the indicated style to the selected text.
FormatAddrFonts	0x0067	Formats the delivery address font for envelopes.
MailMergeEditDataSource	0x0068	Opens a **mail merge data source**.
WW2_PrintMergeCreateDataSource	0x0069	Creates a data file for mail merges that use a header and data file.
WW2_PrintMergeCreateHeaderSource	0x006A	Creates a header file for mail merges that use a

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
		header and data file.
EditUndo	0x006B	Reverses the last action.
EditCut	0x006C	Cuts the selection and puts it on the Clipboard.
EditCopy	0x006D	Copies the selection and puts it on the Clipboard.
EditPaste	0x006E	Inserts the Clipboard contents at the insertion point.
EditPasteSpecial	0x006F	Inserts the Clipboard contents as a linked object, embedded object, or other format.
EditFind	0x0070	Finds the specified text or the specified formatting.
EditFindFont	0x0071	Has no effect.
EditFindPara	0x0072	Has no effect.
EditFindStyle	0x0073	Has no effect.
EditFindClearFormatting	0x0074	Has no effect.
EditReplace	0x0075	Finds the specified text or the specified formatting and replaces it.
EditReplaceFont	0x0076	Has no effect.
EditReplacePara	0x0077	Has no effect.
EditReplaceStyle	0x0078	Has no effect.
EditReplaceClearFormatting	0x0079	Has no effect.
WW7_EditGoTo	0x007A	Jumps to a specified place in the active document.
WW7_EditAutoText	0x007B	Inserts or defines AutoText entries.
EditLinks	0x007C	Allows links to be viewed, updated, opened, or removed.
EditObject	0x007D	Opens the selected object for editing.
ActivateObject	0x007E	Activates an object.
TextToTable	0x007F	Converts the text to table form.
TableToText	0x0080	Converts a table to text.
TableInsertTable	0x0081	Inserts a table.
TableInsertCells	0x0082	Inserts one or more cells into the table.
TableInsertRow	0x0083	Inserts one or more rows into the table.
TableInsertColumn	0x0084	Inserts one or more columns into the table.
TableDeleteCells	0x0085	Deletes the selected cells from the table.
TableDeleteRow	0x0086	Deletes the selected rows from the table.
TableDeleteColumn	0x0087	Deletes the selected columns from the table.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
TableMergeCells	0x0088	Merges the selected table cells into a single cell.
TableSplitCells	0x0089	Splits the selected table cells.
TableSplit	0x008A	Inserts a paragraph mark above the current row in the table.
TableSelectTable	0x008B	Selects an entire table.
TableSelectRow	0x008C	Selects the current row in a table.
TableSelectColumn	0x008D	Selects the current column in a table.
TableRowHeight	0x008E	Changes the height of the rows in a table.
TableColumnWidth	0x008F	Changes the width of the columns in a table.
TableGridlines	0x0090	Toggles table gridlines on and off.
ViewNormal	0x0091	Changes the editing view to normal view.
ViewOutline	0x0092	Displays a document outline.
ViewPage	0x0093	Displays the page as it will be printed and allows editing.
WW2_ViewZoom	0x0094	Scales the editing view.
ViewDraft	0x0095	Displays the document without formatting and pictures for faster editing (toggle).
ViewFieldCodes	0x0096	Shows the field codes or results for all fields (toggle).
Style	0x0097	Applies an existing style or records a style by example.
ToolsCustomize	0x0098	Customizes the application user interface including menus, keyboard and toolbars.
ViewRuler	0x0099	Shows or hides the ruler.
ViewStatusBar	0x009A	Shows or hides the status bar.
NormalViewHeaderArea	0x009B	Shows a list of headers and footers for editing.
ViewFootnoteArea	0x009C	Opens a pane for viewing and editing the footnotes (toggle).
ViewAnnotations	0x009D	Show or hide comment markup balloons.
InsertFrame	0x009E	Inserts an empty frame or encloses the selected item in a frame.
InsertBreak	0x009F	Ends a page, column, or section at the insertion point.
WW2_InsertFootnote	0x00A0	Inserts a footnote reference at the insertion point.
InsertAnnotation	0x00A1	Inserts a comment.
InsertSymbol	0x00A2	Inserts a special character.
InsertPicture	0x00A3	Inserts a picture from a graphics file.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
InsertFile	0x00A4	Inserts the text of another file into the active document.
InsertDateTime	0x00A5	Inserts the current date, time, or both into the active document.
InsertField	0x00A6	Inserts a field in the active document.
InsertMergeField	0x00A7	Inserts a mail merge field at the insertion point.
EditBookmark	0x00A8	Assigns a name to the selection.
MarkIndexEntry	0x00A9	Marks the text to include in the index.
InsertIndex	0x00AA	Collects the index entries into an index.
InsertTableOfContents	0x00AB	Collects the headings or the table of contents entries into a table of contents.
InsertObject	0x00AC	Inserts an equation, chart, drawing, or some other object.
ToolsCreateEnvelope	0x00AD	Creates or prints an envelope.
FormatFont	0x00AE	Changes the appearance of the selected characters.
FormatParagraph	0x00AF	Changes the appearance and line numbering of the selected paragraphs.
FormatSectionLayout	0x00B0	Changes the page format of the selected sections.
FormatColumns	0x00B1	Changes the column format of the selected sections.
FilePageSetup	0x00B2	Changes the page setup of the selected sections.
FormatTabs	0x00B3	Sets and clears the tab stops for the selected paragraphs.
FormatStyle	0x00B4	Applies, creates, or modifies styles.
FormatDefineStyleFont	0x00B5	Has no effect.
FormatDefineStylePara	0x00B6	Has no effect.
FormatDefineStyleTabs	0x00B7	Has no effect.
FormatDefineStyleFrame	0x00B8	Has no effect.
FormatDefineStyleBorders	0x00B9	Has no effect.
FormatDefineStyleLang	0x00BA	Has no effect.
FormatPicture	0x00BB	Changes the picture scaling, size, and cropping information.
ToolsLanguage	0x00BC	Changes the language formatting of the selected characters.
FormatBordersAndShading	0x00BD	Changes the borders and shading of the selected paragraphs, table cells, and pictures.
FormatFrame	0x00BE	Changes the options for frame formatting.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ToolsSpelling	0x00BF	Checks the spelling in the active document.
ToolsSpellSelection	0x00C0	Checks the spelling of the selected text.
ToolsGrammar	0x00C1	Checks the grammar in the active document.
ToolsThesaurus	0x00C2	Finds a synonym for the selected word.
ToolsHyphenation	0x00C3	Changes the hyphenation settings for the active document.
ToolsBulletsNumbers	0x00C4	Changes the numbered and bulleted paragraphs.
ToolsRevisions	0x00C5	Sets track changes for the active document.
ToolsCompareVersions	0x00C6	Compares the active document with an earlier version.
TableSort	0x00C7	Rearranges the selection into a specified order.
ToolsCalculate	0x00C8	Calculates expressions in the selection.
ToolsRepaginate	0x00C9	Recalculates the page breaks.
WW7_ToolsOptions	0x00CA	Changes various categories of the application options.
ToolsOptionsGeneral	0x00CB	Changes the general options.
ToolsOptionsView	0x00CC	Set specific view mode options.
ToolsAdvancedSettings	0x00CE	Changes advanced options.
ToolsOptionsPrint	0x00D0	Changes the printing options.
ToolsOptionsSave	0x00D1	Changes the save settings.
WW2_ToolsOptionsToolbar	0x00D2	Changes the buttons on the toolbar.
ToolsOptionsSpelling	0x00D3	Changes the proofreader options.
ToolsOptionsGrammar	0x00D4	Changes the proofreader options.
ToolsOptionsUserInfo	0x00D5	Changes the user information options.
ToolsRecordMacroToggle	0x00D6	Turns macro recording on or off.
ToolsMacro	0x00D7	Runs, creates, deletes, or revises a macro.
PauseRecorder	0x00D8	Pauses the macro recorder (toggle).
WindowNewWindow	0x00D9	Opens another window for the active document.
WindowArrangeAll	0x00DA	Arranges windows as non-overlapping tiles.
MailMergeEditMainDocument	0x00DB	Switches to a mail merge main document.
WindowList	0x00DC	Switches to the window containing the specified document.
FormatRetAddrFonts	0x00DD	Formats the return address font for envelopes.
Organizer	0x00DE	Manages AutoText entries, styles, macros, and

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
		toolbars.
WW2_TableColumnWidth	0x00DF	Changes the width of the columns in a table.
ToolsOptionsEdit	0x00E0	Changes the editing options.
ToolsOptionsFileLocations	0x00E1	Changes the default locations used to find files.
RecordNextCommand	0x00E2	Records the next command executed.
ToolsAutoCorrectSmartQuotes	0x00E3	Selects or clears the AutoCorrect SmartQuotes check box.
ToolsWordCount	0x00E4	Displays the word count statistics of the active document.
DocSplit	0x00E5	Splits the active window horizontally and then adjusts the split.
DocSize	0x00E6	Changes the size of the active window.
DocMove	0x00E7	Changes the position of the active window.
DocMaximize	0x00E8	Enlarges the active window to full size.
DocRestore	0x00E9	Restores the window to normal size.
DocClose	0x00EA	Prompts to save the document and then closes the active window.
ControlRun	0x00EB	Displays the Control Panel or the Clipboard.
ShrinkSelection	0x00EC	Shrinks the selection to the next smaller unit.
EditSelectAll	0x00ED	Selects the entire document.
InsertPageField	0x00EF	Inserts a page number field.
InsertDateField	0x00F0	Inserts a date field.
InsertTimeField	0x00F1	Inserts a time field.
FormatHeaderFooterLink	0x00F2	Links this header/footer to the previous section.
ClosePane	0x00F3	Closes the active window pane.
OutlinePromote	0x00F4	Promotes the selected paragraphs one heading level.
OutlineDemote	0x00F5	Demotes the selected paragraphs one heading level.
OutlineMoveUp	0x00F6	Moves the selection above the previous item in the outline.
OutlineMoveDown	0x00F7	Moves the selection below the next item in the outline.
NormalStyle	0x00F8	Applies the Normal style.
OutlineExpand	0x00F9	Displays the next level of subtext of the selection.
OutlineCollapse	0x00FA	Hides the lowest subtext of the selection.
ShowHeading1	0x00FB	Displays the level 1 headings only.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ShowHeading2	0x00FC	Displays the level 1 and 2 headings.
ShowHeading3	0x00FD	Displays the level 1 through 3 headings.
ShowHeading4	0x00FE	Displays the level 1 through 4 headings.
ShowHeading5	0x00FF	Displays the level 1 through 5 headings.
ShowHeading6	0x0100	Displays the level 1 through 6 headings.
ShowHeading7	0x0101	Displays the level 1 through 7 headings.
ShowHeading8	0x0102	Displays the level 1 through 8 headings.
ShowHeading9	0x0103	Displays the level 1 through 9 headings.
ShowAllHeadings	0x0104	Displays all of the heading levels and the body text.
OutlineShowFirstLine	0x0105	Toggles between showing the first line of each paragraph only or showing all of the body text in the outline.
OutlineShowFormat	0x0106	Toggles the display of character formatting in outline view.
ShowVars	0x0107	Has no effect.
StepOver	0x0108	Has no effect.
StepIn	0x0109	Has no effect.
ContinueMacro	0x010A	Has no effect.
TraceMacro	0x010B	Has no effect.
EditObjectPrivate	0x010C	Opens the selected object for editing.
NextCell	0x010E	Moves to the next table cell.
PrevCell	0x010F	Moves to the previous table cell.
StartOfRow	0x0110	Moves to the first cell in the current row.
EndOfRow	0x0111	Moves to the last cell in the current row.
StartOfColumn	0x0112	Moves to the first cell in the current column.
EndOfColumn	0x0113	Moves to the last cell in the current column.
ShowAll	0x0114	Shows or hides all nonprinting characters.
WW7_InsertPageBreak	0x0115	Inserts a page break at the insertion point.
WW7_InsertColumnBreak	0x0116	Inserts a column break at the insertion point.
AppMinimize	0x0117	Minimizes the application window to an icon.
AppMaximize	0x0118	Enlarges the application window to full size.
AppRestore	0x0119	Restores the application window to normal size.
DoFieldClick	0x011A	Executes the action associated with the button fields.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
FileClose	0x011B	Closes all of the windows of the active document.
InsertDrawing	0x011C	Inserts a Microsoft Draw object.
InsertChart	0x011D	Inserts a Microsoft Graph object.
SelectCurFont	0x011E	Selects all characters with the same font name and point size.
SelectCurAlignment	0x011F	Selects all paragraphs with the same alignment.
SelectCurSpacing	0x0120	Selects all paragraphs with the same line spacing.
SelectCurIndent	0x0121	Selects all paragraphs with the same indentation.
SelectCurTabs	0x0122	Selects all paragraphs with the same tabs.
SelectCurColor	0x0123	Selects all characters with the same color.
RemoveFrames	0x0124	Removes frame formatting from the selection.
MenuMode	0x0125	Makes the menu bar active.
InsertPageNumbers	0x0126	Adds page numbers to the top or the bottom of the pages.
WW2_ChangeRulerMode	0x0127	Changes the display mode of the ruler (paragraph, table, and document).
EditPicture	0x0128	Uses the specified drawing application to edit the selected picture.
UserDialog	0x0129	Has no effect.
FormatPageNumber	0x012A	Changes the appearance of page numbers.
WW2_FootnoteOptions	0x012B	Changes the options for footnotes.
CopyFile	0x012C	Copies the specified file to the specified destination.
FileNewDefault	0x012D	Creates a new document based on the NORMAL template.
FilePrintDefault	0x012E	Prints the active document using the current defaults.
ViewZoomWholePage	0x012F	Scales the editing view to see the whole page in page layout view.
ViewZoomPageWidth	0x0130	Scales the editing view to see the width of the page.
ViewZoom100	0x0131	Scales the editing view to 100% in normal view.
TogglePortrait	0x0132	Toggles between portrait and landscape mode.
ToolsBulletListDefault	0x0133	Creates a bulleted list based on the current defaults.
ToggleScribbleMode	0x0134	Inserts a pen comment at the location of the insertion point.
ToolsNumberListDefault	0x0135	Creates a numbered list based on the current defaults.
FileAOCEAddMailer	0x0137	Has no effect.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
FileAOCEDeleteMailer	0x0138	Has no effect.
FileAOCEExpandMailer	0x0139	Has no effect.
FileAOCESendMail	0x013B	Has no effect.
FileAOCEReplyMail	0x013C	Has no effect.
FileAOCEReplyAllMail	0x013D	Has no effect.
FileAOCEForwardMail	0x013E	Has no effect.
FileAOCENextLetter	0x013F	Has no effect.
DocMinimize	0x0140	Minimizes the active window to an icon.
FormatAutoFormatBegin	0x0141	Automatically formats a document.
FormatChangeCase	0x0142	Changes the case of the letters in the selection.
ViewToolbars	0x0143	Shows or hides the application toolbars.
TableInsertGeneral	0x0144	Inserts rows, columns, or cells in a table.
TableDeleteGeneral	0x0145	Deletes rows, columns, or cells in a table.
WW2_TableRowHeight	0x0146	Changes the height of the rows in a table.
TableToOrFromText	0x0147	Converts text to a table or a table to text.
EditRedo	0x0149	Redoes the last action that was undone.
EditRedoOrRepeat	0x014A	Redoes the last action that was undone or repeats the last action.
UpdateToc	0x014B	Select method of updating a table of contents or captions.
ViewEndnoteArea	0x0152	Opens a pane for viewing and editing the endnotes (toggle).
MailMergeDataForm	0x0154	Edits a list or table in a form.
InsertDatabase	0x0155	Inserts information from an external data source into the active document.
WW2_InsertTableOfContents	0x0158	Collects the headings or the table of contents entries into a table of contents.
WW2_ToolsHyphenation	0x0159	Hyphenates the current selection.
FormatFrameOrFramePicture	0x015A	Puts the selected picture in a frame or formats a frame.
WW2_ToolsOptionsPrint	0x015B	Has no effect.
TableFormula	0x015C	Inserts a formula field into a table cell.
TextFormField	0x015D	Inserts a text form field.
CheckBoxFormField	0x015E	Inserts a check box form field.
DropDownFormField	0x015F	Inserts a drop-down form field.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
FormFieldOptions	0x0161	Changes the options for a form field.
ProtectForm	0x0162	Toggles protection for the active document.
ApplyFontName	0x0164	Applies the indicated font to the selected text.
InsertCaption	0x0165	Inserts a caption above or below a selected object.
InsertCaptionNumbering	0x0166	Sets the number for a caption type.
InsertAutoCaption	0x0167	Defines which objects are inserted with a caption.
HelpPSSHelp	0x0168	Displays information about the support available for the application.
WW7_DrawTextBox	0x016B	Inserts a text box drawing object.
WW7_ToolsOptionsAutoFormat	0x016D	Changes the AutoFormat options.
DemoteToBodyText	0x016E	Applies the Normal style and converts the selected headings to body text.
InsertCrossReference	0x016F	Inserts a cross-reference.
InsertFootnoteNow	0x0170	Inserts a footnote reference at the insertion point.
InsertEndnoteNow	0x0171	Inserts an endnote reference at the insertion point.
InsertFootnote	0x0172	Inserts a footnote or endnote reference at the insertion point.
NoteOptions	0x0175	Changes the options for footnotes or endnotes.
WW2_FormatCharacter	0x0176	Changes the appearance of the selected characters.
DrawLine	0x0178	Inserts a line drawing object.
DrawRectangle	0x0179	Inserts a rectangle drawing object.
ToolsAutoCorrect	0x017A	Adds or deletes AutoCorrect entries.
ToolsAutoCorrectReplaceText	0x017C	Selects or clears the AutoCorrect ReplaceText check box.
ToolsAutoCorrectInitialCaps	0x017D	Selects or clears the AutoCorrect InitialCaps check box.
ToolsAutoCorrectSentenceCaps	0x017F	Selects or clears the AutoCorrect SentenceCaps check box.
ToolsAutoCorrectDays	0x0180	Selects or clears the AutoCorrect Days check box.
FormatAutoFormat	0x0181	Automatically formats a document.
ToolsOptionsRevisions	0x0182	Changes track changes options.
WW2_ToolsOptionsGeneral	0x0183	Has no effect.
ResetNoteSepOrNotice	0x0184	Resets a separator, continuation separator, or continuation notice to the application default.
FormatBullet	0x0185	Creates a bulleted list.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
FormatNumber	0x0186	Creates a numbered list.
FormatMultilevel	0x0187	Creates a multilevel list.
ConvertObject	0x0188	Converts or activates an object as another type.
TableSortAToZ	0x0189	Sorts records in ascending order (A to Z).
TableSortZToA	0x018A	Sorts records in descending order (Z to A).
WW7_FormatBulletsAndNumbering	0x018D	Creates a numbered or bulleted list.
FormatSimpleNumberDefault	0x018E	Creates a numbered list based on the current defaults.
FormatBulletDefault	0x018F	Creates a bulleted list based on the current defaults.
InsertAddCaption	0x0192	Adds a new caption type.
GoToNextPage	0x0194	Jumps to the next page in the active document.
GoToPreviousPage	0x0195	Jumps to the previous page in the active document.
GoToNextSection	0x0196	Jumps to the next section in the active document.
GoToPreviousSection	0x0197	Jumps to the previous section in the active document.
GoToNextFootnote	0x0198	Jumps to the next footnote in the active document.
GoToPreviousFootnote	0x0199	Jumps to the previous footnote in the active document.
GoToNextEndnote	0x019A	Jumps to the next endnote in the active document.
GoToPreviousEndnote	0x019B	Jumps to the previous endnote in the active document.
GoToNextComment	0x019C	Jumps to the next comment in the active document.
GoToPreviousComment	0x019D	Jumps to the previous comment in the active document.
WW2_FormatDefineStyleChar	0x019E	Has no effect.
WW2_EditFindChar	0x019F	Has no effect.
WW2_EditReplaceChar	0x01A0	Has no effect.
AppMove	0x01A2	Changes the position of the application window.
AppSize	0x01A3	Changes the size of the application window.
Connect	0x01A4	Connects to a network drive.
WW2_EditFind	0x01A5	Has no effect.
WW2_EditReplace	0x01A6	Has no effect.
EditFindLang	0x01AC	Has no effect.
EditReplaceLang	0x01AD	Has no effect.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
MailMergeViewData	0x01AF	Toggles between viewing merge fields and actual data.
ToolsCustomizeKeyboard	0x01B0	Customizes the application key assignments.
ToolsCustomizeMenus	0x01B1	Customizes the application menu assignments.
WW2_ToolsOptionsKeyboard	0x01B2	Remaps keys within the document.
ToolsMergeRevisions	0x01B3	Merges changes from the active document to an earlier version.
ClosePreview	0x01B5	Exits print preview.
SkipNumbering	0x01B6	Makes the selected paragraphs skip numbering.
EditConvertAllFootnotes	0x01B7	Converts all footnotes into endnotes.
EditConvertAllEndnotes	0x01B8	Converts all endnotes into footnotes.
EditSwapAllNotes	0x01B9	Changes all footnotes to endnotes and all endnotes to footnotes.
MarkTableOfContentsEntry	0x01BA	Marks the text to include in the table of contents.
FilePgSetupGX	0x01BC	Has no effect.
FilePrintOneGX	0x01BD	Has no effect.
EditFindTabs	0x01BE	Has no effect.
EditFindBorder	0x01BF	Has no effect.
EditFindFrame	0x01C0	Has no effect.
BorderOutside	0x01C1	Changes the outside borders of the selected paragraphs, table cells, and pictures.
BorderNone	0x01C2	Removes borders from the selected paragraphs, table cells, and pictures.
BorderLineStyle	0x01C3	Changes border line styles of the selected paragraphs, table cells, and pictures.
ShadingPattern	0x01C4	Changes shading pattern of the selected paragraphs, table cells, and pictures.
DrawEllipse	0x01C6	Inserts an ellipse drawing object.
DrawArc	0x01C7	Inserts an arc drawing object.
EditReplaceTabs	0x01C8	Has no effect.
EditReplaceBorder	0x01C9	Has no effect.
EditReplaceFrame	0x01CA	Has no effect.
EditOfficeClipboard	0x01CB	Displays the contents of the shared application clipboard.
EditConvertNotes	0x01CE	Converts selected footnotes into endnotes, or converts selected endnotes into footnotes.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
MarkCitation	0x01CF	Marks the text to include in the table of authorities.
WW2_ToolsRevisionsMark	0x01D0	Has no effect.
DrawGroup	0x01D1	Groups the selected drawing objects.
DrawBringToFront	0x01D2	Brings the selected drawing objects to the front.
DrawSendToBack	0x01D3	Sends the selected drawing objects to the back.
DrawSendBehindText	0x01D4	Sends the selected drawing objects back one layer.
DrawBringInFrontOfText	0x01D5	Brings the selected drawing objects forward one layer.
InsertTableOfAuthorities	0x01D7	Collects the table of authorities entries into a table of authorities.
InsertTableOfFigures	0x01D8	Collects captions into a table of captions.
InsertIndexAndTables	0x01D9	Inserts an index or a table of contents, figures, or authorities into the document.
MailMergeNextRecord	0x01DE	Displays the next record in the active mail merge data source.
MailMergePrevRecord	0x01DF	Displays the previous record in the active mail merge data source.
MailMergeFirstRecord	0x01E0	Displays the first record in the active mail merge data source.
MailMergeLastRecord	0x01E1	Displays the last record in the active mail merge data source.
MailMergeGoToRecord	0x01E2	Displays the specified record in the active mail merge data source.
InsertFormField	0x01E3	Inserts a new form field.
ViewHeader	0x01E4	Displays header in page layout view.
DrawUngroup	0x01E5	Removes the grouping of the selected group of drawing objects.
PasteFormat	0x01E6	Applies the previously copied formatting to selection.
WW2_ToolsOptionsMenus	0x01E7	Has no effect.
FormatDropCap	0x01E8	Formats the first character of current paragraph as a dropped capital.
ToolsCreateLabels	0x01E9	Creates or prints a label or a sheet of labels.
ViewMasterDocument	0x01EA	Switches to master document view.
CreateSubdocument	0x01EB	Transforms the selected outline items into subdocuments.
Language	0x01EC	Changes the language formatting of the selected characters.
ViewFootnoteSeparator	0x01ED	Opens a pane for viewing and editing the footnote

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
		separator.
ViewFootnoteContSeparator	0x01EE	Opens a pane for viewing and editing the footnote continuation separator.
ViewFootnoteContNotice	0x01EF	Opens a pane for viewing and editing the footnote continuation notice.
ViewEndnoteSeparator	0x01F0	Opens a pane for viewing and editing the endnote separator.
ViewEndnoteContSeparator	0x01F1	Opens a pane for viewing and editing the endnote continuation separator.
ViewEndnoteContNotice	0x01F2	Opens a pane for viewing and editing the endnote continuation notice.
WW2_ToolsOptionsView	0x01F3	Has no effect.
DrawBringForward	0x01F4	Brings the selected drawing objects forward.
DrawSendBackward	0x01F5	Sends the selected drawing objects backward.
ViewFootnotes	0x01F6	Opens a pane for viewing and editing the notes (toggle).
ToolsProtectDocument	0x01F7	Sets protection for the active document.
ToolsShrinkToFit	0x01F8	Attempts to make the document fit on one less page.
FormatStyleGallery	0x01F9	Apply styles from templates.
ToolsReviewRevisions	0x01FA	Reviews changes to the active document.
ShowMultiplePages	0x01FD	Show multiple pages.
HelpSearch	0x01FE	Searches for a Help topic by typing or selecting a keyword.
HelpWordPerfectHelpOptions	0x01FF	Has no effect.
MailMergeConvertChevrons	0x0200	Toggles converting Word for the Macintosh mail merge chevrons.
GrowFontOnePoint	0x0201	Increases the font size of the selection by one point.
ShrinkFontOnePoint	0x0202	Decreases the font size of the selection by one point.
Magnifier	0x0203	Toggle zoom-in / zoom-out mode.
FilePrintPreviewFullScreen	0x0204	Toggles full screen.
InsertSound	0x0207	Inserts a sound object into the document.
ToolsProtectUnprotectDocument	0x0208	Toggles protection for the active document.
ToolsUnprotectDocument	0x0209	Removes protection from the active document.
RemoveBulletsNumbers	0x020A	Removes numbers and bullets from the selection.
FileCloseOrCloseAll	0x020B	Closes the file, or if the user is holding down the shift key, closes all files.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
FileCloseAll	0x020C	Closes all of the windows of all documents.
ToolsOptionsCompatibility	0x020D	Changes the document compatibility options.
CopyButtonImage	0x020E	Copy the image of the selected button to the Clipboard.
PasteButtonImage	0x020F	Paste the image on the Clipboard onto the selected button.
ResetButtonImage	0x0210	Reset the image on the selected button to the built-in image.
ApplyAutoTextName	0x0211	Inserts the indicated AutoText entry in the document.
Columns	0x0212	Changes the number of columns in the selected sections.
Condensed	0x0213	Sets the font character spacing of the selection to condensed.
Expanded	0x0214	Sets the font character spacing of the selection to expanded.
FontSize	0x0215	Changes the font size of the selection.
Lowered	0x0216	Lowers the selection below the base line.
Raised	0x0217	Raises the selection above the base line.
FileOpenFile	0x0218	Opens a document.
DrawRoundRectangle	0x0219	Inserts a rounded rectangle drawing object.
DrawFreeformPolygon	0x021A	Inserts a freeform drawing object.
SelectDrawingObjects	0x0221	Allows the selection of multiple drawing objects.
Shading	0x0222	Changes the background shading of paragraphs and table cells.
Borders	0x0223	Changes the borders of paragraphs, table cells, and pictures.
Color	0x0224	Changes the color of the selected text.
DialogEditor	0x0228	Opens the macro dialog editor.
MacroREM	0x0229	Has no effect.
StartMacro	0x022A	Has no effect.
Symbol	0x022B	Inserts a special character.
DrawToggleLayer	0x022C	Switches whether the drawing object appears in the front of or behind the text.
ToolsCustomizeKeyboardShortcut	0x022D	Shortcut method for customizing keyboard settings.
ToolsCustomizeAddMenuShortcut	0x022E	Shortcut method for customizing menus.
DrawFlipHorizontal	0x022F	Flips the selected drawing objects from left to right.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
DrawFlipVertical	0x0230	Flips the selected drawing objects from top to bottom.
DrawRotateRight	0x0231	Rotates the selected drawing objects 90 degrees to the right.
DrawRotateLeft	0x0232	Rotates the selected drawing objects 90 degrees to the left.
TableAutoFormat	0x0233	Applies a set of formatting to a table.
FormatTextFlow	0x0234	Changes text flow direction and character orientation.
WW7_FormatDrawingObject	0x0235	Changes the fill, line, size, and position attributes of the selected drawing objects.
InsertExcelTable	0x0237	Inserts a Microsoft Excel worksheet object.
MailMergeListWordFields	0x0238	Inserts a field at the insertion point.
MailMergeFindRecord	0x0239	Finds a specified record in a mail merge data source.
NormalFontSpacing	0x023B	Removes the expanded or condensed font attribute.
NormalFontPosition	0x023C	Removes the raised or lowered font attribute.
ViewZoom200	0x023D	Scales the editing view to 200 percent in normal view.
ViewZoom75	0x023E	Scales the editing view to 75 percent in normal view.
DrawDisassemblePicture	0x023F	Disassembles the selected metafile picture into drawing objects.
ViewZoom	0x0241	Scales the editing view.
ToolsProtectSection	0x0242	Sets protection for sections of the active document.
OfficeOnTheWeb	0x0243	Opens the Microsoft Office Online web site.
FontSubstitution	0x0245	Changes the font mapping of a document.
ToggleFull	0x0246	Toggles full screen mode on and off.
InsertSubdocument	0x0247	Opens a file and inserts it as a subdocument in a master document.
MergeSubdocument	0x0248	Merges two adjacent subdocuments into one subdocument.
SplitSubdocument	0x0249	Splits the selected part of a subdocument into another subdocument at the same level.
NewToolbar	0x024A	Creates a new toolbar.
ToggleMainTextLayer	0x024B	Toggles showing the main text layer in page layout view.
ShowPrevHeaderFooter	0x024C	Shows the header or footer of the previous section in page layout view.
ShowNextHeaderFooter	0x024D	Shows header or footer of the next section in page layout view.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
GoToHeaderFooter	0x024E	Jump between header and footer.
PromoteList	0x024F	Promotes the selection one level.
DemoteList	0x0250	Demotes the selection one level.
ApplyHeading1	0x0251	Applies Heading 1 style to the selected text.
ApplyHeading2	0x0252	Applies Heading 2 style to the selected text.
ApplyHeading3	0x0253	Applies Heading 3 style to the selected text.
ApplyListBullet	0x0254	Applies List Bullet style to the selected text.
GotoCommentScope	0x0255	Highlights the text associated with an comment reference mark.
TableHeadings	0x0256	Toggles table headings attribute on and off.
OpenSubdocument	0x0257	Opens a subdocument in a new window.
LockDocument	0x0258	Toggles the file lock state of a document.
ToolsCustomizeRemoveMenuShortcut	0x0259	Shortcut method for customizing menus.
FormatDefineStyleNumbers	0x025A	Has no effect.
FormatHeadingNumbering	0x025B	Changes numbering options for heading level styles.
ViewBorderToolbar	0x025C	Shows or hides the Borders/Table toolbar.
ViewDrawingToolbar	0x025D	Shows or hides the Drawing toolbar.
FormatHeadingNumber	0x025E	Modifies Heading Numbering styles.
ToolsEnvelopesAndLabels	0x025F	Creates or prints an envelope, a label, or a sheet of labels.
DrawReshape	0x0260	Displays resizing handles on selected freeform drawing objects. Drag a handle to reshape the object.
MailMergeAskToConvertChevrons	0x0261	Toggles whether to prompt the user about converting Word for the Macintosh mail merge chevrons.
FormatCallout	0x0262	Formats the selected callouts or sets callout defaults.
DrawCallout	0x0263	Inserts a callout drawing object.
TableFormatCell	0x0264	Changes the height and width of the rows and columns in a table.
FileSendMail	0x0265	Sends the active document through electronic mail.
EditButtonImage	0x0266	Edit the image on the selected button.
ToolsCustomizeMenuBar	0x0267	Has no effect.
AutoMarkIndexEntries	0x0268	Inserts index entries using an **automark file**.
InsertEnSpace	0x026A	Inserts an EN space.
InsertEmSpace	0x026B	Inserts an EM space.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
DottedUnderline	0x026C	Underlines the selection with dots (toggle).
ParaKeepLinesTogether	0x026D	Prevents a paragraph from splitting across page boundaries.
ParaKeepWithNext	0x026E	Keeps a paragraph and the following paragraph on the same page.
ParaPageBreakBefore	0x026F	Makes the current paragraph start on a new page.
FileRoutingSlip	0x0270	Has no effect.
EditTOACategory	0x0271	Modifies the category names for the table of authorities.
TableUpdateAutoFormat	0x0272	Updates the table formatting to match the applied formatting set.
ChooseButtonImage	0x0273	Attach an image or text to the selected button.
ParaWidowOrphanControl	0x0274	Prevents a page break from leaving a single line of a paragraph on one page.
ToolsAddRecordDefault	0x0275	Adds a record to a database.
ToolsRemoveRecordDefault	0x0276	Removes a record from a database.
ToolsManageFields	0x0277	Adds or deletes a field from a database.
ViewToggleMasterDocument	0x0278	Switches between outline and master document views.
DrawSnapToGrid	0x0279	Sets up a grid for aligning drawing objects.
DrawAlign	0x027A	Aligns the selected drawing objects with one another or the page.
HelpTipOfTheDay	0x027B	Displays a tip of the day.
FormShading	0x027C	Changes shading options for the current form.
EditUpdateIMEDic	0x027E	Update .**IME**. dictionary.
RemoveSubdocument	0x027F	Merges the contents of the selected subdocuments into the master document that contains them.
CloseViewHeaderFooter	0x0280	Returns to document text.
TableAutoSum	0x0281	Inserts an expression field that automatically sums a table row or column.
MailMergeCreateDataSource	0x0282	Creates a new mail merge data source.
MailMergeCreateHeaderSource	0x0283	Creates a new mail merge header source.
StopMacroRunning	0x0285	Has no effect.
IMEControl	0x0286	Disable .IME.
DrawInsertWordPicture	0x0288	Opens a separate window for creating a picture object or inserts the selected drawing objects into a picture.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
WW7_IncreaseIndent	0x0289	Increases indent or demotes the selection one level.
WW7_DecreaseIndent	0x028A	Decreases indent or promotes the selection one level.
SymbolFont	0x028B	Applies the Symbol font to the selection.
ToggleHeaderFooterLink	0x028C	Links or unlinks this header/footer to or from the previous section.
AutoText	0x028D	Creates or inserts an AutoText entry depending on the selection.
ViewFooter	0x028E	Displays footer in page layout view.
MicrosoftMail	0x0290	Starts or switches to Microsoft Outlook.
MicrosoftExcel	0x0291	Starts or switches to Microsoft Excel.
MicrosoftAccess	0x0292	Starts or switches to Microsoft Access.
MicrosoftSchedule	0x0293	Starts or switches to Microsoft Schedule+.
MicrosoftFoxPro	0x0294	Starts or switches to Microsoft FoxPro.
MicrosoftPowerPoint	0x0295	Starts or switches to Microsoft PowerPoint.
MicrosoftPublisher	0x0296	Starts or switches to Microsoft Publisher.
MicrosoftProject	0x0297	Starts or switches to Microsoft Project.
ListMacros	0x0298	Has no effect.
ScreenRefresh	0x0299	Refreshes the display.
ToolsRecordMacroStart	0x029A	Turns macro recording on or off.
ToolsRecordMacroStop	0x029B	Turns macro recording on or off.
StopMacro	0x029C	Stops recording or running the current macro.
ToggleMacroRun	0x029D	Has no effect.
DrawNudgeUp	0x029E	Moves the selected drawing objects up.
DrawNudgeDown	0x029F	Moves the selected drawing objects down.
DrawNudgeLeft	0x02A0	Moves the selected drawing objects to the left.
DrawNudgeRight	0x02A1	Moves the selected drawing objects to the right.
WW2_ToolsMacro	0x02A2	Runs, creates, deletes, or revises a macro.
MailMergeEditHeaderSource	0x02A3	Opens a mail merge header source.
MailMerge	0x02A4	Combines files to produce form letters, mailing labels, envelopes, and catalogs.
MailMergeCheck	0x02A5	Checks for errors in a mail merge.
MailMergeToDoc	0x02A6	Collects the results of a mail merge in a document.
MailMergeToPrinter	0x02A7	Sends the results of a mail merge to the printer.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
MailMergeHelper	0x02A8	Prepares a main document for a mail merge.
MailMergeQueryOptions	0x02A9	Sets the query options for a mail merge.
InsertWordArt	0x02AA	Inserts a Microsoft WordArt object.
InsertEquation	0x02AB	Inserts a Microsoft Equation object.
RunPrintManager	0x02AC	Displays the Print Manager.
FileMacPageSetup	0x02AD	Has no effect.
FileConfirmConversions	0x02AF	Toggles asking the user to confirm the conversion when opening a file.
HelpContents	0x02B0	Displays Help contents.
WW2_InsertSymbol	0x02B5	Inserts a special character.
FileClosePicture	0x02B6	Closes the active picture document.
WW2_InsertIndex	0x02B7	Collects the index entries into an index.
DrawResetWordPicture	0x02B8	Sets document margins to enclose all drawing objects on the page.
WW2_FormatBordersAndShading	0x02B9	Changes the borders and shading of the selected paragraphs, table cells, and pictures.
OpenOrCloseUpPara	0x02BA	Sets or removes extra spacing above the selected paragraph.
DrawNudgeUpPixel	0x02BC	Moves the selected drawing objects up one pixel.
DrawNudgeDownPixel	0x02BD	Moves the selected drawing objects down one pixel.
DrawNudgeLeftPixel	0x02BE	Moves the selected drawing objects to the left one pixel.
DrawNudgeRightPixel	0x02BF	Moves the selected drawing objects to the right one pixel.
ToolsHyphenationManual	0x02C0	Hyphenates the selection or the entire document.
FixMe	0x02C1	Repairs the installation of the application.
ClearFormField	0x02C2	Deletes the selected form field.
InsertSectionBreak	0x02C3	Ends a section at the insertion point.
DrawUnselect	0x02C4	Unselects a drawn object.
DrawSelectNext	0x02C5	Selects the next drawn object.
DrawSelectPrevious	0x02C6	Selects the previous drawn object.
MicrosoftSystemInfo	0x02C7	Launches the System Information application.
ToolsCustomizeToolbar	0x02CC	Customizes the toolbars.
IndentChar	0x02CF	Increases the indent by width of a character.
UnIndentChar	0x02D0	Decreases the indent by width of a character.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
IndentFirstChar	0x02D1	Increases the hanging indent by width of a character.
UnIndentFirstChar	0x02D2	Decreases the hanging indent by width of a character.
ListCommands	0x02D3	Create a table of commands, with key and menu assignments.
HelpIchitaroHelp	0x02D8	Shows Competitor (Ichitaro, Korean WordPerfect) help.
ChangeByte	0x02DA	Changes between wide and narrow versions of the letters in the selection.
ChangeKana	0x02DB	Changes the characters in the selection between Katakana and Hiragana.
EditCreatePublisher	0x02DC	Has no effect.
EditSubscribeTo	0x02DD	Has no effect.
EditPubOrSubOptions	0x02DE	Has no effect.
EditPublishOptions	0x02DF	Has no effect.
EditSubscribeOptions	0x02E0	Has no effect.
FilePgSetupCustGX	0x02E1	Has no effect.
WW7_DrawVerticalTextBox	0x02E2	Inserts a vertical text box drawing object.
ToolsOptionsTypography	0x02E3	Changes the Typography options.
DistributePara	0x02E4	Distributed. Paragraph.
ViewGridlines	0x02E5	Shows or hides the gridlines.
Highlight	0x02E6	Applies color highlighting to the selection.
FixSpellingChange	0x02E8	Replaces this word by the selected suggestion.
FileProperties	0x02EE	Shows the properties of the active document.
EditCopyAsPicture	0x02EF	Copies the selection and puts it on the Clipboard as a picture.
IndentFirstLine	0x02F2	Increases the hanging indent by width of 2 characters.
UnIndentFirstLine	0x02F3	Decreases the hanging indent by width of 2 characters.
IndentLine	0x02F4	Increases the indent by width of 2 characters.
UnIndentLine	0x02F5	Decreases the indent by width of 2 characters.
InsertAddress	0x02F6	Inserts an address from the user’s Personal Address Book.
NextMisspelling	0x02F7	Find next spelling error.
FilePost	0x02F8	Puts the active document into a Microsoft Exchange folder.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ToolsAutoCorrectExceptions	0x02FA	Adds or deletes AutoCorrect Capitalization exceptions.
MailHideMessageHeader	0x02FB	Shows or hides the mail message header when the application is being used as an e-mail editor.
MailMessageProperties	0x02FC	Sets the properties of the e-mail message.
DotAccent	0x02FD	Formats the selection with dot accents (toggle).
CommaAccent	0x02FE	Formats the selection with comma accents (toggle).
ToolsAutoCorrectCapsLockOff	0x02FF	Selects or clears the AutoCorrect Caps Lock Off check box.
MailMessageReply	0x0300	Replies to a mail message.
MailMessageReplyAll	0x0301	Replies All to a mail message.
MailMessageMove	0x0302	Moves an e-mail message.
MailMessageDelete	0x0303	Deletes an e-mail message.
MailMessagePrevious	0x0304	Goes to the previous e-mail message.
MailMessageNext	0x0305	Goes to the next e-mail message.
MailCheckNames	0x0306	Checks the recipient names of an e-mail message.
MailSelectNames	0x0307	Selects the recipients of an e-mail message.
MailMessageForward	0x0308	Forwards an e-mail message.
ToolsSpellingRecheckDocument	0x0309	Resets spelling results for the current document.
ToolsOptionsAutoFormatAsYouType	0x030A	Changes the AutoFormat As You Type options.
MailMergeUseAddressBook	0x030B	Opens an address book as a data source for mail merge.
EditFindHighlight	0x030C	Has no effect.
EditReplaceHighlight	0x030D	Has no effect.
EditFindNotHighlight	0x030E	Has no effect.
EditReplaceNotHighlight	0x030F	Has no effect.
ToolsHHC	0x0310	Finds a Hangul/Hanja word for the selected word.
UnderlineColor	0x0311	Changes the underline color of the selected text.
ToolsOptionsHHC	0x0312	Changes the **HHC** options.
InsertVerticalFrame	0x0313	Inserts an empty vertical frame or encloses the selected item in a vertical frame.
BorderTLtoBR	0x0314	Changes the top left to bottom right diagonal border of the selected table cells.
BorderTRtoBL	0x0315	Changes the top right to bottom left diagonal border of the selected table cells.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ToolsOptionsFuzzy	0x0316	Changes the fuzzy expressions options.
DrawBrace	0x0317	Inserts a brace drawing object.
DrawBracket	0x0318	Inserts a bracket drawing object.
HelpAW	0x031A	Locates Help topics based on an entered question or request.
HelpMSN	0x031B	Has no effect.
CreateTable	0x031C	Inserts a table.
CharScale	0x031D	Applies character scaling to the selection.
DoubleStrikethrough	0x031E	Makes the selection double strikethrough (toggle).
TopAlign	0x031F	Aligns cell content to the top of cell.
CenterAlign	0x0320	Aligns cell content to the center of cell.
BottomAlign	0x0321	Aligns cell content to the bottom of cell.
ViewOutlineSplitToolbar	0x0324	Shows or hides the Borders/Table toolbar.
DistributeColumn	0x0327	Evenly distributes selected columns.
ViewFormatExToolbar	0x032B	Shows or hides the Extended Formatting toolbar.
InsertNumber	0x032C	Inserts a number in the active document.
ContextHelp	0x032D	Toggles context sensitive help through F1 key.
InsertOfficeDrawing	0x032F	Inserts a Microsoft Draw object.
RedefineStyle	0x0330	Redefines the current style based on the selected text.
ViewOnline	0x0334	Displays the document optimized for reading online.
LetterProperties	0x0335	Formats a Letter Document.
BrowseSel	0x0336	Select the next/previous browse object.
BrowsePrev	0x0337	Jump to the previous browse object.
FormatBulletsAndNumbering	0x0338	Creates a numbered or bulleted list.
ListOutdent	0x0339	Promotes the selection one level.
ListIndent	0x033A	Demotes the selection one level.
ToolsProofing	0x033C	Checks the proofing in the active document.
InsertPageBreak	0x033E	Inserts a page break at the insertion point.
InsertColumnBreak	0x033F	Inserts a column break at the insertion point.
ToolsCreateDirectory	0x0341	Creates a new directory.
BrowseNext	0x0342	Jump to the next browse object.
InsertNumberOfPages	0x0343	Inserts a number of pages field.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
NextInsert	0x0344	Returns to the next insertion point.
TextBoxLinking	0x0348	Creates a forward link to another text box.
TextBoxUnlinking	0x0349	Breaks the forward link to another text box.
GotoNextLinkedTextBox	0x034A	Selects the next linked text box.
GotoPrevLinkedTextBox	0x034B	Selects the previous linked text box.
ToolsSpellingRange	0x034E	Checks the spelling on the range.
ToolsGrammarRange	0x034F	Checks the spelling and grammar on the range.
ViewWeb	0x0350	Displays the document similarly to how a web browser would.
ShowTableGridlines	0x0351	Toggles table gridlines on and off.
BlogBlogPublish	0x0352	Sends the active document to a blog.
BlogBlogPublishDraft	0x0353	Sends the active document to a blog.
BlogBlogOpenExistingDlg	0x0354	Open an existing blog.
BlogBlogInsertCategory	0x0355	Inserts a category dropdown into the document.
TableWrapping	0x0356	Changes the wrapping in a table.
FormatTheme	0x0357	Has no effect.
EditIMEReconversion	0x0359	Reconvert using IME.
HelpShowHide	0x035A	Show/Hide the Office Assistant.
InsertPictureBullet	0x035C	Inserts a picture as a bullet.
TableProperties	0x035D	Changes the height and width of the rows and columns in a table.
EmailSignatureOptions	0x035E	Create or changes AutoSignature entries.
EmailOptions	0x035F	Changes various categories of e-mail options.
ShadingColor	0x0361	Changes the shading color of the selected text.
DistributeGeneral	0x0362	Evenly distributes selected rows/columns in a table.
MergeSplitGeneral	0x0363	Merges or splits the selected table cell(s).
ViewTogglePageBoundaries	0x0367	Switches between showing/hiding vertical margins in Print Layout View.
CreateAutoText	0x0368	Adds an AutoText entry to the active template.
ToggleFormsDesign	0x0369	Toggles Form Design mode.
ToolsAutoSummarizeBegin	0x036A	Automatically generates a summary of the active document.
EmailEnvelope	0x036B	Displays the e-mail envelope.
ViewCode	0x036E	View code for selected control.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
MenuNotesFlow	0x036F	Notes Flow Menu.
UpdateFieldsVBA	0x0370	Updates and displays the results of the selected fields.
FontColor	0x0372	Changes the color of the selected text.
UnlinkFieldsVBA	0x0373	Permanently replaces the field codes with the results.
ToggleMasterSubdocs	0x0374	Switches between hyperlinks and subdocuments.
ToolsGramSettings	0x0375	Customize Grammar Settings.
RemoveCellPartition	0x0378	Removes cell partitions.
ShowPara	0x037A	Shows/hides all non-printing paragraph marks.
DistributeRow	0x037D	Evenly distributes selected rows.
EditGoTo	0x0380	Jumps to a specified place in the active document.
DeleteHyperlink	0x0381	Replaces a hyperlink with its displayed text.
WebOptions	0x0382	Opens the Web Options Dialog.
FixSpellingLang	0x0383	Changes language of this word.
CreateTask	0x0384	Creates a task from the current selection.
DisplayDetails	0x0385	Displays the Details of the selected address.
SpellingAndAutoCorrect	0x0387	Adds selected suggestion as AutoCorrect replacement for this word.
EditPasteAsNestedTable	0x0388	Inserts the Clipboard contents at the insertion point.
ToolsAutoSummarize	0x0389	Automatically generates a summary of the active document.
AutoSummarizeClose	0x038A	Turns AutoSummarize view off.
AutoSummarizeUpdateFileProperties	0x038C	Updates File/Properties information with the current summary.
AutoSummarizePercentOfOriginal	0x038D	Changes the size of the automatic summary.
AutoSummarizeToggleView	0x038E	Switches how the application displays a summary: highlighting summary text, or hiding everything but the summary.
InsertOCX	0x0391	Inserts the selected OCX control or registers a new OCX control.
FormatBackground	0x0392	Displays the format background submenu.
ToolsAutoManager	0x0393	Changes various categories of automatic options, such as AutoCorrect, AutoFormat and so on.
ConvertTextBoxToFrame	0x0394	Converts a single selected textbox into a frame.
OfficeDrawingCommand	0x0395	Executes a Microsoft Office drawing command with the specified arguments.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
FormatObjectCore	0x0396	Changes the properties of the selected objects.
LetterWizard	0x0397	Wizard to create a Letter Document.
HyperlinkOpen	0x0398	Open hyperlink.
WebOpenHyperlink	0x0399	Jump to a location.
WebOpenInNewWindow	0x039A	Open in new window.
WebCopyHyperlink	0x039B	Copy shortcut.
WebAddToFavorites	0x039C	Add to Favorites.
InsertHyperlink	0x039D	Insert hyperlink.
EditHyperlink	0x039E	Edit hyperlink.
WebSelectHyperlink	0x039F	Edit text.
WebOpenFavorites	0x03A0	Open Favorites folder.
WebHideToolbars	0x03A1	Hide other toolbars.
WebOpenStartPage	0x03A2	Open Start Page.
WebGoBack	0x03A3	Backward hyperlink.
FileCloseOrExit	0x03A4	Closes the current document. If only one document is open, the application is exited.
WebGoForward	0x03A5	Forward hyperlink.
WebStopLoading	0x03A6	Stop current jump.
WebRefresh	0x03A7	Refresh current page.
ShowAddInsXDialog	0x03A8	Displays the Office AddIn Manager dialog.
MenuWebFavorites	0x03A9	Represents the "Favorites" menu. Has no effect.
WebAddress	0x03AA	Hyperlink address.
ToolsBusu	0x03AB	Has no effect.
SendToFax	0x03AC	Send this document to fax.
UpdateTocFull	0x03AD	Rebuild a table of contents or captions.
ToolsRevisionMarksAccept	0x03AE	Accepts change in current selection.
ToolsRevisionMarksReject	0x03AF	Rejects change in current selection.
ViewDocumentMap	0x03B0	Toggles state of the Heading Explorer.
FileVersions	0x03B1	Manages the versions of a document.
FormatBackgroundWatermark	0x03B2	Watermark background.
DrawTextBox	0x03B3	Inserts an empty textbox or encloses the selected item in a textbox.
ViewVBCode	0x03B4	Shows the VBA editing environment.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
FormatNumberDefault	0x03B6	Creates a numbered list based on the current defaults.
FormatMultilevelDefault	0x03B7	Creates a numbered list based on the current defaults.
DrawDuplicate	0x03BB	Duplicates the selected drawing objects.
ToolsRevisionMarksToggle	0x03BC	Toggles track changes for the active document.
ToolsBookshelfLookupReference	0x03BD	Looks up a reference for the selected word in Microsoft Bookshelf.
ToolsBookshelfDefineReference	0x03BE	Looks up a definition for the selected word in Microsoft Bookshelf.
ToolsOptionsAutoFormat	0x03BF	Changes the AutoFormat options.
FormatDrawingObject	0x03C0	Changes the properties of the selected drawing objects.
BorderLineWeight	0x03C1	Changes border line weights of the selected paragraphs, table cells, and pictures.
BorderHoriz	0x03C2	Changes the horizontal borders of the selected table cells.
BorderVert	0x03C3	Changes the vertical borders of the selected table cells.
BorderLineColor	0x03C4	Changes border line color of the selected paragraphs, table cells, and pictures.
InsertListNumField	0x03C6	Inserts a ListNum Field.
HtmlResAnchor	0x03C7	Handles Internet Assistant-style hyperlink macro buttons.
WebOpenSearchPage	0x03C8	Open Search Page.
PresentIt	0x03C9	Creates a presentation from the current document.
ToolsRevisionMarksPrev	0x03CA	Find previous change.
ToolsRevisionMarksNext	0x03CB	Find next change.
DeleteAnnotation	0x03CD	Delete comment.
ToolsOptions	0x03CE	Changes various categories of the application options.
SendToOnlineMeetingParticipants	0x03CF	Send this document to Online Meeting participant.
EditPasteAsHyperlink	0x03D0	Inserts the Clipboard contents as a hyperlink object.
BorderAll	0x03D1	Changes all the borders of the selected table cells.
ToolsSpellingHide	0x03D2	Hide background spelling errors.
ToolsGrammarHide	0x03D3	Hide background grammar errors.
FormatChangeCaseFareast	0x03D4	Changes the case of the letters in the selection.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
InsertImagerScan	0x03D5	Inserts one or more images from a scanner or digital camera.
InsertClipArt	0x03D6	Inserts a Microsoft Clip Art Gallery object.
FormatFitText	0x03D7	Apply Fit Text Property.
EditAutoText	0x03D9	Inserts or defines AutoText entries.
FormatPhoneticGuide	0x03DA	Inserts a Phonetic Guide field in the active document.
FormatCombineCharacters	0x03DB	Combine Characters.
PostcardWizard	0x03DC	Starts the postcard wizard.
ToolsDictionary	0x03DD	Translates the selected word.
ToolsConsistency	0x03E0	Checks the consistency in the active document.
SetDrawingDefaults	0x03E1	Changes the default drawing object properties.
AutoScroll	0x03E2	Starts scrolling the active document.
EditWrapBoundary	0x03E3	Edit the wrapping boundary for a picture or drawing object.
DrawVerticalTextBox	0x03E4	Inserts an empty vertical text box or encloses the selected item in a vertical textbox.
DefaultCharBorder	0x03E5	Default character border.
MenuWebGo	0x03E6	Represents the web options menu. Has no effect.
WW7_ToolsGrammar	0x03E8	Checks the proofing in the active document.
ToolsAutoCorrectHECorrect	0x03E9	Hangul and alphabet correction.
WebAddHyperlnkToFavorites	0x03EA	Add to Favorites.
FormatBackgroundSwatch	0x03EB	Changes the background of the document.
FormatBackgroundNone	0x03EC	Removes the background from the document.
FormatBackgroundMoreColors	0x03ED	Provides more color choices for the background color.
FormatBackgroundFillEffect	0x03EE	Provides fill effects for the background color.
FileSaveVersion	0x03EF	Saves a new version of a document.
WebToolbar	0x03F0	Toggle Web toolbar.
ToggleTextFlow	0x03F1	Changes text flow direction and character orientation.
IncreaseIndent	0x03F2	Increases indent or demotes the selection one level.
DecreaseIndent	0x03F3	Decreases indent or promotes the selection one level.
FileSaveHtml	0x03F4	Saves the file as an HTML document.
DefaultCharShading	0x03F7	Default character shading.
ToolsFixSynonym	0x03FA	Fixes a spelling mistake with a synonym suggestion.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ToolsOptionsBidi	0x0405	Changes the Bidirectional options.
ViewSecurity	0x0419	View document security options.
ToolsInsertScript	0x041A	Has no effect.
RemoveAllScripts	0x041B	Has no effect.
MicrosoftScriptEditor	0x041C	Has no effect.
RunToggle	0x041D	Toggles the insertion point between right-to-left and left-to-right runs.
LtrPara	0x041E	Set paragraph orientation to left-to-right.
RtlPara	0x041F	Set paragraph orientation to right-to-left.
RtlRun	0x0422	Makes the current run right-to-left.
LtrRun	0x0423	Makes the current run left-to-right.
BoldRun	0x0424	Makes the current run in the selection bold (toggle).
ItalicRun	0x0425	Makes the current run in the selection italic (toggle).
FormattingProperties	0x0426	Shows or hides Formatting Properties.
HelpContentsArabic	0x0427	Displays Help in a context of bidirectional editing.
RTLMacroDialogs	0x0428	Makes macro dialogs display right-to-left.
LTRMacroDialogs	0x0429	Makes macro dialogs display left-to-right.
InsertHorizontalLine	0x042A	Inserts a horizontal line.
InsertGraphicalHorizontalLine	0x042B	Inserts a picture horizontal line.
FramesetWizard	0x042C	Turns the current window into a frameset.
FrameSplitAbove	0x042D	Splits the active frame, adding the new frame above the current.
FrameSplitBelow	0x042E	Splits the active frame, adding the new frame below the current.
FrameSplitLeft	0x042F	Splits the active frame, adding new frame left of the current.
FrameSplitRight	0x0430	Splits the active frame, adding new frame right of the current.
FrameRemoveSplit	0x0431	Removes the current frame.
FrameProperties	0x0432	Changes the properties of the frame.
TableSelectCell	0x0433	Selects the current cell in a table.
TableInsertRowBelow	0x0434	Inserts one or more rows into the table below the current row.
TableInsertColumnRight	0x0435	Inserts one or more columns into the table to the right of the current column.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
TableDeleteTable	0x0436	Deletes the selected table.
TableInsertTableEG	0x0437	Inserts a table.
TableOptions	0x0438	Changes the height and width of the rows and columns in a table.
CellOptions	0x0439	Changes the margins and other options of a table cell.
EmailSend	0x043A	Executes the e-mail Send command of the e-mail envelope.
EmailSelectNames	0x043B	Displays the e-mail address book.
EmailCheckNames	0x043C	Verifies the recipient names in the e-mail envelope.
EmailSelectToNames	0x043D	Displays the e-mail address book to add recipients to the "To" field.
EmailSelectCcNames	0x043E	Displays the e-mail address book to add recipients to the "Cc" field.
EmailSelectBccNames	0x043F	Displays the e-mail address book to add recipients to the "Bcc" field.
EmailFocusSubject	0x0440	Switches focus to the subject field of the e-mail envelope.
EmailMessageOptions	0x0441	Displays the options dialog of the e-mail envelope.
EmailFlag	0x0442	Displays the message flag dialog of the envelope.
EmailSaveAttachment	0x0443	Saves the attachments of an e-mail envelope message.
FileNewEmail	0x0444	Creates a new e-mail message.
WebPagePreview	0x0445	Displays full pages in a Web browser.
TableInsertRowAbove	0x0448	Inserts one or more rows into the table.
PrivFunctionkey1	0x0449	Private function for f1 key.
PrivFunctionkey2	0x044A	Private function for f2 key.
PrivFunctionkey3	0x044B	Private function for f3 key.
PrivFunctionkey4	0x044C	Private function for f4 key.
PrivFunctionkey5	0x044D	Private function for f5 key.
PrivFunctionkey6	0x044E	Private function for f6 key.
PrivFunctionkey7	0x044F	Private function for f7 key.
PrivFunctionkey8	0x0450	Private function for f8 key.
PrivFunctionkey9	0x0451	Private function for f9 key.
PrivFunctionkey10	0x0452	Private function for f10 key.
PrivFunctionkey11	0x0453	Private function for f11 key.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
PrivFunctionkey12	0x0454	Private function for f12 key.
FileSaveFrameAs	0x0455	Saves a copy of the current frame document in a separate file.
ShowScriptAnchor	0x0456	Has no effect.
FramesetTOC	0x0457	Create a frameset table of content.
DiacriticColor	0x0458	Changes the color of the diacritics.
FileNewWeb	0x0459	Creates a new document based on the Normal template.
FormatThemeName	0x045A	Has no effect.
FileNewPrint	0x045B	Creates a new document based on the Normal template.
FileNewDialog	0x045C	Creates a new document based on the Normal template.
HTMLSourceRefresh	0x045E	Has no effect.
ToggleWebDesign	0x045F	Toggles Web Design mode.
HTMLSourceDoNotRefresh	0x0460	Has no effect.
ShowConsistency	0x0461	Show next formatting inconsistency.
InsertHTMLCheckBox	0x0462	Has no effect.
InsertHTMLOptionButton	0x0463	Has no effect.
InsertHTMLDropdownBox	0x0464	Has no effect.
InsertHTMLListBox	0x0465	Has no effect.
InsertHTMLTextBox	0x0466	Has no effect.
InsertHTMLTextArea	0x0467	Has no effect.
InsertHTMLSubmit	0x0468	Has no effect.
InsertHTMLImageSubmit	0x0469	Has no effect.
InsertHTMLReset	0x046A	Has no effect.
InsertHTMLHidden	0x046B	Has no effect.
InsertHTMLPassword	0x046C	Has no effect.
InsertHTMLMovie	0x046D	Has no effect.
InsertHTMLBGSound	0x046E	Has no effect.
InsertHTMLMarquee	0x046F	Has no effect.
OnlineMeeting	0x0470	Has no effect.
ShowAllFareast	0x0471	Shows or hides all nonprinting characters.
AutoFitContent	0x0475	Auto-Fit table to the contents.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
AutoFitWindow	0x0476	Auto-Fit table to the window.
AutoFitFixed	0x0478	Set table size to a fixed width.
TopRightAlign	0x0479	Aligns cell content to the top-logical right of cell.
TopCenterAlign	0x047A	Aligns cell content to the top-center of cell.
TopLeftAlign	0x047B	Aligns cell content to the top-logical left of cell.
MiddleRightAlign	0x047C	Aligns cell content to the middle-logical right of cell.
MiddleCenterAlign	0x047D	Aligns cell content to the middle-center of cell.
MiddleLeftAlign	0x047E	Aligns cell content to the middle-logical left of cell.
BottomRightAlign	0x047F	Aligns cell content to the bottom-logical right of cell.
BottomCenterAlign	0x0480	Aligns cell content to the bottom-center of cell.
BottomLeftAlign	0x0481	Aligns cell content to the bottom-logical left of cell.
ViewHTMLSource	0x0482	Has no effect.
ToolsTCSCTranslation	0x0484	Translates from Traditional Chinese to Simplified Chinese or vice-versa depending on the choice of the user.
TableWizard	0x0485	Invokes the Table Wizard add-in (Korean and Chinese).
HanjaDictionary	0x0486	Has no effect.
FormatHorizontalInVertical	0x0488	Apply Horizontal in Vertical property.
FormatTwoLinesInOne	0x0489	Apply Two Lines in One property.
FormatEncloseCharacters	0x048A	Inserts an enclosed character.
UnderlineStyle	0x048B	Formats the selection with a continuous underline.
FileSaveAsWebPage	0x048C	Saves a copy of the document in a separate file.
DrawingGrid	0x0490	Tunnel to SnapToGrid dialog.
ToolsTCSCTranslate	0x0491	Translates from Traditional Chinese to Simplified Chinese.
ToolsSCTCTranslate	0x0492	Translates from Simplified Chinese to Traditional Chinese.
ToolsTranslateChinese	0x0493	Translates from Traditional Chinese to Simplified Chinese on a computer set up with Taiwanese settings; otherwise translates from Simplified Chinese to Traditional Chinese.
ShowAllConsistency	0x0494	Show all format inconsistencies.
InsertSpecialSymbol	0x0496	Inserts a special character.
EnvelopeWizard	0x0497	Invokes the Envelope Wizard add-in (Chinese).
GreetingSentence	0x0498	Invokes the Japanese Greeting Wizard.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ViewOutlineMaster	0x0499	Displays a document outline.
ScheduleMeeting	0x049A	Schedules an Online Meeting.
WebDiscussions	0x049B	Starts Web Server Discussions.
EditPaste2	0x049C	Inserts the Clipboard contents at the insertion point.
ToolsProtect	0x04D8	Sets protection for the active document or selection.
FileUndoCheckout	0x04D9	Undo the Check Out of a Document.
ShowTableTools	0x04DA	Shows Table Tools in the Ribbon.
ShowPictureTools	0x04DB	Shows Picture Tools in the Ribbon.
SelectSimilarFormatting	0x04DC	Select all similar formatting.
MailMergeShadeFields	0x04DD	Toggles shading of merge fields.
MailMergeWizard	0x04DE	Invokes Mail Merge.
EditPasteOption	0x04DF	Inserts the Clipboard contents at the insertion point using specific recovery option.
FormatStyleVisibility	0x04E0	Changes the visibility state of the document's styles.
JapaneseGreetingOpeningSentence	0x04E1	Japanese Greeting Wizard Opening Sentence.
JapaneseGreetingClosingSentence	0x04E2	Japanese Greeting Wizard Closing Sentence.
JapaneseGreetingPreviousGreeting	0x04E3	Japanese Greeting Wizard Previous Greeting.
ModifyProperty	0x04E4	Brings up a dialog to modify a particular property.
ApplyPropertyOfSurrounding	0x04E5	Matches formatting of current selection to formatting of surrounding text for a particular property.
TranslatePane	0x04E6	Opens the translation pane.
ContinueNumbering	0x04E7	Continues paragraph numbering.
ToolsSpeech	0x04EA	Turns Speech Recognition on or off.
MailAsPlainText	0x04EB	Converts the current message to plain text.
MailAsHTML	0x04EC	Converts the current message to HTML.
CssLinks	0x04ED	Manages external CSS links.
ToolsFixHHC	0x04EE	Insert converted Hangul or Hanja text.
LineSpacing	0x04EF	Applies line spacing to the selection.
MailAsRTF	0x04F0	Converts the current message to RTF.
FileNewContext	0x04F1	Creates a new document based on the NORMAL template.
ViewSignatures	0x04F3	View the signatures in this document.
ReturnReview	0x04F4	Send this document under review.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
FileVersionsLocal	0x04F5	Manages the local versions of a document.
EndReview	0x04F6	End the review for this document.
NormalizeText	0x04F8	Make text consistent with the rest.
IgnoreConsistenceError	0x04F9	Ignore formatting inconsistency error.
IgnoreAllConsistenceError	0x04FA	Ignore all formatting inconsistency errors.
ShrinkMultiSel	0x04FB	Shrinks a multiple selection to the piece that was selected last.
FileCheckout	0x04FD	Check out a document.
FileCheckin	0x04FE	Check in a document.
LearnWords	0x04FF	Use words from document to improve speech recognition.
EditPictureEdit	0x0500	Converts the selected picture into a Drawing Canvas.
FormatDefineStyleTable	0x0502	Has no effect.
FormatDefineStyleStripes	0x0503	Has no effect.
ViewChanges	0x0504	Show or hide markup balloons.
DisplayFinalDoc	0x0505	Show insertions inline and deletions in balloons.
DisplayOriginalDoc	0x0506	Show deletions inline and insertions in balloons.
ShowChangesAndComments	0x0508	Show or hide markup balloons.
ShowComments	0x0509	Show or hide comment balloons.
ShowInsertionsAndDeletions	0x050A	Show or hide markup balloons.
ShowFormatting	0x050B	Show or hide markup balloons.
PreviousChangeOrComment	0x050D	Go to the previous insertion, deletion, or comment.
NextChangeOrComment	0x050E	Go to the next insertion, deletion, or comment.
AcceptChangesSelected	0x050F	Accepts change in current selection.
AcceptAllChangesShown	0x0510	Accepts all changes that are highlighted in the current filter settings.
AcceptAllChangesInDoc	0x0511	Accepts all changes in document, ignoring filter settings.
RejectChangesSelected	0x0512	Rejects changes and deletes comments in current selection.
RejectAllChangesShown	0x0513	Rejects all changes that are highlighted in the current filter settings.
RejectAllChangesInDoc	0x0514	Rejects all changes in document, ignoring filter settings.
DeleteAllCommentsShown	0x0515	Deletes all comments that are highlighted in the current filter settings.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
DeleteAllCommentsInDoc	0x0516	Deletes all comments in document, ignoring filter settings.
InsertNewComment	0x0517	Insert comment (includes menu).
MailMergeFieldMapping	0x0518	Mail Merge field mapping.
MailMergeAddressBlock	0x0519	Mail Merge Address Block.
MailMergeGreetingLine	0x051A	Mail Merge Greeting Line.
MailMergeInsertFields	0x051B	Mail Merge Insert Fields.
MailMergeRecipients	0x051C	Mail Merge Recipients.
MMEmailOptions	0x051D	Mail Merge E-mail Options Dialog.
MMNewDocOptions	0x051E	Mail Merge New Document Merge Options Dialog.
MMPrintOptions	0x051F	Mail Merge Print Merge Options Dialog.
MMFaxOptions	0x0520	Mail Merge Fax Options Dialog.
ViewTaskPane	0x0521	Shows or hides the Task Pane.
MailMergeEditAddressBlock	0x0523	Edit Address Block.
MailMergeEditGreetingLine	0x0524	Edit Greeting Line.
ApplyPropertyOfOriginal	0x0525	Matches formatting of current selection to formatting of original selection for a particular property.
ApplyFormattingOfSurrounding	0x0529	Applies formatting of surrounding text to current selection.
ApplyFormattingOfOriginal	0x052A	Applies formatting of original selection to current selection.
LettersWizardJToolbar	0x052B	Displays or hides the Japanese Greeting Wizard Toolbar.
InsertWebComponent	0x052C	Has no effect.
MailMergePropagateLabel	0x052D	Populate all mail merge labels in the document.
MailMergeFindEntry	0x052E	Finds a specified entry in a mail merge data source.
ShowSmPane	0x052F	Displays the Document Updates Pane.
SignatureLineMenuSign	0x0530	Signs with a **digital signature**.
ResetFormField	0x0531	Resets the selected form field to its default value.
DisplaySharedWorkspacePane	0x0532	Displays the Document Management pane.
FileVersionsServer	0x0533	Manages the server versions of a document.
DisplayForReview	0x0534	Selects viewing mode for revisions and comments.
AnnotationEdit	0x0535	Edit comment.
ShowAllAuthors	0x0539	Show or hide markup balloons for all authors.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
Translate	0x053A	Opens the translation pane.
MailMergeSetDocumentType	0x053B	Sets or clears the Mail Merge document type.
FormatField	0x053C	Inserts a field in the active document.
ReplaceEmailSignature	0x053D	Replaces the current AutoSignature with a different one.
IncreaseParagraphSpacing	0x053E	Increases paragraph spacing by 6 points.
DecreaseParagraphSpacing	0x053F	Decreases paragraph spacing by 6 points.
ReplyToAnnotation	0x0540	Reply to comment.
ToolsWordCountRecount	0x0541	Updates the word count statistics of the active document.
ToolsWordCountList	0x0542	Displays the word count statistics of the active document.
FormatStyleModify	0x0543	Modifies selected style.
FormatStyleByExample	0x0544	Creates a style out of the currently selected text.
SelectNumber	0x0545	Selects the paragraph number.
RestartNumbering	0x0546	Restarts paragraph numbering.
FixUIMChange	0x0547	Replaces this word by the selected suggestion.
UIMCorrectionUI	0x0548	Brings up the correction UI for the Tablet Input Panel.
FixUIMDeleteWord	0x0549	Removes the word.
ClearFormatting	0x054A	Clears formatting and styles from selected text.
ToolsOptionsEditCopyPaste	0x054C	Changes the editing options.
TxbxAutosize	0x054D	Changes the selected drawing object to autosize.
EditPasteAppendTable	0x054E	Inserts the clipboard contents at the insertion point.
ReviewingPane	0x054F	Opens a summary pane for viewing and editing document revisions (toggle).
OutlinePromoteHeading1	0x0550	Promotes the selected text to Heading 1 style.
ToolsOptionsSecurity	0x0551	Changes security options.
FileSearch	0x0553	Brings up the Search UI workpane.
FormattingPane	0x0554	Applies, creates, or modifies styles and formatting.
DeleteStyle	0x0555	Deletes the current style.
RenameStyle	0x0556	Renames the current style.
LabelOptions	0x0557	Label Options Dialog.
EnvelopeSetup	0x0558	Envelopes Option Dialog.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
MailMergeToEMail	0x0559	Sends the results of the mail merge to an e-mail message.
MailMergeToFax	0x055A	Sends the results of the mail merge to Fax.
MailMergeToolbar	0x055B	Displays or hides the Mail Merge Toolbar.
MailMergeCreateList	0x055C	Create an Office Address List.
MailMergeEditList	0x055D	Edit an Office Address List.
TableAutoFormatStyle	0x055F	Applies a table style to a table.
LicenseVerification	0x0561	Displays the dialog box for activating the product.
FormatConsistencyCheck	0x0562	Check for formatting consistency.
SendForReview	0x0563	Send this document for review.
SignOutOfPassport	0x0564	Signs out of Windows Live ID.
ShowRepairs	0x0565	Shows all repairs made to the document during Crash Recovery.
ToolsEServices	0x0567	Opens the eServices dialog.
DeleteStructure	0x0568	Remove XML Element.
ViewXMLStructure	0x056C	Show XML Structure Pane.
GotoTableOfContents	0x056D	Selects the first table of contents in the document.
UpdateTableOfContents	0x056E	Updates the first table of contents in the document.
OutlineLevel	0x056F	Sets the selected paragraphs to the heading level.
ShowLevel	0x0570	Displays the selected level headings only.
ToggleCharacterCode	0x0571	Toggles a character code and a character.
ToolsOptionsSmartTag	0x0573	Changes the Smart Tag options.
EmailFocusIntroduction	0x0574	Switches focus to the introduction field of the e-mail envelope.
EditPasteFromExcel	0x0575	Inserts the Clipboard contents at the insertion point.
InsertStyleSeparator	0x0576	Joins two paragraphs together creating leading emphasis.
FixBrokenText	0x0577	Has no effect.
ReadingModePageview	0x0587	Show pages as they will look if printed.
ToggleXMLTagView	0x0588	Toggle XML Tag View on or off.
SchemaLibrary	0x0589	Displays the Schema Library dialog.
ResearchLookup	0x058A	Looks up the word in the research tool.
WindowArrangeSideBySide	0x058B	Arranges two windows side by side.
SqmDialog	0x058C	Opens the Customer Feedback Options dialog.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
InsertInkComment	0x058D	Insert ink comment.
StyleLockDown	0x058E	Locks styles in a document.
SyncScrollSideBySide	0x058F	Enables synchronous scrolling of two windows side-by-side.
ResetSideBySide	0x0590	Resets window position for side-by-side.
XMLOptions	0x0591	Changes XML settings for this document.
XMLDocument	0x0592	Applies XML Transforms to this document.
FormattingRestrictions	0x0593	Style lock down settings.
FilePermissionMenu	0x0596	File Permission Menu.
FPUnprotected	0x0597	"Unprotected" template (DRM).
FPConfidential	0x0598	"Confidential" template (DRM).
FPAdminTemplates	0x059C	Administrator-defined template (DRM).
MyPermission	0x059D	Displays the DRM usage permissions for the user.
ToggleThumbnail	0x059E	Toggles thumbnail view.
ToolsThesaurusRR	0x059F	Displays synonyms for the selected word in the Research pane.
DoNotDistribute	0x05A0	Permission toggle button on toolbar.
ToggleReadingMode2Pages	0x05A2	Toggles 2 Pages view.
ToggleReadingModeInk	0x05A3	Enables Ink Annotation.
ReadingModeInkOff	0x05A4	Unlocks document for ink.
InsertSoundComment	0x05A5	Inserts a sound object into the document.
EditFindReadingMode	0x05A6	Finds the specified text or the specified formatting.
UseBalloons	0x05A7	Show all revisions in balloons.
NeverUseBalloons	0x05A8	Show all revisions inline.
NoInsertionDeletionBalloons	0x05A9	Show comments and formatting revisions in balloons.
ShowInkAnnotations	0x05AA	Show or hide ink annotations.
DeleteAllInkAnnotations	0x05AB	Delete all ink annotations.
ToggleReadingModeHelp	0x05AC	Help for Ink Annotation.
HelpContactUs	0x05AD	Brings up the Web browser and displays the Contact Us page.
HelpCheckForUpdates	0x05AE	Brings up the Web browser and displays the Product Update page.
BlogBlogInsertCategories	0x05AF	Inserts a category into the document.
ToggleToolbars	0x05B0	Toggles Toolbars.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ReadingMode	0x05B1	Toggles full screen reading.
ApplyStructure	0x05B2	Apply XML Element.
Research	0x05B3	Initiates the Research pane.
XmlAttr	0x05B4	Modify attribute settings of an XML element.
FPSelectUser	0x05B5	Select user in permission menu.
ViewDocumentMapReadingMode	0x05B6	Toggles state of the Heading Explorer.
ReadingModeMini	0x05B7	Switch to full screen reading.
ReadingModeLookup	0x05B8	Lookup tools for reading.
ReadingModeGrowFont	0x05B9	Increases the font size for full screen reading.
ReadingModeShrinkFont	0x05BA	Decreases the font size for full screen reading.
FaxService	0x05BB	Send this document to fax over the Internet.
GettingStartedPane	0x05BC	Has no effect.
FilePermission	0x05BD	Restricts permission for a document.
DocumentActionsPane	0x05BE	Smart Document Pane.
ReadingModeLayout	0x05BF	Switch to full screen reading.
AnnotInkPen	0x05C0	Ink Comment Pen.
AnnotInkEraser	0x05C1	Ink Comment Eraser.
CopyInkAsText	0x05C2	Copies the ink selection and puts its text equivalent on the Clipboard.
InsertInkAnnotations	0x05C3	Insert ink annotation.
EmailChooseAccount	0x05C4	Allows choosing an e-mail account.
EmailAttachmentOptions	0x05C5	Toggles display of the Attachment Options task pane.
InkEraser	0x05C6	Ink Eraser.
CloseReadingMode	0x05C8	Stops full screen reading.
InkAnnotationEraser	0x05C9	Ink Eraser.
DocInspector	0x05CA	Document Inspector.
GoToFurthestReadPg	0x05CB	Goes to furthest read page.
GoToFirstPg	0x05CC	Goes to first page.
GoToLastPg	0x05CD	Goes to last page.
BackHistoryItem	0x05CE	Goes back to most recent screen.
ForwardHistoryItem	0x05CF	Goes forward to next visited screen.
JumpToScrn	0x05D0	Jump to screen label for screen navigator popup menu.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
JumpToHeading	0x05D1	Jump to Heading label from screen navigator popup menu.
SaveAsQuickFormatSet	0x05D5	Saves the current Quick Style list as a new Quick Style set.
InsertAlignmentTab	0x05DB	Inserts an alignment tab at the insertion point.
ResetParagraphFormatting	0x076C	Resets paragraph formatting.
CharacterRemoveStyle	0x076D	Clears character style from selection.
RestoreCharacterStyle	0x076E	Restores character style and removes direct formatting.
CharacterClearFormatting	0x076F	Clears character properties from formatting.
SeparateList	0x0770	Separates current paragraph into a new list.
JoinToPreviousList	0x0771	Joins to previous list.
SetNumberingValue	0x0774	Sets the numbering value.
EquationToggle	0x0775	Insert an equation.
EquationProfessionalFormat	0x0776	Convert to Professional Format.
EquationLinearFormat	0x0777	Convert to Linear Format.
AdjustListIndents	0x0778	Changes the position of the list.
ShowTasks	0x0779	Shows workflow tasks for this Document.
InsertSignatureLine	0x077A	Insert digital signature line.
EquationMathAutoCorrect	0x077B	Add or delete Math AutoCorrect entries.
InsertCitation	0x077C	Insert citation.
InsertBibliography	0x077D	Insert bibliography.
SelectBibliographyStyle	0x077E	Select bibliography style.
BibliographySourceManager	0x0780	Opens the Source Manager dialog box.
EquationInsertSymbol	0x0781	Insert equation symbol.
BibliographyCreateSource	0x0782	Opens the Create Source dialog box.
LockPolicyLabel	0x0783	Locks .**policy labels**. for this document.
UnlockPolicyLabel	0x0784	Unlocks .policy labels. for this document.
InsertPolicyLabel	0x0785	Inserts .policy labels. for this document.
FillPolicyLabel	0x0786	Fills in the .policy labels. for this document.
InsertPolicyBarcode	0x0787	Inserts barcode.
InsertBuildingBlockIP	0x0789	Inserts the building block at the insertion point.
InsertBuildingBlockHeader	0x078A	Inserts the building block in the header.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
InsertBuildingBlockFooter	0x078B	Inserts the building block in the footer.
InsertBuildingBlockBeginSection	0x078C	Inserts the building block at the beginning of the current section.
InsertBuildingBlockEndSection	0x078D	Inserts the building block at the end of the current section.
InsertBuildingBlockBeginDocument	0x078E	Inserts the building block at the beginning of the document.
InsertBuildingBlockEndDocument	0x078F	Inserts the building block at the end of a document.
AdvertisePublishAs	0x0790	Advertise Publish Export to PDF and XPS.
ShowMarkupArea	0x0791	Show or hide markup area highlight.
SwitchNavigationWindow	0x0793	Choose navigation window.
ToolsAutoCorrectManager	0x0794	Adds or deletes AutoCorrect or Math AutoCorrect entries.
ReadingModeAllowEditing	0x0795	Allow or disallow typing while reading.
ReadingModePageMarginsType	0x0796	Hide the margins on the printed page to display larger text.
EquationInsert	0x0797	Insert an equation.
StartWorkflow	0x0798	Starts a workflow for this document.
DropCapGallery	0x0799	Opens the list of drop cap styles.
PageOrientationGallery	0x079B	Opens the list of options for page orientation.
FormatStyleManagement	0x079C	Manage the document or stylesheet.
UpdateStyle	0x079D	Updates the current style based on the selected text.
NewStyle	0x079E	New quick style from selection.
FormattingPaneCurrent	0x079F	Lists the current formatting in the document.
ListAdvanceToVBA	0x07A0	Advances the numbering value.
ResetAdvanceToVBA	0x07A1	Resets the value of the number to advance to.
DownloadPictures	0x07A2	Reloads the e-mail message, allowing linked pictures to be downloaded from the Internet.
ViewZoomTwoPage	0x07A3	Scales the editing view to see the two pages in page layout view.
SymbolMRUGallery	0x07A6	Symbol MRU Gallery.
QuickFormatsGallery	0x07A7	Opens the list of Quick Styles.
QuickFormatsThemeGallery	0x07A8	Opens the list of Quick Style sets.
ClearAllFormatting	0x07A9	Clears formatting and styles from selected text.
TogglePanningHand	0x07AA	Displays the panning state of the document.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
BulletsGallery	0x07AB	Opens the Bullet gallery.
NumberingGallery	0x07AC	Opens the Numbering Gallery.
MenuShowSourceDocuments	0x07AD	Shows or hides source documents.
PageMarginsGallery	0x07B1	Opens the list of options for page margins.
CharScaleDialog	0x07B2	Opens the list of font scaling percentages.
AllShapesGallery	0x07B4	Displays the shapes that are available to insert.
RotateObjectGallery	0x07B5	Opens the list of options for rotating objects.
LineStyleGallery	0x07B6	Opens the list of line styles.
LineWidthGallery	0x07B7	Opens the list of line widths.
ArrowStyleGallery	0x07B8	Opens the list of arrow styles.
ChangeShapesGallery	0x07B9	Displays the shapes that are available to substitute.
TexturesGallery	0x07BA	Opens the list of textures.
FontColorPicker	0x07BB	Opens the list of font colors.
ColumnsGallery	0x07BC	Opens the list of preset column layouts.
EquationIncreaseAlignment	0x07BD	Increase alignment point after a manual break.
EquationDecreaseAlignment	0x07BE	Decrease alignment point after a manual break.
EquationChangeStyle	0x07BF	Change equation style (Display or Inline).
DocEncryption	0x07C0	Add document encryption.
BlogBlogAccountOptionsDlg	0x07C1	Changes blog account settings.
EquationInsertRowBefore	0x07C2	Insert a row into a matrix object.
EquationInsertRowAfter	0x07C3	Insert a row into a matrix object.
FileSendBlog	0x07C4	Sends the active document to a blog.
EquationInsertColumnBefore	0x07C5	Insert a column into a matrix object.
EquationInsertColumnAfter	0x07C6	Insert a column into a matrix object.
EquationDeleteRow	0x07C7	Delete a row from a matrix object.
EquationDeleteColumn	0x07C8	Delete a column from a matrix object.
EquationVerticalCenter	0x07C9	Set equation vertical alignment to Center.
EquationVerticalTop	0x07CA	Set equation vertical alignment to Top.
EquationVerticalBottom	0x07CB	Set equation vertical alignment to Bottom.
EquationHorizontalCenter	0x07CC	Set equation horizontal alignment to Center.
EquationHorizontalLeft	0x07CD	Set equation horizontal alignment to Left.
EquationHorizontalRight	0x07CE	Set equation horizontal alignment to Right.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
EquationShowHideLowerLimit	0x07CF	Show/Hide N-ary lower limit.
EquationShowHideUpperLimit	0x07D0	Show/Hide N-ary upper limit.
EquationShowHideRadicalDegree	0x07D1	Show/Hide the radical degree.
EquationShowHideOpeningDelimiter	0x07D2	Show/Hide the left character.
EquationShowHideClosingDelimiter	0x07D3	Show/Hide the right character.
EquationAutoProfessionalFormat	0x07D4	Automatically convert equation to Professional Format.
SignatureLineMenuDetails	0x07D5	Digital signature line details.
SignatureLineMenuSetup	0x07D6	Digital signature line setup.
SignatureLineMenuUnSign	0x07D7	Removes digital signature.
EquationFractionGallery	0x07D8	Equation fraction gallery.
EquationIntegralGallery	0x07D9	Equation integral gallery.
EquationRadicalGallery	0x07DA	Equation radical gallery.
EquationNaryGallery	0x07DB	Equation N-ary gallery.
EquationDelimiterGallery	0x07DC	Equation delimiter gallery.
EquationScriptGallery	0x07DD	Equation script gallery.
NextComment	0x07DE	Go to the next comment.
PreviousComment	0x07DF	Go to the previous comment.
DefineNewBullet	0x07E0	Defines a new bullet.
DefineNewNumber	0x07E1	Defines a new number format.
CreateBuildingBlockFromSel	0x07E2	Creates a building block from the current selection.
FooterGallery	0x07E3	Footer Gallery.
HeaderGallery	0x07E4	Header Gallery.
CoverPageGallery	0x07E5	Cover Page Gallery.
LegoPageNumGallery	0x07E7	Page Numbers Gallery.
LegoPageNumPageGallery	0x07E8	Page Numbers (Page) Gallery.
LegoWatermarkGallery	0x07E9	Watermark Gallery.
LegoPageNumTopGallery	0x07EA	Page Numbers (Top) Gallery.
LegoPageNumBottomGallery	0x07EB	Page Numbers (Bottom) Gallery.
LegoEquationsGallery	0x07EC	Equations Gallery.
LegoTablesGallery	0x07EE	Tables Gallery.
LegoCommonPartsGallery	0x07F0	Common Parts Gallery.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
CreateCommonFieldBlockFromSel	0x07F3	Creates a new Common Field building block from the current selection.
CreateCoverPageBlockFromSel	0x07F4	Creates a new Cover Page building block from the current selection.
CreateEquationBlockFromSel	0x07F5	Creates a new Equation building block from the current selection.
CreateFooterBlockFromSel	0x07F6	Creates a new Footer building block from the current selection.
CreateHeaderBlockFromSel	0x07F7	Creates a new Header building block from the current selection.
CreatePageNumFromSel	0x07F9	Creates a new Page Number building block from the current selection.
CreatePageNumTopFromSel	0x07FA	Creates a new Page Number (Top) from the current selection.
CreatePageNumBottomFromSel	0x07FB	Creates a new Page Number (Bottom) from the current selection.
CreateTableBlockFromSel	0x07FC	Creates a new Table building block from the current selection.
CreatePageNumPageBlockFromSel	0x07FD	Creates a new Page Number (Page) from the current selection.
CreateWaterMarkBlockFromSel	0x07FF	Creates a new Watermark Building Block from the current selection.
EquationEdit	0x0800	Insert/Edit an equation.
DefaultCondensed	0x0801	Sets the font character spacing of the selection to condensed.
DefaultExpanded	0x0802	Sets the font character spacing of the selection to expanded.
EquationSymbolsGallery	0x0803	Equation symbols gallery.
WordSearchLibraries	0x080D	Search libraries.
InsertOCXDialog	0x080E	Inserts the selected ActiveX control.
ToggleXMLStructure	0x080F	Shows/Hides XML Structure Pane.
XmlSchema	0x0810	Changes the XML Schema options.
XmlExpansionPacks	0x0811	Changes the XML Expansion Pack options.
OartCommand	0x0812	Execute an OfficeArt undo or redo command.
BuildingBlockOrganizer	0x0813	Manages Building Block entries.
CompareDocumentsCompare	0x0814	Compare two versions of a document (legal blackline).
CompareDocumentsCombine	0x0815	Combine revisions from multiple authors into a single document.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
CompareDocumentsLastMajor	0x0816	Compare this document with the last major version published on the server.
CompareDocumentsLastMinor	0x0817	Compare this document with the last version saved on the server.
CompareDocumentsVersion	0x0818	Compare this document with a specific version saved on the server.
InsertSignatureLineMenuItem	0x0819	Insert digital signature line.
UpdateFieldsTable	0x081A	Updates and displays the results of the selected fields.
UpdateFieldsIndex	0x081B	Updates and displays the results of the selected fields.
ToolsHyphenationAutoOn	0x081C	Changes the automatic hyphenation setting for the active document.
ToolsHyphenationAutoOff	0x081D	Changes the automatic hyphenation setting for the active document.
MailMergeClearDocumentType	0x081E	Clears the Mail Merge document type.
MailMergeSetDocTypeFormLetter	0x081F	Sets the Mail Merge document type to Form Letter.
MailMergeSetDocTypeEmail	0x0820	Sets the Mail Merge document type to E-mail.
MailMergeSetDocTypeFax	0x0821	Sets the Mail Merge document type to Fax.
MailMergeSetDocTypeEnvelope	0x0822	Sets the Mail Merge document type to Envelope.
MailMergeSetDocTypeLabel	0x0823	Sets the Mail Merge document type to Label.
MailMergeSetDocTypeDirectory	0x0824	Sets the Mail Merge document type to Directory.
UpdatePolicyLabels	0x0825	Updates .policy labels.
BlogBlogOpenBlogSite	0x0826	Opens the blog's Web site.
StyleQuickFormat	0x0827	Add or remove the current style from the Quick Style list.
RefTipLangGallery	0x0828	Translation ScreenTip Gallery.
RefTipSelectLang	0x0829	Show or Hide Translation ScreenTip.
EquationInsertEmptyStructure	0x082C	Insert equation structure.
RemoveSimilarFormatting	0x082D	Removes all similar formatting.
TogglePropertyPanel	0x082E	Turns on or off the Property Editor.
OutlineLevelGallery	0x082F	Outline Level Gallery.
BreaksGallery	0x0830	Breaks Gallery.
ToolsLineNumOff	0x0831	Turns off line numbering for the current document.
ToolsLineNumContinuous	0x0832	Turns off line numbering for the current document.
ToolsLineNumRestPage	0x0833	Turns off line numbering for the current document.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ToolsLineNumResetSection	0x0834	Turns off line numbering for the current document.
ToolsLineNumSuppress	0x0835	Turns off line numbering for the current document.
TocOutlineLevelGallery	0x0836	Outline Level Gallery for the Table of Contents.
AcceptChangesAndAdvance	0x0837	Accepts change in current selection.
RejectChangesAndAdvance	0x0838	Rejects changes and deletes comments in current selection.
CreateSharedWorkspace	0x0839	Creates a document workspace.
SaveToDocMgmtServer	0x083A	Saves to Document Management Server.
DisplayDocumentManagementPane	0x083B	Displays the Document Management Pane.
FreezeLayout	0x083C	Freeze wrapping width.
NavBack	0x083D	Jump back to the previous page in full screen reading.
NavForward	0x083E	Jump forward to the next page in full screen reading.
MenuManageDocument	0x083F	Manage.
MenuShareDocument	0x0840	Shares a copy.
MenuFinalizeDocument	0x0841	Finalize Document.
MenuSignaturesDocument	0x0842	View any digital signatures for this document.
MarkAsReadOnly	0x0843	Marks as Final.
SignDocument	0x0844	Sign this document.
AddDigitalSignature	0x0845	Add a digital signature.
ShowReviewerFilter	0x0846	Menu for showing reviewers.
SigningServices	0x0847	Signing services.
BibInsertSource	0x0848	Insert New Bibliography Source.
ControlProperties	0x0849	Shows the properties of the current control.
FillColorPicker	0x084A	Fill Color Picker.
LineColorPicker	0x084B	Line Color Picker.
ToggleDocumentText	0x084C	Shows or hides the main text layer in page layout view.
HeadFootDiffFirstPage	0x084D	Turns on a different header and footer for the first page.
HeadFootDiffOddEvenPage	0x084E	Turns on a different header and footer for odd and even pages.
InsertNewPage	0x084F	Inserts a new page break at the insertion point.
HideOutline	0x0850	Turns off the document outline.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
BrightnessGallery	0x0851	Brightness Gallery.
ContrastGallery	0x0852	Contrast Gallery.
ChangeCaseGallery	0x0853	Change Case Gallery.
ShadingColorPicker	0x0856	Shading Color Picker.
BringForward	0x0857	Brings the selected drawing objects forward.
BringToFront	0x0858	Brings the selected drawing objects to the front.
SendBackward	0x0859	Sends the selected drawing objects backward.
SendToBack	0x085A	Sends the selected drawing objects to the back.
EquationInsertArgumentBefore	0x085C	Insert a new argument.
EquationInsertArgumentAfter	0x085D	Insert a new argument.
EquationDeleteArgument	0x085E	Delete an argument.
EquationRemoveStructure	0x085F	Remove the equation structure.
EquationRemoveSubscript	0x0860	Remove the subscript.
EquationRemoveSuperscript	0x0861	Remove the superscript.
EquationStackedFraction	0x0862	Stacked fraction.
EquationNoBarFraction	0x0863	No-Bar fraction.
EquationSkewedFraction	0x0864	Skewed fraction.
EquationLinearFraction	0x0865	Linear fraction.
EquationStretchDelimiters	0x0866	Stretch delimiter characters.
EquationShowHidePlaceholders	0x0867	Show or hide placeholders in a matrix.
EquationScriptAlignment	0x0868	Change scripts alignment.
OutlookInsertFile	0x0869	Launches the Insert file attachment dialog for e-mail.
EquationMatchDelimiters	0x086A	Match delimiters to argument height.
EquationNaryLimitLocation	0x086B	Change N-ary limits location.
EquationLimitLocation	0x086C	Change limit location.
EquationBarLocation	0x086E	Change bar location.
EquationStretchNaryOperator	0x086F	Stretch N-ary characters.
EquationGroupingCharacterLocation	0x0870	Change grouping character location.
EquationArrayExpansion	0x0871	Expand equation array to the column width.
EquationExpansion	0x0872	Expand equation to equation array width.
EquationIncreaseArgumentSize	0x0873	Increase argument size.
EquationDecreaseArgumentSize	0x0874	Decrease argument size.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
EquationRecognizedFunctions	0x0875	Add or delete equation recognized functions.
SearchOfficeOnline	0x0876	Opens the Search Office Online page.
BrowseForThemes	0x0877	Opens a dialog to browse for themes.
ListLevelGallery	0x0878	Opens the List Level Gallery.
OutlineNumberingGallery	0x0879	Opens the Multilevel List Gallery.
DefineNewList	0x087A	Defines a new list.
DefineNewListStyle	0x087B	Defines a new list style.
AddToContacts	0x087C	Adds selected business card to Contacts.
PropertiesGallery	0x087D	Properties Gallery.
ToggleDocumentActionBar	0x087E	Shows or hides the Message Bar.
StyleApplyPane	0x087F	Applies, creates, or modifies styles and formatting.
EquationManualBreak	0x0881	Insert or remove a manual break in equations.
EquationAlignThisCharacter	0x0882	Insert or remove an alignment point in equations.
EquationAlignAtEquals	0x0883	Insert or remove an alignment point in equations.
EmailStationeryOptions	0x0884	Creates or changes Stationery entries.
SetListLevelVBA	0x0885	Sets the list level.
ApplyQuickFormat	0x0887	Applies the selected style from the Quick Style set.
ApplyQuickStyleSet	0x0888	Applies the selected Quick Style set.
OutlookViewZoom	0x0889	Scales the editing view.
DeleteBuildingBlock	0x088A	Deletes the surrounding building block.
EquationNormalText	0x088B	Make the selection Normal Text (toggle).
EquationFunctionGallery	0x088C	Equation function gallery.
EquationAccentGallery	0x088D	Equation accent gallery.
EquationLimitGallery	0x088E	Equation limit gallery.
EquationOperatorGallery	0x088F	Equation operator gallery.
EquationMatrixGallery	0x0890	Equation matrix gallery.
MenuReadingTools	0x0891	Reading tools for full screen reading.
GoToNextReadingPage	0x0892	Moves to the next page in full screen reading.
GoToPrevReadingPage	0x0893	Moves to the previous page in full screen reading.
ReadingMode1Page	0x0894	Show 1 Page view.
ReadingMode2Pages	0x0895	Show 2 Pages view.
MenuReadingViewOptions	0x0896	View options for full screen reading.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ScrnNav	0x0897	Display Screen Navigator Menu.
ReadingModePageMargins	0x0898	Show the actual page unaltered.
ReadingModePageNoMargins	0x0899	Zoom in, making the text larger, and suppress the margins to make sure the page remains visible.
ReadingModePageAutoMargins	0x089A	Hide the margins if the page display is too small to read.
ToggleDontOpenAttachInFullScreen	0x089C	Prevents opening of attachments in full screen.
TrackChangesOptions	0x089F	Changes track changes options.
ColorPickerShadowE1o	0x08A0	Opens the shadow color picker.
ColorPicker3DE1o	0x08A1	Opens the 3-D color picker.
GradientGallery	0x08A2	Gradient Gallery.
BarCodeGroup	0x08A3	Has no effect.
AsianLayoutFlyout	0x08A4	Asian Layout Menu.
JustifyFlyout	0x08A5	Displays the Justify menu for East Asian languages.
TOAGroup	0x08A6	Has no effect.
JapaneseGreetingFlyout	0x08A7	Japanese Greeting flyout anchor.
JustifyParaSpecial	0x08A8	Aligns the paragraph at both the left and the right indent.
JustifyParaLow	0x08A9	Aligns the paragraph - Arabic setting.
JustifyParaMedium	0x08AA	Aligns the paragraph - Arabic setting.
JustifyParaHigh	0x08AB	Aligns the paragraph - Arabic setting.
JustifyParaThai	0x08AC	Aligns the paragraph - Thai setting.
IndentLeftSpinner	0x08AD	Has no effect.
IndentRightSpinner	0x08AE	Has no effect.
SpacingBeforeSpinner	0x08AF	Has no effect.
SpacingAfterSpinner	0x08B0	Has no effect.
HeaderPositionSpinner	0x08B1	Has no effect.
FooterPositionSpinner	0x08B2	Has no effect.
SpacingLabel	0x08B3	Has no effect.
IndentLabel	0x08B4	Has no effect.
InsertFormControlsGallery	0x08B5	Opens the list of form controls.
ShapeHeightSpinner	0x08B6	Displays the Shape Height spin box.
ShapeWidthSpinner	0x08B7	Displays the Shape Width spin box.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
HighlightColorPicker	0x08B8	Opens the highlight color picker.
BorderColorPicker	0x08B9	Opens the border color picker.
BackgroundColorPicker	0x08BA	Opens the background color picker.
GoToHeader	0x08BB	Move between the header and footer.
GoToFooter	0x08BC	Move between the header and footer.
FormatBackgroundColor	0x08BD	Sets the document background color.
CancelHighlightMode	0x08BE	Applies color highlighting to the selection.
UnderlineGallery	0x08BF	Opens the list of underline styles.
UnderlineColorPicker	0x08C0	Opens the underline color picker.
TextFlowGallery	0x08C1	Opens the list of text flow options.
CellAlignmentGallery	0x08C2	Opens the list of table cell alignment options.
PicturePositionGallery	0x08C3	Opens the list of picture position options.
InkingGroup	0x08C4	Has no effect.
AdvancedBrightnessContrast	0x08C5	Changes the properties of the selected drawing objects.
RecolorGallery	0x08C7	Recolor Gallery.
ShadowStyleGallery	0x08C8	Opens the list of shadow styles.
Style3DGallery	0x08C9	Opens the list of 3-D style options.
Direction3DGallery	0x08CA	Opens the list of 3-D direction options.
DepthGallery	0x08CB	Opens the extrusion depth gallery.
SurfaceMatGallery	0x08CC	Opens the 3D surface material gallery.
Lighting3DGallery	0x08CD	Opens the list of 3-D lighting options.
WordArtGallery	0x08CE	Opens the list of WordArt options.
InsertWordArtGallery	0x08CF	Represents a Microsoft WordArt Gallery.
PageSizeGallery	0x08D0	Opens the list of page size options.
InsertTableGallery	0x08D1	Opens the list of table templates.
ShapeStyleGallery	0x08D2	Opens the list of shape styles.
WordArtShapeGallery	0x08D3	Opens the list of WordArt shapes.
XMLGroup	0x08D4	Has no effect.
ReviewingPaneHorizontal	0x08D6	Shows or hides a summary pane for viewing and editing document revisions (horizontal).
ReviewingPaneVertical	0x08D7	Shows or hides a summary pane for viewing and editing document revisions (vertical).

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
EquationScriptLocation	0x08D8	Change scripts location.
EquationInsertStructure	0x08D9	Insert equation structure.
BulletsNumberingStyleDialog	0x08DA	Bullets and Numbering Style Definition Dialog Box.
SaveCurrentTheme	0x08DB	Saves the current theme.
AutoTextGallery	0x08DE	AutoText Gallery.
TextBoxGallery	0x08DF	Text Box Gallery.
BibliographyGallery	0x08E0	Bibliography Gallery.
CreateAutoTextBlockFromSel	0x08E1	Creates a new AutoText Building Block from the current selection.
CreateTextBoxBlockFromSel	0x08E2	Creates a new Text Box Building Block from the current selection.
CreateLayoutBlockFromSel	0x08E3	Creates a new Layout Building Block from the current selection.
SaveCoverPageBlock	0x08E4	Saves the current cover page as a new building block.
SaveHeaderBlock	0x08E5	Saves the current header as a new building block.
SaveFooterBlock	0x08E6	Saves the current footer as a new building block.
SavePageNumTopBlock	0x08E7	Saves the current page number (top) as a new building block.
SavePageNumBottomBlock	0x08E8	Saves the current page number (bottom) as a new building block.
SavePageNumBlock	0x08E9	Saves the current page number as a new building block.
ViewHeaderOnly	0x08EA	Displays the header in page layout view.
EquationLeftJustification	0x08EB	Left-align equation.
EquationRightJustification	0x08EC	Right-align equation.
EquationCenteredJustification	0x08ED	Center equation.
EquationCenteredAsGroupJustification	0x08EE	Center equations as a group.
UxGalWordTableStyles	0x08EF	Opens the list of table styles.
WordTableStylesHeaderRow	0x08F0	Header Row.
WordTableStylesTotalRow	0x08F1	Total Row.
WordTableStylesFirstColumn	0x08F2	First Column.
WordTableStylesLastColumn	0x08F3	Last Column.
WordTableStylesBandedRows	0x08F4	Banded Rows.
WordTableStylesBandedColumns	0x08F5	Banded Columns.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ClearTableStyle	0x08F6	Clears table style formatting.
ApplyTableStyle	0x08F7	Applies the selected table style.
ModifyTableStyle	0x08F8	Modifies the table style.
CheckCompatibility	0x08F9	Check document compatibility.
CompareTranslationBaseDocuments	0x08FA	View changes in the source document.
FontSchemePicker	0x08FB	Opens the font scheme picker.
ColorSchemePicker	0x08FC	Opens the color scheme picker.
StyleMatrixPicker	0x08FD	Opens the style matrix picker.
ThemeGallery	0x08FE	Opens the list of available themes.
EquationMatrixSpacing	0x08FF	Set the spacing of a matrix.
EquationEquationArraySpacing	0x0900	Set the spacing of an equation array.
DrawingAdvancedLayout	0x0901	Changes the advanced layout properties of the selected drawing objects.
ReadingModeToPrintView	0x0902	Switch from full screen reading mode to print view.
LineSpacingMenu	0x0904	Applies line spacing to the selection.
FileSendPdf	0x0905	Sends the active document through e-mail as PDF attachment.
FileSendXps	0x0906	Sends the active document through e-mail as XPS attachment.
CreateNewColorScheme	0x0909	Opens the create new color scheme dialog.
FileSaveWordDotx	0x090A	Save file as a [ECMA-376] template.
FileSaveWordDocx	0x090B	Save file as a [ECMA-376] document.
FileSaveWord11	0x090C	Save file in Word Binary File format.
InsertPicture3	0x090D	Inserts a picture.
SaveEquation	0x090E	Saves the current Equation as a new building block.
ViewFooterOnly	0x090F	Displays footer in page layout view.
EngWritingAssistant	0x0910	English Assistant.
TableOfContentsGallery	0x0911	Table Of Contents Gallery.
FileSaveAsOtherFormats	0x0912	Saves a copy of the document in a separate file.
CreateTableOfContentsFromSel	0x0915	Creates a new table of contents building block from the current selection.
SaveTableOfContentsBlock	0x0919	Saves the current table of contents as a new building block.
TextboxPositionGallery	0x091D	Opens the list of textbox position options.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
TextboxStyleGallery	0x091E	Opens the list of textbox styles.
TableColumnWidthSpinner	0x091F	Changes the width of the columns in a table.
TableRowHeightSpinner	0x0920	Changes the height of the row in a table.
RibbonFilePermissionMenu	0x0921	File Permission Menu.
MailMergeInsertMergeKeyword	0x0922	Mail Merge Insert Merge Keywords.
InsertTableOfContentsMenu	0x0923	Collects the headings or the table of contents entries into a table of contents.
WordSetDefaultPaste	0x0924	Allows setting the default paste action.
ReadingTrackChanges	0x0926	Menu for tracking changes.
ReadingFlyoutAnchorShowAcetateMarkup	0x0927	Show comments and changes.
ReadingInkTools	0x0928	Menu for Ink tools.
ViewEmailSource	0x0929	View the HTML source of this e-mail message.
ParagraphRemoveStyle	0x092A	Clears paragraph style from selection (restores the normal style).
RestoreParagraphStyle	0x092B	Restores paragraph style and removes direct formatting.
MSWordBibAddNewPlaceholder	0x092C	Add new placeholder.
DocExport	0x092D	Publish current document as XPS or PDF.
RemoveWatermark	0x092E	Removes the Watermarks from the current section.
RemoveCoverPage	0x092F	Removes the Cover Page from the document.
RemoveHeader	0x0930	Removes the header in the current section.
RemoveFooter	0x0931	Removes the footer in the current section.
RemovePageNumbers	0x0932	Removes Page Number building block from the document.
RemoveCurrentBuildingBlock	0x0933	Removes the current building block from the document.
RemoveTableOfContents	0x0934	Removes Table of Contents building block from the document.
ApplyQFSetInitial	0x0935	Applies the initial Quick Style set.
ApplyQFSetTemplate	0x0936	Applies the document template Quick Style set.
CreateNewFontScheme	0x0937	Opens the Create New Font Scheme dialog.
RemoveCitation	0x0938	Remove bibliography citation.
EditCitation	0x0939	Edit bibliography citation.
EditSource	0x093A	Opens the Edit Source dialog box.
BibliographyCitationToText	0x093B	Converts bibliography citation to static text.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
BibliographyEditSource	0x093C	Opens the Edit Source dialog box.
SaveOssThemeToTemplate	0x093D	Save OSS Theme to Template.
LoadOssThemeFromTemplate	0x093E	Load OSS Theme to Template.
OutlookInsertFile2	0x093F	Inserts the text of another file into the active document.
UpgradeDocument	0x0940	Upgrade Document to current file format.
UpdateFieldsToa	0x0947	Updates and displays the results of the selected fields.
UpdateFieldsTof	0x0948	Updates and displays the results of the selected fields.
NavigateMove	0x0949	Navigate to the opposite Move location.
ContentControlGroup	0x094A	Group the selection into a rich text content control with locked contents.
FormatPageBordersAndShading	0x094B	Changes the borders and shading of the selected paragraphs, table cells, and pictures.
DrawVerticalTextBox2	0x094C	Inserts an empty vertical text box or encloses the selected item in a vertical textbox.
ViewPageFromOutline	0x094D	Displays the page as it will be printed and allows editing.
StylePaneNewStyle	0x094E	Creates a new style out of the currently selected text.
ContentControlRichText	0x094F	Insert a rich text content control.
ContentControlText	0x0950	Insert a plain text content control.
ContentControlPicture	0x0951	Insert a picture content control.
ContentControlComboBox	0x0952	Insert a combo box content control.
ContentControlDropdownList	0x0953	Insert a dropdown content control.
ContentControlBuildingBlockGallery	0x0954	Insert a building block content control.
ContentControlDate	0x0955	Insert a date picker content control.
ToggleRibbon	0x0956	Shows or hides the Ribbon.
InkColorPicker	0x0957	Ink Color Picker.
EATextBoxMenu	0x0958	Insert Textbox menu.
DrawTextBox2	0x0959	Inserts an empty textbox or encloses the selected item in a textbox.
BBPropertiesDlg	0x095A	Building block properties dialog.
EquationsOptions	0x095B	Equation Options.
ReapplyTableStyle	0x095C	Reapplies the selected table style (keeping direct formatting intact).

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
CustomHeaderGallery	0x095D	Custom Header Gallery.
CustomFooterGallery	0x095E	Custom Footer Gallery.
CustomCoverPageGallery	0x095F	Custom Cover Page Gallery.
CustomPageNumGallery	0x0960	Custom Page Number Gallery.
CustomPageNumTopGallery	0x0961	Custom Page Number Top Gallery.
CustomPageNumBottomGallery	0x0962	Custom Page Number Bottom Gallery.
CustomPageNumPageGallery	0x0963	Custom Page Number Page Gallery.
CustomWatermarkGallery	0x0964	Custom Watermark Gallery.
CustomEquationsGallery	0x0965	Custom Equations Gallery.
CustomTablesGallery	0x0966	Custom Tables Gallery.
CustomQuickPartsGallery	0x0967	Custom Quick Parts Gallery.
CustomAutoTextGallery	0x0968	Custom AutoText Gallery.
CustomTextBoxGallery	0x0969	Custom Text Box Gallery.
CustomTableOfContentsGallery	0x096A	Custom Table of Contents Gallery.
CustomBibliographyGallery	0x096B	Custom Bibliography Gallery.
Custom1Gallery	0x096C	Custom 1 Gallery.
Custom2Gallery	0x096D	Custom 2 Gallery.
Custom3Gallery	0x096E	Custom 3 Gallery.
Custom4Gallery	0x096F	Custom 4 Gallery.
Custom5Gallery	0x0970	Custom 5 Gallery.
CreateBibliographyFromSel	0x0971	Creates a new bibliography building block from the current selection.
SaveBibliographyBlock	0x0972	Saves the current bibliography as a new building block.
MailMergeUseOutlookContacts	0x0974	Opens Outlook contacts as a data source for mail merge.
ChineseTranslationGroup	0x0976	Has no effect.
TableInsertCells2	0x0977	Inserts one or more cells into the table.
ContentControlUngroup	0x0978	Remove a content control group.
BibliographyEditCitationButton	0x0979	Edit bibliography Citation.
BibliographyEditSourceButton	0x097A	Opens the Edit Source dialog box.
BibliographyEditCitationToolbar	0x097B	Edit bibliography Citation.
BibliographyEditSourceToolbar	0x097C	Opens the Edit Source dialog box.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
EquationShowHideBorderTop	0x097D	Show or hide the top edge.
EquationShowHideBorderBottom	0x097E	Show or hide the bottom edge.
EquationShowHideBorderLeft	0x097F	Show or hide the left edge.
EquationShowHideBorderRight	0x0980	Show or hide the right edge.
EquationShowHideBorderHorizontalStrike	0x0981	Add or remove horizontal strike.
EquationShowHideBorderVerticalStrike	0x0982	Add or remove vertical strike.
EquationShowHideBorderTLBRStrike	0x0983	Add or remove strike from top left.
EquationShowHideBorderBLTRStrike	0x0984	Add or remove strike from bottom left.
BibliographyBibliographyToText	0x0985	Converts bibliography to static text.
QFSetAsDefault	0x0986	Saves the current Quick Styles to the document template.
CompatChkr	0x0987	Compatibility check.
MailMergeInsertFieldsFlyout	0x0988	Mail Merge Insert Fields.
AcceptChangesOrAdvance	0x0989	Accepts change in current selection.
RejectChangesOrAdvance	0x098A	Rejects changes and deletes comments in current selection.
NavBackMenu	0x098B	Menu for jumping back to the previous page in full screen reading.
NavForwardMenu	0x098C	Menu for jumping forward to the next page in full screen reading.
ReadModeShowMarkup	0x098D	Menu for viewing mode for revisions and comments in reading mode.
ReadModeMarkupFinal	0x098E	Menu item for showing final view in reading mode.
ReadModeMarkupFinalMarkup	0x098F	Menu item for showing final+markup view in reading mode.
ReadModeMarkupOriginal	0x0990	Menu item for Original view in reading mode.
ReadModeMarkupOriginalMarkup	0x0991	Menu item for Original+markup view in reading mode.
OpenOrCloseParaAbove	0x0992	Sets or removes extra spacing above the selected paragraph.
OpenOrCloseParaBelow	0x0993	Sets or removes extra spacing below the selected paragraph.
OpenParaAbove	0x0994	Adds extra spacing above the selected paragraph.
CloseParaAbove	0x0995	Removes extra spacing above the selected paragraph.
OpenParaBelow	0x0996	Adds extra spacing below the selected paragraph.
CloseParaBelow	0x0997	Removes extra spacing below the selected

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
		paragraph.
NextPane	0x0999	Switches to the next window pane or taskpane.
PrevPane	0x099A	Switches to the previous window pane or taskpane.
CheckDocumentParts	0x099B	Goes to Office Online to Check for New Document Building Blocks.
BibliographyFilterLanguages	0x099C	Filter Languages.
RaiseTextBaseline	0x099F	Moves text baseline up.
LowerTextBaseline	0x09A0	Moves text baseline down.
BibUpdateLang	0x09A1	Update Bibliography Language.
TableStyleNew	0x09A2	Creates a new table style.
Zoom100	0x09A3	Scales the current view to 100%.
UpdateBibliography	0x09A6	Update bibliography.
RibbonReviewProtectDocumentMenu	0x09A7	Review Protect Document Menu.
RibbonReviewRestrictFormatting	0x09A8	Restrict Formatting and Editing in the Protect Document menu.
ToggleOptimizeForLayout	0x09A9	Toggles optimize for layout option.
CharLeft	0x0FA0	Moves the insertion point to the left one character.
CharRight	0x0FA1	Moves the insertion point to the right one character.
WordLeft	0x0FA2	Moves the insertion point to the left one word.
WordRight	0x0FA3	Moves the insertion point to the right one word.
SentLeft	0x0FA4	Moves the insertion point to the beginning of the previous sentence.
SentRight	0x0FA5	Moves the insertion point to beginning of the next sentence.
ParaUp	0x0FA6	Moves the insertion point to the beginning of the previous paragraph.
ParaDown	0x0FA7	Moves the insertion point to the beginning of the next paragraph.
LineUp	0x0FA8	Moves the insertion point up one line.
LineDown	0x0FA9	Moves the insertion point down one line.
PageUp	0x0FAA	Moves the insertion point and document display to the previous screen of text.
PageDown	0x0FAB	Moves the insertion point and document display to the next screen of text.
StartOfLine	0x0FAC	Moves the insertion point to the beginning of the current line.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
EndOfLine	0x0FAD	Moves the insertion point to the end of the current line.
StartOfWindow	0x0FAE	Moves the insertion point to the beginning of the first visible line on the screen.
EndOfWindow	0x0FAF	Moves the insertion point to the end of the last visible line on the screen.
StartOfDocument	0x0FB0	Moves the insertion point to the beginning of the first line of the document.
EndOfDocument	0x0FB1	Moves the insertion point to the end of the last line of the document.
CharLeftExtend	0x0FB2	Extends the selection to the left one character.
CharRightExtend	0x0FB3	Extends the selection to the right one character.
WordLeftExtend	0x0FB4	Extends the selection to the left one word.
WordRightExtend	0x0FB5	Extends the selection to the right one word.
SentLeftExtend	0x0FB6	Extends the selection to the beginning of the previous sentence.
SentRightExtend	0x0FB7	Extends the selection to beginning of the next sentence.
ParaUpExtend	0x0FB8	Extends the selection to the beginning of the previous paragraph.
ParaDownExtend	0x0FB9	Extends the selection to the beginning of the next paragraph.
LineUpExtend	0x0FBA	Extends the selection up one line.
LineDownExtend	0x0FBB	Extends the selection down one line.
PageUpExtend	0x0FBC	Extends the selection and changes the document display to the previous screen of text.
PageDownExtend	0x0FBD	Extends the selection and changes the document display to the next screen of text.
StartOfLineExtend	0x0FBE	Extends the selection to the beginning of the current line.
EndOfLineExtend	0x0FBF	Extends the selection to the end of the current line.
StartOfWindowExtend	0x0FC0	Extends the selection to the beginning of the first visible line on the screen.
EndOfWindowExtend	0x0FC1	Extends the selection to the end of the last visible line on the screen.
StartOfDocExtend	0x0FC2	Extends the selection to the beginning of the first line of the document.
EndOfDocExtend	0x0FC3	Extends the selection to the end of the last line of the document.
File1	0x0FC5	Opens this document.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
File2	0x0FC6	Opens this document.
File3	0x0FC7	Opens this document.
File4	0x0FC8	Opens this document.
File5	0x0FC9	Opens this document.
File6	0x0FCA	Opens this document.
File7	0x0FCB	Opens this document.
File8	0x0FCC	Opens this document.
File9	0x0FCD	Opens this document.
MailMergeInsertAsk	0x0FCF	Inserts an Ask field at the insertion point.
MailMergeInsertFillIn	0x0FD0	Inserts a Fill-in field at the insertion point.
MailMergeInsertIf	0x0FD1	Inserts an If field at the insertion point.
MailMergeInsertMergeRec	0x0FD2	Inserts a MergeRec field at the insertion point.
MailMergeInsertMergeSeq	0x0FD3	Inserts a MergeSeq field at the insertion point.
MailMergeInsertNext	0x0FD4	Inserts a Next field at the insertion point.
MailMergeInsertNextIf	0x0FD5	Inserts a NextIf field at the insertion point.
MailMergeInsertSet	0x0FD6	Inserts a Set field at the insertion point.
MailMergeInsertSkipIf	0x0FD7	Inserts a SkipIf field at the insertion point.
BorderTop	0x0FDE	Changes the top borders of the selected paragraphs, table cells, and pictures.
BorderLeft	0x0FDF	Changes the left border of the selected paragraphs, table cells, and pictures.
BorderBottom	0x0FE0	Changes the bottom border of the selected paragraphs, table cells, and pictures.
BorderRight	0x0FE1	Changes the right border of the selected paragraphs, table cells, and pictures.
BorderInside	0x0FE2	Changes the inside borders of the selected paragraphs, table cells, and pictures.
ShowMe	0x0FE4	Gives an in-depth explanation of the suggested tip.
AutomaticChange	0x0FE6	Performs the suggested AutoFormat action.
FormatDrawingObjectWrapSquare	0x0FF8	Changes the selected drawing objects to square wrapping.
FormatDrawingObjectWrapTight	0x0FF9	Changes the selected drawing objects to tight wrapping.
FormatDrawingObjectWrapThrough	0x0FFA	Changes the selected drawing objects to tight through wrapping.
FormatDrawingObjectWrapNone	0x0FFB	Changes the selected drawing objects to no

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
		wrapping.
FormatDrawingObjectWrapTopBottom	0x0FFC	Changes the selected drawing objects to top/bottom wrapping.
MicrosoftOnTheWeb1	0x0FFE	Browse to an application-related Web site.
MicrosoftOnTheWeb2	0x0FFF	Browse to an application-related Web site.
MicrosoftOnTheWeb3	0x1000	Browse to an application-related Web site.
MicrosoftOnTheWeb4	0x1001	Browse to an application-related Web site.
MicrosoftOnTheWeb5	0x1002	Browse to an application-related Web site.
MicrosoftOnTheWeb6	0x1003	Browse to an application-related Web site.
MicrosoftOnTheWeb7	0x1004	Browse to an application-related Web site.
MicrosoftOnTheWeb8	0x1005	Browse to an application-related Web site.
MicrosoftOnTheWeb9	0x1006	Browse to an application-related Web site.
MicrosoftOnTheWeb10	0x1007	Browse to an application-related Web site.
MicrosoftOnTheWeb11	0x1008	Browse to an application-related Web site.
MicrosoftOnTheWeb12	0x1009	Browse to an application-related Web site.
MicrosoftOnTheWeb13	0x100A	Browse to an application-related Web site.
MicrosoftOnTheWeb14	0x100B	Browse to an application related Web site.
MicrosoftOnTheWeb15	0x100C	Browse to an application related Web site.
MicrosoftOnTheWeb16	0x100D	Browse to an application related Web site.
MicrosoftOnTheWeb17	0x100E	Browse to an application related Web site.
FormatDrawingObjectWrapFront	0x100F	Changes the selected drawing objects to no wrapping in front of text.
FormatDrawingObjectWrapBehind	0x1010	Changes the selected drawing objects to no wrapping behind text.
FormatDrawingObjectWrapInline	0x1011	Changes the selected drawing object to inline wrapping.
File10	0x10CC	Opens this document.
File11	0x10CD	Opens this document.
File12	0x10CE	Opens this document.
File13	0x10CF	Opens this document.
File14	0x10D0	Opens this document.
File15	0x10D1	Opens this document.
File16	0x10D2	Opens this document.
File17	0x10D3	Opens this document.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
File18	0x10D4	Opens this document.
File19	0x10D5	Opens this document.
File20	0x10D6	Opens this document.
File21	0x10D7	Opens this document.
File22	0x10D8	Opens this document.
File23	0x10D9	Opens this document.
File24	0x10DA	Opens this document.
File25	0x10DB	Opens this document.
File26	0x10DC	Opens this document.
File27	0x10DD	Opens this document.
File28	0x10DE	Opens this document.
File29	0x10DF	Opens this document.
File30	0x10E0	Opens this document.
File31	0x10E1	Opens this document.
File32	0x10E2	Opens this document.
File33	0x10E3	Opens this document.
File34	0x10E4	Opens this document.
File35	0x10E5	Opens this document.
File36	0x10E6	Opens this document.
File37	0x10E7	Opens this document.
File38	0x10E8	Opens this document.
File39	0x10E9	Opens this document.
File40	0x10EA	Opens this document.
File41	0x10EB	Opens this document.
File42	0x10EC	Opens this document.
File43	0x10ED	Opens this document.
File44	0x10EE	Opens this document.
File45	0x10EF	Opens this document.
File46	0x10F0	Opens this document.
File47	0x10F1	Opens this document.
File48	0x10F2	Opens this document.
File49	0x10F3	Opens this document.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
File50	0x10F4	Opens this document.
PageSetupMargins	0x10F5	Changes the page setup of the selected sections.
PageSetupPaper	0x10F6	Changes the page setup of the selected sections.
PageSetupLayout	0x10F7	Changes the page setup of the selected sections.
LegacyFileMru	0x10F8	Opens this document.
MenuFile	0x1644	File Menu.
MenuEdit	0x1645	Edit Menu.
MenuView	0x1646	View Menu.
MenuInsert	0x1647	Insert Menu.
MenuFormat	0x1648	Format Menu.
MenuTools	0x1649	Tools Menu.
MenuTable	0x164A	Table Menu.
MenuWindow	0x164B	Window Menu.
MenuHelp	0x164C	Help Menu.
MenuWork	0x164D	Work Menu.
MenuFont	0x164E	Font Menu.
MenuLanguage	0x1650	Language Submenu.
MenuMicrosoftOnTheWeb	0x1651	Microsoft On the Web Menu.
MenuBorder	0x1652	Has no effect.
MenuInsertTextBox	0x1653	Insert Textbox Submenu.
MenuInsertFrame	0x1654	Insert Frame Submenu.
MenuDraw	0x1655	Has no effect.
DrawMenuTextWrapping	0x1656	Has no effect.
DrawMenuOrder	0x1657	Has no effect.
DrawMenuGrouping	0x1658	Has no effect.
DrawMenuAlignDistribute	0x1659	Has no effect.
DrawMenuRotateFlip	0x165A	Has no effect.
DrawMenuNudge	0x165B	Has no effect.
FormatFillColor	0x165C	Applies the most recently used fill color to the selected AutoShape.
FormatLineColor	0x165D	Applies the most recently used line color to the selected AutoShape.
DrawMenuShadows	0x165E	Has no effect.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
FormatLineStyle	0x165F	Has no effect.
DrawMenuLineDash	0x1660	Has no effect.
DrawMenuArrows	0x1661	Has no effect.
DrawMenu3D	0x1662	Has no effect.
DrawMenuShadowColor	0x1663	Applies the most recently used shadow color to the selected AutoShape.
DrawMenuImageControl	0x1664	Has no effect.
DrawMenuChangeShape	0x1665	Has no effect.
DrawMenuChangeShape0	0x1666	Has no effect.
DrawMenuChangeShape1	0x1667	Has no effect.
DrawMenuChangeShape2	0x1668	Has no effect.
DrawMenuChangeShape3	0x1669	Has no effect.
DrawMenuChangeShape4	0x166A	Has no effect.
DrawMenuAutoShapes	0x166B	Has no effect.
DrawMenuMoreShapes1	0x166C	Has no effect.
DrawMenuMoreShapes2	0x166D	Has no effect.
DrawMenuMoreShapes3	0x166E	Has no effect.
DrawMenuMoreShapes4	0x166F	Has no effect.
DrawMenuMoreShapes5	0x1670	Has no effect.
DrawMenuMoreShapes6	0x1671	Has no effect.
DrawMenuTextShape	0x1672	Has no effect.
DrawMenuTextAlignment	0x1673	Has no effect.
DrawMenuTextTracking	0x1674	Has no effect.
DrawMenu3DDepth	0x1675	Has no effect.
DrawMenu3DDirection	0x1676	Has no effect.
DrawMenu3DColor	0x1677	Applies the most recently used 3-D color to the selected AutoShape.
DrawMenu3DLighting	0x1678	Has no effect.
DrawMenu3DSurface	0x1679	Has no effect.
MenuOrgChartSelect	0x167A	Has no effect.
MenuTableInsert	0x167B	Macro Submenu.
MenuTableDelete	0x167C	Macro Submenu.
AutoSignatureList	0x167D	Email AutoSignatures menu.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
MenuFrameset	0x167E	Format Frameset Submenu.
FilePreview	0x167F	File Preview Menu.
MenuFixSpellingLang	0x1680	Represents a menu. Has no effect.
MenuRevisions	0x1681	Revisions Submenu.
MenuFormatBackground	0x1682	Format Background Submenu.
MenuFixSpellingAC	0x1683	Represents a menu. Has no effect.
MenuPicture	0x1684	Insert Picture Submenu.
MenuAutoText	0x1685	Insert AutoText Submenu.
MenuMacro	0x1686	Macro Submenu.
MenuPowerTalk	0x1687	PowerTalk Submenu.
MenuHyperlinkSub	0x1688	Hyperlink.
MenuCellVerticalAlign	0x1689	Cell Vertical Alignment Submenu.
MenuEditObject	0x168A	Represents a menu. Has no effect.
MenuSendTo	0x168B	Represents a menu. Has no effect.
MenuAutoTextList	0x168D	Has no effect.
MenuTableSelect	0x1696	Macro Submenu.
MenuTableConvert	0x1697	Macro Submenu.
MenuTableInsertPalette	0x1698	Has no effect.
FixHHCMenu	0x1699	Represents a menu. Has no effect.
MenuTableAutoFitShort	0x169A	Macro Submenu.
MenuTableAutoFitLong	0x169B	Macro Submenu.
MenuCellAlignment	0x169C	Has no effect.
MenuTableInsertLong	0x169D	Macro Submenu.
MenuCollaboration	0x169E	Collaboration Submenu.
MenuAsianLayout	0x169F	Asian Layout Submenu.
FixSynonymMenu	0x16A0	Represents a menu. Has no effect.
MenuOrgChartLayout	0x16AB	Has no effect.
DrawMenuMoreShapes7	0x16AC	Has no effect.
MenuReference	0x16AE	Insert Reference Submenu.
MenuLettersMail	0x16AF	Tools Letters and Mailings Submenu.
MenuClear	0x16B0	Clear Submenu.
MenuDiagramLayout	0x16B1	Diagram Layout.

Copyright © 2025 Microsoft Corporation

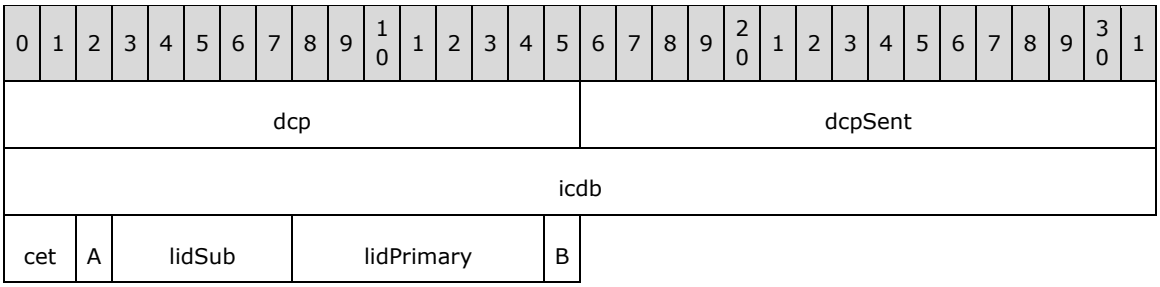
Name	Value	Meaning
MenuShowChanges	0x16B3	Fine tune which balloons are shown.
MenuShowReviewers	0x16B4	Fine tune which balloons are shown.
ResolveMenu	0x16B5	Accept/Reject Changes and Delete Comments.
MenuOrgChartInsert	0x16B6	Inserts an additional box to the organization chart.
MenuDiagramConvertTo	0x16B7	Convert To.
ApplyXMLStructureMenu	0x16B8	Represents a menu. Has no effect.
FormatInkColor	0x16B9	Brings up the format ink color dialog.
MenuVersion	0x16BA	Manages the versions of a document.
FormatInkAnnotColor	0x16BB	Brings up the format ink annotation color dialog.
MenuShowBalloons	0x16BC	Fine tune which balloons are shown.
InsertInkSplitMenu	0x16BD	Adds the Ink Tools tab to the Ribbon.
ReadingModeViewAllMenu	0x16C0	Produces a submenu of Heading1 or 2.
EquationVerticalMenu	0x16C2	Equation vertical alignment menu.
EquationHorizontalMenu	0x16C3	Equation horizontal alignment menu.
RefTipLangMenu	0x16C4	Translation.
MenuTableInsertIntoTable	0x16C5	Menu for inserting rows, columns, or cells into a table.
MenuCellAlignmentNoTearoff	0x16C6	Menu for table cell alignment in dialog boxes.
EquationJustificationMenu	0x16C7	Equation justification.
EquationInsertMenu	0x16C8	Matrix insert menu.
EquationDeleteMenu	0x16C9	Matrix delete menu.
EquationBorderPropertiesMenu	0x16CA	Equation border properties menu.
MenuWordQFStyles	0x16CB	Quick formatting menu.
WW2_FileTemplates	0x17A6	Changes the active template and the template options.
TrustCenter	0x17C7	Changes various security and privacy options.
OfficeCenter	0x17D0	Changes various categories of the application options.
InsertOCXCheckbox	0x1BA5	Inserts a Checkbox Control.
InsertOCXSpin	0x1BA6	Inserts a Spin Control.
InsertOCXScrollbar	0x1BA7	Inserts a Scrollbar Control.
InsertOCXLabel	0x1BA8	Inserts a Label Control.
InsertOCXTextBox	0x1BA9	Inserts a Text Box Control.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
InsertOCXButton	0x1BAA	Inserts a Button Control.
InsertOCXOptionButton	0x1BAB	Inserts a RadioButton Control.
InsertOCXListBox	0x1BAC	Inserts a Listbox Control.
InsertOCXDropdownCombo	0x1BAD	Inserts a Combobox Control.
InsertOCXToggleButton	0x1BAE	Inserts a Toggle Button Control.
ViewControlToolbox	0x1BAF	Shows or hides the Control Toolbox.
ShowPropertyBrowser	0x1BB0	Shows the Property Browser.
InsertOCXFrame	0x1BB1	Inserts a Frame Control.
InsertOCXImage	0x1BB2	Inserts an Image Control.
ToolbarLabel	0x1BB4	Represents a toolbar label control. Has no effect.
ViewWebToolbox	0x1BC4	Shows or hides the Web Toolbox.
ChangeMailFormat	0x1BC9	Changes the current message format.
DeleteSchema	0x1BD1	Deletes an XML Schema from the document.
AlignLeft	0x1BDD	Aligns the selected drawing objects to the left.
AlignCenterHorizontal	0x1BDE	Aligns the selected drawing objects horizontally to the center.
AlignRight	0x1BDF	Aligns the selected drawing objects to the right.
AlignTop	0x1BE0	Aligns the selected drawing objects to the top.
AlignCenterVertical	0x1BE1	Aligns the selected drawing objects vertically to the center.
AlignBottom	0x1BE2	Aligns the selected drawing objects to the bottom.
PPPropertyEditorDlg	0x1BE3	Show property editor dialog.

2.9.76 FCKS

The **FCKS** structure contains information about a **grammar checker cookie**. The grammar checker cookie itself is contained within the data that corresponds to the **fcCookieData** member of FibRgFcLcb97.



[MS-DOC] - v20250218
Word (.doc) Binary File Format
Copyright © 2025 Microsoft Corporation
Release: February 18, 2025

dcp (2 bytes): An integer that specifies the number of characters that are spanned by the text corresponding to the given grammar checker cookie. If **fHeader** is equal to 0x01, this value MUST be ignored.

dcpSent (2 bytes): An integer that specifies the number of characters between the start of the text that corresponds to the given grammar checker cookie and the start of the sentence which contains the text. If **fHeader** is equal to 0x01, this value MUST be ignored.

icdb (4 bytes): An unsigned integer that specifies the byte offset within the RgCdb that is specified by the **fcCookieData** member of FibRgFcLcb97, at which the data corresponding to this grammar checker cookie is located.

cet (2 bits): The error type that corresponds to the grammar checker cookie. The error types are interpreted as follows.

Value	Meaning
0x0	Default (not typo, homonym, or consistency)
0x1	Typo
0x2	Homonym
0x3	Consistency

If **fHeader** is equal to 0x1, this value MUST be ignored.

A - fError (1 bit): A bit that indicates whether the grammar checker cookie corresponds to a **grammar checker** error that is displayed to the user. If **fHeader** is equal to 0x1, this value MUST be ignored.

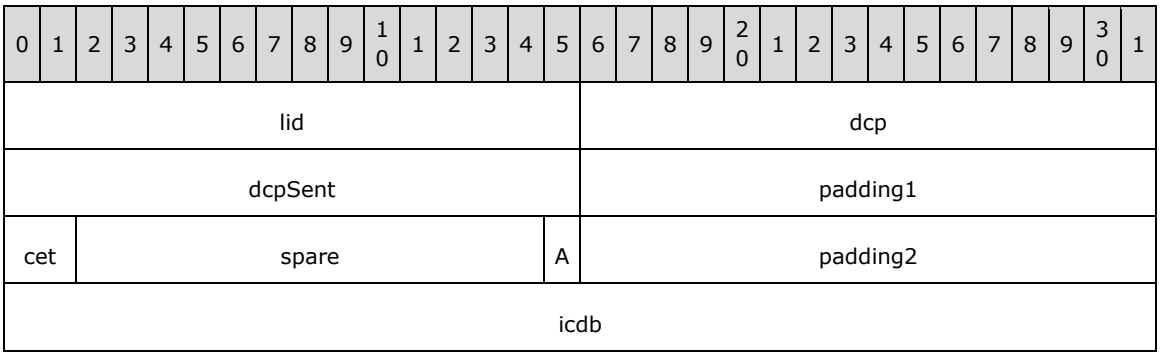
lidSub (5 bits): The 10th through 14th least significant bits of the language ID component of the LCID of the grammar checker which created the given grammar checker cookie, as specified in [MS-LCID].

lidPrimary (7 bits): The 7 least significant bits of the language ID component of the LCID of the grammar checker which created the given grammar checker cookie, as specified in [MS-LCID].

B - fHeader (1 bit): A bit indicating whether this is a special entry containing implementation-specific data for the grammar checker which created this grammar checker cookie. There MUST be only one entry with **fHeader** set to 0x1 by a given grammar checker in a document.

2.9.77 FCKSOLD

The **FCKSOLD** structure contains information about a **grammar checker cookie**. The grammar checker cookie itself is contained within the data that corresponds to the **fcCookieData** member of FibRgFcLcb97.



Copyright © 2025 Microsoft Corporation

lid (2 bytes): A LID that corresponds to the **grammar checker** that created the given grammar checker cookie.

dcp (2 bytes): An integer that specifies the number of characters that are spanned by the text corresponding to the given grammar checker cookie. This value MUST be greater than or equal to zero.

dcpSent (2 bytes): An integer that specifies the number of characters between the start of the text that corresponds to the given grammar checker cookie and the start of the sentence that contains the text. This value MUST be less than or equal to zero.

padding1 (2 bytes): This value is undefined and MUST be ignored.

cet (2 bits): An error type that corresponds to the grammar checker cookie. The error types are interpreted as follows.

Value	Meaning
0x0	Default (not typo, homonym, or consistency)
0x1	Typo
0x2	Homonym
0x3	Consistency

spare (13 bits): This value is undefined and MUST be ignored.

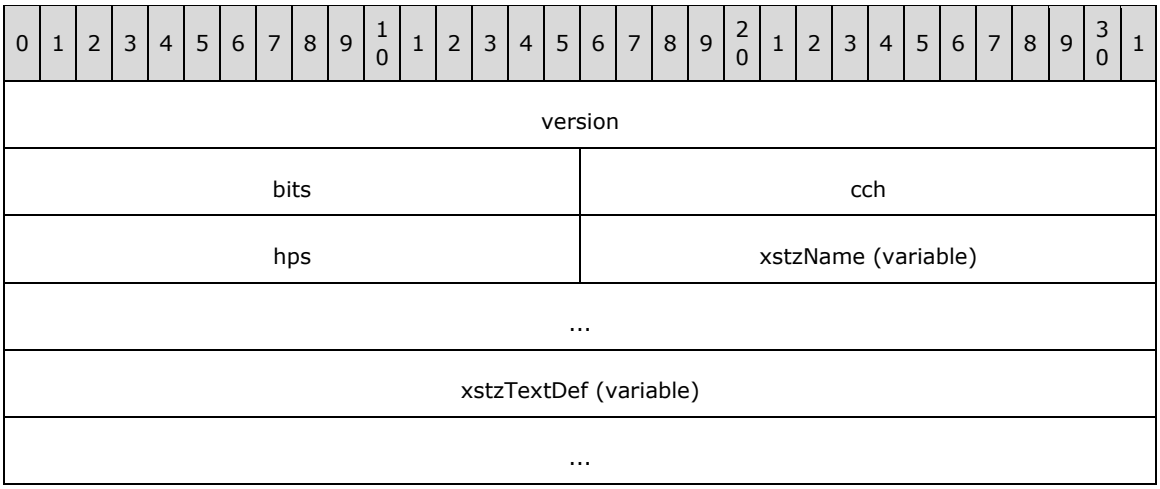
A - fError (1 bit): A bit that indicates whether the grammar checker cookie corresponds to a grammar checker error that is intended to be displayed to the user.

padding2 (2 bytes): This value is undefined and MUST be ignored.

icdb (4 bytes): An unsigned integer that specifies the byte offset within the RgCdb that is specified by the **fcCookieData** member of FibRgFcLcb97 at which the data corresponding to this grammar checker cookie is located.

2.9.78 FFData

The **FFData** structure specifies **form field** data for a text box, check box, or drop-down list box.



Copyright © 2025 Microsoft Corporation

wDef (optional)	xstzTextFormat (variable)
...	
xstzHelpText (variable)	
...	
xstzStatText (variable)	
...	
xstzEntryMcr (variable)	
...	
xstzExitMcr (variable)	
...	
hsttbDropList (variable)	
...	

version (4 bytes): An unsigned integer that MUST be 0xFFFFFFFF.

bits (2 bytes): An FFDataBits that specifies the type and state of this form field.

cch (2 bytes): An unsigned integer that specifies the maximum length, in characters, of the value of the textbox. This value MUST NOT exceed 32767. A value of 0 means there is no maximum length of the value of the textbox. If **bits.iType** is not iTypeText (0), this value MUST be 0.

hps (2 bytes): An unsigned integer. If **bits.iType** is iTypeChck (1), **hps** specifies the size, in half-points, of the checkbox and MUST be between 2 and 3168, inclusive. If **bits.iType** is not iTypeChck (1), **hps** is undefined and MUST be ignored.

xstzName (variable): An Xstz that specifies the name of this form field. **xstzName.cch** MUST NOT exceed 20.

xstzTextDef (variable): An optional Xstz that specifies the default text of this textbox. This structure MUST exist if and only if **bits.iType** is iTypeTxt (0). **xstzTextDef.cch** MUST NOT exceed 255. If **bits.iTypeTxt** is either iTypeTxtCurDate (3) or iTypeTxtCurTime (4), **xstzTextDef** MUST be an empty string. If **bits.iTypeTxt** is iTypeTxtCalc (5), **xstzTextDef** specifies an expression to calculate.

wDef (2 bytes): An optional unsigned integer that specifies the default state of the checkbox or dropdown list box. This value MUST exist if and only if **bits.iType** is iTypeChck (1) or iTypeDrop (2). If **bits.iType** is iTypeChck (1), **wDef** MUST be 0 or 1 and specify the default state of the checkbox as unchecked or checked, respectively. If **bits.iType** is iTypeDrop (2), **wDef** MUST be less than the number of items in the dropdown list box and specify the default item selected (zero-based index).

xstzTextFormat (variable): An Xstz that specifies the string format of the textbox. **xstzTextFormat** MUST be an empty string if **bits.iType** is not iTypeTxt (0).

Copyright © 2025 Microsoft Corporation

xstzTextFormat.cch MUST NOT exceed 64. Valid formatting strings are specified in [ECMA-376] part 1, section 17.16.20 format (Text Box Form Field Formatting).

xstzHelpText (variable): An Xstz that specifies the help text for the form field. The value of **xstzHelpText.cch** MUST NOT exceed 255.

xstzStatText (variable): An Xstz that specifies the status bar text for the form field. The value of **xstzStatText.cch** MUST NOT exceed 138.

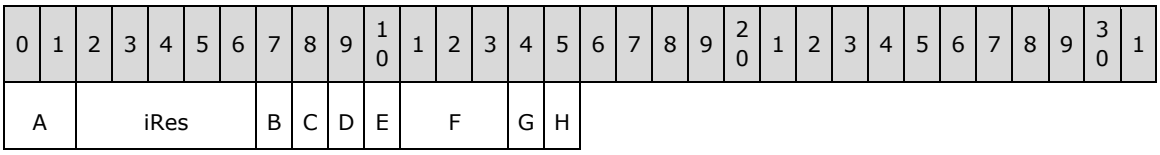
xstzEntryMcr (variable): An Xstz that specifies a macro to run on entry of the form field. The value of **xstzEntryMcr.cch** MUST NOT exceed 32.

xstzExitMcr (variable): An Xstz that specifies a macro to run after the value of the form field changes. The value of **xstzExitMcr.cch** MUST NOT exceed 32.

hsttbDropList (variable): An optional STTB that specifies the entries in the dropdown list box. This MUST exist if and only if **bits.iType** is iTypeDrop (2). The entries are **Unicode** strings and do not have extra data. This MUST NOT exceed 25 elements.

2.9.79 FFDataBits

The **FFDataBits** structure specifies the type and properties for a **form field** that is specified by a FFData.



A - iType (2 bits): An unsigned integer that specifies the type of the form field. This value MUST be one of the following.

Value	Name	Description
0	iTypeText	Specifies that the form field is a textbox.
1	iTypeChck	Specifies that the form field is a checkbox.
2	iTypeDrop	Specifies that the form field is a dropdown list box.

iRes (5 bits): An unsigned integer. If **iType** is iTypeText (0), then **iRes** MUST be 0. If **iType** is iTypeChck (1), **iRes** specifies the state of the checkbox and MUST be 0 (unchecked), 1 (checked), or 25 (undefined). Undefined checkboxes are treated as unchecked. If **iType** is iTypeDrop (2), **iRes** specifies the current selected list box item. A value of 25 specifies the selection is undefined. Otherwise, **iRes** is a zero-based index into FFData**.hsttbDropList**.

B - fOwnHelp (1 bit): A bit that specifies whether the form field has custom help text in FFData**.xstzHelpText**. If **fOwnHelp** is 0, FFData**.xstzHelpText** contains an empty or auto-generated string.

C - fOwnStat (1 bit): A bit that specifies whether the form field has custom status bar text in FFData**.xstzStatText**. If **fOwnStat** is 0, FFData**.xstzStatText** contains an empty or auto-generated string.

D - fProt (1 bit): A bit that specifies whether the form field is protected and its value cannot be changed.

E - iSize (1 bit): A bit that specifies whether the size of a checkbox is automatically determined by the text size where the checkbox is located. This value MUST be 0 if **iType** is not iTypeChck (1).

Copyright © 2025 Microsoft Corporation

F - iTypeTxt (3 bits): An unsigned integer that specifies the type of the textbox. This MUST be one of the following values.

Value	Name	Description
0	iTypeTxtReg	Specifies that the textbox value is regular text.
1	iTypeTxtNum	Specifies that the textbox value is a number.
2	iTypeTxtDate	Specifies that the textbox value is a date or time.
3	iTypeTxtCurDate	Specifies that the textbox value is the current date.
4	iTypeTxtCurTime	Specifies that the textbox value is the current time.
5	iTypeTxtCalc	Specifies that the textbox value is calculated from an expression. The expression is given by FFData**.xstzTextDef**.

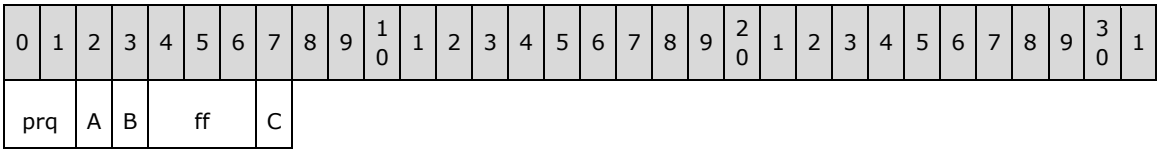
If **iType** is not iTypeText (0), **iTypeTxt** MUST be 0 and MUST be ignored.

G - fRecalc (1 bit): A bit that specifies whether the value of the field is automatically calculated after the field is modified.

H - fHasListBox (1 bit): A bit that specifies that the form field has a list box. This value MUST be 1 if **iType** is iTypeDrop (2). Otherwise, this value MUST be 0.

2.9.80 FFID

The **FFID** structure specifies the font family and **character pitch** for a font.



prq (2 bits): A 2-bit field that specifies character pitch. This MUST contain one of the following values.

Value	Meaning
0x00	Default pitch.
0x01	Fixed pitch.
0x02	Variable pitch.

A - fTrueType (1 bit): A bit that specifies whether the font is a **TrueType font**.

B - unused1 (1 bit): This bit is undefined and MUST be ignored.

ff (3 bits): A bit field that specifies the font family type as described in [MSDN-FONTS]. This field MUST contain one of the following values.

Value	Meaning
0x00	Font family is unspecified for this font.
0x01	Roman (Serif).
0x02	Swiss (Sans-serif).
0x03	Modern (Monospace).
0x04	Script (Cursive).
0x05	Decorative (Fantasy).

Copyright © 2025 Microsoft Corporation

C - unused2 (1 bit): This field MUST be zero and MUST be ignored.

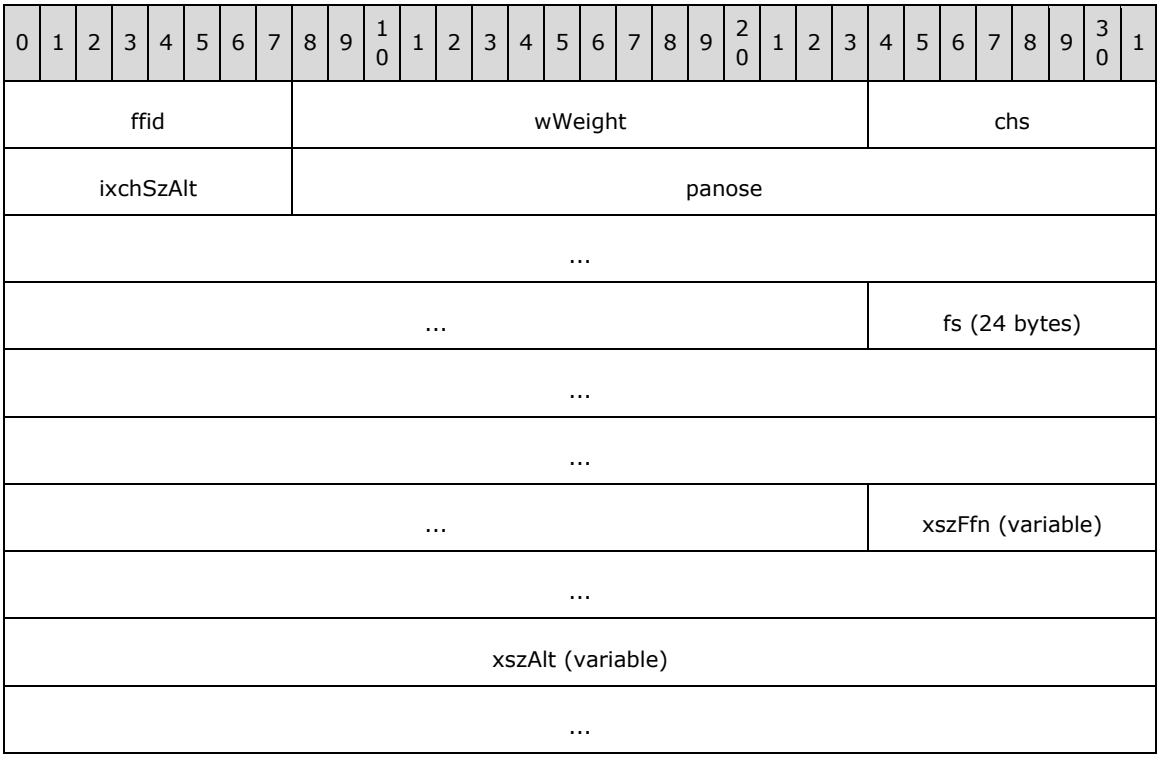
2.9.81 FFM

The **FFM** enumeration specifies the type of font substitution that is needed for the associated text. Font substitution is needed when certain language characters are not supported by the current font for the text, so a different font needs to be picked that supports the characters.

Name	Value	Meaning
ffmNone	0x00	No font substitution is needed for this text.
ffmDefault	0x01	Substitute a font using default heuristics.
ffmUILang	0x02	Substitute a font using the best font for the language of the text.
ffmUIDialog	0x04	Substitute a font using the same font that the user interface text is displayed in, if appropriate.

2.9.82 FFN

The **FFN** structure specifies information about a font that is used in the document. This information MUST be complete for each font. In addition to specifying a specific named font, this information is intended for the purpose of font substitution when that named font is not available.



ffid (1 byte): An FFID that specifies the font family.

wWeight (2 bytes): A signed integer that specifies the visual weight of the font. This value MUST be between 0 and 1000. A value of 700 corresponds to bold text. A value of 400 corresponds to normal text.

Copyright © 2025 Microsoft Corporation

chs (1 byte): An unsigned integer that specifies the **character set** that is used by the font. This MUST be one of the following values.

Value	Meaning
0	ANSI_CHARSET
1	DEFAULT_CHARSET
2	SYMBOL_CHARSET
128	SHIFTJIS_CHARSET
129	HANGEUL_CHARSET
129	HANGUL_CHARSET
134	GB2312_CHARSET
136	CHINESEBIG5_CHARSET
255	OEM_CHARSET
130	JOHAB_CHARSET
177	HEBREW_CHARSET
178	ARABIC_CHARSET
161	GREEK_CHARSET
162	TURKISH_CHARSET
163	VIETNAMESE_CHARSET
222	THAI_CHARSET
238	EASTEUROPE_CHARSET
204	RUSSIAN_CHARSET
77	MAC_CHARSET
186	BALTIC_CHARSET

ixchSzAlt (1 byte): An unsigned integer that specifies the zero-based index into the **xszFfn**. If nonzero, this value specifies the location within **xszFfn** where **xszAlt** begins.

panose (10 bytes): A **Panose** that specifies font attributes for TrueType fonts.

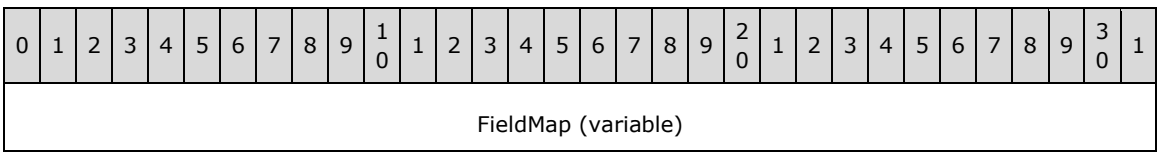
fs (24 bytes): A **FontSignature**, as specified in [MC-FONTSIGNATURE], that specifies the **Unicode Subset Bitfields** of the font, as specified in [MC-USB], and **Code Page Bitfields**, as specified in [MC-CPB].

xszFfn (variable): A null-terminated Unicode string that MUST contain the name of the font.

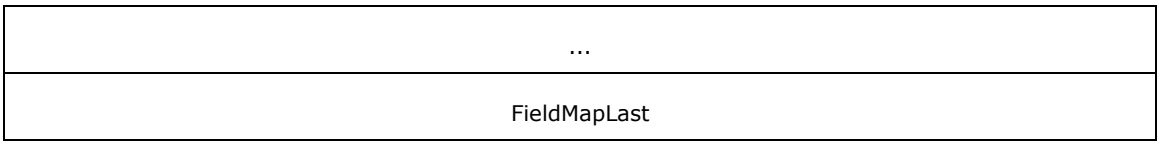
xszAlt (variable): A null-terminated Unicode string that specifies the name of an alternative font, intended for font substitution if the font specified by xszFfn is not available. This field, if it exists, begins immediately after the terminating null character of xszFfn. If **ixchSzAlt** is nonzero, this string MUST exist, otherwise it MUST NOT exist.

2.9.83 FieldMapBase

The **FieldMapBase** structure contains a **FieldMap** which is followed by a marker that specifies where the **FieldMap** ends (**FieldMapLast**). A **FieldMapBase** MUST correspond with one of 30 standard mail merge address fields, which are defined for ODSOPropertyBase.OdsoProp when ODSOPropertyBase.id is equal to 0x0016.



Copyright © 2025 Microsoft Corporation

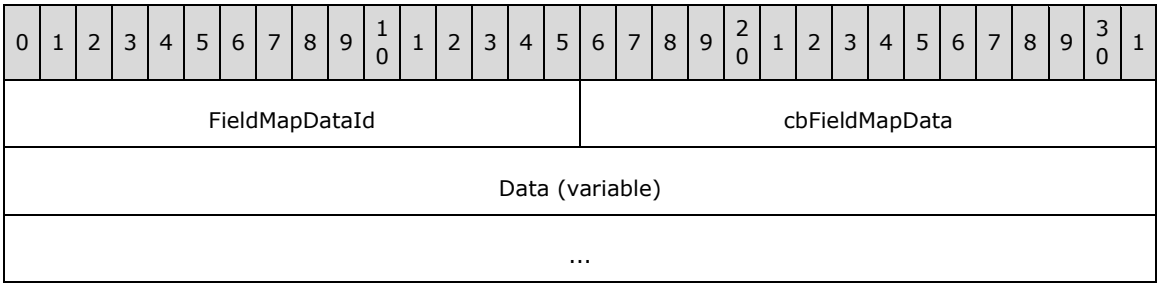


FieldMap (variable): An array of FieldMapDataItem. Data that specifies the mapping between one of 30 standard mail merge address fields and a column in the data source.

FieldMapLast (4 bytes): Contains a FieldMapTerminator that specifies that there is no further data to read for the current **FieldMap**.

2.9.84 FieldMapDataItem

The **FieldMapDataItem** structure contains information about a mail merge field mapping. All **FieldMapDataItems** that apply to a particular field mapping are grouped together. When a **FieldMapTerminator** is encountered, there is no further data about this field mapping, and any subsequent **FieldMapDataItem** structures are associated with subsequent field mappings.



FieldMapDataId (2 bytes): An unsigned integer that specifies the type of this **FieldMapDataItem**. This value MUST be 0x0001, 0x0002, 0x0003, or 0x0004.

cbFieldMapData (2 bytes): An unsigned integer that specifies the size, in bytes, of the following **Data** element.

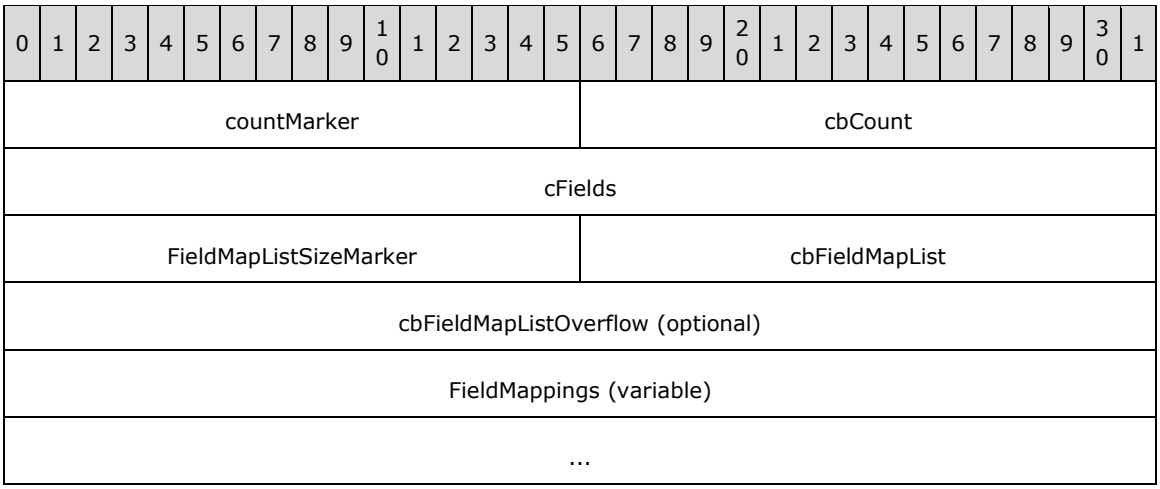
Data (variable): Contains the actual data for this **FieldMapDataItem**. The meaning of the data depends on the preceding **FieldMapDataId** and is specified as follows.

FieldMapDataId	Data
0x0001	An unsigned integer that specifies the mail merge field is being mapped to a data source column. This value MUST be 0x00000001.
0x0002	A Unicode string that specifies the name of the data source column to which this merge field is being mapped. The string is not null-terminated.
0x0003	A Unicode string that specifies the name of the standard mail merge field to which the data source column is being mapped. The string is not null-terminated. This string MUST be ignored.
0x0004	An unsigned integer that specifies the zero-based index of the data source column to which this merge field is being mapped. If the value is 0xFFFFFFFF, this **FieldMapDataItem** MUST be ignored.

2.9.85 FieldMapInfo

The **FieldMapInfo** structure specifies information about how fields from a mail merge data source are mapped to standard mail merge address fields, which are defined for **ODSOPropertyBase**.**OdsoProp** when **ODSOPropertyBase.id** is equal to 0x0016.

Copyright © 2025 Microsoft Corporation



countMarker (2 bytes): An unsigned integer that specifies that the count of **FieldMappings** follows. This value MUST be 0.

cbCount (2 bytes): An unsigned integer that specifies the size, in bytes, of the following mapped field count. This value MUST be 0x0004.

cFields (4 bytes): An unsigned integer that specifies the number of elements in the **FieldMappings** array. This value MUST be 30.

FieldMapListSizeMarker (2 bytes): An unsigned integer that specifies that the size of the **FieldMappings** array that follows. This value MUST be 0x0001.

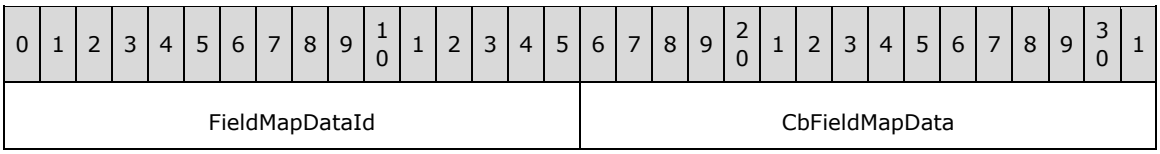
cbFieldMapList (2 bytes): An unsigned integer that specifies the size, in bytes, of the **FieldMappings** array. If the size is greater than 0xFFFE, this value MUST be 0xFFFF.

cbFieldMapListOverflow (4 bytes): An unsigned integer that specifies the size in bytes of the **FieldMappings** array. This value is only present if **cbFieldMapList** is set to 0xFFFF.

FieldMappings (variable): An array of FieldMapBase. Each FieldMapBase element in this array maps a column in the mail merge data source to a corresponding standard mail merge address field. There are 30 standard mail merge address fields, which are defined for **ODSOPropertyBase.OdsoProp** when **ODSOPropertyBase.id** is equal to 0x0016.

2.9.86 FieldMapTerminator

The **FieldMapTerminator** structure marks the end of the **FieldMapDataItem** structures that apply to an element of the **FieldMap** field of a FieldMapBase.



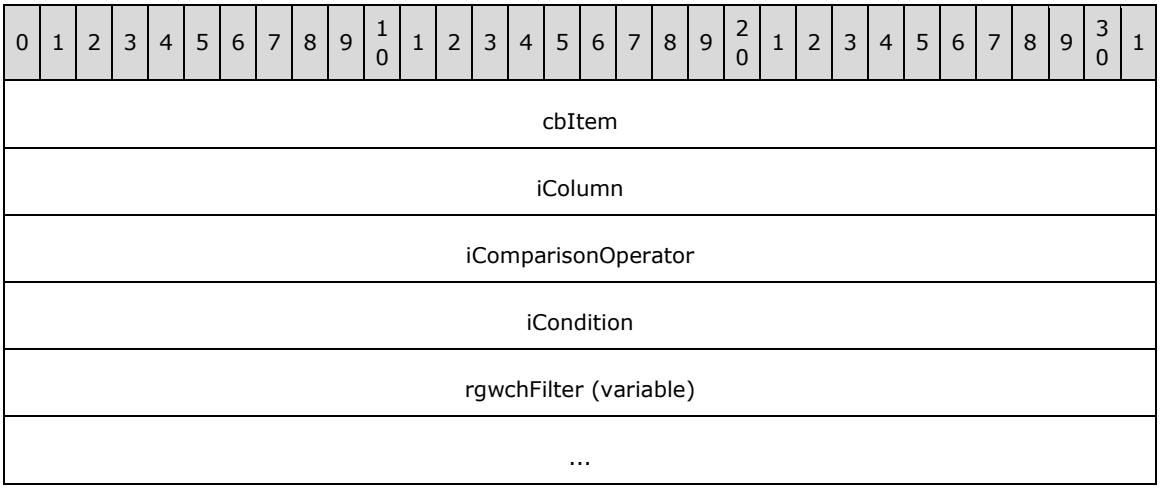
FieldMapDataId (2 bytes): An unsigned integer that specifies there is no further data to read for the current **FieldMap**. This value MUST be 0.

CbFieldMapData (2 bytes): This value MUST be 0.

Copyright © 2025 Microsoft Corporation

2.9.87 FilterDataItem

The **FilterDataItem** structure contains data that is used to filter a list of mail merge recipients.



cbItem (4 bytes): An unsigned integer that specifies the size, in bytes, of this **FilterDataItem**.

iColumn (4 bytes): An unsigned integer that specifies the zero-based index of the database column to which this filter applies. This value MUST be greater than or equal to zero and MUST be less than or equal to 254.

iComparisonOperator (4 bytes): An unsigned integer that specifies the comparison operator to be used for the comparison. This MUST be one of the following values.

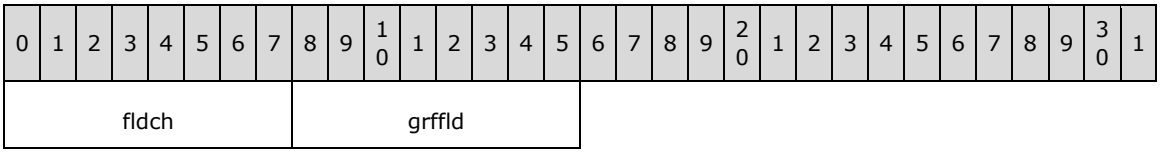
Value	Meaning
0x00000000	Equal.
0x00000001	Not equal.
0x00000002	Less than.
0x00000003	Greater than.
0x00000004	Less than or equal.
0x00000005	Greater than or equal.
0x00000006	Empty.
0x00000007	Not empty.

iCondition (4 bytes): An unsigned integer that specifies how this comparison is combined with other comparisons in the filter. This value MUST be 0 (logical **AND**) or 1 (logical **OR**).

rgwchFilter (variable): A Unicode string that specifies the value to be used as the basis for the comparison. The string is null-terminated and MUST contain no more than 212 characters.

2.9.88 Fld

The **Fld** structure specifies a field character.



Copyright © 2025 Microsoft Corporation

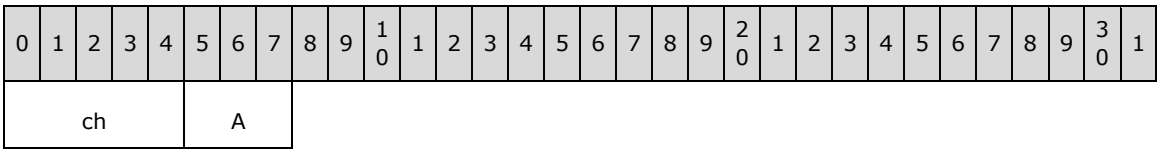
fldch (1 byte): An **fldch** whose **ch** member controls the interpretation of **grffld**. This value MUST be 0x13, 0x14, or 0x15.

grffld (1 byte): The meaning of this field is dependent on the value of **fldch**, as defined following.

fldch.ch	Meaning
0x13	**grffld** is an unsigned integer that indicates the kind of field this was the last time that an application parsed it. The values are specified in flt.
0x14	**grffld** is unused and MUST be ignored.
0x15	**grffld** is a grffldEnd.

2.9.89 fldch

The **fldch** structure determines the type of the field character.



ch (5 bits): An unsigned integer whose value MUST be either 0x13, 0x14, or 0x15. This value controls the interpretation of the **grffld** member of the containing Fld.

A - reserved (3 bits): Three reserved bits, which an application MUST ignore.

2.9.90 flt

The **flt** enumeration is an index to a **field type**. Most of the field type indices that are listed in the following table are mapped to entries in [ECMA-376] part 1, section 17.16.5. Values that are not specified following MUST NOT be used.

Value	Name	Meaning
0x01	Not Named	Specifies that the field was unable to be parsed.
0x02	Not Named	Specifies that the field represents a **REF** field where the keyword has been omitted. The **REF** field is specified in [ECMA-376] part 1, section17.16.5.51.
0x03	REF	Specified in [ECMA-376] part 1, section 17.16.5.51
0x05	FTNREF	This field is identical to NOTEREF specified in [ECMA-376] part 1, section 17.16.5.40.
0x06	SET	Specified in [ECMA-376] part 1, section 17.16.5.57.
0x07	IF	Specified in [ECMA-376] part 1, section 17.16.5.26.
0x08	INDEX	Specified in [ECMA-376] part 1, section 17.16.5.29.
0x0A	STYLEREF	Specified in [ECMA-376] part 1, section 17.16.5.59.
0x0C	SEQ	Specified in [ECMA-376] part 1, section 17.16.5.56.

Copyright © 2025 Microsoft Corporation

Value	Name	Meaning
0x0D	TOC	Specified in [ECMA-376] part 1, section 17.16.5.68.
0x0E	INFO	Specified in [ECMA-376] part 4, section 14.10.4.7.
0x0F	TITLE	Specified in [ECMA-376] part 1, section 17.16.5.66.
0x10	SUBJECT	Specified in [ECMA-376] part 1, section 17.16.5.60.
0x11	AUTHOR	Specified in [ECMA-376] part 1, section 17.16.5.4.
0x12	KEYWORDS	Specified in [ECMA-376] part 1, section 17.16.5.30.
0x13	COMMENTS	Specified in [ECMA-376] part 1, section 17.16.5.9.
0x14	LASTSAVEDBY	Specified in [ECMA-376] part 1, section 17.16.5.31.
0x15	CREATEDATE	Specified in [ECMA-376] part 1, section 17.16.5.11.
0x16	SAVEDATE	Specified in [ECMA-376] part 1, section 17.16.5.53.
0x17	PRINTDATE	Specified in [ECMA-376] part 1, section 17.16.5.47.
0x18	REVNUM	Specified in [ECMA-376] part 1, section 17.16.5.52.
0x19	EDITTIME	Specified in [ECMA-376] part 1, section 17.16.5.16.
0x1A	NUMPAGES	Specified in [ECMA-376] part 1, section 17.16.5.42.
0x1B	NUMWORDS	Specified in [ECMA-376] part 1, section 17.16.5.43.
0x1C	NUMCHARS	Specified in [ECMA-376] part 1, section 17.16.5.41.
0x1D	FILENAME	Specified in [ECMA-376] part 1, section 17.16.5.17.
0x1E	TEMPLATE	Specified in [ECMA-376] part 1, section 17.16.5.64.
0x1F	DATE	Specified in [ECMA-376] part 1, section 17.16.5.13.
0x20	TIME	Specified in [ECMA-376] part 1, section 17.16.5.65.
0x21	PAGE	Specified in [ECMA-376] part 1, section 17.16.5.44.
0x22	=	Specified in [ECMA-376] part 1, section 17.16.3.3.
0x23	QUOTE	Specified in [ECMA-376] part 1, section 17.16.5.49.
0x24	INCLUDE	This field is identical to **INCLUDETEXT** specified in [ECMA-376] part 1, section 17.16.5.28.
0x25	PAGEREF	Specified in [ECMA-376] part 1, section 17.16.5.45.
0x26	ASK	Specified in [ECMA-376] part 1, section 17.16.5.3.
0x27	FILLIN	Specified in [ECMA-376] part 1, section 17.16.5.19.
0x28	DATA	Usage: DATA *datafile [headerfile]* Specifies that this field SHOULD<223> redirect the mail merge data and header files to the ones specified.
0x29	NEXT	Specified in [ECMA-376] part 1, section 17.16.5.38.

Copyright © 2025 Microsoft Corporation

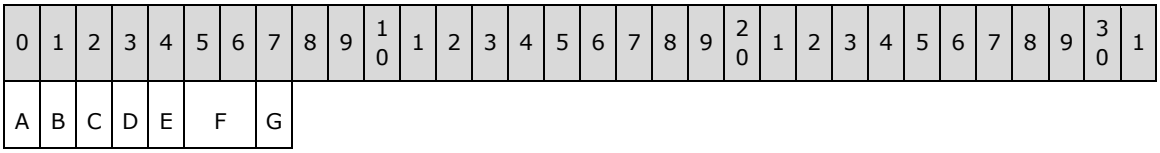
Value	Name	Meaning
0x2A	NEXTIF	Specified in [ECMA-376] part 1, section 17.16.5.39.
0x2B	SKIPIF	Specified in [ECMA-376] part 1, section 17.16.5.58.
0x2C	MERGEREC	Specified in [ECMA-376] part 1, section 17.16.5.36.
0x2D	DDE	Specified in [MS-OE376] part 2, section 1.3.2.1.
0x2E	DDEAUTO	Specified in [MS-OE376] part 2, section 1.3.2.2.
0x2F	GLOSSARY	This field is identical to **AUTOTEXT** specified in [ECMA-376] part 1, section 17.16.5.5.
0x30	PRINT	Specified in [ECMA-376] part 1, section 17.16.5.46.
0x31	EQ	Specified in [ECMA-376] part 4, section 14.10.4.6.
0x32	GOTOBUTTON	Specified in [ECMA-376] part 1, section 17.16.5.23.
0x33	MACROBUTTON	Specified in [ECMA-376] part 1, section 17.16.5.34.
0x34	AUTONUMOUT	Specified in [ECMA-376] part 4, section 14.10.4.3.
0x35	AUTONUMLGL	Specified in [ECMA-376] part 4, section 14.10.4.2.
0x36	AUTONUM	Specified in [ECMA-376] part 4, section 14.10.4.1.
0x37	IMPORT	Identical to the INCLUDEPICTURE field specified in [ECMA-376] part 1, section 17.16.5.27.
0x38	LINK	Specified in [ECMA-376] part 1, section 17.16.5.32.
0x39	SYMBOL	Specified in [ECMA-376] part 1, section 17.16.5.61.
0x3A	EMBED	Specifies that the field represents an embedded **OLE object**.
0x3B	MERGEFIELD	Specified in [ECMA-376] part 1, section 17.16.5.35.
0x3C	USERNAME	Specified in [ECMA-376] part 1, section 17.16.5.71.
0x3D	USERINITIALS	Specified in [ECMA-376] part 1, section 17.16.5.70.
0x3E	USERADDRESS	Specified in [ECMA-376] part 1, section 17.16.5.69.
0x3F	BARCODE	Specified in [ECMA-376] part 4, section 14.10.4.4.
0x40	DOCVARIABLE	Specified in [ECMA-376] part 1, section 17.16.5.15.
0x41	SECTION	Specified in [ECMA-376] part 1, section 17.16.5.54.
0x42	SECTIONPAGES	Specified in [ECMA-376] part 1, section 17.16.5.55.
0x43	INCLUDEPICTURE	Specified in [ECMA-376] part 1, section 17.16.5.27.
0x44	INCLUDETEXT	Specified in [ECMA-376] part 1, section 17.16.5.28.
0x45	FILESIZE	Specified in [ECMA-376] part 1, section 17.16.5.18.
0x46	FORMTEXT	Specified in [ECMA-376] part 1, section 17.16.5.22.
0x47	FORMCHECKBOX	Specified in [ECMA-376] part 1, section 17.16.5.20.
0x48	NOTEREF	Specified in [ECMA-376] part 1, section 17.16.5.40.

Copyright © 2025 Microsoft Corporation

Value	Name	Meaning
0x49	TOA	Specified in [ECMA-376] part 1, section 17.16.5.67.
0x4B	MERGESEQ	Specified in [ECMA-376] part 1, section 17.16.5.37.
0x4F	AUTOTEXT	Specified in [ECMA-376] part 1, section 17.16.5.5.
0x50	COMPARE	Specified in [ECMA-376] part 1, section 17.16.5.10.
0x51	ADDIN	Specifies that the field contains data created by an add-in.
0x53	FORMDROPDOWN	Specified in [ECMA-376] part 1, section 17.16.5.21.
0x54	ADVANCE	Specified in [ECMA-376] part 1, section 17.16.5.2.
0x55	DOCPROPERTY	Specified in [ECMA-376] part 1, section 17.16.5.14.
0x57	CONTROL	Specifies that the field represents an OCX control.
0x58	HYPERLINK	Specified in [ECMA-376] part 1, section 17.16.5.25.
0x59	AUTOTEXTLIST	Specified in [ECMA-376] part 1, section 17.16.5.6.
0x5A	LISTNUM	Specified in [ECMA-376] part 1, section 17.16.5.33.
0x5B	HTMLCONTROL	Specifies the field represents an HTML control.
0x5C	BIDIOUTLINE	Specified in [ECMA-376] part 4, section 14.10.4.5.
0x5D	ADDRESSBLOCK	Specified in [ECMA-376] part 1, section 17.16.5.1.
0x5E	GREETINGLINE	Specified in [ECMA-376] part 1, section 17.16.5.24.
0x5F	SHAPE	This field is identical to **QUOTE** specified in [ECMA-376] part 1, section 17.16.5.49.

2.9.91 FNFB

The **FNFB** structure describes the file systems for which a given path is valid.



A - fFAT (1 bit): A bit that specifies whether the path is valid on **FAT** file systems. If **fNonFileSys** is nonzero, this value MUST be 0.

B - unused1 (1 bit): This bit is undefined and MUST be ignored.

C - unused2 (1 bit): This bit is undefined and MUST be ignored.

D - fNTFS (1 bit): A bit that specifies whether the path is valid on **NTFS** file systems. If **fNonFileSys** is nonzero, this MUST be zero.

E - fNonFileSys (1 bit): A bit that specifies whether the path is not a native file system path. If this value is nonzero, the path is not a native file system path, and therefore requires an external file

Copyright © 2025 Microsoft Corporation

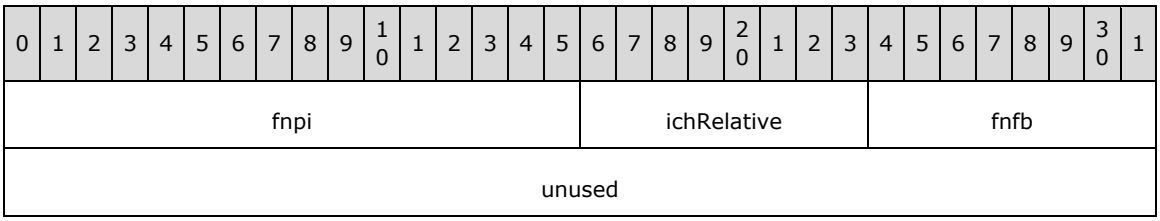
I/O protocol. If this value is zero, the path is native and can be used by the native Windows file I/O API.

F - unused3 (2 bits): This field is undefined and MUST be ignored.

G - unused4 (1 bit): This field is undefined and MUST be ignored.

2.9.92 FNIF

The **FNIF** structure contains information about a file name (see SttbFnm) so that the path, type, and file system of the file can be determined.



fnpi (2 bytes): An FNPI that specifies the type and the identifier of the file name, which is unique within the scope of **fnpi.fnpt**. This is used to define these values, not to reference a file name.

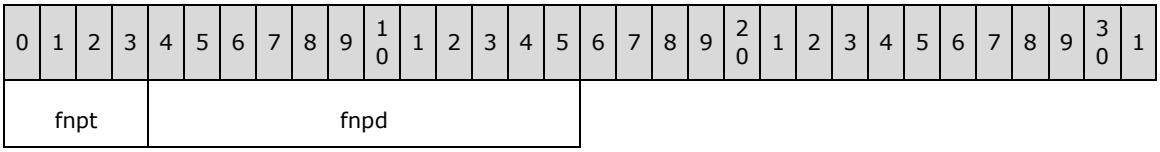
ichRelative (1 byte): An unsigned integer that specifies a character offset into the file name string. The segment of the file name string that starts at this character offset specifies the path of the file relative to the folder that contains the **document**. If the file name does not contain such a path, this value MUST be 0xFF.

fnfb (1 byte): An FNFB that specifies on what file systems the file name is valid.

unused (4 bytes): This field is undefined and MUST be ignored.

2.9.93 FNPI

The **FNPI** structure contains a type and an identifier for a file name. This structure can be used to define the type and identifier of a file name in SttbFnm, or it can be used to reference the file name in SttbFnm that has an identical **fnpi** in the appended FNIF. The definition of each **FNPI** specifies how it is used.



fnpt (4 bits): A signed integer that specifies the type of a file name. This MUST be one of the following values.

Value	Meaning
3	The file name refers to a **mail merge data source** file. This document MUST be a **mail merge** document.
5	The file name refers to a **subdocument**. This document MUST be a **master document**.

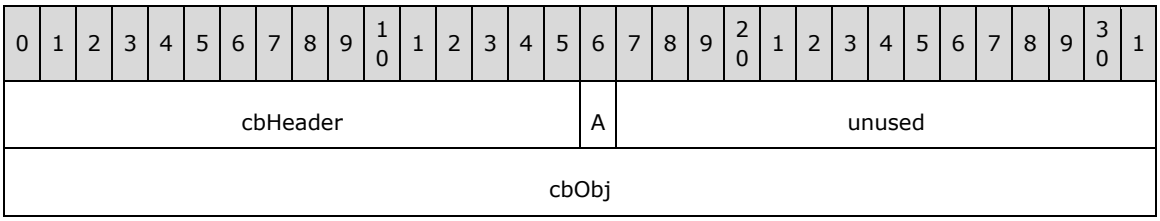
fnpd (12 bits): A signed integer that specifies an identifier for a file name. This value MUST NOT be 0xFFF.

Copyright © 2025 Microsoft Corporation

2.9.94 FOBJH

The **FOBJH** structure specifies size and compression information about the **OLE object** storage that immediately follows it in the Data stream of a file that is encrypted with Office Binary Document RC4 CryptoAPI Encryption (section 2.2.6.3). Every OLE object storage in the Data stream MUST be preceded by an FOBJH.

If **fCompressed** is 1, the bytes of the OLE object storage are compressed by the algorithm specified in [RFC1950].



cbHeader (2 bytes): A signed integer that specifies the size, in bytes, of the FOBJH. This value MUST be 8.

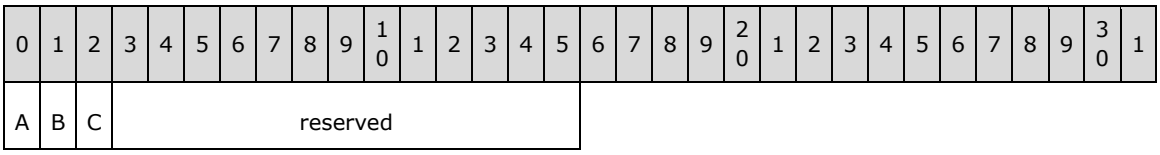
A - fCompressed (1 bit): Specifies whether the OLE object storage that follows this FOBJH is compressed.

unused (15 bits): This field is undefined and MUST be ignored.

cbObj (4 bytes): A signed integer that specifies the size, in bytes, of the FOBJH and the OLE object storage that follows it.

2.9.95 FrameTextFlowOperand

The **FrameTextFlowOperand** structure specifies the direction of text flow for a frame.



A - fVertical (1 bit): A bit that specifies that text flows vertically instead of horizontally.

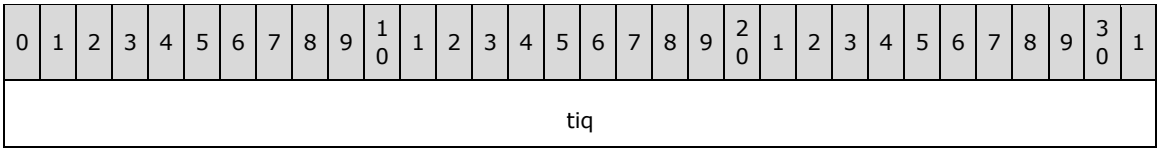
B - fBackwards (1 bit): A bit that specifies that vertical text flow is from bottom to top. If this bit is set, **fVertical** MUST also be set.

C - fRotateFont (1 bit): A bit that specifies that non-Latin text flow is rotated 90 degrees counter-clockwise.

reserved (13 bits): This value MUST be 0 and MUST be ignored.

2.9.96 FSDAP

The **FSDAP** structure specifies information about an attribute on a **structured document tag** in the document.



Copyright © 2025 Microsoft Corporation

...	
cch	rgValue (variable)
...	

tiq (8 bytes): A TIQ that specifies further information about the attribute represented by this FSDAP.

cch (2 bytes): An unsigned integer that specifies the count of characters in **rgValue**, not including its null terminator.

rgValue (variable): A null-terminated sequence of **Unicode** characters that specifies the value of the attribute represented by this FSDAP.

2.9.97 Fsnk

The **Fsnk** enumeration provides a 32-bit integer that specifies what kind of **DofrFsn** a record is. A field of this type MUST contain one of the following values.

Name	Value	Meaning
fsnkNil	0x00000000	No specified record kind.
fsnkFrameset	0x00000001	A record that has this **fsnk** value applies to the most recent **DofrFsn** record with **fsnk** equal to **fsnkFrame**, unless it appears before the first **DofrFsn** record with **fsnk** equal to **fsnkFrame**, in which case it applies to the outermost frame. This record type supplies more details about how that frame handles its child frames.
fsnkFrame	0x00000002	This record contains basic specifications for a frame. Records that have this **fsnk** value MUST appear before any other records that describe that frame.

2.9.98 Fssd

The **Fssd** structure specifies the position and units of a frame divider position.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
Units																															
Val																															

Units (4 bytes): An **FssUnits** element that specifies how to interpret **Val**.

Val (4 bytes): The position of the divider. This value can be interpreted in several ways, as specified by **Units**. If **Units** is set to **iFssUnitsNil**, this value MUST be ignored.

2.9.99 FssUnits

The **FssUnits** enumerated type specifies the units in an **Fssd**. A field of this type MUST contain one of the following values.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
iFssUnitsNil	0x00000000	No units are specified.
iFssUnitsPxl	0x00000001	The value is in pixels.
iFssUnitsPct	0x00000002	The value is a percentage of the size of the parent frame.
iFssUnitsRel	0x00000003	The value is a relative position. The actual position is a fraction of the parent frame size with this value as the numerator and the sum of all relative sizes for this row or column as the denominator.

2.9.100 FTO

The **FTO** enumerated type identifies the feature that is responsible to create a given smart tag in a document.

Name	Value	Meaning
ftoUnknown	0x0000	Not known.
ftoGrammar	0x0001	The grammar checker.
ftoScanDll	0x0002	An external scanning DLL.
ftoVB	0x0003	Visual Basic for Applications (VBA) script.

2.9.101 Fts

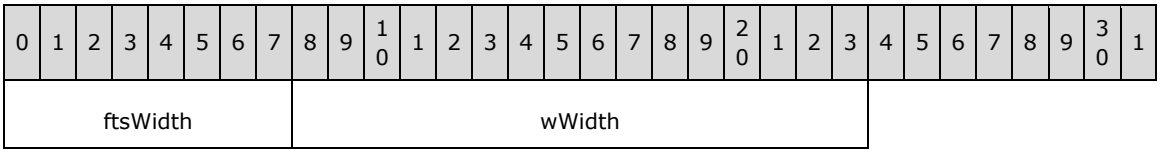
The **Fts** enumeration specifies how the preferred width for a table, table indent, table cell, **cell** margin, or **cell spacing** is defined. Any Table SPRM that specifies a preferred table width, table indent, cell width, **cell margin**, or cell spacing MUST also specify an **Fts** value to determine how the size is defined. Some **Fts** values are disallowed for some Sprms.

Name	Value	Meaning
ftsNil	0x00	The size is undefined and MUST be ignored.
ftsAuto	0x01	No preferred width is specified. The width is derived from other table measurements where a preferred size is specified, as well as from the size of the table contents, and the constraining size of the container of the table.
ftsPercent	0x02	The preferred width is measured in units of 1/50th of a percent (that is, a value of 50 translates to 1 percent). When specifying the preferred width of a portion of a table, such as a cell, spacing or indent, the percentage is relative to the width of the entire table. When specifying the preferred width of an entire table, the percentage is relative to the width of the page, less any margin or gutter space. Alternatively, if the table is nested inside another table, the percentage is relative to the width of the cell in the containing table, less cell margins.
ftsDxa	0x03	The preferred width of the table, indent, cell, cell margin, or cell spacing is an absolute width measured in **twips**.
ftsDxaSys	0x13	The preferred cell spacing is an absolute width measured in twips. ftsDxaSys is used when cell spacing is applied as a result of applying a table border.

Copyright © 2025 Microsoft Corporation

2.9.102 FtsWWidth_Indent

The **FtsWWidth_Indent** structure specifies the preferred width of indentation for a table.



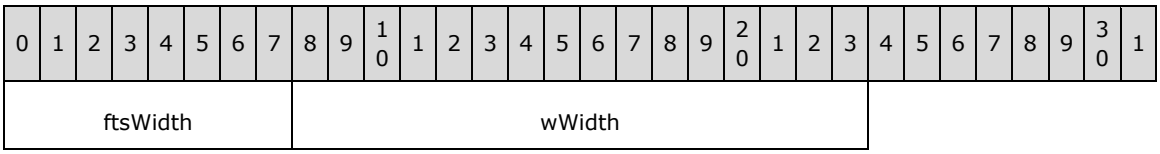
ftsWidth (1 byte): A value from the **Fts** enumeration that specifies the units of measurement for the **wWidth** value. **ftsWidth** MUST NOT be ftsPercent. **ftsWidth** MUST NOT be ftsDxaSys.

wWidth (2 bytes): An integer that specifies the preferred size of the indent. The size is evaluated differently depending on the value of **ftsWidth**.

ftsWidth value	wWidth meaning
ftsNil	**wWidth** is not used and MUST be zero.
ftsAuto	**wWidth** is not used and MUST be zero.
ftsPercent	This value of **ftsWidth** is not allowed.
ftsDxa	**wWidth** is measured in **twips**. It MUST be greater than or equal to -31,560 (-21 $^{11}/_{12}$ inches). It MUST be less than or equal to 31,680 (22 inches), less the width of the table. That is, the **logical right** edge of the table, calculated as the sum of this indentation and the width of the table (or the sum of the widths of the cells), MUST be less than or equal to 31,680 (22 inches).

2.9.103 FtsWWidth_Table

The **FtsWWidth_Table** structure specifies the preferred horizontal width of a table.



ftsWidth (1 byte): A value from the **Fts** enumeration that specifies the units of measurement for the **wWidth** value. The **ftsWidth** value MUST NOT be ftsDxaSys.

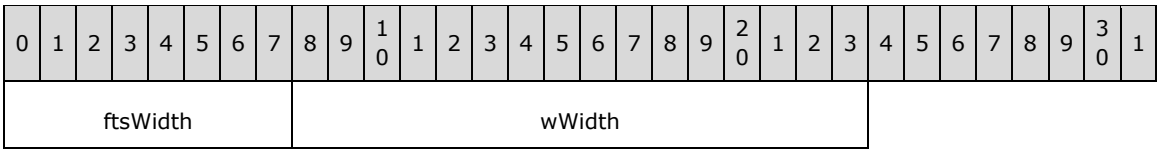
wWidth (2 bytes): An integer that specifies the preferred width. The size is evaluated differently depending on the value of **ftsWidth**.

ftsWidth value	wWidth meaning
ftsNil	**wWidth** is not used and MUST be zero.
ftsAuto	**wWidth** is not used and MUST be zero.
ftsPercent	**wWidth** MUST be non-negative and MUST be less than or equal to 30,000 (600%).
ftsDxa	**wWidth** MUST be non-negative and MUST be less than or equal to 31,680 (22 inches).

2.9.104 FtsWWidth_TablePart

The **FtsWWidth_TablePart** structure specifies the preferred horizontal width of an internal part of a table.

Copyright © 2025 Microsoft Corporation



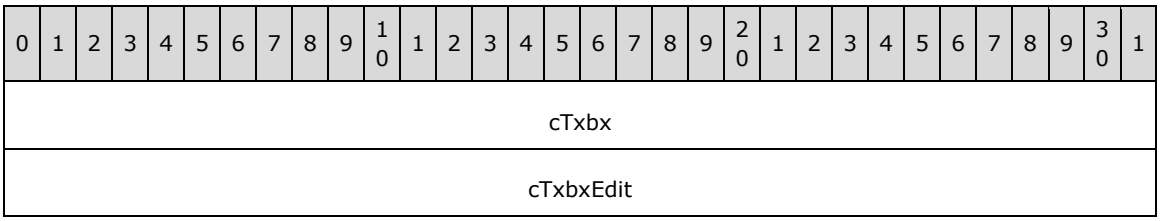
ftsWidth (1 byte): A value from the **Fts** enumeration that specifies the units of measurement for the **wWidth** value. The **ftsWidth** value MUST NOT be ftsDxaSys.

wWidth (2 bytes): An integer that specifies the preferred width. The size is evaluated differently depending on the value of **ftsWidth**.

ftsWidth value	wWidth meaning
ftsNil	**wWidth** is undefined and MUST be ignored.
ftsAuto	**wWidth** is not used and MUST be zero.
ftsPercent	**wWidth** MUST be non-negative and MUST be less than or equal to 5000 (100%).
ftsDxa	**wWidth** MUST be non-negative and MUST be less than or equal to 31,680 (22 inches).

2.9.105 FTXBXNonReusable

The **FTXBXNonReusable** structure is used within the **FTXBXS** structure when that structure describes a real textbox. A real textbox is any shape object into which text is added, and that is the first or only shape in a linked chain.

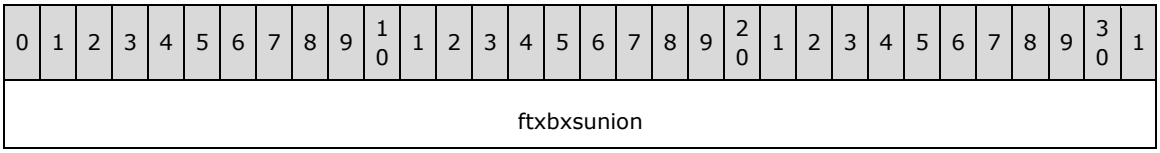


cTxbx (4 bytes): An integer that specifies how many shapes are in the chain into which the textbox text can flow. This number MUST be greater than zero and MUST match the length of the chain starting with the shape that is identified by the **lid** field in the FTXBXS structure and continuing through each linked shape.

cTxbxEdit (4 bytes): This value MUST be 0 and MUST be ignored.

2.9.106 FTXBXS

The **FTXBXS** structure is used by PlcftxbxTxt and by PlcfHdrtxbxTxt to associate ranges of text from the Textboxes Document and the Header Textboxes Document, respectively, with shape objects. In addition to the actual textboxes, there are 1 or more extra FTXBXS structures that can be reused by the application when creating new actual textboxes. The last FTXBXS in the PLC MUST be a reusable structure rather than an actual textbox. Additional reusable FTXBXS structures can occur at any index in the PLC.



Copyright © 2025 Microsoft Corporation

...	
fReusable	itxbxsDest
...	lid
...	txidUndo
...	

ftxbxsunion (8 bytes): If **fReusable** is "true", **ftxbxsunion** is an FTXBXSReusable structure. Also, if this is the last FTXBXS structure in the PLC, **ftxbxsunion** is an FTXBXSReusable structure, regardless of the **fReusable** flag. Otherwise, **ftxbxsunion** is an FTXBXNonReusable structure.

fReusable (2 bytes): An integer that specifies whether this structure describes an actual textbox or an extra structure that is available for reuse by the application. **fReusable** MUST be either zero ("false"), or it MUST have the 0x0001 bit set. When nonzero ("true"), bits other than 0x0001 MUST be ignored.

When **fReusable** is zero, this FTXBXS structure describes an actual textbox. The bounding CPs in PlcftxbxTxt or PlcfHdrtxbxTxt MUST be more than one character position apart, except when this is the last FTXBXS structure in the PLC. In that case there is no restriction on the character range specified by the bounding CPs in PlcftxbxTxt or PlcfHdrtxbxTxt. Text within this CP range MUST be ignored.

When **fReusable** is nonzero, this FTXBXS structure describes a reusable spare textbox structure. The bounding CPs in PlcftxbxTxt or PlcfHdrtxbxTxt MUST be one character position apart. When this is the last FTXBXS structure in the PLC, **fReusable** MUST be ignored and treated as if it were set to 0x0001 for the purposes of **ftxbxsunion** and **lid**.

itxbxsDest (4 bytes): This field MUST be ignored.

lid (4 bytes): An integer that specifies which shape object the textbox text begins in. When **fReusable** is "true", **lid** MUST be zero and MUST be ignored.

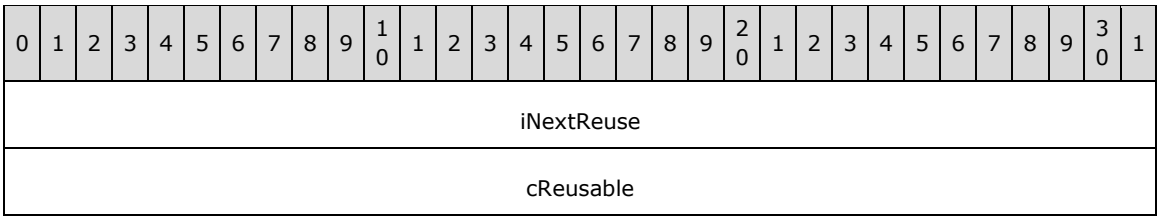
When **fReusable** is "false", **lid** MUST match the **OfficeArtFSP.spid** shape identifier in an **OfficeArtSpContainer** structure as specified by [MS-ODRAW] section 2.2.14. Furthermore, the **MSOPSText_lTxid** property of the **OfficeArtSpContainer**, as specified in [MS-ODRAW] section 2.3.21.1, MUST be a 4-byte integer where the high 2 bytes divided by 0x10000 gives the 1-based index of this **FTXBXS** structure in its **PLC**, and where the low 2 bytes are 0x0000.

txidUndo (4 bytes): This value MUST be 0 and MUST be ignored.

2.9.107 FTXBXSReusable

The **FTXBXSReusable** structure is used within the **FTXBXS** structure when it describes a spare structure that can be reused by the application and converted into an actual textbox. An **FTXBXS** structure can become reusable when the shape is deleted or linked after another shape in a chain. Additionally, the final **FTXBXS** structure in a **PLC** is reusable. All reusable **FTXBXS** structures in a **PLC** are part of a single chain, with the last **FTXBXS** structure in a **PLC** being the first item in the chain.

Copyright © 2025 Microsoft Corporation

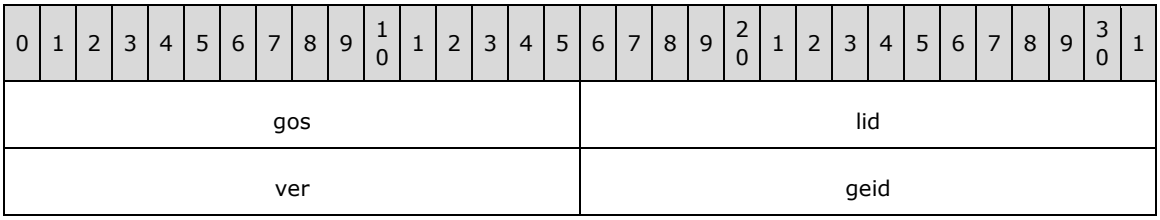


iNextReuse (4 bytes): An integer that specifies the index of the next reusable item in the chain. If this is the last **FTXBXS** structure in the chain, this value MUST be -1. Otherwise, this value MUST be non-negative, and MUST be less than the number of **FTXBXS** structures in the **PLC**. Furthermore, the **FTXBXS** structure at that index MUST be flagged as reusable, and MUST have a **cReusable** value that is 1 less than the **cReusable** value from this structure.

cReusable (4 bytes): An integer that specifies how many reusable **FTXBXS** structures are in the chain after this one. If this is the last **FTXBXS** structure in the chain, this value MUST be 0. Otherwise, it MUST be greater than zero, and MUST be less than the number of **FTXBXS** structures in the **PLC**.

2.9.108 GOSL

The **GOSL** structure specifies the option set for a grammar checker implementing the **CGAPI** interface, as well as information to identify the corresponding grammar checker.



gos (2 bytes): An unsigned integer that specifies a CGAPI option set. **gos** is implementation-specific to the grammar checker identified by **lid**, **ver**, and **ceid**. By default, the value is 0x0001.

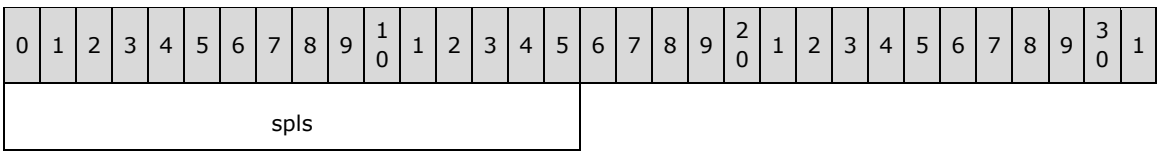
lid (2 bytes): A **LID** that specifies the language of the associated grammar checker.

ver (2 bytes): An unsigned integer that is the version number of the associated grammar checker, as it is specified through CGAPI.

geid (2 bytes): An unsigned integer that is the company identifier of the associated grammar checker, as it is specified through CGAPI.

2.9.109 GrammarSpls

The **GrammarSpls** structure is an **SPLS** structure that specifies the state of the grammar checker over a range of text. Some states that are possible in a generic **SPLS** are not allowed in a **GrammarSpls** structure.



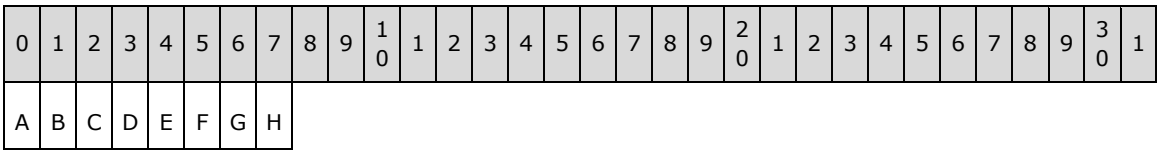
spls (2 bytes): An **SPLS** structure. The **spls.fExtend** field MUST be zero if the **spls.fError** field is zero. The **spls.splf** field MUST be one of the following:

Copyright © 2025 Microsoft Corporation

- splfMaybeDirty
- splfDirty
- splfEdit
- splfForeign
- splfClean
- splfErrorMin
- splfRepeatWord
- splfUnknownWord

2.9.110 grffldEnd

The **grffldEnd** structure describes the properties of the field.



A - fDiffer (1 bit): If this bit is set, the field shows results if the document-level setting is to show field instructions, and shows instructions if the document-level setting is to show field results.

B - fZombieEmbed (1 bit): If this bit is set, the field result contains an **OLE object**, but the field type is not able to generate OLE objects.

C - fResultsDirty (1 bit): If this bit is set, the field results were either edited or formatted since the last time that an application calculated the field.

D - fResultsEdited (1 bit): If this bit is set, the field results were edited since the last time that an application calculated the field.

E - fLocked (1 bit): If this bit is set, this field does not recalculate.

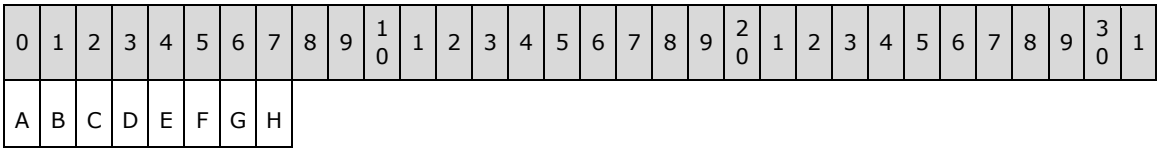
F - fPrivateResult (1 bit): If this bit is set, the field result is not intended to be visible to the user.

G - fNested (1 bit): This bit MUST be set if this field is contained in another field.

H - fHasSep (1 bit): This bit MUST be set if this field has a separator.

2.9.111 grfhic

The **grfhic** structure is a set of HTML incompatibility flags that specify the HTML incompatibilities of a list structure. The values specify possible incompatibilities between an LVL or LVLF and HTML lists. The values do not define list properties.



Copyright © 2025 Microsoft Corporation

A - fhicChecked (1 bit): A bit that specifies whether the list structure that contains this **grfhic** structure is checked for HTML incompatibilities.

B - fhicFormat (1 bit): A bit that specifies whether the numbering sequence or format of an LVL is unsupported by HTML at the time of the most recent HTML compatibility check. The numbering sequence or format of an LVL is unsupported by HTML if one or more of the following conditions are "true".

- **LVL.lvlf.nfc** is greater than 0x04
- **LVL.lvlf.fLegal** is nonzero
- **LVL.lvlf.fNoRestart** is nonzero
- **LVL.lvlf.ixchFollow** is nonzero

If **fhicChecked** is zero, this MUST be ignored. If the structure that contains this **grfhic** is not an LVLF, this MUST be ignored.

C - fhicListText (1 bit): A bit that specifies whether the string specified by LVL.**xst** was not of the standard form "#." (a level number placeholder followed by a period) at the time of the most recent HTML compatibility check. If **fhicChecked** is zero, this MUST be ignored. If the structure that contains this grfhic is not an LVLF, this MUST be ignored.

D - fhicPeriod (1 bit): A bit that specifies whether something other than a period was the last character of the number text specified by LVL.**xst** at the time of the most recent HTML compatibility check. If **fhicChecked** is zero, this MUST be ignored. If the structure that contains this grfhic is not an LVLF, this MUST be ignored.

E - fhicLeft1 (1 bit): A bit that specifies whether the indents specified by LVL.**grpprlPapx** were different than the standard HTML indents at the time of the most recent HTML compatibility check. The indents that are specified by LVL.**grpprlPapx** are different than the standard HTML indents if one or more of the conditions in the following list are "true":

- The logical left indent of the first line of the paragraph properties that are specified by LVL.**grpprlPapx** (see sprmPDxaLeft1) is not equal to -360.
- The logical left indent of the paragraph properties that are specified by LVL.**grpprlPapx** (see sprmPDxaLeft) is not equal to 720 * (*iLvl* + 1), where *iLvl* is the zero-based level of the list that LVL corresponds to.

If **fhicChecked** is zero, this MUST be ignored. If the structure that contains this grfhic is not a LVLF, this MUST be ignored.

F - fhicListTab (1 bit): A bit that specifies whether the first added custom tab stop of the paragraph properties specified by LVL.**grpprlPapx** (see sprmPChgTabs and sprmPChgTabsPapx) was not equal to the logical left indent of the paragraph properties specified by LVL.**grpprlPapx** (see sprmPDxaLeft) at the time of the most recent HTML compatibility check. If LVL.**grpprlPapx** does not add any custom tabs, this MUST be zero. If **fhicChecked** is zero, this MUST be ignored. If the structure that contains this **grfhic** is not an LVLF, this MUST be ignored.

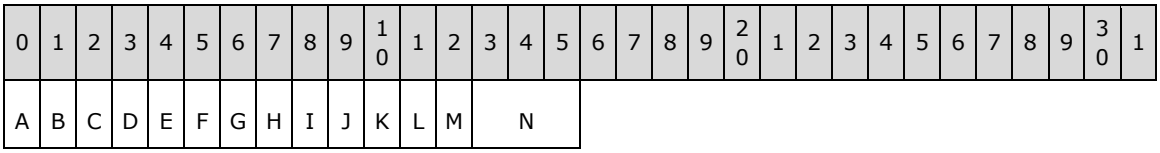
G - unused (1 bit): This bit is undefined and MUST be ignored.

H - fhicBullet (1 bit): A bit that specifies whether the level used bullets instead of numbers at the time of the most recent HTML compatibility check. A level uses bullets if LVL.**lvlf**.**nfc** is equal to 0x17. If **fhicChecked** is zero, this MUST be ignored. If the structure that contains this **grfhic** is not an LVLF, this MUST be ignored.

Copyright © 2025 Microsoft Corporation

2.9.112 GRFSTD

The **GRFSTD** structure specifies the general properties of a style.



A - fAutoRedef (1 bit): Specifies whether user formatting modifications are automatically merged into the paragraph style definition, as specified in [ECMA-376] part 1, section 17.7.4.2 (autoRedefine).

B - fHidden (1 bit): Specifies whether this style is not shown in the application UI, as specified in [ECMA-376] part 1, section 17.7.4.4 (hidden).

C - f97LidsSet (1 bit): Specifies whether sprmCRgLid0_80 and sprmCRgLid1_80 were applied, as appropriate, to this paragraph or character style for compatibility with applications that do not support sprmCRgLid0, sprmCRgLid1, and sprmCFNoProof. If this value is 1, the compatibility Sprms have already been applied for this style. If this value is 0, the compatibility Sprms need to be applied to the formatting properties of the current style or a base style. This value SHOULD<224> be 0.

D - fCopyLang (1 bit): If **f97LidsSet** is 1, this value specifies whether the applied compatibility sprmCRgLid0_80 or sprmCRgLid1_80 specified an actual language or a special LID value (0x0400) signifying that no proofing is needed for the text. This MUST be ignored if **f97LidsSet** is 0.

E - fPersonalCompose (1 bit): Specifies whether this character style can be used to automatically format the new message text in a new e-mail, as specified in [ECMA-376] part 1, section 17.7.4.12 (personalCompose). This MUST be ignored if this is not a character style.

F - fPersonalReply (1 bit): Specifies whether this character style can be used to automatically format the new message text when replying to an e-mail, as specified in [ECMA-376] part 1, section 17.7.4.13 (personalReply). This MUST be ignored if this is not a character style.

G - fPersonal (1 bit): Specifies whether this character style was applied to format all message text from one or more users in an e-mail, as specified in [ECMA-376] part 1, section 17.7.4.11 (personal). This MUST be ignored if this is not a character style.

H - fNoHtmlExport (1 bit): This value MUST be 0 and MUST be ignored.

I - fSemiHidden (1 bit): Specifies whether this style is not shown in the simplified main styles UI of the application, as specified in [ECMA-376] part 1, section 17.7.4.16 (semiHidden).

J - fLocked (1 bit): Specifies whether this style is prevented from being applied by using the application UI, as specified in [ECMA-376] part 1, section 17.7.4.7 (locked).

K - fInternalUse (1 bit): This bit is undefined and MUST be ignored.

L - fUnhideWhenUsed (1 bit): Specifies whether the **fSemiHidden** property is to be set to 0 when this style is used, as specified in [ECMA-376] part 1, section 17.7.4.20 (unhideWhenUsed).

M - fQFormat (1 bit): Specifies whether this style is shown in the Ribbon Style gallery, as specified in [ECMA-376] part 1, section 17.7.4.14 (qFormat).

N - fReserved (3 bits): This value MUST be 0 and MUST be ignored.

Copyright © 2025 Microsoft Corporation

2.9.113 GrLPUpxSw

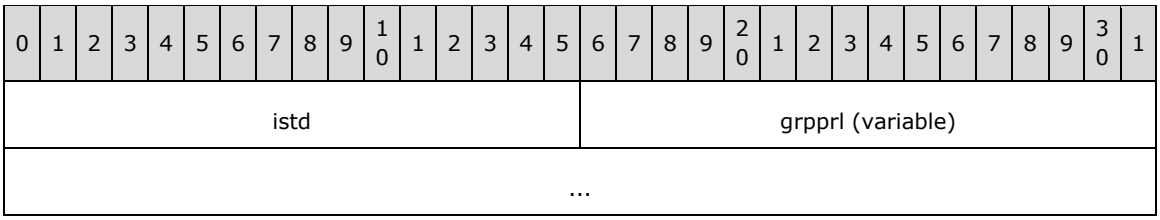
The **GrLPUpxSw** structure is an array of variable-size structures that specify the formatting of the style.

The content of the **GrLPUpxSw** structure depends on the type of the style (the **stk** member of StdfBase); see the following.

Value	Meaning
stkPara	stk value 1; the GrLPUpxSw contains a StkParaGRLPUPX.
stkChar	stk value 2; the GrLPUpxSw contains a StkCharGRLPUPX.
stkTable	stk value 3; the GrLPUpxSw contains a StkTableGRLPUPX.
stkList	stk value 4; the GrLPUpxSw contains a StkListGRLPUPX.

2.9.114 GrpPrlAndIstd

The **GrpPrlAndIstd** structure specifies the style and properties that are applied to a paragraph, a table row, or a table cell.

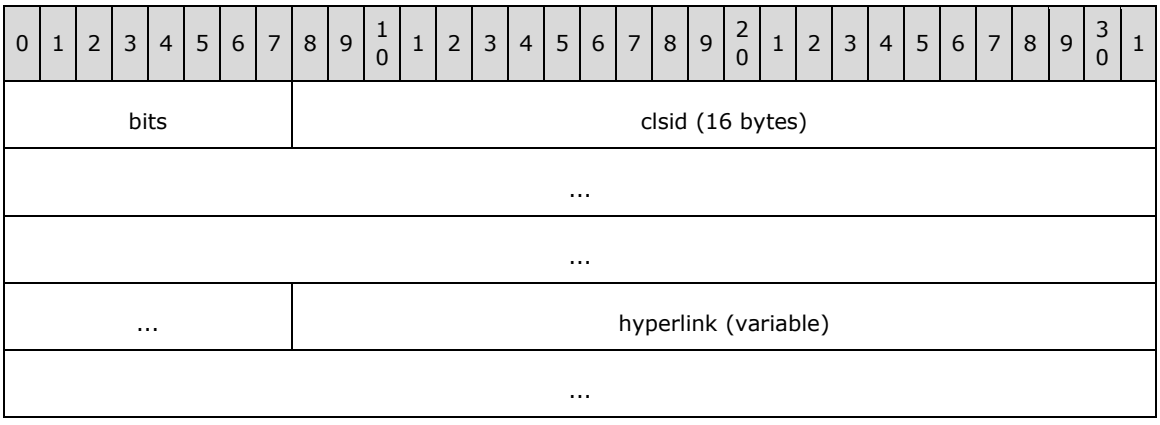


istd (2 bytes): An integer that specifies the style that is applied to this paragraph, cell marker or table row marker. See Applying Properties for more details about how to interpret this value.

grpprl (variable): An array of Prl elements. Specifies the properties of this paragraph, table row, or table cell. This array MUST contain a whole number of Prl elements.

2.9.115 HFD

The **HFD** structure specifies hyperlink field data including how to handle the hyperlink when it is traversed and a location in this document or an external document or webpage.



Copyright © 2025 Microsoft Corporation

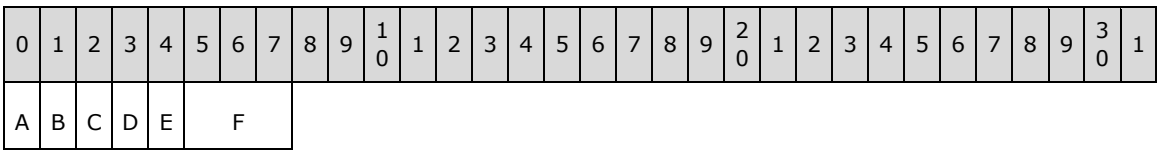
bits (1 byte): An HFDBits that specifies how to handle the hyperlink when it is traversed.

clsid (16 bytes): A **CLSID** that specifies the **COM** component that is used to create the **hyperlink**.

hyperlink (variable): A Hyperlink Object as specified in [MS-OSHARED] section 2.3.7.1. This object specifies a location in this document or an external document or webpage.

2.9.116 HFDBits

The **HFDBits** structure specifies how to handle a hyperlink when it is traversed.



A - fNew (1 bit): A bit that specifies if the hyperlink is to be opened in a new window.

B - fNoHist (1 bit): A bit that specifies if the navigation history is preserved when traversing this hyperlink. This value is 1 if the navigation history is not preserved and 0 if the navigation history is preserved.

C - fImageMap (1 bit): A bit that specifies if the hyperlink is a location in an **HTML image map**.

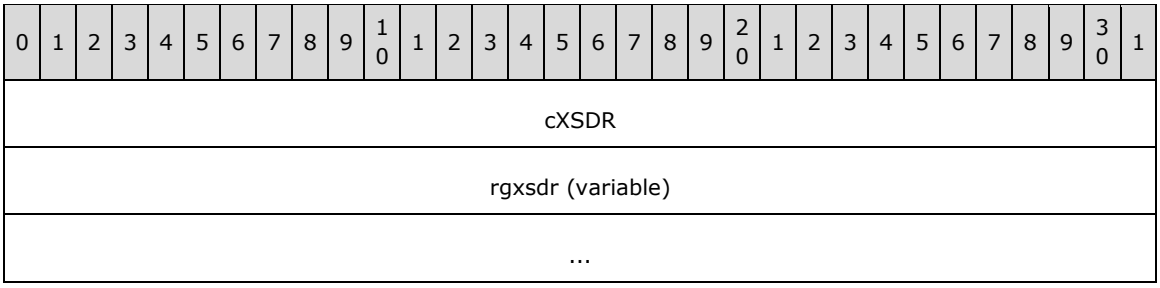
D - fLocation (1 bit): A bit that specifies if the hyperlink contains a specific location in the target document.

E - fTooltip (1 bit): A bit that specifies if the hyperlink contains a **ScreenTip** string.

F - unused (3 bits): This value MUST be 0 and MUST be ignored.

2.9.117 Hplxsdr

The **Hplxsdr** structure contains the schema definition references of the document. Each individual reference consists of a **Uniform Resource Identifier (URI)**, **manifest** location, table of elements, and table of attributes.



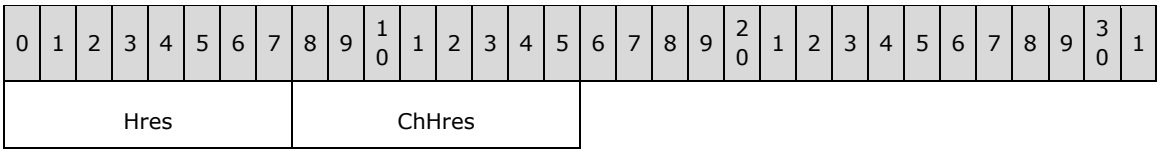
cXSDR (4 bytes): A signed integer that specifies the number of schema definition references. The minimum value is 0.

rgxsdr (variable): An array of XSDR.

2.9.118 HresiOperand

The **HresiOperand** structure specifies how word-breaking is handled.

Copyright © 2025 Microsoft Corporation



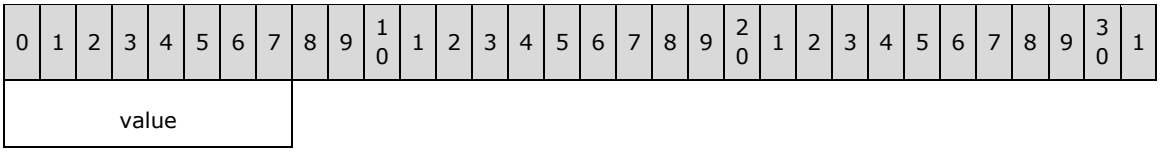
Hres (1 byte): An unsigned integer that specifies the word-breaking method. This property MUST specify one of the following values. By default, normal word-breaking is used.

Value	Name	Description
0x01	hresNormal	Normal word-breaking: Insert a hyphen and continue word on the next line.
0x02	hresAddBefore	Similar to Normal but also add **ChHres** before the hyphen.
0x03	hresChangeBefore	Similar to Normal but also change the character before the hyphen to **ChHres**.
0x04	hresDeleteBefore	Similar to Normal but also delete the character before the hyphen.
0x05	hresChangeAfter	Similar to Normal but also change the character after the hyphen to **ChHres**.
0x06	hresDelAndChange	Similar to Normal but also delete two characters before the hyphen and replace them both with **ChHres**.

ChHres (1 byte): An unsigned integer that specifies the **ASCII** character to be added to the text in addition to the hyphen. If **Hres** is set to **hresNormal**, **ChHres** MUST be 0x00; otherwise it MUST be a valid character.

2.9.119 Ico

The **Ico** structure specifies an entry in the color palette that is listed in the following table.



value (1 byte): An unsigned integer which maps to a COLORREF according to the following. The value MUST be less than 0x11.

	COLORREF			
Value	**Red**	**Green**	**Blue**	**fAuto**
0x00	0x00	0x00	0x00	0xFF
0x01	0x00	0x00	0x00	0x00
0x02	0x00	0x00	0xFF	0x00
0x03	0x00	0xFF	0xFF	0x00
0x04	0x00	0xFF	0x00	0x00
0x05	0xFF	0x00	0xFF	0x00
0x06	0xFF	0x00	0x00	0x00
0x07	0xFF	0xFF	0x00	0x00
0x08	0xFF	0xFF	0xFF	0x00

Copyright © 2025 Microsoft Corporation

Value	COLORREF			
	Red	**Green**	**Blue**	**fAuto**
0x09	0x00	0x00	0x80	0x00
0x0A	0x00	0x80	0x80	0x00
0x0B	0x00	0x80	0x00	0x00
0x0C	0x80	0x00	0x80	0x00
0x0D	0x80	0x00	0x80	0x00
0x0E	0x80	0x80	0x00	0x00
0x0F	0x80	0x80	0x80	0x00
0x10	0xC0	0xC0	0xC0	0x00

2.9.120 IDPCI

The **IDPCI** structure specifies the kind of formatting that the **format consistency checker** flagged within a region of text in the document. The possible values are showing following.

Name	Value	Meaning
idpciFmt	0x00000000	Character formatting in the region is inconsistent with formatting in the rest of the document.
idpciStyChar	0x00000001	Character style in the region is identical to a character style elsewhere in the document.
idpciPapc	0x00000002	Paragraph formatting in the region is inconsistent with formatting in the rest of the document.
idpciStyPara	0x00000003	Paragraph style in the region is identical to a paragraph style elsewhere in the document.
idpciLvl	0x00000004	Formatting of items in a numbered or bulleted list in the region is inconsistent with formatting in the rest of the document.
idpciStyList	0x00000005	Bulleted or numbered list style in the region is identical to a bulleted or numbered list style elsewhere in the document.
idpciStyTable	0x00000006	Table style in the region is identical to a table style elsewhere in the document.
idpciRevChar	0x00000007	(Revised Character) Characters in the region were changed while revision marking was on.
idpciRevPara	0x00000008	(Revised Paragraph) Paragraphs in the region were changed while revision marking was on.
idpciRevTable	0x00000009	(Revised Table) Tables in the region were changed while revision marking was on.
idpciRevSect	0x0000000A	(Revised Section) Sections in the region were changed while revision marking was on.
idpciImage	0x0000000B	A picture defined inline in the region has been combined, to save space, with an identical picture defined elsewhere in the document.

2.9.121 Ipat

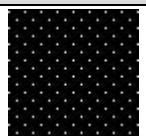
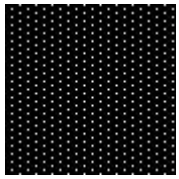
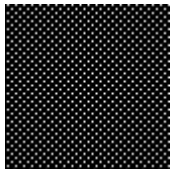
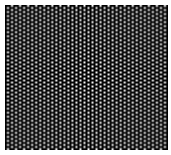
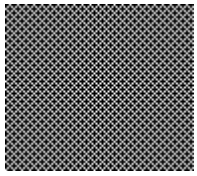
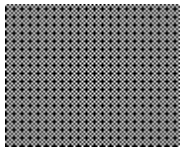
The **Ipat** enumeration is an index to a **shading pattern**. Most pattern indices listed in the following table are mapped to entries of ST_Shd, as specified in [ECMA-376] part 1, section 17.18.78 ST_Shd

Copyright © 2025 Microsoft Corporation

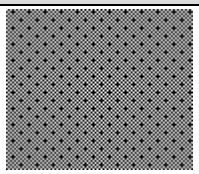
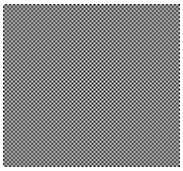
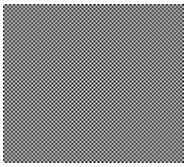
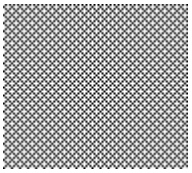
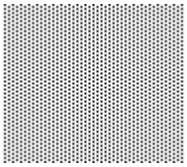
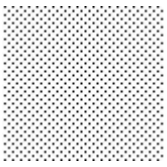
(Shading Patterns). All pattern indices that are not mapped to an ST_Shd value are not supported by the [ECMA-376] format and are lost if converted from the MS-DOC format to the [ECMA-376] format; these pattern values SHOULD NOT<225> be used.

Name	Value	Meaning
ipatAuto	0x0000	Clear, ST_Shd: clear
ipatSolid	0x0001	Solid ST_Shd: solid
ipatPct5	0x0002	5%, ST_Shd: pct5
ipatPct10	0x0003	10%, ST_Shd: pct10
ipatPct20	0x0004	20%, ST_Shd: pct20
ipatPct25	0x0005	25%, ST_Shd: pct25
ipatPct30	0x0006	30%, ST_Shd: pct30
ipatPct40	0x0007	40%, ST_Shd: pct40
ipatPct50	0x0008	50%, ST_Shd: pct50
ipatPct60	0x0009	60%, ST_Shd: pct60
ipatPct70	0x000A	70%, ST_Shd: pct70
ipatPct75	0x000B	75%, ST_Shd: pct75
ipatPct80	0x000C	80%, ST_Shd: pct80
ipatPct90	0x000D	90%, ST_Shd: pct90
ipatDkHorizontal	0x000E	Horizontal Stripe, ST_Shd: horzStripe
ipatDkVertical	0x000F	Vertical Stripe, ST_Shd: vertStripe
ipatDkForeDiag	0x0010	Reverse Diagonal Stripe, ST_Shd: reverseDiagStripe
ipatDkBackDiag	0x0011	Diagonal Stripe, ST_Shd: diagStripe
ipatDkCross	0x0012	Horizontal Cross, ST_Shd: horzCross
ipatDkDiagCross	0x0013	Diagonal Cross, ST_Shd: diagCross
ipatHorizontal	0x0014	Thin Horizontal Stripe, ST_Shd: thinHorzStripe
ipatVertical	0x0015	Thin Vertical Stripe, ST_Shd: thinVertStripe
ipatForeDiag	0x0016	Thin Reverse Diagonal Stripe, ST_Shd: thinReverseDiagStripe
ipatBackDiag	0x0017	Thin Diagonal Stripe, ST_Shd: thinDiagStripe
ipatCross	0x0018	Thin Horizontal Cross, ST_Shd: thinHorzCross
ipatDiagCross	0x0019	Thin Diagonal Cross, ST_Shd: thinDiagCross
ipatPctNew2	0x0023	Specifies that the pattern used for the current shaded region shall be a 2.5% fill pattern, as follows:

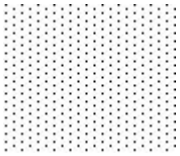
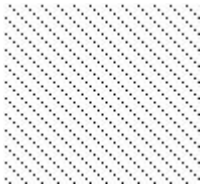
Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
		
ipatPctNew7	0x0024	Specifies that the pattern used for the current shaded region shall be a 7.5% fill pattern, as follows: 
ipatPctNew12	0x0025	12.5%, ST_Shd: pct12
ipatPctNew15	0x0026	15%, ST_Shd: pct15
ipatPctNew17	0x0027	Specifies that the pattern used for the current shaded region shall be a 17.5% fill pattern, as follows: 
ipatPctNew22	0x0028	Specifies that the pattern used for the current shaded region shall be a 22.5% fill pattern, as follows: 
ipatPctNew27	0x0029	Specifies that the pattern used for the current shaded region shall be a 27.5% fill pattern, as follows: 
ipatPctNew32	0x002A	Specifies that the pattern used for the current shaded region shall be a 32.5% fill pattern, as follows: 
ipatPctNew35	0x002B	35%, ST_Shd: pct35
ipatPctNew37	0x002C	37.5%, ST_Shd: pct37
ipatPctNew42	0x002D	Specifies that the pattern used for the current shaded region shall be a 42.5% fill pattern, as follows:

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
		
ipatPctNew45	0x002E	45%, ST_Shd: pct45
ipatPctNew47	0x002F	Specifies that the pattern used for the current shaded region shall be a 47.5% fill pattern, as follows: 
ipatPctNew52	0x0030	Specifies that the pattern used for the current shaded region shall be a 52.5% fill pattern, as follows: 
ipatPctNew55	0x0031	55%, ST_Shd: pct55
ipatPctNew57	0x0032	Specifies that the pattern used for the current shaded region shall be a 57.5% fill pattern, as follows: 
ipatPctNew62	0x0033	62.5%, ST_Shd: pct62
ipatPctNew65	0x0034	65%, ST_Shd: pct65
ipatPctNew67	0x0035	Specifies that the pattern used for the current shaded region shall be a 67.5% fill pattern, as follows: 
ipatPctNew72	0x0036	Specifies that the pattern used for the current shaded region shall be a 72.5% fill pattern, as follows: 

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
ipatPctNew77	0x0037	Specifies that the pattern used for the current shaded region shall be a 77.5% fill pattern, as follows: 
ipatPctNew82	0x0038	Specifies that the pattern used for the current shaded region shall be an 82.5% fill pattern, as follows: 
ipatPctNew85	0x0039	85%, ST_Shd: pct85
ipatPctNew87	0x003A	87.5%, ST_Shd: pct87
ipatPctNew92	0x003B	Specifies that the pattern used for the current shaded region shall be a 92.5% fill pattern, as follows: 
ipatPctNew95	0x003C	95%, ST_Shd: pct95
ipatPctNew97	0x003D	Specifies that the pattern used for the current shaded region shall be a 97.5% fill pattern, as follows: 
ipatNil	0xFFFF	Nil, ST_Shd: nil

2.9.122 IScrollType

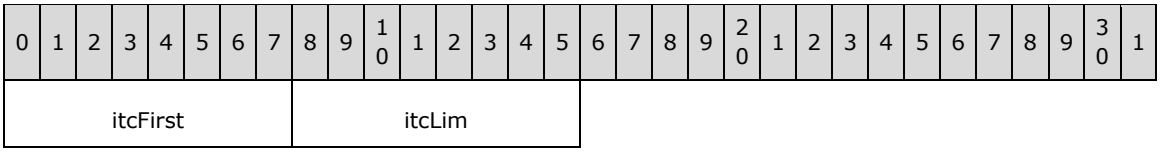
The **IScrollType** enumerated type specifies the scrollbar behavior for a frame. A field of this type MUST contain one of the following values.

Name	Value	Meaning
iScrollAuto	0x00000000	A scrollbar appears only if it is needed.
iScrollYes	0x00000001	A scrollbar appears even if not needed.
iScrollNo	0x00000002	The frame never has a scrollbar.

Copyright © 2025 Microsoft Corporation

2.9.123 ItcFirstLim

The **ItcFirstLim** structure specifies a range of cells in a table row. The range is inclusive of the first index, and exclusive of the second. The first cell in a row is at index 0. The maximum number of cells in a row is 63.

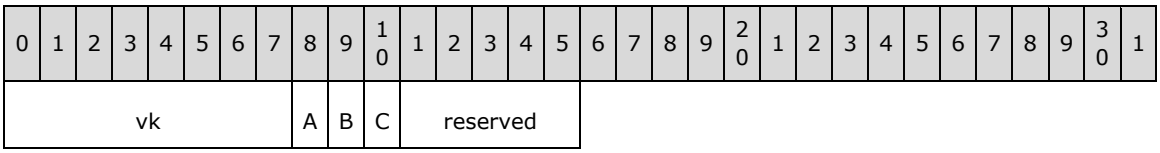


itcFirst (8 bits): An integer value that specifies the index of the first cell in a contiguous range. The cell at this index is inside the range. This value MUST be non-negative and MUST be less than the number of cells in the row.

itcLim (8 bits): An integer value that specifies the index of the first cell beyond the contiguous range. The cell at this index is outside the range. This value MUST be greater than or equal to **itcFirst** and MUST be less than or equal to the number of cells in the row. When **itcLim** is equal to **itcFirst**, the range contains zero cells.

2.9.124 Kcm

The **Kcm** structure specifies a shortcut key combination through a **virtual key code** and modifiers.



vk (1 byte): An integer that specifies the Virtual key code for this shortcut key combination.

A - fkmShift (1 bit): Specifies whether the SHIFT key is pressed in this shortcut key combination.

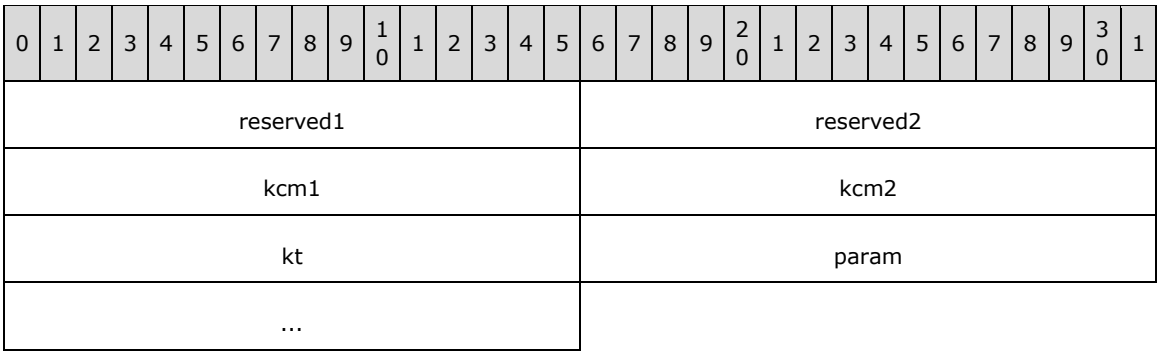
B - fkmControl (1 bit): Specifies whether the CTRL key is pressed in this shortcut key combination.

C - fkmAlt (1 bit): Specifies whether the ALT key is pressed in this shortcut key combination.

reserved (5 bits): This value MUST be 0.

2.9.125 Kme

The **Kme** structure specifies a mapping of a shortcut key to a command to be executed.



reserved1 (2 bytes): This value MUST be 0.

Copyright © 2025 Microsoft Corporation

reserved2 (2 bytes): This value MUST be 0.

kcm1 (2 bytes): A Kcm that specifies the **primary shortcut key**.

kcm2 (2 bytes): A Kcm that specifies the **secondary shortcut key**, or 0x00FF if there is no secondary shortcut key.

kt (2 bytes): A Kt that specifies the type of action to be taken when the key combination is pressed.

param (4 bytes): The meaning of this field depends on the value of **kt**, as follows.

kt	param
ktCid	A Cid that specifies a command to be executed.
ktChar	A 4-byte unsigned integer that specifies a single character to be inserted. This value MUST be between 0 and 65535.
ktMask	This MUST be ignored.

2.9.126 Kt

The **Kt** enumeration specifies the type of action to be taken when a shortcut key combination is pressed. This enumeration is used by the **Kme** structure.

Name	Value	Meaning
ktCid	0x0000	Execute a command specified by a Cid.
ktChar	0x0001	Insert a single character.
ktMask	0x0003	Perform the default action (as if the key combination is unassigned).

2.9.127 Kul

The **Kul** enumeration specifies the style of underlining for text.

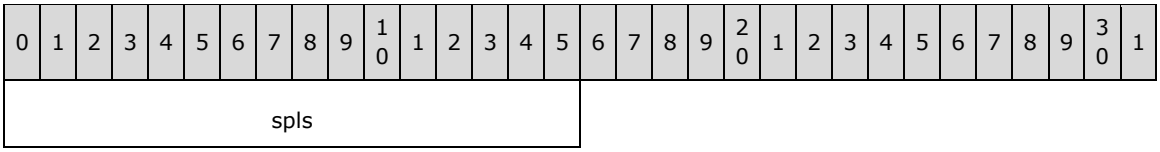
Name	Value	Meaning
kulNone	0x00	No underlining.
kulSingle	0x01	Normal single underline.
kulWords	0x02	Underline words only.
kulDouble	0x03	Double underline.
kulDotted	0x04	Dotted underline.
kulThick	0x06	Heavy underline.
kulDash	0x07	Dashed underline.
kulDotDash	0x09	Dot-dash underline.
kulDotDotDash	0x0A	Dot-dot-dash underline.
kulWavy	0x0B	Wavy underline.
kulDottedHeavy	0x14	Heavy dotted underline.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
kulDashHeavy	0x17	Heavy dashed underline.
kulDotDashHeavy	0x19	Heavy dot-dash underline.
kulDotDotDashHeavy	0x1A	Heavy dot-dot-dash underline.
kulWavyHeavy	0x1B	Heavy wavy underline.
kulDashLong	0x27	Long-dash underline.
kulWavyDouble	0x2B	Wavy double underline.
kulDashLongHeavy	0x37	Heavy long-dash underline.

2.9.128 LadSpls

The **LadSpls** structure is an **SPLS** structure that specifies the state of the language auto-detection over a range of text. Some states that are possible in a generic **SPLS** are not allowed in a **LadSpls** structure.



spls (2 bytes): An **SPLS** structure. The **spls.fExtend** and **spls.fTypo** fields are not used and MUST be zero. The **spls.splf** field MUST be one of the following:

- splfMaybeDirty
- splfDirty
- splfEdit
- splfForeign
- splfClean
- splfNoLAD

2.9.129 LBCOperand

The **LBCOperand** enumeration specifies where text continues after a line break. When a line is shortened or broken into multiple text regions by the presence of a picture, shape, or another object, the operand specifies the location at which the text continues. If a line is not broken by an object, the following values have no effect and the text simply continues on the next line.

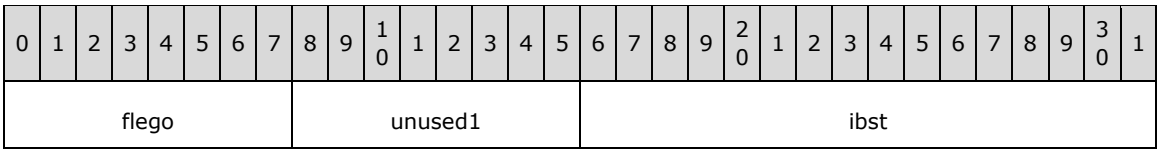
Name	Value	Meaning
lbrNone	0x00	Text continues in the next available region of the current line, in logical reading order, or on the next line if no more regions are left.
lbrLeft	0x01	If the line break is located to the logical left of the object, text restarts in the next available region of the current line, in logical reading order, or on the next line if no more regions are left. If the line break is located to the logical right of the object, text restarts on the next available line that is not broken by an object. In this case, the use of this value has the same

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
		result as the use of the value **lbrBoth**.
lbrRight	0x02	If the line break is located to the logical right of the object, text restarts in the next available region of the current line, in logical reading order, or on the next line if no more regions are left. If the line break is located to the logical left of the object, text restarts on the next available line that is not broken by an object. In this case, the use of this value has the same result as the use of the value **lbrBoth**.
lbrBoth	0x03	Text restarts on the next available line that is not broken by the presence of a picture, shape, or another object.

2.9.130 LEGOXTR_V11

The **LEGOXTR_V11** structure contains information about an **AutoText** item.



flego (1 byte): An unsigned integer that specifies the type of an AutoText item. This MUST be one of the following values.

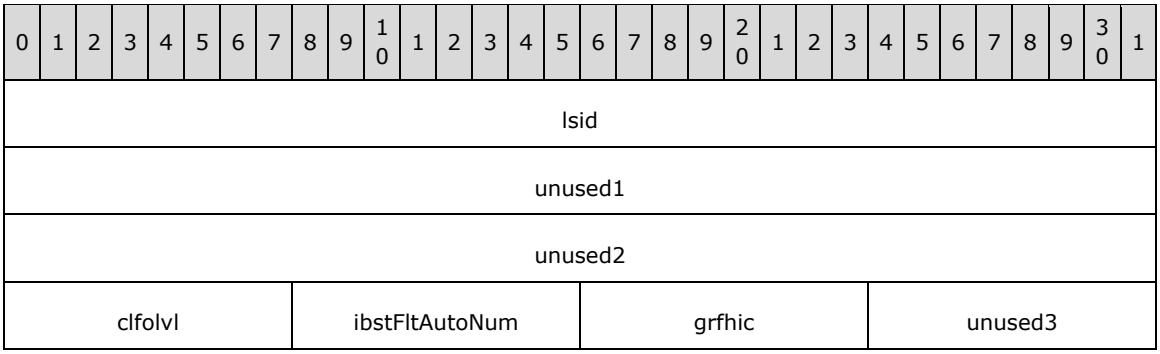
Value	Meaning
0x00	The item is a named AutoText item.
0x0A	The item is a formatted text **AutoCorrect** item.

unused1 (1 byte): This field MUST be ignored.

ibst (2 bytes): A signed integer that specifies a zero-based index into SttbGlsyStyle. The string at this index is the name of the **style** that is used by the AutoText item. If this integer is equal to 0xFFFF, there is no style used by the AutoText item. If **flego** is nonzero, this MUST be equal to 0xFFFF.

2.9.131 LFO

The **LFO** structure specifies the **LSTF** element that corresponds to a list that contains a paragraph. An **LFO** can also specify formatting information that overrides the **LSTF** element to which it corresponds.



Copyright © 2025 Microsoft Corporation

lsid (4 bytes): A signed integer that specifies the list identifier of an LSTF. This **LFO** corresponds to the LSTF in PlfLst.**rgLstf** that has an **lsid** whose value is equal to this value.

unused1 (4 bytes): This field MUST be ignored.

unused2 (4 bytes): This field MUST be ignored.

clfolvl (1 byte): An unsigned integer that specifies the count of **LFOLVL** elements that are stored in the **rgLfoLvl** field of the **LFOData** element that corresponds to this **LFO** structure.

ibstFltAutoNum (1 byte): An unsigned integer that specifies the **field** that this **LFO** represents. This MUST be one of the following values.

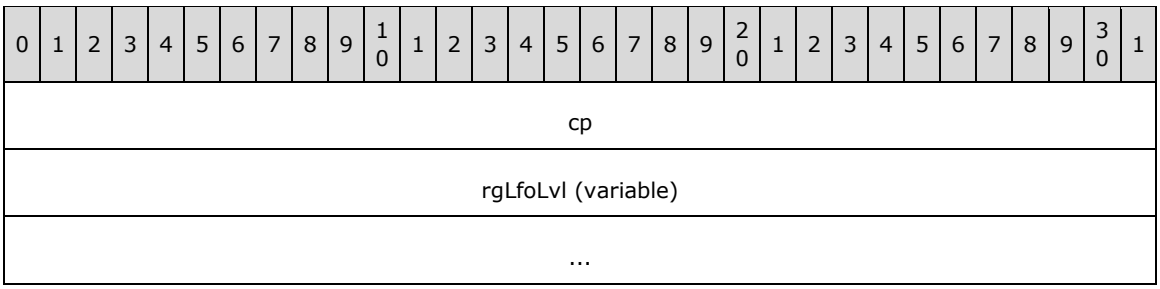
Value	Meaning
0x00	This LFO is not used for any field. The **fAutoNum** of the related LSTF MUST be set to 0.
0xFC	This LFO is used for the AUTONUMLGL field (see AUTONUMLGL in flt). The **fAutoNum** of the related LSTF MUST be set to 1.
0xFD	This LFO is used for the AUTONUMOUT field (see AUTONUMOUT in flt). The **fAutoNum** of the related LSTF MUST be set to 1.
0xFE	This LFO is used for the AUTONUM field (see AUTONUM in flt). The **fAutoNum** of the related LSTF MUST be set to 1.
0xFF	This LFO is not used for any field. The **fAutoNum** of the related LSTF MUST be set to 0.

grfhic (1 byte): A **grfhic** that specifies HTML incompatibilities.

unused3 (1 byte): This field MUST be ignored.

2.9.132 LFOData

The **LFOData** structure contains the Main Document CP of the corresponding LFO, as well as an array of LVL override data.



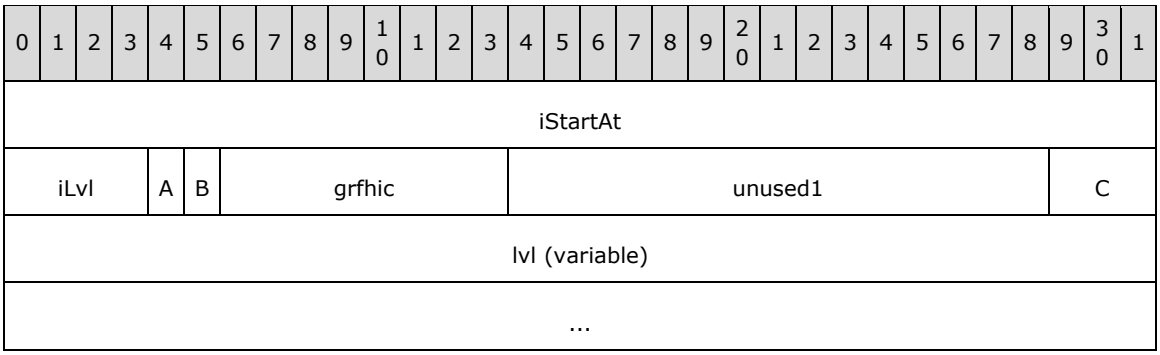
cp (4 bytes): A CP that specifies the position of the first paragraph in the Main Document whose **iLfo** property (see sprmPIlfo) specifies the corresponding LFO. If this is equal to 0xFFFFFFFF, this MUST be ignored.

rgLfoLvl (variable): An array of LFOLVL. The **cLfolvl** field of the corresponding LFO specifies the count of elements in this array.

Copyright © 2025 Microsoft Corporation

2.9.133 LFOLVL

The **LFOLVL** structure contains information that is used to override the formatting information of a corresponding **LVL**.



iStartAt (4 bytes): If fStartAt is set to 0x1, this is a signed integer that specifies the start-at value that overrides lvlf.iStartAt of the corresponding LVL. This value MUST be less than or equal to 0x7FFF and MUST be greater than or equal to zero. If both fStartAt and fFormatting are set to 0x1, or if fStartAt is set to 0x0, this value is undefined and MUST be ignored.

iLvl (4 bits): An unsigned integer that specifies the zero-based level of the list that this overrides. This LFOLVL overrides the LVL that specifies the level formatting of this level of the LSTF that is specified by the **lsid** field of the LFO to which this LFOLVL corresponds. This value MUST be less than or equal to 0x08.

A - fStartAt (1 bit): A bit that specifies whether this LFOLVL overrides the start-at value of the level.

B - fFormatting (1 bit): A bit that specifies whether **lvl** is an LVL that overrides the corresponding LVL.

grfhic (8 bits): A grfhic that specifies the HTML incompatibilities of the overriding level formatting.

unused1 (15 bits): This MUST be ignored.

C - unused2 (3 bits): This MUST be ignored.

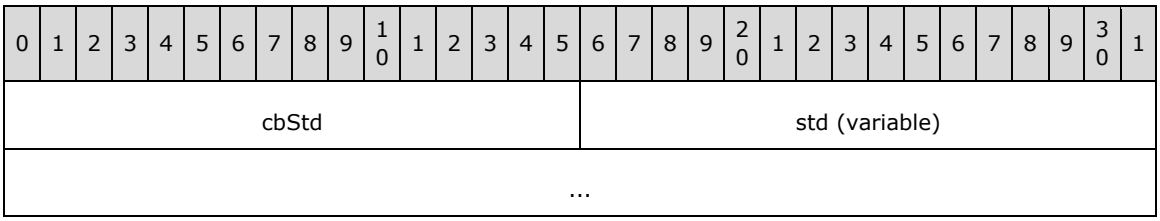
lvl (variable): If **fFormatting** is set to 0x1, this is an LVL that completely overrides the LVL to which this LFOLVL corresponds. If **fFormatting** is not set to 0x1, this does not exist.

2.9.134 LID

The **LID** structure is an unsigned 16-bit integer that specifies a language code.

2.9.135 LPStd

The **LPStd** structure specifies a length-prefixed style definition.



Copyright © 2025 Microsoft Corporation

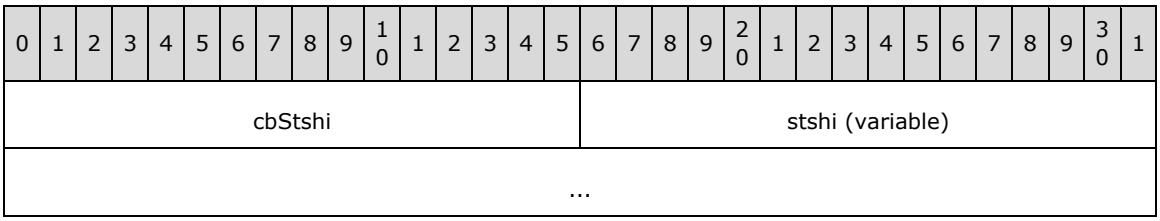
cbStd (2 bytes): A signed integer that specifies the size, in bytes, of **std**. This value MUST NOT be less than 0. LPStd structures are stored on even-byte boundaries, but this length MUST NOT include this padding.

A style definition can be empty, in which case **cbStd** MUST be 0.

std (variable): An STD that specifies the style definition.

2.9.136 LPStshi

The **LPStshi** structure specifies general stylesheet information.

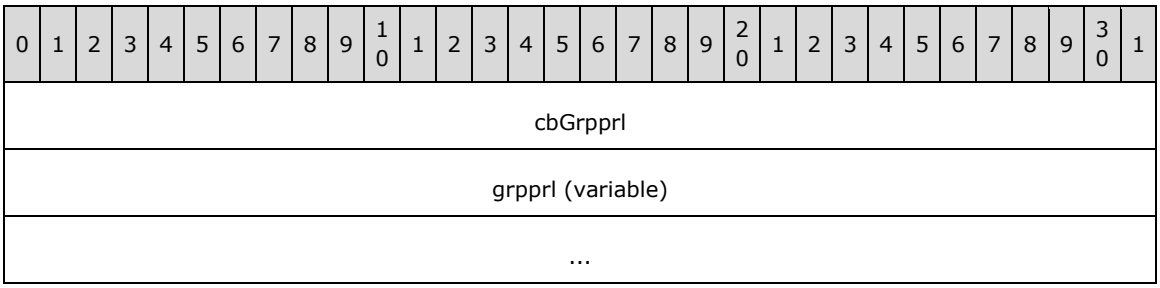


cbStshi (2 bytes): An unsigned integer that specifies the size, in bytes, of **stshi**.

stshi (variable): A stshi that specifies general stylesheet information.

2.9.137 LPStshiGrpPrl

The **LPStshiGrpPrl** structure specifies an array of formatting properties.

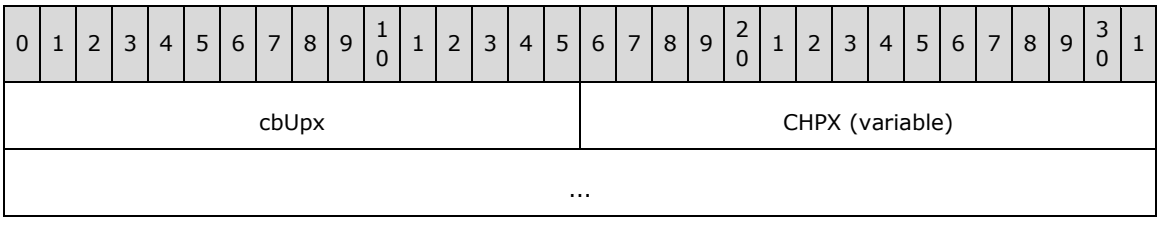


cbGrpprl (4 bytes): A signed 32-bit integer that specifies the size, in bytes, of **grpprl**.

grpprl (variable): An array of Prl elements that specify formatting properties.

2.9.138 LPUpxChpx

The **LPUpxChpx** structure specifies character formatting properties. This structure is padded to an even length, but the length in **cbUpx** MUST NOT include this padding.



cbUpx (2 bytes): An unsigned integer that specifies the size, in bytes, of **CHPX**. This value does not include the padding.

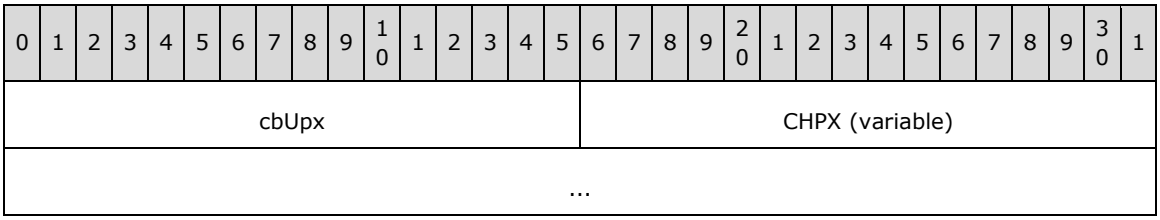
Copyright © 2025 Microsoft Corporation

CHPX (variable): A UpxChpx that specifies character formatting properties.

2.9.139 LPUpxChpxRM

The **LPUpxChpxRM** structure that specifies character formatting properties for revision-marked style formatting.

The structure is padded to an even length.



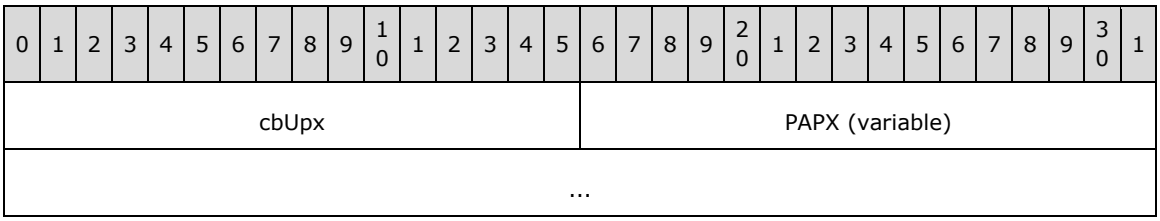
cbUpx (2 bytes): An unsigned integer that specifies the length, in bytes, of **CHPX**. This value MUST NOT include padding.

CHPX (variable): A UpxChpx that specifies character formatting properties.

2.9.140 LPUpxPapx

The **LPUpxPapx** structure specifies paragraph formatting properties.

The structure is padded to an even length, but the length in **cbUpx** MUST NOT include this padding.



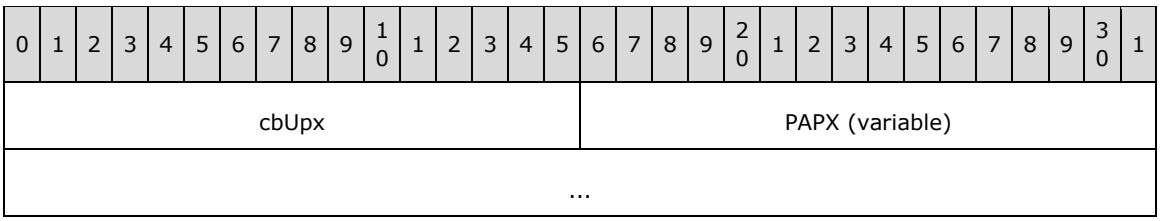
cbUpx (2 bytes): An unsigned integer that specifies the size, in bytes, of **PAPX**, not including the (potential) padding.

PAPX (variable): A UpxPapx that specifies paragraph formatting properties.

2.9.141 LPUpxPapxRM

The **LPUpxPapxRM** structure specifies the paragraph formatting properties that are used for revision-marked style formatting.

The structure is padded to be an even length, but the length in **cbUpx** MUST NOT include this padding.



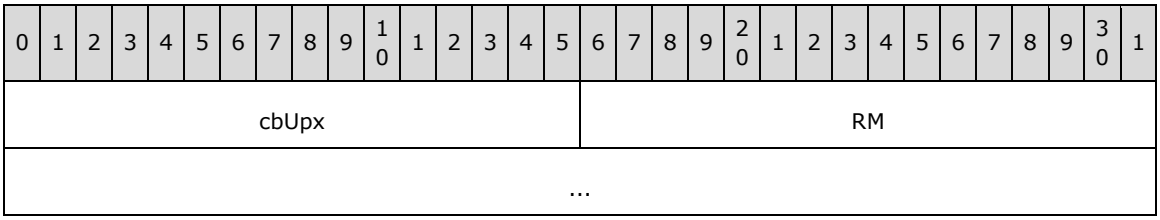
Copyright © 2025 Microsoft Corporation

cbUpx (2 bytes): An unsigned 16-bit integer that specifies the size, in bytes, of **PAPX**. This value does not include any specified padding.

PAPX (variable): A UpxPapx that specifies paragraph formatting properties.

2.9.142 LPUpxRm

The **LPUpxRm** structure specifies revision-marking information.

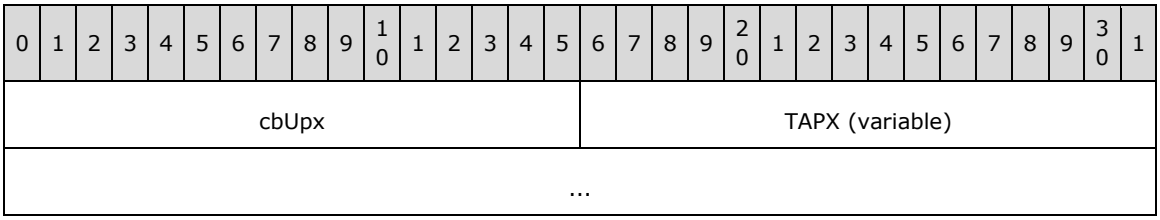


cbUpx (2 bytes): An unsigned 16-bit integer that specifies the size, in bytes, of **RM**. This value MUST be 0x0006.

RM (6 bytes): An UpxRm that specifies revision-marking information.

2.9.143 LPUpxTapx

The **LPUpxTapx** structure specifies table formatting properties. This structure is padded to an even length, but the length in **cbUpx** MUST NOT include this padding.

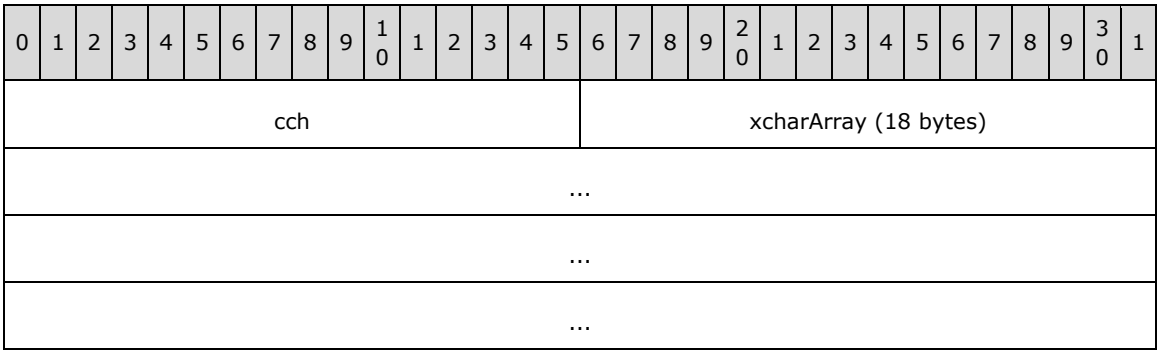


cbUpx (2 bytes): An unsigned integer that specifies the size, in bytes, of **TAPX**. This value does not include padding.

TAPX (variable): A UpxTapx that specifies table formatting properties.

2.9.144 LPXCharBuffer9

The **LPXCharBuffer9** structure is a length-prefixed buffer for up to 9 **Unicode** characters. The text is not null-terminated.



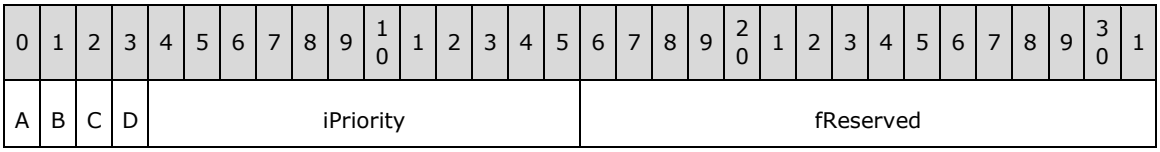
Copyright © 2025 Microsoft Corporation

cch (2 bytes): An unsigned integer that specifies the number of characters from the buffer that are actually used. This value MUST be less than or equal to 9.

xcharArray (18 bytes): An array of 16-bit Unicode characters. The first **cch** characters make a Unicode string. The remaining characters MUST be ignored.

2.9.145 LSD

The **LSD** structure specifies the properties to be used for latent application-defined styles (see StshiLsd) when they are created.



A - fLocked (1 bit): Specifies the value that the **fLocked** field of GRFSTD is set to when this latent style is instantiated.

B - fSemiHidden (1 bit): Specifies the value that the **fSemiHidden** field of GRFSTD is set to when this latent style is instantiated.

C - fUnhideWhenUsed (1 bit): Specifies the value that the **fUnhideWhenUsed** field of GRFSTD is set to when this latent style is instantiated.

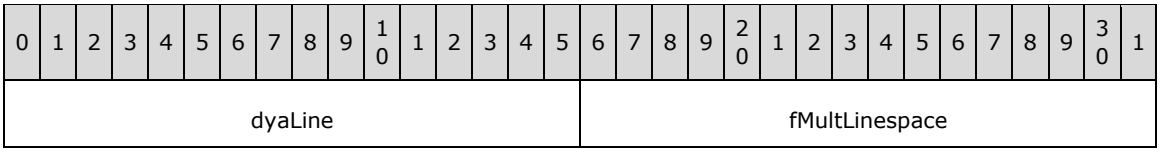
D - fQFormat (1 bit): Specifies the value that the **fQFormat** field of GRFSTD is set to when this latent style is instantiated.

iPriority (12 bits): An unsigned integer that specifies the value that the **iPriority** field of StdfPost2000 is set to when this latent style is instantiated. This MUST be a value between 0x0000 and 0x0063, inclusive.

fReserved (16 bits): This value MUST be 0 and MUST be ignored.

2.9.146 LSPD

The **LSPD** structure specifies the spacing between lines in a paragraph.



dyaLine (16 bits): An integer that specifies the spacing between lines, based on the following rules:

- **dyaLine** MUST either be between 0x0000 and 0x7BC0 or between 0x8440 and 0xFFFF.
- When **dyaLine** is between 0x8440 and 0xFFFF, the line spacing, in **twips**, is exactly 0x10000 minus **dyaLine**.
- When **fMultLinespace** is 0x0001 and **dyaLine** is between 0x0000 and 0x7BC0, a spacing multiplier is used to determine line spacing for this paragraph. The spacing multiplier is **dyaLine**/240. For example, a spacing multiplier value of 1 specifies single spacing; a spacing multiplier value of 2 specifies double spacing; and so on. The actual line spacing, in twips, is the spacing multiplier times the font size, in twips.

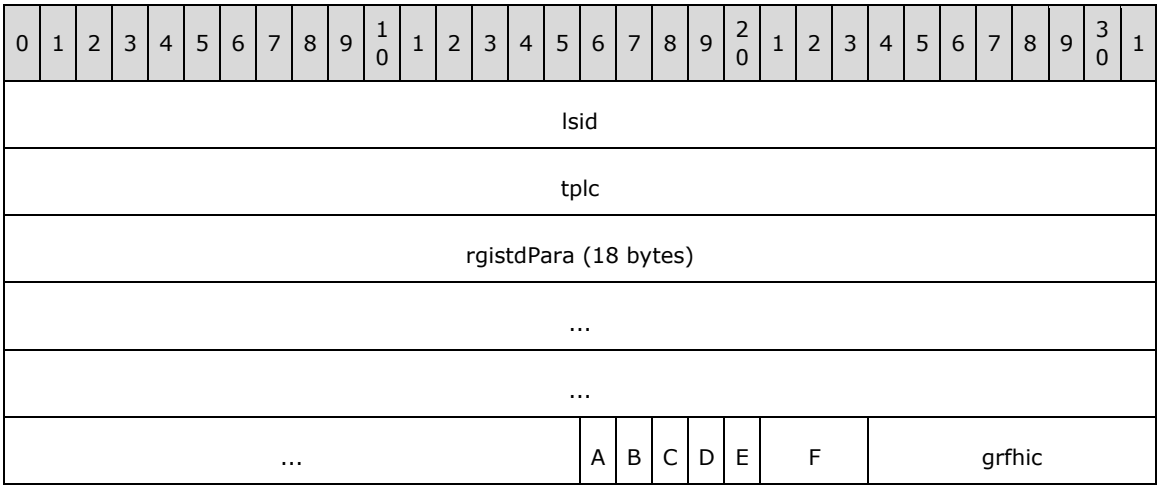
Copyright © 2025 Microsoft Corporation

- When **fMultLinespace** is 0x0000 and **dyaLine** is between 0x0000 and 0x7BC0, the line spacing, in twips, is **dyaLine** or the number of twips necessary for single spacing, whichever value is greater.

fMultLinespace (16 bits): An integer which MUST be either 0x0000 or 0x0001.

2.9.147 LSTF

The **LSTF** structure contains formatting properties that apply to an entire list.



lsid (4 bytes): A signed integer that specifies the list identifier. This MUST be unique for each LSTF. This value MUST NOT be 0xFFFFFFFF.

tplc (4 bytes): A Tplc that specifies a unique identifier for this LSTF that MAY<226> be used for user interface purposes. If **fHybrid** is nonzero, this MUST be ignored.

rgistdPara (18 bytes): An array of nine 16-bit signed integers. Each element of **rgistdPara** specifies the ISTD of the style that is linked to the corresponding level in the list. If no style is linked to a given level, the value of the corresponding element of **rgistdPara** MUST be 0x0FFF.

A - fSimpleList (1 bit): A bit that, when set to 0x1, specifies that this LSTF represents a simple (one-level) list that has one corresponding LVL (see the **fcPlfLst** field of FibRgFcLcb97). Otherwise, this LSTF represents a multi-level list that has nine corresponding LVLs.

B - unused1 (1 bit): This bit MUST be ignored.

C - fAutoNum (1 bit): A bit that specifies whether the list that this LSTF represents is used for the AUTONUMOUT, AUTONUMLGL, and AUTONUM fields (see AUTONUMOUT, AUTONUMLGL, and AUTONUM in flt).

D - unused2 (1 bit): This bit MUST be ignored.

E - fHybrid (1 bit): A bit that specifies whether the list this LSTF defines is a **hybrid list**.

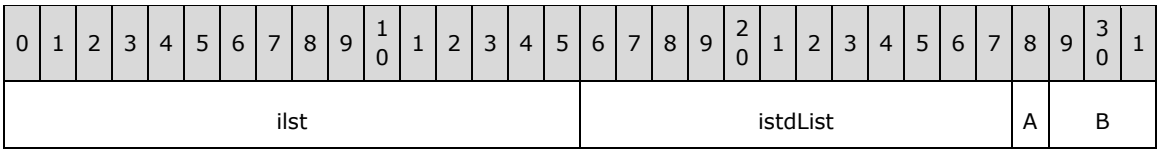
F - reserved1 (3 bits): This MUST be zero, and MUST be ignored.

grfhic (1 byte): A grfhic that specifies the HTML incompatibilities of the list.

2.9.148 Lstsf

The **Lstsf** structure specifies a list style.

Copyright © 2025 Microsoft Corporation



ilst (2 bytes): An unsigned integer that specifies a zero-based index into the **PlfIst**.

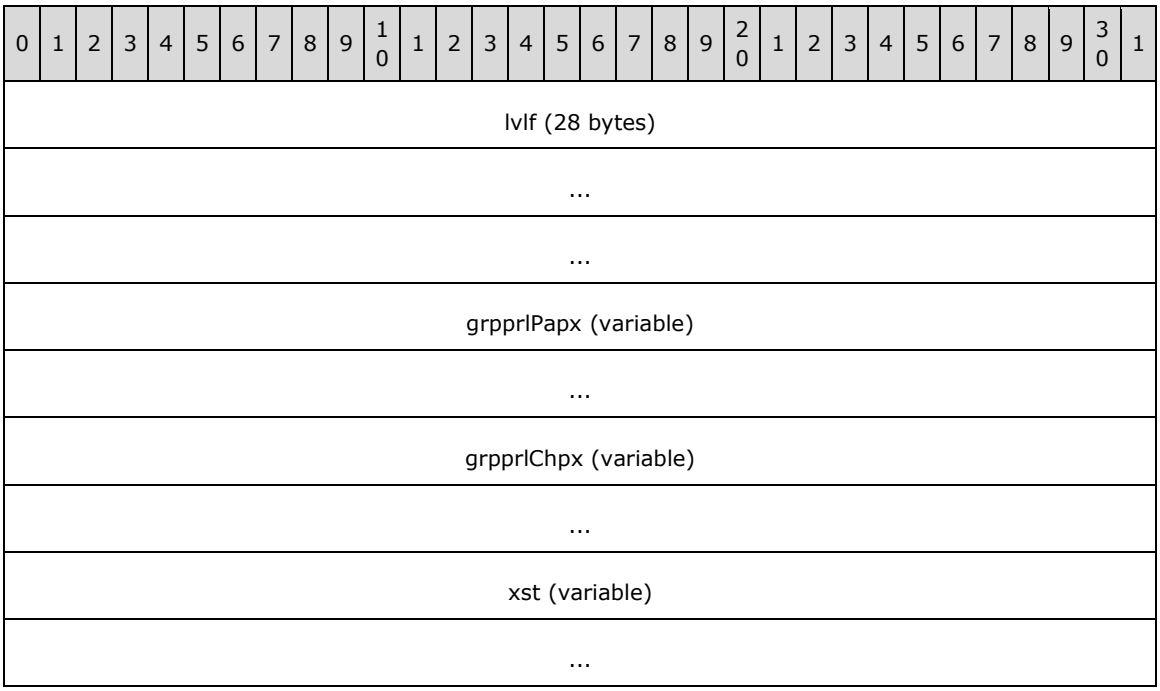
istdList (12 bits): An unsigned integer that specifies the **ISTD** for the list style. To determine the text properties, see Determining Properties of a Style (section 2.4.6.5).

A - fStyleDef (1 bit): A bit flag that specifies the type of this list definition. If **fStyleDef** is "true", this **Lstsf** is a list style definition, meaning that a custom numbered or bulleted list style was defined. In this case, **ilst** specifies which custom list style is to be used. If **fStyleDef** is "false", it means that a standard list style is used. In this case, **istdList** specifies which standard style to use.

B - fUnused (3 bits): This field MUST be zero and MUST be ignored.

2.9.149 LVL

The **LVL** structure contains formatting information about a specific level in a list. When a paragraph is formatted as part of this level, each placeholder in **xst** is replaced with the inherited level number of the most recent or current paragraph in the same list that is in the zero-based level specified by that placeholder. The level number that replaces a placeholder is formatted according to the **lvlf.nfc** of the **LVL** structure that corresponds to the level that the placeholder specifies, unless the **lvlf.fLegal** of this **LVL** structure is nonzero.



lvlf (28 bytes): An **LVLF** structure that specifies formatting information for this level.

grpprlPapx (variable): An array of Prl elements that specifies the paragraph formatting of a paragraph in this level. The size of **grpprlPapx** is specified by **lvlf.cbGrpprlPapx**.

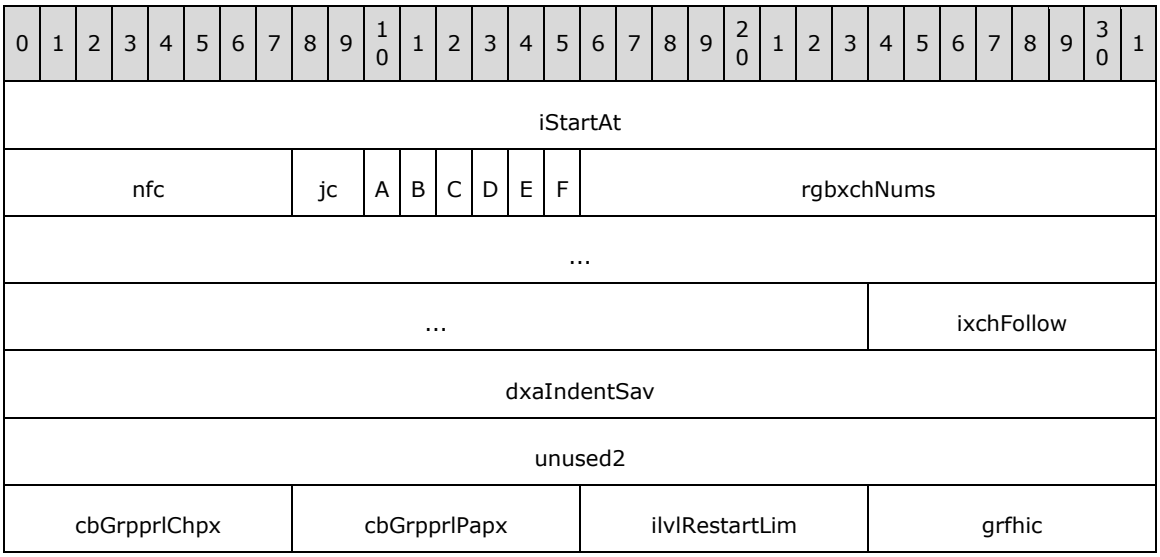
Copyright © 2025 Microsoft Corporation

grpprlChpx (variable): An array of Prl elements that specifies the character formatting of the **number text** that begins each paragraph in this level. The size of **grpprlChpx** is specified by **lvlf.cbGrpprlChpx**.

xst (variable): An Xst that specifies the number text that begins each paragraph in this level. This can contain placeholders for level numbers that are inherited from the other paragraphs in the list. Any element in the **rgtchar** field of this Xst can be a placeholder. Each placeholder is an unsigned 2-byte integer that specifies the zero-based level that the placeholder is for. Each placeholder MUST have a value that is less than or equal to the zero-based level of the list that this **LVL** represents. The indexes of the placeholders are specified by **lvlf.rgbxchNums**. Placeholders that correspond to levels that do not have a number sequence (see **lvlf.nfc**) MUST be ignored. If this level uses bullets (see **lvlf.nfc**), the **cch** field of this Xst MUST be equal to 0x0001, and this MUST NOT contain any placeholders.

2.9.150 LVLF

The **LVLF** structure contains formatting properties for an individual level in a list.



iStartAt (4 bytes): A signed integer that specifies the beginning number for the number sequence belonging to this level. This value MUST be less than or equal to 0x7FFF and MUST be greater than or equal to zero. If this level does not have a number sequence (see **nfc**), this MUST be ignored.

nfc (1 byte): An **MSONFC**, as specified in [MS-OSHARED] section 2.2.1.3, that specifies the format of the level numbers that replace the placeholders for this level in the **xst** fields of the LVLs in this list. This value MUST NOT be equal to 0x08, 0x09, 0x0F, or 0x13. If this is equal to 0xFF or 0x17, this level does not have a number sequence and therefore has no number formatting. If this is equal to 0x17, the level uses bullets.

jc (2 bits): An unsigned integer that specifies the justification of this level. This MUST be one of the following values.

Value	Meaning
0x0	Left justified
0x1	Center justified

Copyright © 2025 Microsoft Corporation

Value	Meaning
0x2	Right justified

A - fLegal (1 bit): A bit that specifies whether this level overrides the **nfc** of all inherited level numbers. If the original **nfc** of a level number is **msonfcArabicLZ**, it is preserved. Otherwise, the **nfc** of the level number is overridden by **msonfcArabic**.

B - fNoRestart (1 bit): A bit that specifies whether the number sequence of the level does not restart after a level is encountered that is more significant than the level to which this **LVLF** corresponds. If this is nonzero, **ilvlRestartLim** specifies the levels after which the number sequence of this level restarts. Otherwise, this number sequence of this level restarts when a more significant level is encountered. If this level does not have a number sequence (see **nfc**), this MUST be ignored.

C - fIndentSav (1 bit): A bit that specifies whether the level indented the text it was applied to and that the indent needs to be removed when numbering is removed. The indent to be removed is stored in **dxaIndentSav**.

D - fConverted (1 bit): A bit that specifies whether the **nfc** of this **LVLF** structure was previously a temporary value used for **bidirectional compatibility** that was converted into a standard **MSONFC**, as specified in [MS-OSHARED] section 2.2.1.3.

E - unused1 (1 bit): This bit MUST be ignored.

F - fTentative (1 bit): A bit that specifies whether the format of the level is tentative. This is used to describe the levels of a **hybrid list** that are not in use or displayed. If the **fHybrid** bit of the LSTF of the list is zero, this MUST be ignored.

rgbxchNums (9 bytes): An array of 8-bit integers. Each integer specifies a one-based character offset to a level placeholder in the **xst.rgtchar** of the **LVL** that contains this **LVLF**. This array is zero-terminated, unless it is full. The count of elements in this array, before to the first terminating zero, MUST be less than or equal to the one-based level of the list to which this **LVL** corresponds. The integers in this array, before the first terminating zero, MUST be in ascending order, and MUST be unique.

ixchFollow (1 byte): An unsigned integer that specifies the character that follows the number text. This MUST be one of the following values.

Value	Meaning
0x0	A tab follows the number text.
0x1	A space follows the number text.
0x2	Nothing follows the number text.

dxaIndentSav (4 bytes): If **fIndentSav** is nonzero, this is a signed integer that specifies the size, in **twips**, of the indent that needs to be removed when the numbering is removed. This MUST be less than or equal to 0x00007BC0 or greater than or equal to 0xFFFF8440. If **fIndentSav** is zero, this MUST be ignored.

unused2 (4 bytes): This field MUST be ignored.

cbGrpprlChpx (1 byte): An unsigned integer that specifies the size, in bytes, of the **grpprlChpx** in the **LVL** that contains this **LVLF**.

Copyright © 2025 Microsoft Corporation

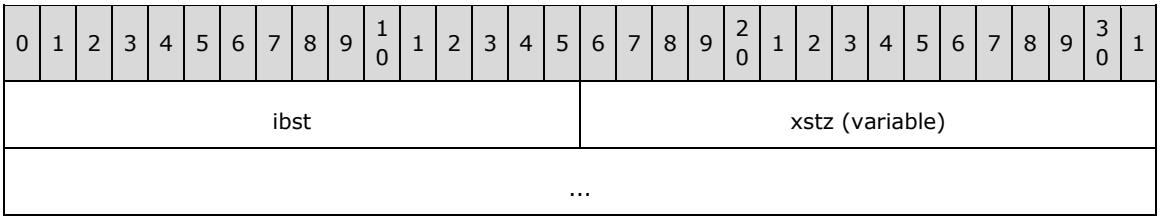
cbGrpprlPapx (1 byte): An unsigned integer that specifies the size, in bytes, of the **grpprlPapx** in the **LVL** that contains this **LVLF**.

ilvlRestartLim (1 byte): An unsigned integer that specifies the first (most-significant) zero-based level after which the number sequence of this level does not restart. The number sequence of this level does restart after any level that is more significant than the specified level. This MUST be less than or equal to the zero-based level of the list to which this **LVLF** corresponds. If **fNoRestart** is zero, this MUST be ignored. If this level does not have a number sequence (see **nfc**), this MUST be ignored.

grfhic (1 byte): A **grfhic** that specifies the HTML incompatibilities of the level.

2.9.151 MacroName

The **MacroName** structure specifies a single entry in the macro name table, as defined in MacroNames.

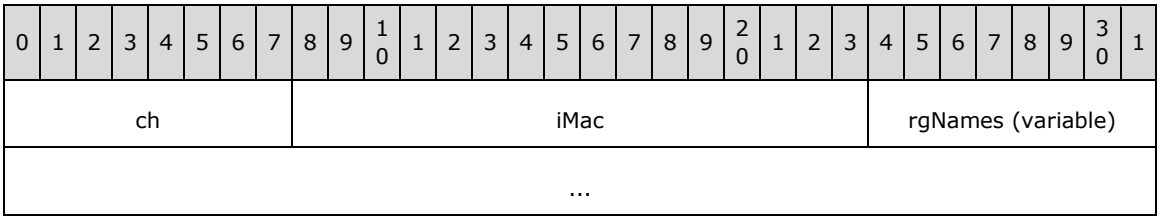


ibst (2 bytes): An unsigned integer that specifies the index of the current entry in the macro name table. This MUST NOT be the same as the index of any other entry.

xstz (variable): An **Xstz** structure that specifies the name of the macro. The length of the string, excluding the terminating null character, MUST NOT exceed 255 characters.

2.9.152 MacroNames

The **MacroNames** structure specifies the macro name table. This structure is used in a sequence of structures that specify command-related customizations. For more information, see the **Tcg255** structure.



ch (1 byte): An unsigned integer that identifies this structure as a **MacroNames** structure. This value MUST be 17.

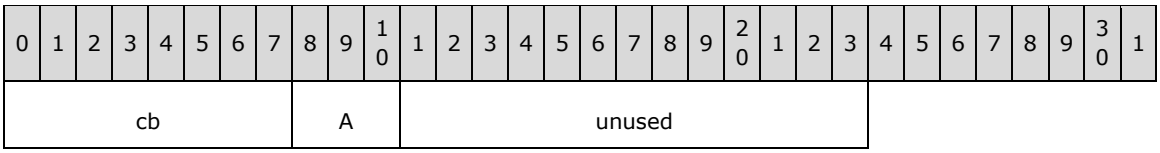
iMac (2 bytes): An unsigned integer that specifies the number of **MacroName** structures in **rgNames**.

rgNames (variable): An array of **MacroName** structures. The number of structures is specified by **iMac**.

2.9.153 MathPrOperand

The **MathPrOperand** structure is an operand to sprmCFMathPr. This operand specifies the justification for equations.

Copyright © 2025 Microsoft Corporation



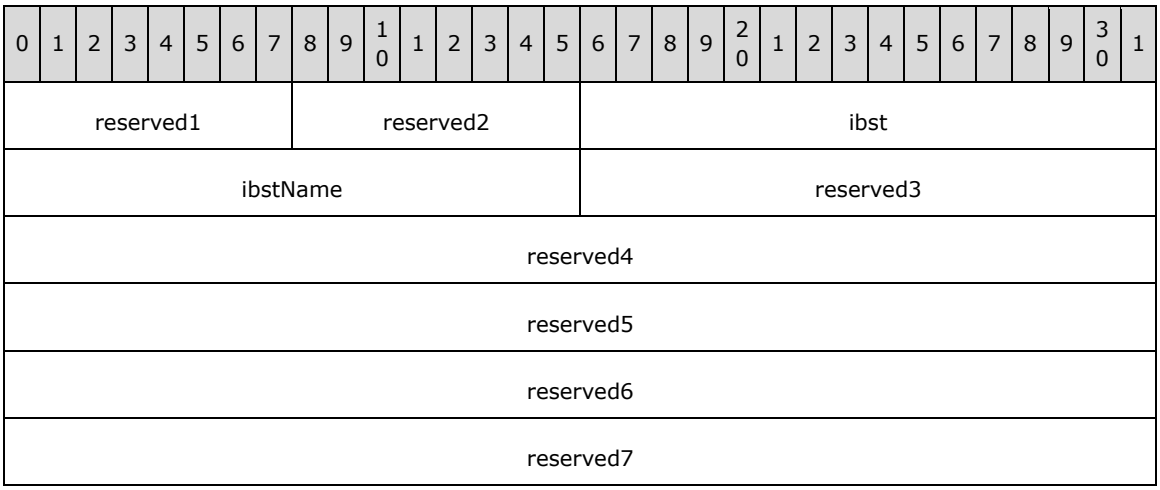
cb (1 byte): The size of this structure, in bytes, not including this byte. This value MUST be 0x02.

A - jcMath (3 bits): Specifies the justification. The valid values and their meanings are specified in the **mthbpjc** member of DOPMTH.

unused (13 bits): This field is undefined and MUST be ignored.

2.9.154 Mcd

The **Mcd** structure specifies a macro.



reserved1 (1 byte): A signed integer that MUST be 0x56.

reserved2 (1 byte): This value MUST be 0.

ibst (2 bytes): An unsigned integer that specifies the name of the macro. The macro name is specified by MacroName.**xstz** of the MacroName entry in the MacroNames, such that MacroName.**ibst** equals **ibst**. MacroNames MUST contain such an entry.

ibstName (2 bytes): An unsigned integer that specifies the index into the Command String Table (TcgSttbf.**sttbf**) where the name and arguments of the macro are specified.

reserved3 (2 bytes): An unsigned integer that MUST be 0xFFFF.

reserved4 (4 bytes): This field MUST be ignored.

reserved5 (4 bytes): This field MUST be 0.

reserved6 (4 bytes): This field MUST be ignored.

reserved7 (4 bytes): This field MUST be ignored.

2.9.155 MDP

The **MDP** structure contains information that is needed to display information about an e-mail message and its author.

Copyright © 2025 Microsoft Corporation

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
dttm																															
reserved1																ibstAuthor															

dttm (4 bytes): A **DTTM** structure that specifies the date and time at which an e-mail message was created.

reserved1 (2 bytes): This field MUST be zero, and MUST be ignored.

ibstAuthor (2 bytes): A signed integer that specifies the index into the SttbfRMark structure of the author of the message.

2.9.156 MFPF

The **MFPF** structure specifies the type of picture data that is stored.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
mm																xExt															
yExt																swHMF															

mm (2 bytes): A signed integer that specifies the format of the picture data. This MUST be one of the following values.

Name	Value	Meaning
MM_SHAPE	0x0064	Shape object
MM_SHAPEFILE	0x0066	Shape file

xExt (2 bytes): This field is unused and MUST be ignored.

yExt (2 bytes): This field is unused and MUST be ignored.

swHMF (2 bytes): This field MUST be zero and MUST be ignored.

2.9.157 NilBrc

The **NilBrc** structure is a special value of a **Brc** structure that specifies that the region in question has no border. It is one possible value of the **BrcMayBeNil** structure. It is defined as its own type because the values it contains are not valid for **Brc** structures in general.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
colorref																															
nilBrc																															

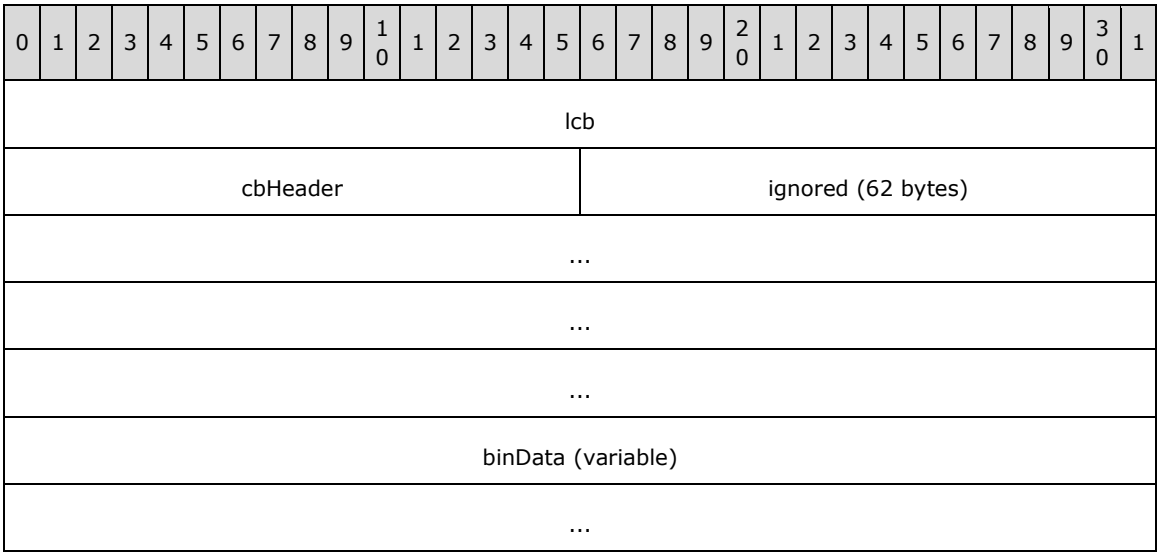
colorref (4 bytes): This field is unused and MUST be ignored.

Copyright © 2025 Microsoft Corporation

nilBrc (4 bytes): This value MUST be 0xFFFFFFFF.

2.9.158 NilPICFAndBinData

The **NilPICFAndBinData** structure that holds header information and binary data for a hyperlink, **form field**, or add-in field. The **NilPICFAndBinData** structure MUST be stored in the Data Stream.



lcb (4 bytes): A signed integer that specifies the size, in bytes, of this structure.

cbHeader (2 bytes): An unsigned integer that specifies the number of bytes from the beginning of this structure to the beginning of **binData**. This value MUST be 0x44.

ignored (62 bytes): This field MUST be 0 and MUST be ignored.

binData (variable): The interpretation of the **binData** element depends on the **field type** of the field containing the picture character and is given by the following.

Field Type	Data Type
REF	HFD
PAGEREF	HFD
FORMTEXT	FFData
FORMCHECKBOX	FFData
NOTEREF	HFD
PRIVATE	Custom binary data that is specified by the add-in that inserted this field.
ADDIN	Custom binary data that is specified by the

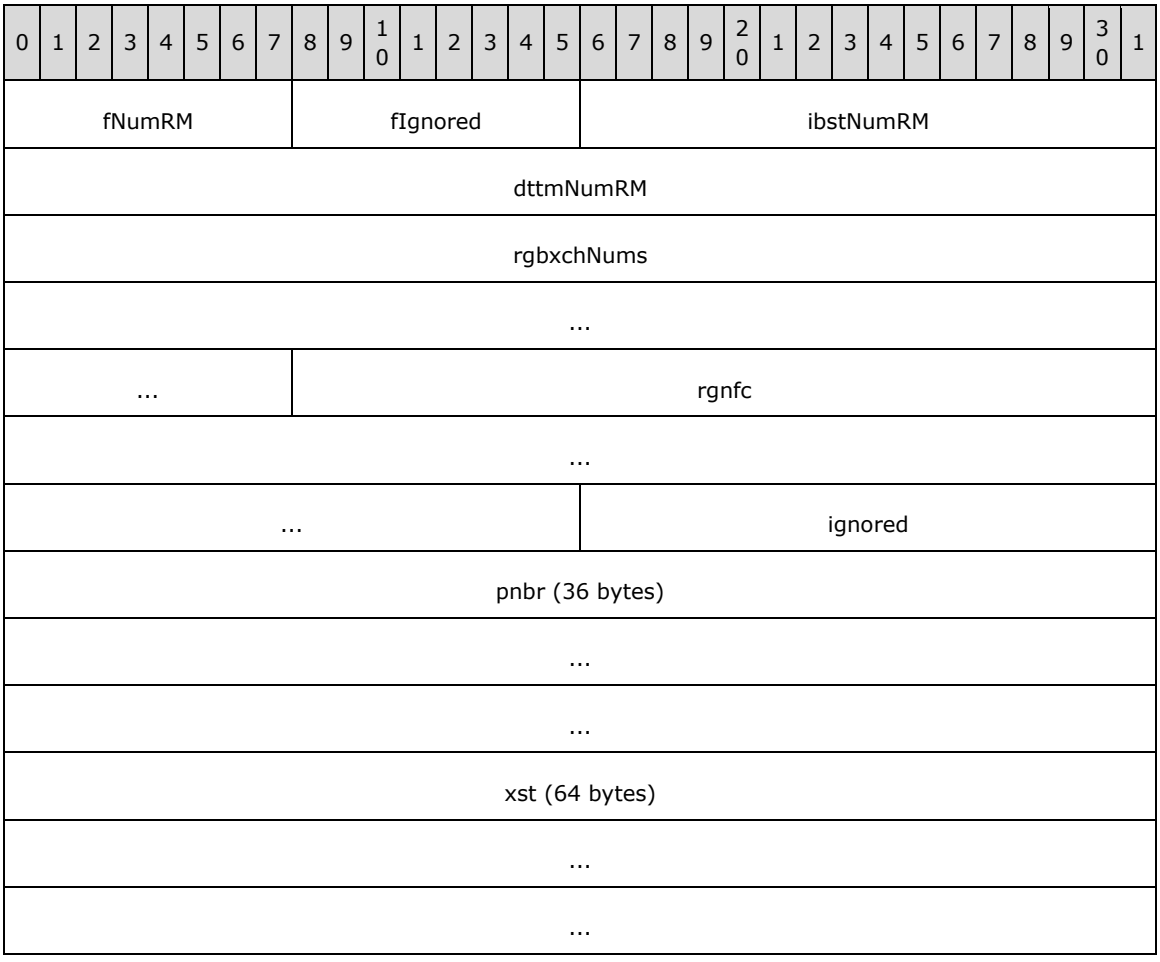
Copyright © 2025 Microsoft Corporation

Field Type	Data Type
	add-in that inserted this field.
FORMDROPDOWN	FFData
HYPERLINK	HFD

The **NilPICFAndBinData** structure is invalid if it describes a picture character that is not inside a field or is inside a field with a field type other than those specified in the preceding table. The size of **binData** is **lcb** –**cbHeader**. The data MAY<227> be invalid. If the data is invalid, it MUST be ignored.

2.9.159 NumRM

The **NumRM** structure is a numbering revision mark that specifies information about a numbering revision for a paragraph.



fNumRM (1 byte): A **Bool8** value that specifies whether the paragraph was already numbered when revision mark tracking was turned on.

fIgnored (1 byte): This field MUST be ignored.

Copyright © 2025 Microsoft Corporation

ibstNumRM (2 bytes): An integer that specifies an index for the numbering revision in the revision mark author array that is contained in the **SttbfRMark** structure.

dttmNumRM (4 bytes): A **DTTM** structure that specifies the date and time at which the numbering revision occurred.

rgbxchNums (9 bytes): An array of **BYTE** elements. Each unsigned integer in the array specifies an index into **xst**. The index is the location of a paragraph number placeholder for the numbering level that corresponds to the index. For example, **xst**[**rgbxchNums**[0]] is the location in **xst** of the first level placeholder. The text to display at the location depends on the numeric value of the level of the paragraph, as specified by **pnbr**[0] and the numbering format at **rgnfc**[0]. A value of zero specifies that the numbering level at the corresponding index is not in use.

rgnfc (9 bytes): An array of 8-bit **MSONFC** elements, as specified in [MS-OSHARED] section 2.2.1.3. Each **MSONFC** element that is contained in the array specifies the format of the numeric value for the corresponding level placeholder in **xst**. For example, for the second numbering level, the value of **rgnfc**[1] specifies the format of **pnbr**[1], which is inserted into **xst** at the level placeholder location that is specified by **rgbxchNums**[1].

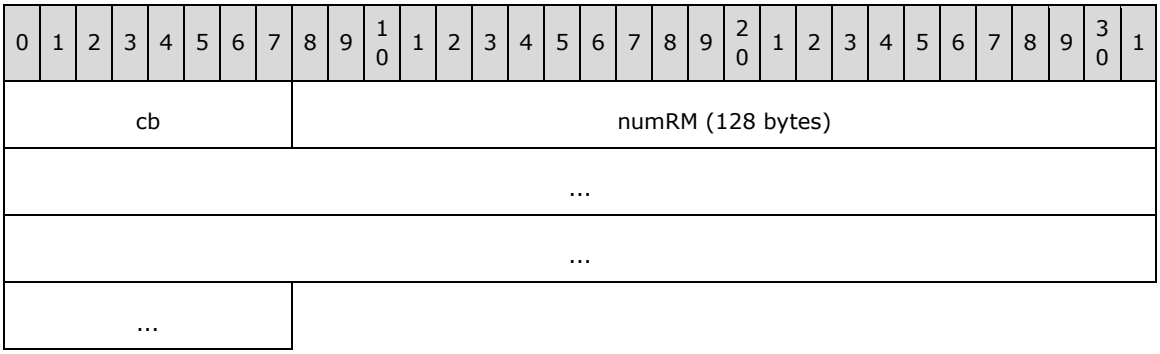
ignored (2 bytes): This field MUST be ignored.

pnbr (36 bytes): An array of **LONG** elements. Each unsigned integer in the array specifies the numeric value for the corresponding level placeholder in **xst**.

xst (64 bytes): An array of **USHORT** elements. A string that specifies the format of the numbering for the paragraph. The first position in the array is an integer that specifies the length of the format string. The format string begins at the second position and contains level placeholders for the numbering level text to be inserted. The locations of level placeholders are specified by **rgbxchNums**. To create the final display string, the text is specified by **rgnfc**, and **pnbr** is inserted at the corresponding location in **xst**.

2.9.160 NumRMOperand

The **NumRMOperand** structure is the operand for the sprmPNumRM value that contains information about a numbering revision mark.



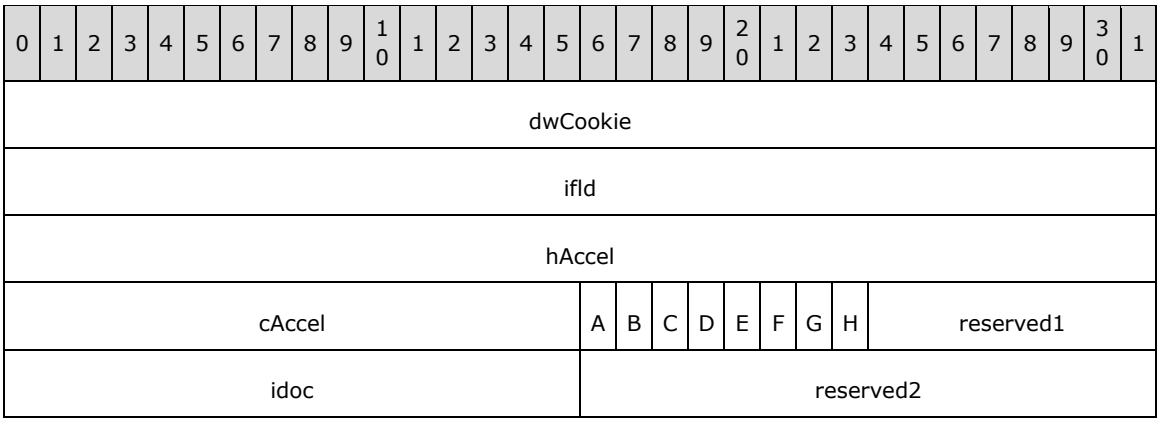
cb (1 byte): An unsigned integer that specifies the size, in bytes, of the **NumRM** structure. This value MUST be 128.

numRM (128 bytes): A **NumRM** that specifies the properties of the numbering revision mark.

2.9.161 OcxInfo

The **OcxInfo** structure specifies an **OLE control** (such as a checkbox, radio button, and so on) in the document. The data that is contained in **OcxInfo** structures SHOULD<228> be ignored.

Copyright © 2025 Microsoft Corporation



dwCookie (4 bytes): An integer value that specifies the index location of this **OcxInfo** in the **RgxOcxInfo** array. This value MUST be unique for all **OcxInfo** structures in the document.

ifld (4 bytes): An unsigned integer value that specifies an index location in the **Plcfld** structure. The value MUST be a valid **FLD** index in the correct **Plcfld** structure.

The Plcfld that is used is dependent on the value of **idoc**, as specified following.

Value	Location
1	The Main Document (FibRgFcLcb97.**fcPlcfFldMom**).
2	The Header Document (FibRgFcLcb97.**fcPlcfFldHdr**).
3	The Footnote Document (FibRgFcLcb97.**fcPlcfFldFtn**).
4	The Textbox Document (FibRgFcLcb97.**fcPlcfFldTxbx**).
6	The Endnote Document (FibRgFcLcb97.**fcPlcfFldEdn**).
7	The Comment Document (FibRgFcLcb97.**fcPlcfFldAtn**).
8	The Header Textbox Document (FibRgFcLcb97.**fcPlcfHdrtxbxTxt**).

hAccel (4 bytes): This value is undefined and MUST be ignored.

cAccel (2 bytes): An unsigned integer that specifies the number of entries in the **accelerator key** table of this control.

A - fifld (1 bit): This field MUST have a value of 1.

B - fEatsReturn (1 bit): Specifies whether this control is a sink for the ENTER key.

C - fEatsEscape (1 bit): Specifies whether this control is a sink for the ESC key.

D - fDefaultButton (1 bit): Specifies whether this control is the default button.

E - fCancelButton (1 bit): Specifies whether this control is the default CANCEL button.

F - fFailedLoad (1 bit): Specifies whether an error occurred during the loading of this control. A value of 1 specifies that this control MUST be ignored.

G - fRTL (1 bit): Specifies whether this control has special display handling for **right-to-left** languages.

H - fCorrupt (1 bit): Specifies whether this control is corrupted. A value of 1 specifies that this control MUST be ignored.

reserved1 (8 bits): Undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

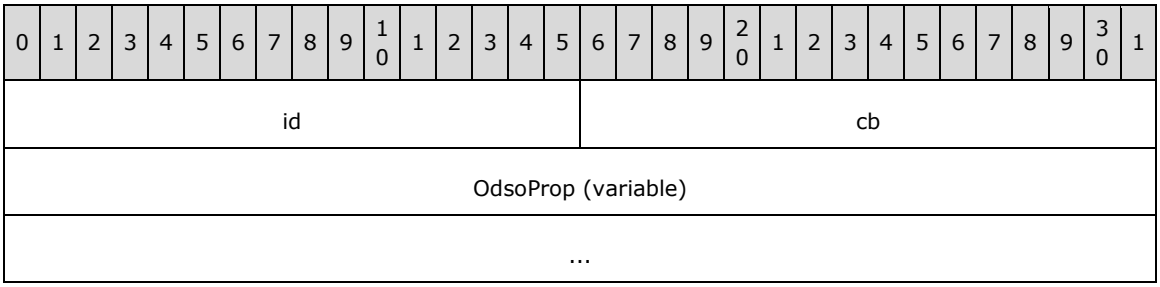
idoc (2 bytes): An integer that specifies where **ifld** can be found. The value MUST be one of the following.

Value	Location
1	The Main Document.
2	The Header Document.
3	The Footnote Document.
4	The Textbox Document.
6	The Endnote Document.
7	The Comment Document.
8	The Header Textbox Document.

reserved2 (2 bytes): Undefined and MUST be ignored.

2.9.162 ODSOPropertyBase

The **ODSOPropertyBase** structure contains an Office Data Source Object property type (**id**), size (**cb**), and value (**OdsoProp**). An Office Data Source Object is used to perform the mail merge.



id (2 bytes): An unsigned integer that specifies the type of the Office Data Source Object property (**OdsoProp**). This MUST be one of the following values: 0x0000, 0x0001, 0x0002, 0x0010, 0x0011, 0x0012, 0x0013, 0x0014, 0x0015, 0x0016, or 0x0017.

cb (2 bytes): An unsigned integer that specifies the size, in bytes, of the **OdsoProp** value or, if the size is greater than 0xFFFE, this value MUST be 0xFFFF.

OdsoProp (variable): If **cb** equals 0xFFFF, this contains an object of type **ODSOPropertyLarge**; otherwise it contains an object of type **ODSOPropertyStandard**. The data that is contained in the **OdsoProp** element is dependent on the **id** field and is defined following.

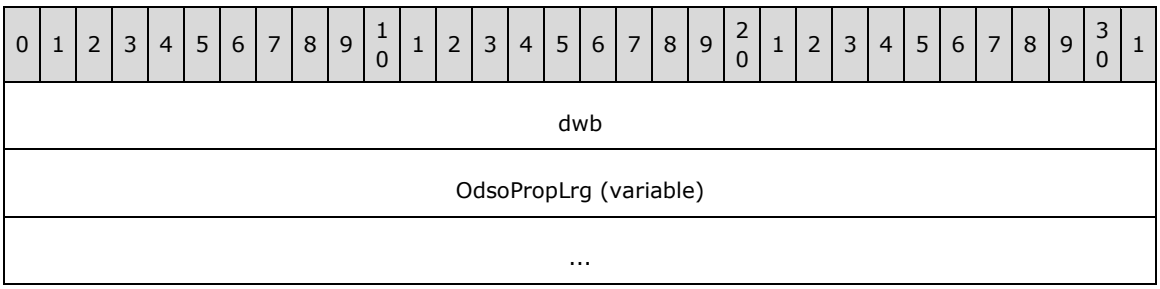
id	Meaning of data in OdsoProp
0x0000	A Unicode string, that specifies a Universal Data Link (UDL), that contains a data source **connection string**. The string is not null terminated.
0x0001	A Unicode string that specifies the set of data to be used when a data source includes multiple data sets. The string is not null terminated.
0x0002	A Unicode string that specifies the name of the file to be used as a data source. The string is not null terminated.
0x0010	A 4-byte unsigned integer that specifies the type of data source connection. The value stored in the file is not used by the application, as it is reset after loading the file, based on the connection information in **OdsoProp**s 0x0000, 0x0001, and 0x0002. This MUST be a value between 0 and 7.
0x0011	A 2-byte unsigned integer that specifies a Unicode character used as a column delimiter for a text data source.
0x0012	A 4-byte unsigned integer that specifies whether the first row is a header row of column names. A value of 0x00000001 specifies that the first row contains column names; a value of

Copyright © 2025 Microsoft Corporation

id	Meaning of data in OdsoProp
	0x00000000 specifies that it does not.
0x0013	The property contains an array of **FilterDataItem** structures that are used to filter the list of recipients.
0x0014	The property contains up to three **SortColumnAndDirection** items that are used to sort the list of recipients.
0x0015	The property contains a **RecipientInfo** structure.
0x0016	The property contains a **FieldMapInfo** structure that specifies which database columns are mapped to each of 30 standard mail merge address fields. The **FieldMapDataItem** structures MUST appear in the following order and all items MUST be present: 1. Unique Identifier 2. Courtesy Title 3. First Name 4. Middle Name 5. Last Name 6. Suffix 7. Nickname 8. Job Title 9. Company 10. Address 1 11. Address 2 12. City 13. State 14. Postal Code 15. Country or Region 16. Business Phone 17. Business Fax 18. Home Phone 19. Home Fax 20. E-mail Address 21. Web Page 22. Spouse Courtesy Title 23. Spouse First Name 24. Spouse Middle Name 25. Spouse Last Name 26. Spouse Nickname 27. Phonetic Guide for First Name 28. Phonetic Guide for Last Name 29. Address 3 30. Department
0x0017	A 2-byte unsigned integer that specifies which step of the mail merge wizard the application last displayed. This MUST be a value between 1 and 6.

2.9.163 ODSOPropertyLarge

The **ODSOPropertyLarge** structure contains an ODSO property that is at least 0xFFFF bytes in size. See specifications of the ODSO property types under **ODSOPropertyBase.id**.



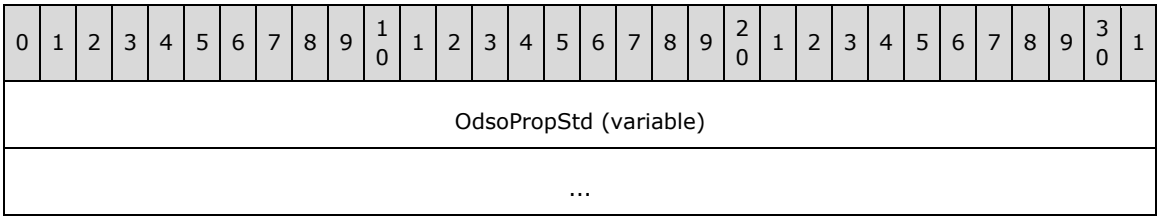
Copyright © 2025 Microsoft Corporation

dwb (4 bytes): An unsigned integer that specifies the size, in bytes, of the **OdsoPropLrg** element.

OdsoPropLrg (variable): Contains the data for this property.

2.9.164 ODSOPropertyStandard

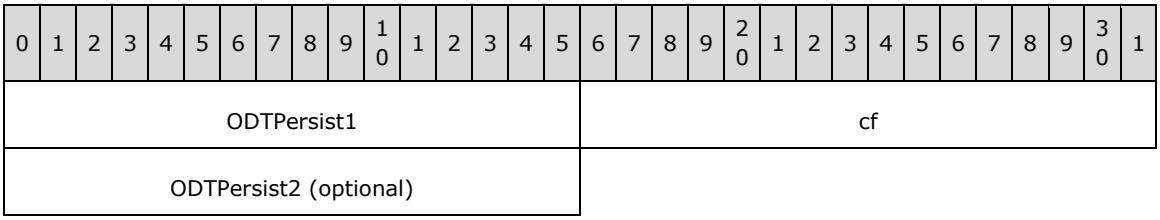
The **ODSOPropertyStandard** structure contains an ODSO property that is less than 0xFFFF bytes in size. See descriptions of the ODSO property types under **ODSOPropertyBase.id**.



OdsoPropStd (variable): Contains the data for this property.

2.9.165 ODT

The **ODT** structure stores information about an **OLE object**. Each OLE object in a Word Binary file is stored in a storage within the ObjectPool storage. Each of these storages has an ObjInfo stream which contains an **ODT** structure.



ODTPersist1 (2 bytes): An **ODTPersist1** structure that specifies information about the OLE object.

cf (2 bytes): An unsigned integer that specifies the format this OLE object uses to transmit data to the host application. Valid values and their meanings are:

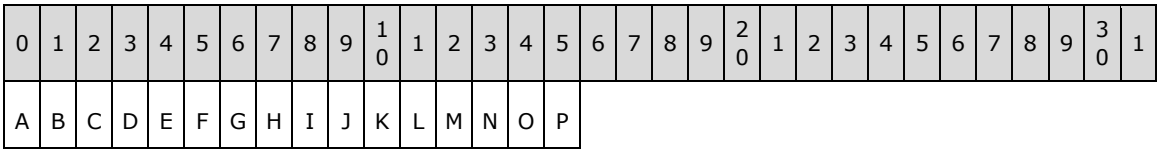
Value	Meaning
0x0001	Rich Text Format
0x0002	Text format
0x0003	Metafile or Enhanced Metafile, depending on **ODTPersist2.fStoredAsEMF**
0x0004	Bitmap
0x0005	Device Independent Bitmap
0x000A	HTML format
0x0014	**Unicode** text format

Copyright © 2025 Microsoft Corporation

ODTPersist2 (2 bytes): An **ODTPersist2** structure that specifies additional information about the OLE object. This member does not exist if the ObjInfo stream containing this **ODT** structure is not large enough to accommodate it.

2.9.166 ODTPersist1

The **ODTPersist1** structure is a collection of bits that specify information about an **OLE object**.



A - reserved1 (1 bit): Undefined and MUST be ignored.

B - fDefHandler (1 bit): If this bit is 1, then the application MUST assume that this OLE object's **class identifier (CLSID)** is {00020907-0000-0000-C000-000000000046}.

C - reserved2 (1 bit): Undefined and MUST be ignored.

D - reserved3 (1 bit): Undefined and MUST be ignored.

E - fLink (1 bit): A bit that specifies whether this OLE object is a link.

F - reserved4 (1 bit): Undefined and MUST be ignored.

G - fIcon (1 bit): A bit that specifies whether this OLE object is being represented by an icon.

H - fIsOle1 (1 bit): A bit that specifies whether this OLE object is only compatible with OLE 1. If this bit is zero, then the object is compatible with OLE 2.

I - fManual (1 bit): A bit that specifies whether the user has requested that this OLE object only be updated in response to a user action. If **fManual** is zero, then the user has requested that this OLE object update automatically. If **fLink** is zero, then **fManual** is undefined and MUST be ignored.

J - fRecomposeOnResize (1 bit): A bit that specifies whether this OLE object has requested to be notified when it is resized by its container.

K - reserved5 (1 bit): MUST be zero and MUST be ignored.

L - reserved6 (1 bit): MUST be zero and MUST be ignored.

M - fOCX (1 bit): A bit that specifies whether this object is an **OLE control**.

N - fStream (1 bit): If **fOCX** is zero, then this bit MUST be zero. If **fOCX** is 1, then **fStream** is a bit that specifies whether this OLE control stores its data in a single stream instead of a storage. If **fStream** is 1, then the data for the OLE control is in a stream called "\003OCXDATA" where \003 is the character with value 0x0003, not the string literal "\003".

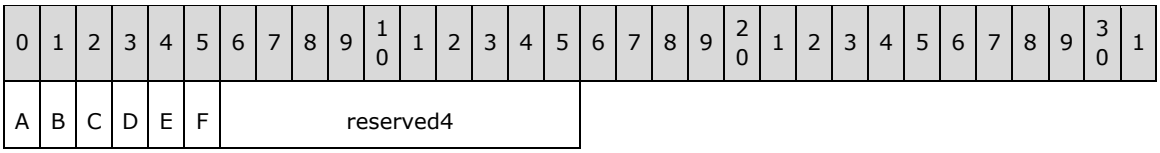
O - reserved7 (1 bit): Undefined and MUST be ignored.

P - fViewObject (1 bit): A bit that specifies whether this OLE object supports the IViewObject interface as described in [MSDOCS-IViewObject].

Copyright © 2025 Microsoft Corporation

2.9.167 ODTPersist2

The **ODTPersist2** structure is a collection of bits that specify information about an **OLE object**.



A - fEMF (1 bit): A bit that specifies that the presentation of this OLE object in the document is in the Enhanced Metafile format. This is different from **fStoredAsEMF** in the case of an object being represented as an icon. For icons, the icon can be an Enhanced Metafile even if the OLE object does not support the Enhanced Metafile format.

B - reserved1 (1 bit): MUST be zero and MUST be ignored.

C - fQueriedEMF (1 bit): A bit that specifies whether the application that saved this Word Binary file had queried this OLE object to determine whether it supported the Enhanced Metafile format.

D - fStoredAsEMF (1 bit): A bit that specifies that this OLE object supports the Enhanced Metafile format.

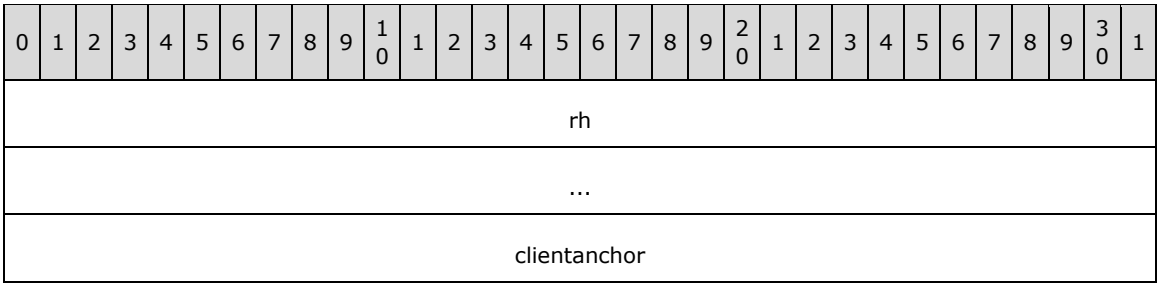
E - reserved2 (1 bit): Undefined and MUST be ignored.

F - reserved3 (1 bit): Undefined and MUST be ignored.

reserved4 (10 bits): Undefined and MUST be ignored.

2.9.168 OfficeArtClientAnchor

The **OfficeArtClientAnchor** structure is used by **OfficeArtSpContainer**, as specified in [MS-ODRAW] section 2.2.14, that specifies the location of a drawing.



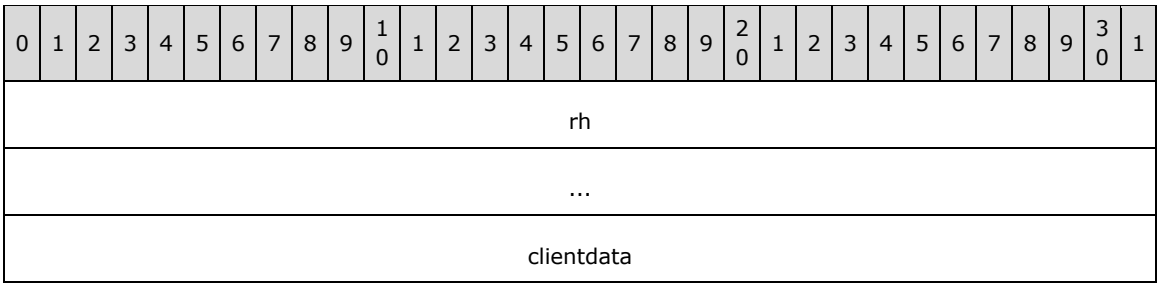
rh (8 bytes): An **OfficeArtRecordHeader**, as specified in [MS-ODRAW] section 2.2.1, that specifies information about the structure.

clientanchor (4 bytes): A 4-byte integer that specifies a valid index into the **aCP** field of the corresponding PlcfSpa. The **CP** at this index is the location of the drawing. A value of -1 specifies an invalid value.

2.9.169 OfficeArtClientData

The **OfficeArtClientData** structure is used by the **OfficeArtSpContainer**, as specified in [MS-ODRAW] section 2.2.14.

Copyright © 2025 Microsoft Corporation

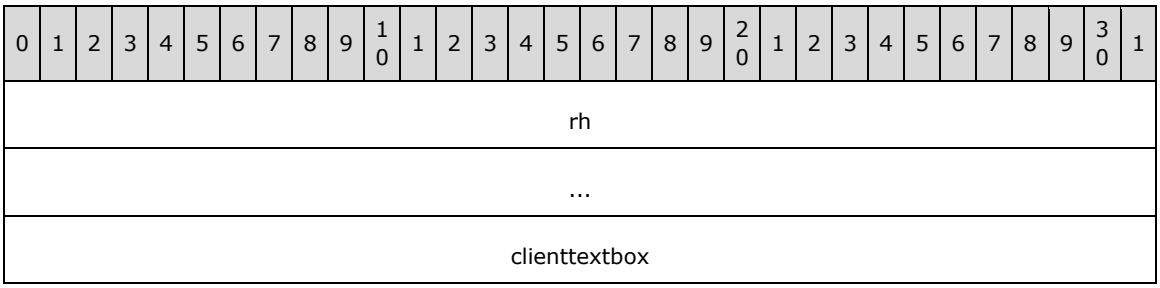


rh (8 bytes): An **OfficeArtRecordHeader**, as specified in [MS-ODRAW] section 2.2.1, that specifies information about the structure.

clientdata (4 bytes): An integer that SHOULD<229> be ignored.

2.9.170 OfficeArtClientTextbox

The **OfficeArtClientTextbox** structure used by **OfficeArtSpContainer**, as specified in [MS-ODRAW] section 2.2.14, that specifies the text identifier for a shape.

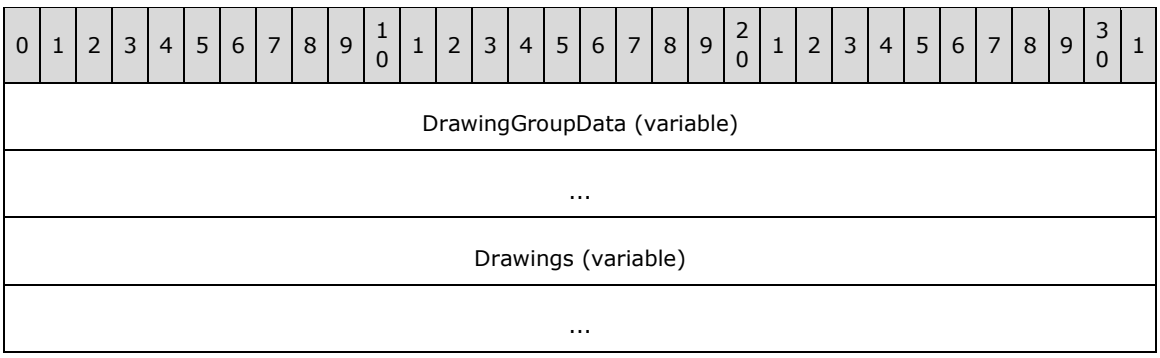


rh (8 bytes): An **OfficeArtRecordHeader**, as specified in [MS-ODRAW] section 2.2.1, that specifies information about the structure.

clienttextbox (4 bytes): A 4-byte unsigned integer that specifies the text identifier of the shape, as specified in [MS-ODRAW] section 2.3.21.1. This value specifies the location of the text for the textbox in the following manner: Dividing the high 2 bytes by 0x10000 specifies a 1-based index into **PlcfTxbxTxt** of the **FTXBXS** structure where the text for this textbox is located. The low 2 bytes specify the zero-based index in the textbox chain that the textbox occupies.

2.9.171 OfficeArtContent

The **OfficeArtContent** structure specifies information about a drawing in the document. The delay stream that is referenced in [MS-ODRAW] is the WordDocument stream.



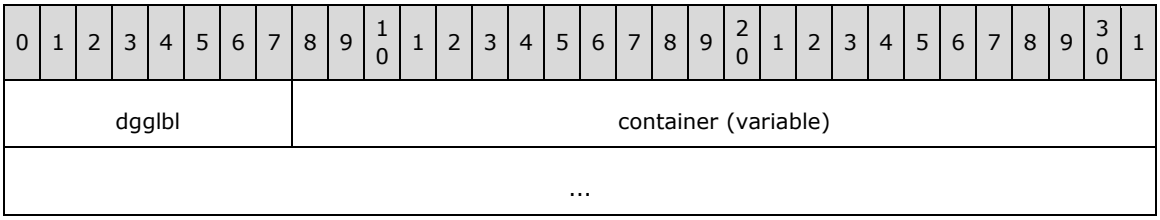
Copyright © 2025 Microsoft Corporation

DrawingGroupData (variable): An **OfficeArtDggContainer** element, as specified in [MS-ODRAW] section 2.2.12, that contains the drawing group information for the document.

Drawings (variable): An array of **OfficeArtWordDrawing** elements that specifies information about the drawings in the document. Drawings for the Main Document are located at index 0 of this array. Drawings for the Header Document are located at index 1 of this array.

2.9.172 OfficeArtWordDrawing

The **OfficeArtWordDrawing** structure specifies information about the drawings in the document. The delay stream that is referenced in [MS-ODRAW] is the WordDocument stream.

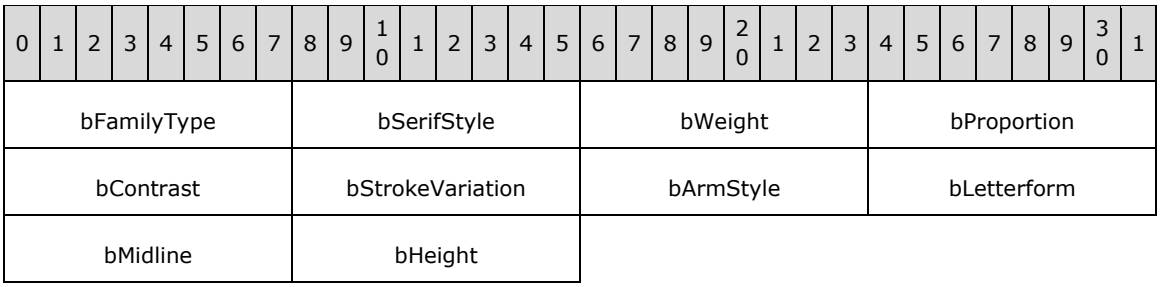


dgglbl (1 byte): An unsigned integer that specifies where **container** is located. A value of 0x00 specifies that **container** is in the Main Document. A value of 0x01 specifies that **container** is in the Header Document.

container (variable): An **OfficeArtDgContainer**, as specified in [MS-ODRAW] section 2.2.13, that specifies the information about the drawings.

2.9.173 PANOSE

The **PANOSE** structure defines the PANOSE font classification values for a **TrueType font**, as specified in [PANOSE]. These characteristics are used to associate the font with other fonts of similar appearance but different names.



bFamilyType (1 byte):

For Latin fonts, this field MUST have one of the following values.

Value	Meaning
PAN_ANY (0)	Any.
PAN_NO_FIT (1)	No fit.
PAN_FAMILY_TEXT_DISPLAY (2)	Text and display.
PAN_FAMILY_SCRIPT (3)	Script.
PAN_FAMILY_DECORATIVE (4)	Decorative.
PAN_FAMILY_PICTORIAL (5)	Pictorial.

Copyright © 2025 Microsoft Corporation

bSerifStyle (1 byte): Specifies the serif style. For Latin fonts, this field MUST have one of the following values.

Value	Meaning
PAN_ANY (0)	Any.
PAN_NO_FIT (1)	No fit.
PAN_SERIF_COVE (2)	Cove.
PAN_SERIF_OBTUSE_COVE (3)	Obtuse cove.
PAN_SERIF_SQUARE_COVE (4)	Square cove.
PAN_SERIF_OBTUSE_SQUARE_COVE (5)	Obtuse square cove.
PAN_SERIF_SQUARE (6)	Square.
PAN_SERIF_THIN (7)	Thin.
PAN_SERIF_BONE (8)	Bone.
PAN_SERIF_EXAGGERATED (9)	Exaggerated.
PAN_SERIF_TRIANGLE (10)	Triangle.
PAN_SERIF_NORMAL_SANS (11)	Normal sans serif.
PAN_SERIF_OBTUSE_SANS (12)	Obtuse sans serif.
PAN_SERIF_PERP_SANS (13)	Perp sans serif.
PAN_SERIF_FLARED (14)	Flared.
PAN_SERIF_ROUNDED (15)	Rounded.

bWeight (1 byte): For Latin fonts, this field MUST have one of the following values.

Value	Meaning
PAN_ANY (0)	Any.
PAN_NO_FIT (1)	No fit.
PAN_WEIGHT_VERY_LIGHT (2)	Very light.
PAN_WEIGHT_LIGHT (3)	Light.
PAN_WEIGHT_THIN (4)	Thin.
PAN_WEIGHT_BOOK (5)	Book.
PAN_WEIGHT_MEDIUM (6)	Medium.
PAN_WEIGHT_DEMI (7)	Demibold.
PAN_WEIGHT_BOLD (8)	Bold.
PAN_WEIGHT_HEAVY (9)	Heavy.
PAN_WEIGHT_BLACK (10)	Black.

Copyright © 2025 Microsoft Corporation

Value	Meaning
PAN_WEIGHT_NORD (11)	Nord.

bProportion (1 byte): For Latin fonts, this field MUST have one of the following values.

Value	Meaning
PAN_ANY (0)	Any.
PAN_NO_FIT (1)	No fit.
PAN_PROP_OLD_STYLE (2)	Old Style.
PAN_PROP_MODERN (3)	Modern.
PAN_PROP_EVEN_WIDTH (4)	Even Width.
PAN_PROP_EXPANDED (5)	Expanded.
PAN_PROP_CONDENSED (6)	Condensed.
PAN_PROP_VERY_EXPANDED (7)	Very Expanded.
PAN_PROP_VERY_CONDENSED (8)	Very Condensed.
PAN_PROP_MONOSPACED (9)	Monospaced.

bContrast (1 byte): For Latin fonts, this field MUST have one of the following values.

Value	Meaning
PAN_ANY (0)	Any.
PAN_NO_FIT (1)	No fit.
PAN_CONTRAST_NONE (2)	None.
PAN_CONTRAST_VERY_LOW (3)	Very low.
PAN_CONTRAST_LOW (4)	Low.
PAN_CONTRAST_MEDIUM_LOW (5)	Medium low.
PAN_CONTRAST_MEDIUM (6)	Medium.
PAN_CONTRAST_MEDIUM_HIGH (7)	Medium high.
PAN_CONTRAST_HIGH (8)	High.
PAN_CONTRAST_VERY_HIGH (9)	Very high.

bStrokeVariation (1 byte): For Latin fonts, this field MUST have one of the following values.

Value	Meaning
PAN_ANY (0)	Any.

Copyright © 2025 Microsoft Corporation

Value	Meaning
PAN_NO_FIT (1)	No fit.
2	No Variation.
3	Gradual/diagonal.
4	Gradual/transitional.
5	Gradual/vertical.
6	Gradual/horizontal.
7	Rapid/vertical.
8	Rapid/horizontal.
9	Instant/Vertical.
10	Instant/Horizontal.

bArmStyle (1 byte): For Latin fonts, this field MUST have one of the following values.

Value	Meaning
PAN_ANY (0)	Any.
PAN_NO_FIT (1)	No fit.
PAN_STRAIGHT_ARMS_HORZ (2)	Straight arms/horizontal.
PAN_STRAIGHT_ARMS_WEDGE (3)	Straight arms/wedge.
PAN_STRAIGHT_ARMS_VERT (4)	Straight arms/vertical.
PAN_STRAIGHT_ARMS_SINGLE_SERIF (5)	Straight arms/single-serif.
PAN_STRAIGHT_ARMS_DOUBLE_SERIF (6)	Straight arms/double-serif.
PAN_BENT_ARMS_HORZ (7)	Non-straight arms/horizontal.
PAN_BENT_ARMS_WEDGE (8)	Non-straight arms/wedge.
PAN_BENT_ARMS_VERT (9)	Non-straight arms/vertical.
PAN_BENT_ARMS_SINGLE_SERIF (10)	Non-straight arms/single-serif.
PAN_BENT_ARMS_DOUBLE_SERIF (11)	Non-straight arms/double-serif.

bLetterform (1 byte): For Latin fonts, this field MUST have one of the following values.

Value	Meaning
PAN_ANY (0)	Any.
PAN_NO_FIT (1)	No fit.
PAN_LETT_NORMAL_CONTACT (2)	Normal/Contact.
PAN_LETT_NORMAL_WEIGHTED (3)	Normal/Weighted.

Copyright © 2025 Microsoft Corporation

Value	Meaning
PAN_LETT_NORMAL_BOXED (4)	Normal/Boxed.
PAN_LETT_NORMAL_FLATTENED (5)	Normal/Flattened.
PAN_LETT_NORMAL_ROUNDED (6)	Normal/Rounded.
PAN_LETT_NORMAL_OFF_CENTER (7)	Normal/Off-Center.
PAN_LETT_NORMAL_SQUARE (8)	Normal/Square.
PAN_LETT_OBLIQUE_CONTACT (9)	Oblique/Contact.
PAN_LETT_OBLIQUE_WEIGHTED (10)	Oblique/Weighted.
PAN_LETT_OBLIQUE_BOXED (11)	Oblique/Boxed.
PAN_LETT_OBLIQUE_FLATTENED (12)	Oblique/Flattened.
PAN_LETT_OBLIQUE_ROUNDED (13)	Oblique/Rounded.
PAN_LETT_OBLIQUE_OFF_CENTER (14)	Oblique/Off-Center.
PAN_LETT_OBLIQUE_SQUARE (15)	Oblique/Square.

bMidline (1 byte): For Latin fonts, this field MUST have one of the following values.

Value	Meaning
PAN_ANY (0)	Any.
PAN_NO_FIT (1)	No fit.
PAN_MIDLINE_STANDARD_TRIMMED (2)	Standard/Trimmed.
PAN_MIDLINE_STANDARD_POINTED (3)	Standard/Pointed.
PAN_MIDLINE_STANDARD_SERIFED (4)	Standard/Serifed.
PAN_MIDLINE_HIGH_TRIMMED (5)	High/Trimmed.
PAN_MIDLINE_HIGH_POINTED (6)	High/Pointed.
PAN_MIDLINE_HIGH_SERIFED (7)	High/Serifed.
PAN_MIDLINE_CONSTANT_TRIMMED (8)	Constant/Trimmed.
PAN_MIDLINE_CONSTANT_POINTED (9)	Constant/Pointed.
PAN_MIDLINE_CONSTANT_SERIFED (10)	Constant/Serifed.
PAN_MIDLINE_LOW_TRIMMED (11)	Low/Trimmed.
PAN_MIDLINE_LOW_POINTED (12)	Low/Pointed.
PAN_MIDLINE_LOW_SERIFED (13)	Low/Serifed.

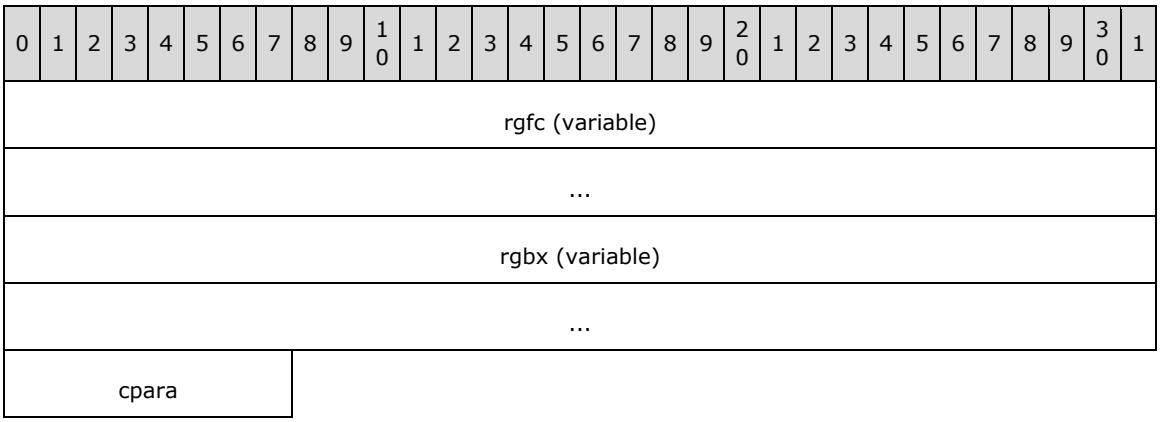
bHeight (1 byte): For Latin fonts, this field MUST have one of the following values.

Copyright © 2025 Microsoft Corporation

Value	Meaning
PAN_ANY (0)	Any.
PAN_NO_FIT (1)	No fit.
PAN_XHEIGHT_CONSTANT_SMALL (2)	Constant/small.
PAN_XHEIGHT_CONSTANT_STD (3)	Constant/standard.
PAN_XHEIGHT_CONSTANT_LARGE (4)	Constant/large.
PAN_XHEIGHT_DUCKING_SMALL (5)	Ducking/small.
PAN_XHEIGHT_DUCKING_STD (6)	Ducking/standard.
PAN_XHEIGHT_DUCKING_LARGE (7)	Ducking/large.

2.9.174 PapxFkp

The **PapxFkp** structure maps paragraphs, table rows, and table cells to their properties. A **PapxFkp** structure is 512 bytes in size, with **cpara** in the last byte. The elements of **rgbx** specify the locations of **PapxInFkp** structures that start at offsets between the end of **rgbx** and **cpara** within this **PapxFkp** structure.



rgfc (variable): An array of 4-byte unsigned integers. Each element of this array specifies an offset in the WordDocument Stream where a paragraph of text begins, or where an **end of row mark** exists. This array MUST be sorted in ascending order and MUST NOT contain duplicates. Each paragraph begins immediately after the end of the previous paragraph. The count of elements that this array contains is **cpara** incremented by 1. The last element does not specify the beginning of a paragraph; instead it specifies the end of the last paragraph.

rgbx (variable): An array of **BxPap**, followed by **PapxInFkp** structures. The elements of this array, which has **cpara** elements and parallels **rgfc**, each specify the offset of one of the **PapxInFkp** structures in this **PapxFkp** structure.

Each **PapxInFkp** specifies the paragraph properties for the paragraph at the corresponding offset in **rgfc** or the table properties for the table row whose end of row mark is located at the corresponding offset in **rgfc**.

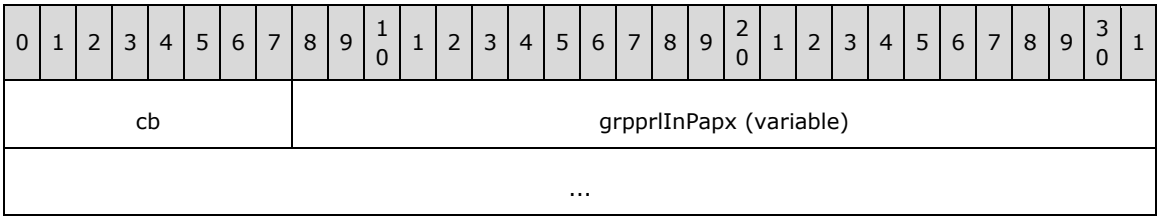
cpara (1 byte): An unsigned integer that specifies the total number of paragraphs, table rows, or table cells for which this **PapxFkp** structure specifies formatting. This field occupies the last byte

Copyright © 2025 Microsoft Corporation

of the **PapxFkp** structure The value of this field MUST be at least 0x01, and MUST NOT exceed 0x1D because that would cause **rgfc** and **rgb** to expand and PapxFkp to exceed 512 bytes.

2.9.175 PapxInFkp

The **PapxInFkp** structure specifies a set of text properties.



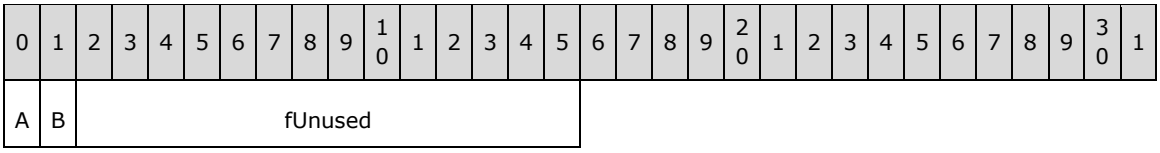
cb (1 byte): An unsigned integer that specifies the size of the **grpprlInPapx**. If this value is not 0, the **grpprlInPapx** is 2×cb-1 bytes long. If this value is 0, the size is specified by the first byte of **grpprlInPapx**.

grpprlInPapx (variable): If **cb** is 0, the first byte of **grpprlInPapx** (call it cb') is an unsigned integer that specifies the size of the rest of **grpprlInPapx**. **cb'** MUST be at least 1. After **cb'**, there are 2×**cb'** more bytes in **grpprlInPapx**. The bytes after **cb'** form a GrpPrlAndIstd.

If **cb** is nonzero, **grpprlInPapx** is GrpPrlAndIstd.

2.9.176 PbiGrfOperand

The **PbiGrfOperand** structure specifies the properties of a picture bullet.



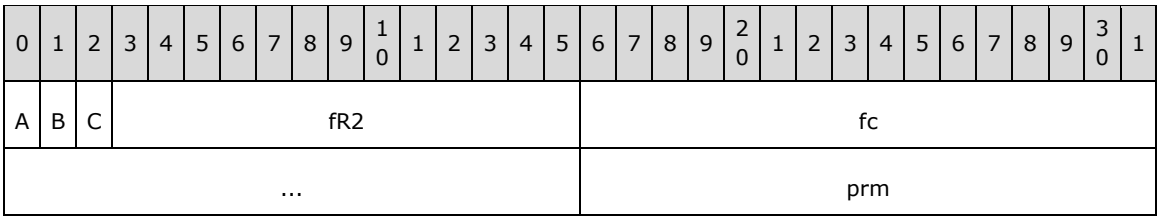
A - fPicBullet (1 bit): Specifies whether the bullet is a picture bullet.

B - fNoAutoSize (1 bit): Specifies whether the size of the picture changes automatically to match the size of the text that follows the bullet.

fUnused (14 bits): This field is undefined and MUST be ignored.

2.9.177 Pcd

The **Pcd** structure specifies the location of text in the WordDocument Stream and additional properties for this text. A **Pcd** structure is an element of a **PlcPcd** structure.



A - fNoParaLast (1 bit): If this bit is 1, the text MUST NOT contain a paragraph mark.

B - fR1 (1 bit): This field is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

C - fDirty (1 bit): This field MUST be 0.

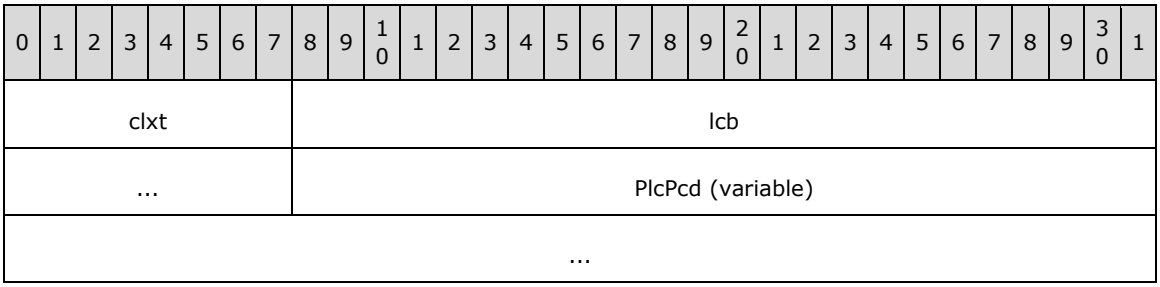
fR2 (13 bits): This field is undefined and MUST be ignored.

fc (4 bytes): An **FcCompressed** structure that specifies the location of the text in the WordDocument Stream.

prm (2 bytes): A **Prm** structure that specifies additional properties for this text. These properties are used as part of the algorithms in sections 2.4.6.1 (Direct Paragraph Formatting) and 2.4.6.2 (Direct Character Formatting).

2.9.178 Pcdt

The **Pcdt** structure contains a **PlcPcd** structure and specifies its size.



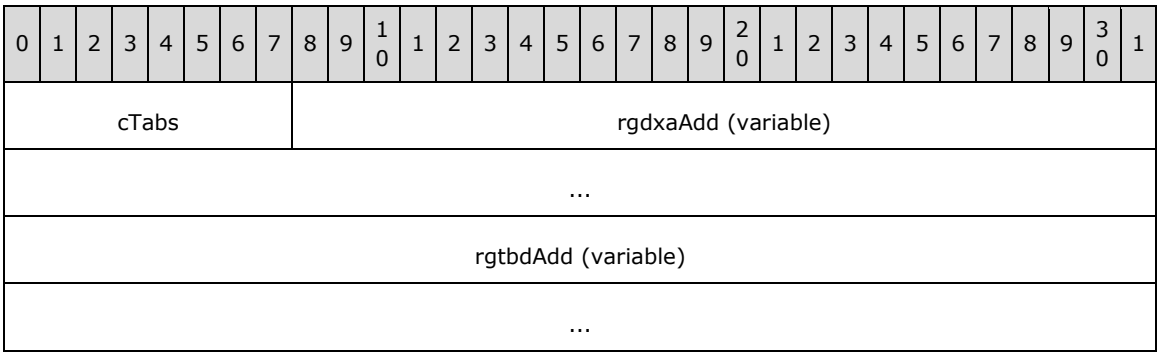
clxt (1 byte): This value MUST be 0x02.

lcb (4 bytes): An unsigned integer that specifies the size, in bytes, of the **PlcPcd** structure.

PlcPcd (variable): A **PlcPcd** structure. As with all **Plc** elements, the size that is specified by **lcb** MUST result in a whole number of **Pcd** structures in this **PlcPcd** structure.

2.9.179 PChgTabsAdd

The **PChgTabsAdd** structure specifies the locations and properties of custom tab stops.



cTabs (1 byte): An unsigned integer that specifies the number of records in **rgdxaAdd** and **rgtbdAdd**. This value MUST be less than or equal to 64.

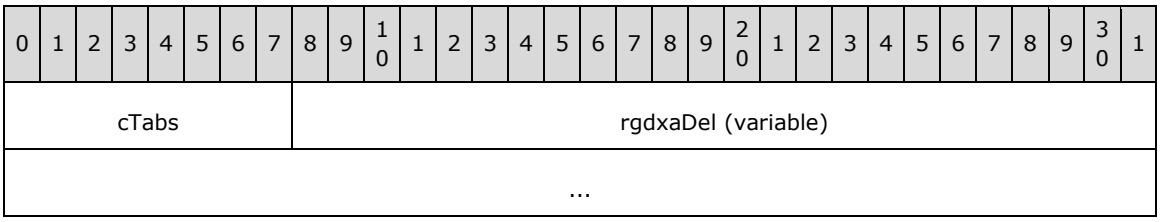
rgdxaAdd (variable): An array of **XAS** integer values. The number of records is specified by **cTabs**. The values in this array MUST be in ascending order. Each **XAS** value specifies a location at which to add a custom tab stop.

Copyright © 2025 Microsoft Corporation

rgtbdAdd (variable): An array of **TBD** structures. The number of records is specified by **cTabs**. Each **TBD** specifies the alignment and leader attributes of the custom tab stop at the location that is specified at the corresponding index in **rgdxaAdd**.

2.9.180 PChgTabsDel

The **PChgTabsDel** structure specifies the locations at which custom tab stops are ignored.

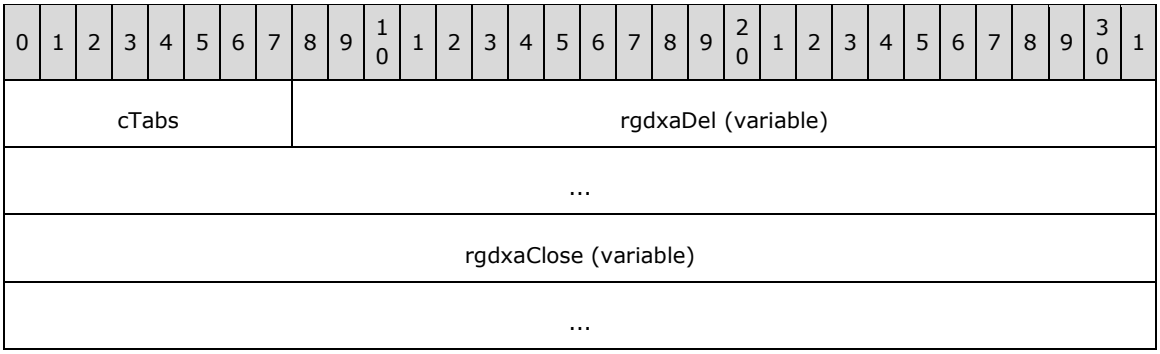


cTabs (1 byte): An unsigned integer that specifies the number of records in **rgdxaDel**. This value MUST be less than or equal to 64.

rgdxaDel (variable): An array of **XAS**. The number of records is specified by **cTabs**. The elements contained in the array MUST be in ascending order. Each XAS specifies a location at which to ignore any custom tab stop within 25 **twips**.

2.9.181 PChgTabsDelClose

The **PChgTabsDelClose** structure specifies the locations at which custom tab stops are ignored.



cTabs (1 byte): An unsigned integer that specifies the number of records in **rgdxaDel** and **rgdxaClose**. This value MUST be greater than or equal to 0, and less than or equal to 64.

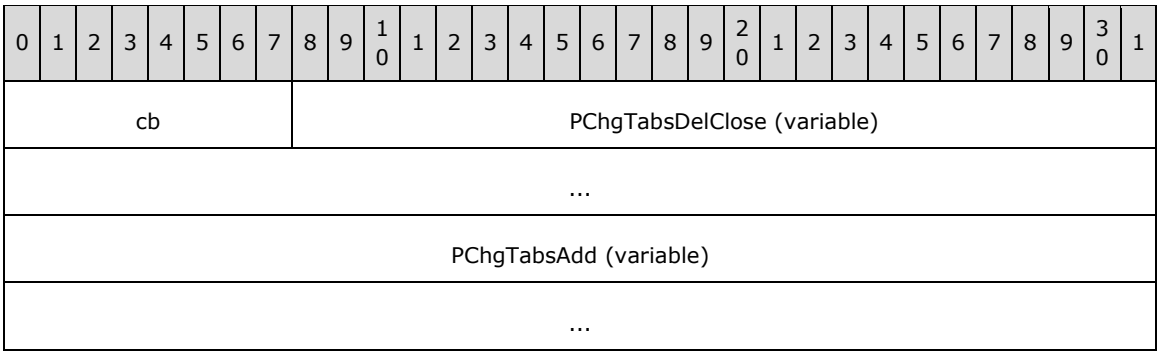
rgdxaDel (variable): An array of 16-bit integers. The number of records is specified by **cTabs**. The integers contained in the array MUST be in ascending order. Each integer SHOULD<230> be greater than or equal to -31680. Each integer MUST be less than or equal to 31680. Each integer specifies a location at which to ignore any custom tab stop within 25 **twips**.

rgdxaClose (variable): An array of **XAS_plusOne**. The number of records is specified by **cTabs**. Each entry in **rgdxaClose** specifies a distance, in twips in both directions, from the corresponding entry in **rgdxaDel**. All tab stops inside this range are deleted. Any entry in **rgdxaClose** that has a value of less than 0x0019 is treated as though the value was 0x0019.

Copyright © 2025 Microsoft Corporation

2.9.182 PChgTabsOperand

The **PChgTabsOperand** structure is used by **sprmPChgTabs** to specify a list of custom tab stops to add and another list of custom tab stops to ignore.



cb (1 byte): An unsigned integer that specifies the size of the operand. This value MUST be greater than or equal to 2 and less than or equal to 255. A value that is less than 255 specifies the size of the operand in bytes, not including **cb**. A value of 255 specifies that this instance of sprmPChgTabs MAY<231> be ignored and that the size of the remainder of this operand, in bytes, is calculated by using the following formula:

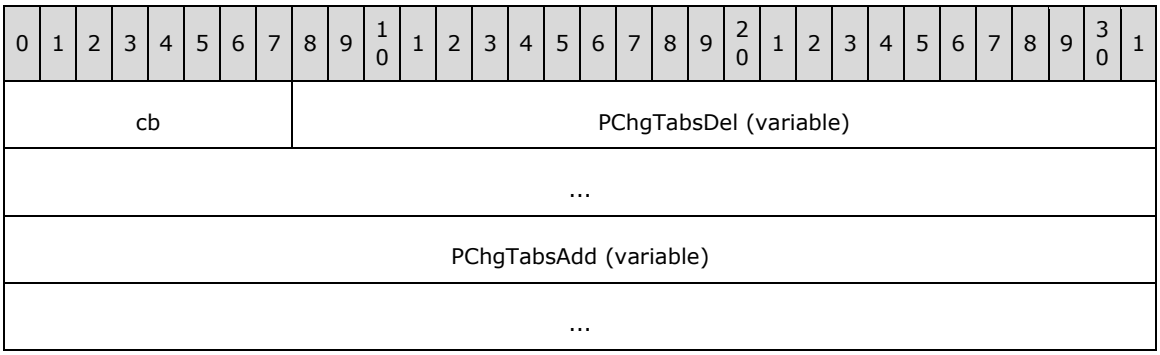
$$cb = 4 \times \mathbf{PChgTabsDelClose.cTabs} + 3 \times \mathbf{PChgTabsAdd.cTabs}$$

PChgTabsDelClose (variable): A PchgTabsDelClose that specifies the locations of custom tab stops to ignore.

PChgTabsAdd (variable): A PChgTabsAdd that specifies the locations and properties of custom tab stops to add.

2.9.183 PChgTabsPapxOperand

The **PChgTabsPapxOperand** structure is used by sprmPChgTabsPapx to specify custom tab stops to be added or ignored.



cb (1 byte): An unsigned integer that specifies the size of the operand in bytes, not including **cb**. This value MUST be greater than or equal to 2 and less than or equal to 255.

PChgTabsDel (variable): A **PChgTabsDel** structure that specifies the locations at which custom tab stops are ignored.

PChgTabsAdd (variable): A **PChgTabsAdd** structure that specifies the locations and properties of custom tab stops to be added.

Copyright © 2025 Microsoft Corporation

2.9.184 PgbApplyTo

The **PgbApplyTo** enumeration is used to specify the pages to which a **page border** applies.

Name	Value	Meaning
pgbAllPages	0x0	The page border applies to all pages in the **section**.
pgbFirstPage	0x1	The page border applies only to the first page of the section.
pgbAllButFirst	0x2	The page border applies to all but the first page of the section.

2.9.185 PgbOffsetFrom

The **PgbOffsetFrom** enumeration is used to specify the location from which the offset of a **page border** is measured.

Name	Value	Meaning
pgbFromText	0x0	The offset of the page border is measured from the text.
pgbFromEdge	0x1	The offset of the page border is measured from the edge of the page.

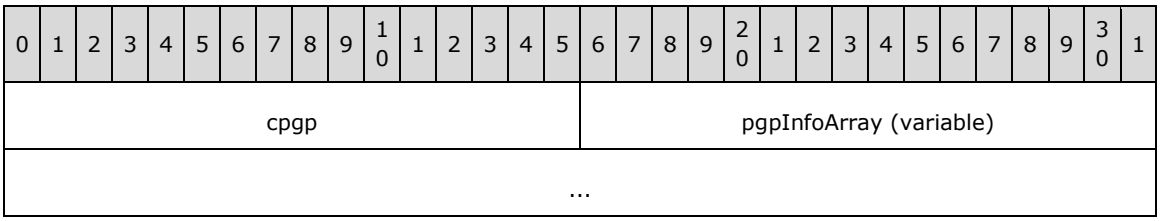
2.9.186 PgbPageDepth

The **PgbPageDepth** enumeration is used to specify the "depth" of a **page border** in relation to other page elements.

Name	Value	Meaning
pgbAtFront	0x0	The page border is positioned in front of the text and other content.
pgbAtBack	0x1	The page border is positioned behind the text and other content.

2.9.187 PGPArray

The **PGPArray** structure is a collection of the **PGPInfo** entries in the document.



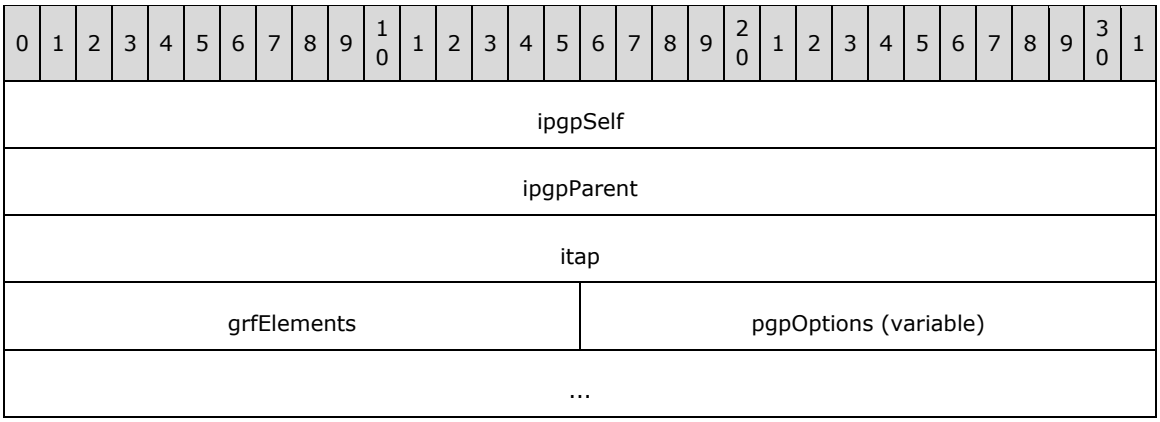
cpgp (2 bytes): The count of **PGPInfo** entries to read.

pgpInfoArray (variable): An array of **PGPInfo** structures. This array contains **cpgp** elements.

2.9.188 PGPInfo

The **PGPInfo** structure describes the border and margin properties that can be applied to a contiguous range of paragraphs.

Copyright © 2025 Microsoft Corporation



ipgpSelf (4 bytes): A unique 4-byte value that is used to identify this entry. This value MUST NOT be 0.

ipgpParent (4 bytes): This is the identifier of the immediate parent **PGPInfo** structure. A value of 0 indicates that there is no parent and that, therefore, this is an outermost PGPInfo.

itap (4 bytes): The table depth to which this **PGPInfo** structure is applied. **PGPInfo** structures can be applied to paragraphs that are within a table cell.

grfElements (2 bytes): A bit field that describes how to read in the variable length **pgpOptions**. The meanings of the bits are as follows.

Bit value	Meaning
0x0001	PGPOptions.**dxaLeft** is present.
0x0002	PGPOptions.**dxaRight** is present.
0x0004	PGPOptions.**dyaBefore** is present.
0x0008	PGPOptions.**dyaAfter** is present.
0x0010	PGPOptions.**brcLeft** is present.
0x0020	PGPOptions.**brcRight** is present.
0x0040	PGPOptions.**brcTop** is present.
0x0080	PGPOptions.**brcBottom** is present.
0x0100	PGPOptions.**type** is present.

pgpOptions (variable): A PGPOptions structure that describes all the relevant paragraph properties that are different than the defaults.

2.9.189 PGPOptions

The **PGPOptions** structure is a variable-sized container of the PGPInfo properties that are to be changed from their default values. The members that are present in the file are indicated by PGPInfo.**grfElements**.

Copyright © 2025 Microsoft Corporation

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
cbOption (optional)																dxaLeft (optional)															
...																dxaRight (optional)															
...																dyaBefore (optional)															
...																dyaAfter (optional)															
...																brcLeft (optional)															
...																															
...																brcRight (optional)															
...																															
...																brcTop (optional)															
...																															
...																brcBottom (optional)															
...																															
...																type (optional)															

cbOption (2 bytes): If **PGPInfo.grfElements** is nonzero, this is the byte size of the remaining **PGPOptions** data in the file.

dxaLeft (4 bytes): If **PGPInfo.grfElements** & 0x0001 is nonzero, this is the size of the left margin to apply, measured in 1/20 point increments. Otherwise, the default value of 0 is used.

dxaRight (4 bytes): If **PGPInfo.grfElements** & 0x0002 is nonzero, this is the size of the right margin to apply, measured in 1/20 point increments. Otherwise, the default value of 0 is used.

dyaBefore (4 bytes): If **PGPInfo.grfElements** & 0x0004 is nonzero, this is the size of the top margin to apply, measured in 1/20 point increments. Otherwise, the default of 0 is used.

dyaAfter (4 bytes): If **PGPInfo.grfElements** & 0x0008 is nonzero, this is the size of the bottom margin to apply, measured in 1/20 point increments. Otherwise, the default value of 0 is used.

brcLeft (8 bytes): If **PGPInfo.grfElements** & 0x0010 is nonzero, this is the **Brc** that describes the left border of the **PGPInfo**. Otherwise, there is no left border.

brcRight (8 bytes): If **PGPInfo.grfElements** & 0x0020 is nonzero, this is the **Brc** that describes the right border of the **PGPInfo**. Otherwise, there is no right border.

brcTop (8 bytes): If **PGPInfo.grfElements** & 0x0040 is nonzero, this is the **Brc** that describes the top border of the **PGPInfo**. Otherwise, there is no top border.

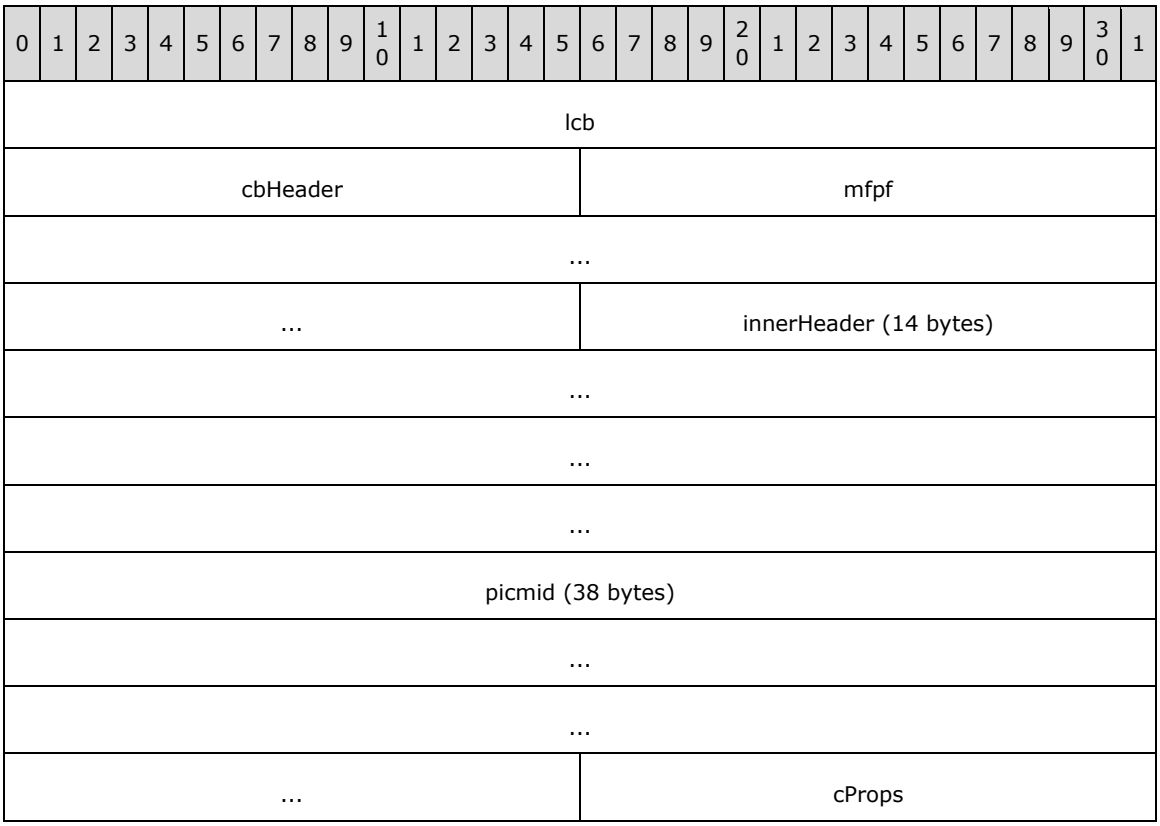
brcBottom (8 bytes): If **PGPInfo**.**grfElements** & 0x0080 is nonzero, this is the **Brc** that describes the bottom border of the **PGPInfo**. Otherwise, there is no bottom border.

Copyright © 2025 Microsoft Corporation

type (2 bytes): If **PGPInfo.grfElements** & 0x0100 is nonzero, this value MUST be 0, 1 or 2. If this value is 1, this **PGPInfo** is represented as a <BLOCKQUOTE> element when saved as HTML. If this value is 2, this PGPInfo is represented as a <BODY> element, provided that it is applied at a point where the <BODY> element is legal in HTML. If this value is not present or is 0, it is assumed that this **PGPInfo** represents a <DIV> element.

2.9.190 PICF

The **PICF** structure specifies the type of a picture, as well as the size of the picture and information about its border.



lcb (4 bytes): A signed integer that specifies the size, in bytes, of this **PICF** structure and the subsequent data.

cbHeader (2 bytes): An unsigned integer that specifies the size, in bytes, of this **PICF** structure. This value MUST be 0x44.

mfpf (8 bytes): An **MFPF** structure that specifies the storage format of the picture.

innerHeader (14 bytes): A **PICF_Shape** structure that specifies additional header information.

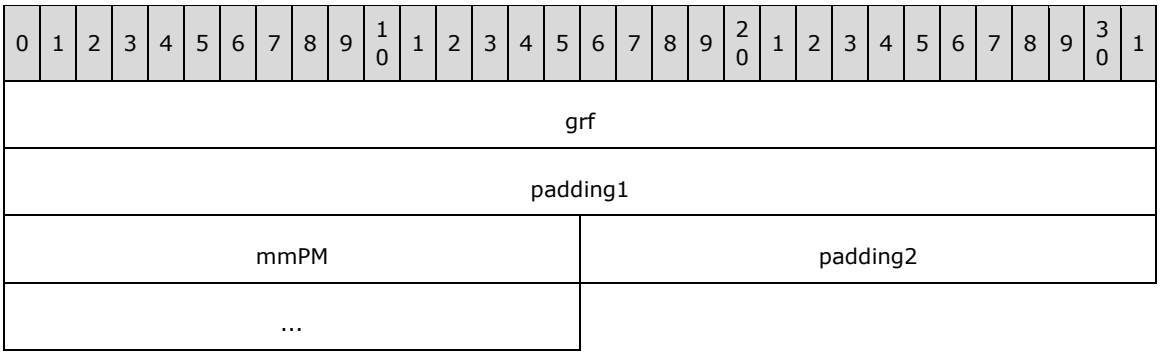
picmid (38 bytes): A **PICMID** structure that specifies the size and border information of the picture.

cProps (2 bytes): This value MUST be 0 and MUST be ignored.

2.9.191 PICF_Shape

The **PICF_Shape** structure specifies additional header information for pictures of type **MM_SHAPE** or **MM_SHAPEFILE**.

Copyright © 2025 Microsoft Corporation



grf (4 bytes): This field MUST be ignored.

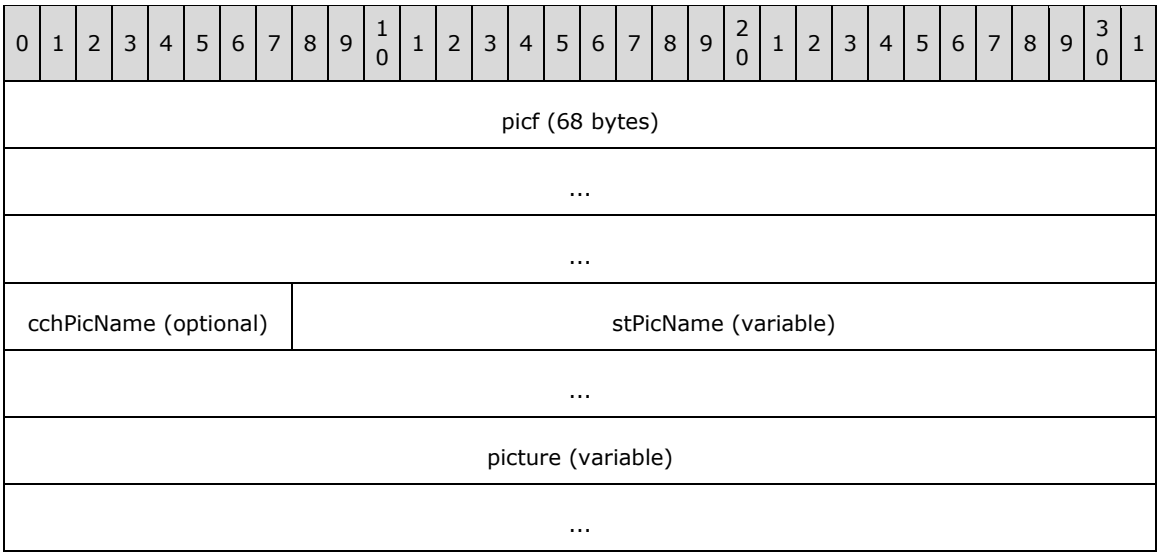
padding1 (4 bytes): This value MUST be 0 and MUST be ignored.

mmPM (2 bytes): This field MUST be ignored.

padding2 (4 bytes): This field MUST be zero and MUST be ignored.

2.9.192 PICFAndOfficeArtData

The **PICFAndOfficeArtData** structure specifies header information and binary data for a picture. These structures MUST be stored in the Data Stream at locations that are specified by the sprmCPicLocation value. The range of text that is described by the **Chpx** structure which contains the sprmCPicLocation value MUST contain the picture character (U+0001).



picf (68 bytes): A **PICF** structure that specifies the type of the picture, as well as the picture size and border information.

cchPicName (1 byte): An optional unsigned integer that specifies the size of **stPicName**. This value MUST exist if and only if **picf.mfpf.mm** is MM_SHAPEFILE (0x0066).

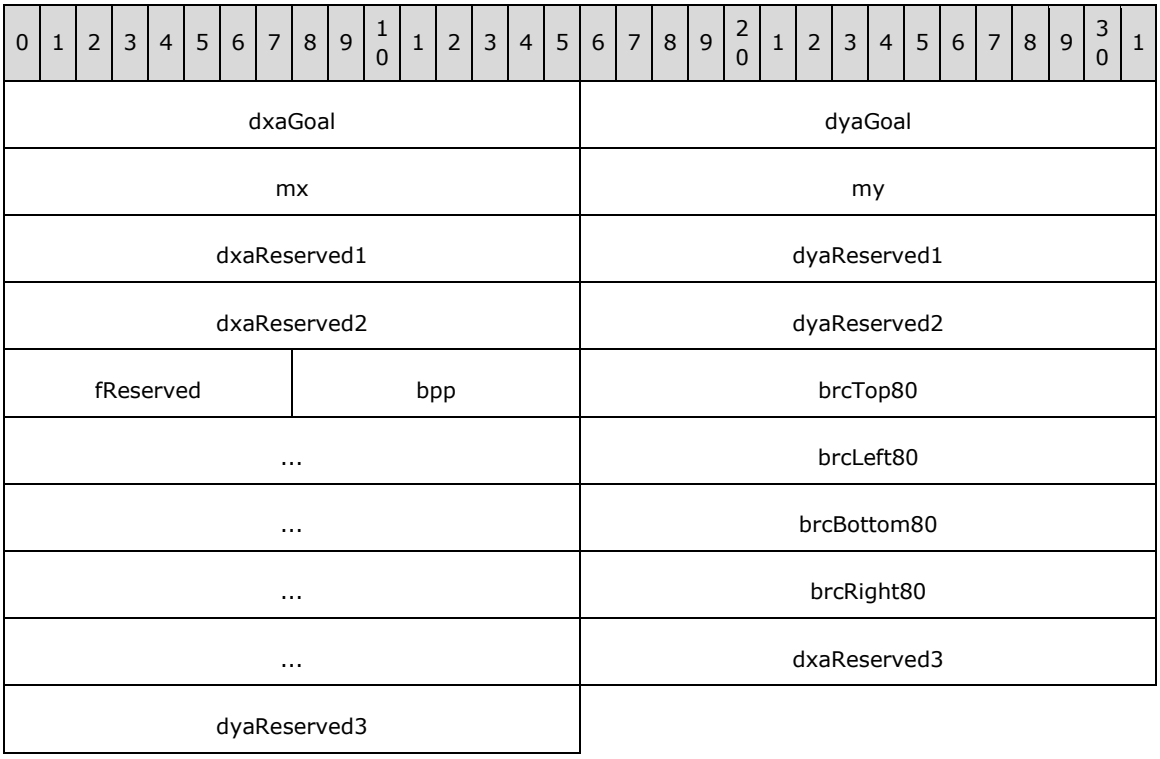
stPicName (variable): An optional string of ANSI characters that specifies the full path and file name of the picture. This value MUST exist if and only if **picf.mfpf.mm** is **MM_SHAPEFILE** (0x0066). The length of the string is equal to **cchPicName** and is not null-terminated.

Copyright © 2025 Microsoft Corporation

picture (variable): An **OfficeArtInlineSpContainer**, as specified in [MS-ODRAW] section 2.2.15, that specifies the image.

2.9.193 PICMID

The **PICMID** structure specifies the size and border information for a picture.



dxaGoal (2 bytes): A signed integer that specifies the initial width of the picture, in **twips**, before cropping or scaling occurs. This value MUST be greater than zero.

dyaGoal (2 bytes): A signed integer that specifies the initial height of the picture, in twips, before cropping or scaling occurs. This value MUST be greater than zero.

mx (2 bytes): An unsigned integer that specifies the ratio, measured in tenths of a percent, between the final display width and the initial picture width that is specified by **dxaGoal**. If the picture is not cropped, **mx** values that are greater than 1000 cause the picture to stretch in width, while values that are less than 1000 cause the picture to shrink in width.

If the picture is horizontally cropped and the **mx** value is not adjusted accordingly, the picture is scaled. To counteract the new dimensions of a cropped image and avoid scaling, set **mx** to the value of ((**dxaGoal** – (left-crop + right-crop)) / **dxaGoal**.

The final display width MUST be at least 15 twips and MUST NOT exceed 31680 twips (22 inches) after cropping and scaling.

my (2 bytes): An unsigned integer that specifies the ratio, measured in tenths of a percent, between the final display height and the initial picture height that was specified by **dyaGoal**. If the picture is not cropped, **my** values that are greater than 1000 cause the picture to stretch in height, while values of less than 1000 cause the picture to shrink.

Copyright © 2025 Microsoft Corporation

If the picture is vertically cropped and the **my** value is not adjusted accordingly, the picture is scaled. To counteract the new dimensions of a cropped image and avoid scaling, set the **my** value to the value of ((**dyaGoal** – (top-crop + bottom-crop)) / **dyaGoal**.

The final display height MUST be at least 15 twips and MUST NOT exceed 31680 twips (22 inches) after cropping and scaling.

dxaReserved1 (2 bytes): This value MUST be 0 and MUST be ignored.

dyaReserved1 (2 bytes): This value MUST be 0 and MUST be ignored.

dxaReserved2 (2 bytes): This value MUST be 0 and MUST be ignored.

dyaReserved2 (2 bytes): This value MUST be 0 and MUST be ignored.

fReserved (8 bits): This value MUST be 0 and MUST be ignored.

bpp (8 bits): This field is unused and MUST be ignored.

brcTop80 (4 bytes): A **Brc80** structure that specifies what border to render above the picture.

brcLeft80 (4 bytes): A **Brc80** structure that specifies what border to render to the left of the picture.

brcBottom80 (4 bytes): A **Brc80** structure that specifies what border to render below the picture.

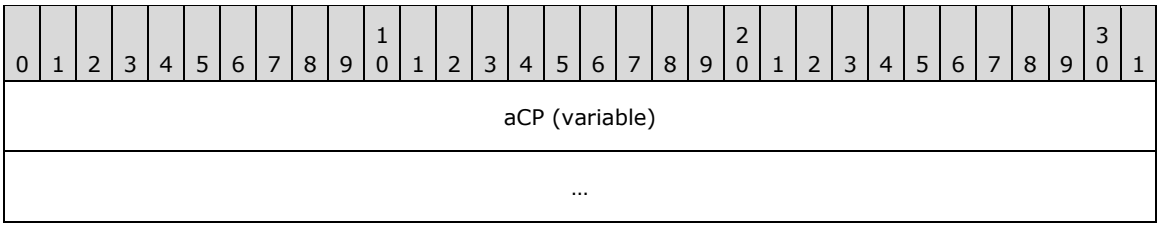
brcRight80 (4 bytes): A **Brc80** structure that specifies what border to render to the right of the picture.

dxaReserved3 (2 bytes): This value MUST be 0 and MUST be ignored.

dyaReserved3 (2 bytes): This value MUST be 0 and MUST be ignored.

2.9.194 PlcfGlsy

The **PlcfGlsy** structure is a PLC that contains only CPs and no additional data. The count of CPs in a **PlcfGlsy** structure MUST be equal to a number that represents the count of strings in the corresponding **SttbfGlsy** incremented by 2. A **PlcfGlsy** MUST NOT contain duplicate CPs.

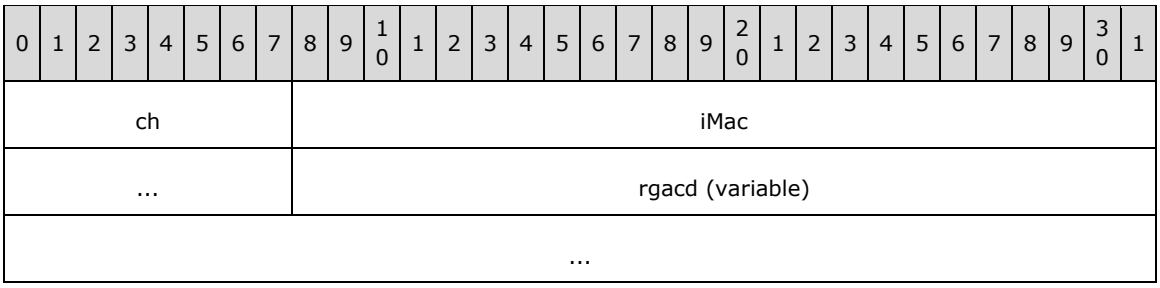


aCP (variable): An array of CP. Each CP is an offset into the main document. Each CP MUST be greater than or equal to zero, and MUST be less than **FibRgLw97.ccpText**. Each CP specifies the beginning of a range of text that constitutes the contents of an **AutoText** item. Each AutoText item corresponds to its respective entry in the parallel AutoText item string table **SttbfGlsy**. The range of text ends immediately before the next CP. The last CP MUST be ignored, and the second to last CP does not begin a new text range; it only terminates the text range that started with the previous CP.

2.9.195 PlfAcd

The **PlfAcd** structure specifies the **allocated commands** in a sequence of command-related customizations. For more information, see **Tcg255**.

Copyright © 2025 Microsoft Corporation



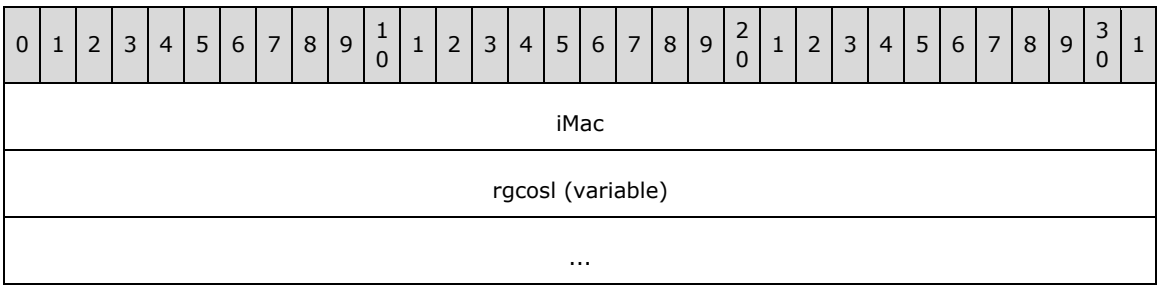
ch (1 byte): An unsigned integer value that identifies this structure as **PlfAcd**. This value MUST be 2.

iMac (4 bytes): A signed integer value that specifies the number of allocated command descriptor structures, as specified in **Acd**, in **rgacd**. This value MUST be greater than or equal to 0.

rgacd (variable): An array of **Acd** structures. The number of structures that are contained in this array is specified by **iMac**.

2.9.196 PlfCosl

The **PlfCosl** structure is a list of **COSL** that is specified as an array and its associated count of elements. Each element specifies the option set to use for a grammar checker that implements the **NLCheck** interface. An option set specifies a value for each grammar option.

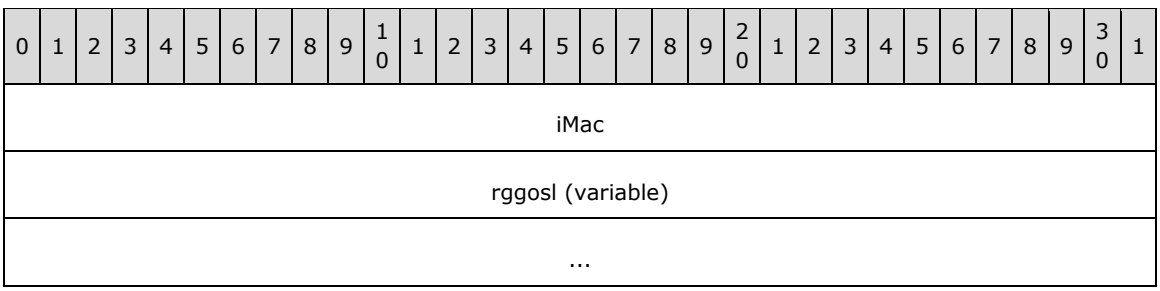


iMac (4 bytes): A signed integer that specifies the number of entries in **rgcosl**. This value MUST be greater than or equal to zero.

rgcosl (variable): An array of COSL.

2.9.197 PlfGosl

The **PlfGosl** structure is a list of **GOSL** structures that are specified as an array, and its associated count of elements. Each element specifies the option set to use for a grammar checker that implements the **CGAPI** interface. An option set specifies a value for each grammar option.



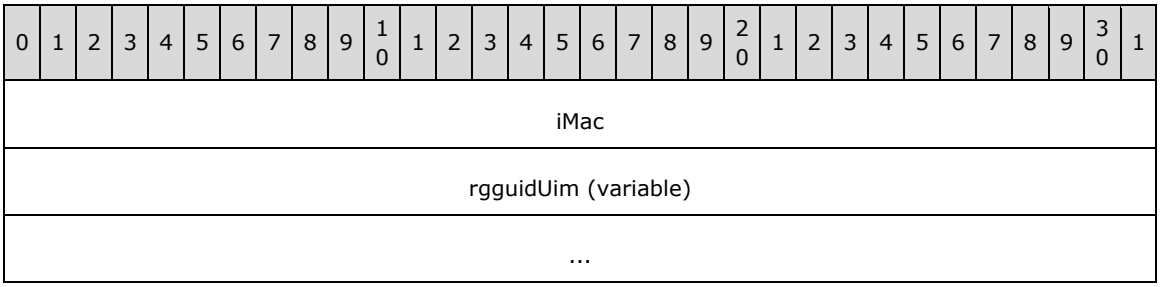
iMac (4 bytes): A signed integer that represents the count of entries in **rgcosl**. This value MUST be greater than or equal to zero.

Copyright © 2025 Microsoft Corporation

rggosl (variable): An array of **GOSL** structures.

2.9.198 PlfguidUim

The **PlfguidUim** structure specifies an array of GUIDs which are referenced by the **UIM** structures of **PlcfUim**.

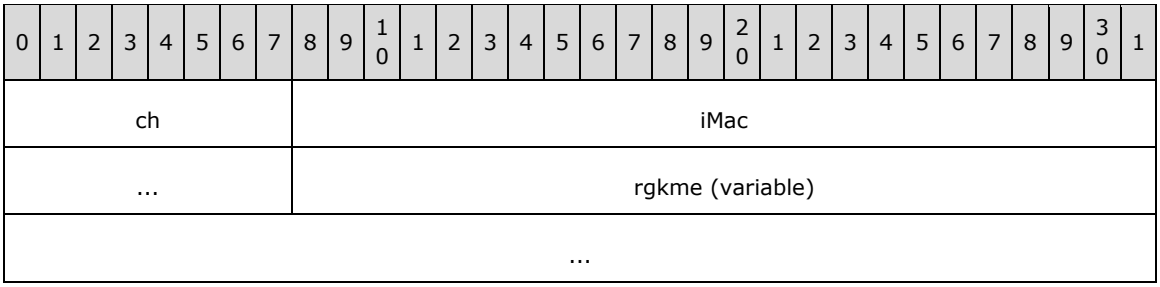


iMac (4 bytes): An unsigned integer that specifies the number of GUIDs in **rgguidUim**.

rgguidUim (variable): An array of 16-byte GUIDs that specify the service category or CLSID of the service providing data referenced by a UIM structure.

2.9.199 PlfKme

The **PlfKme** structure specifies keyboard mappings. This structure is used in the sequence of structures that specify command-related customizations. For more information, see the Tcg255 structure.



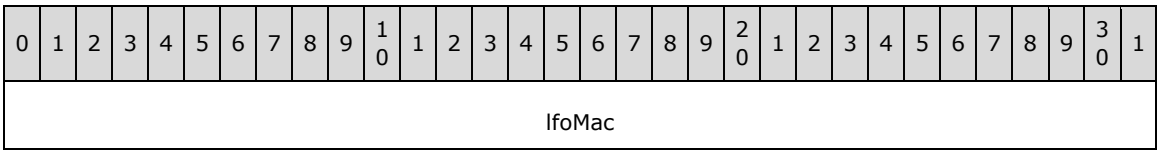
ch (1 byte): An unsigned integer that identifies this structure as **PlfKme**. This value MUST be either 3 or 4. A value of 3 indicates regular keyboard key map entries. A value of 4 indicates invalid keyboard key map entries. For more information, see the Tcg255.**rgtcgData** field.

iMac (4 bytes): A signed integer that specifies the number of keyboard key map entries, as specified in **Kme**, in **rgkme**. This value MUST be greater than or equal to 0.

rgkme (variable): An array of **Kme** structures. The number of structures is specified by **iMac**.

2.9.200 PlfLfo

The **PlfLfo** structure contains the list format override data for the document.



Copyright © 2025 Microsoft Corporation

rgLfo (variable)
...
rgLfoData (variable)
...

lfoMac (4 bytes): An unsigned integer that specifies the count of elements in both the **rgLfo** and **rgLfoData** arrays.

rgLfo (variable): An array of **LFO** structures. The number of elements in this array is specified by **lfoMac**.

rgLfoData (variable): An array of **LFOData** that is parallel to **rgLfo**. The number of elements that are contained in this array is specified by **lfoMac**.

2.9.201 PlfLst

The **PlfLst** structure contains the list formatting information for the document.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
cLst																rgLstf (variable)															
...																															

cLst (2 bytes): A signed integer that specifies the count of LSTF structures that are contained in **rgLstf**.

rgLstf (variable): An array of LSTF. The number of elements that are contained in this array is specified by **cLst**.

2.9.202 PlfMcd

The **PlfMcd** structure specifies macro commands. This structure is used in the sequence of structures that specify command-related customizations. For more information, see Tcg255.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
ch								iMac																							
...								rgmcd (variable)																							
...																															

ch (1 byte): An unsigned integer that identifies this structure as **PlfMcd**. This value MUST be 1.

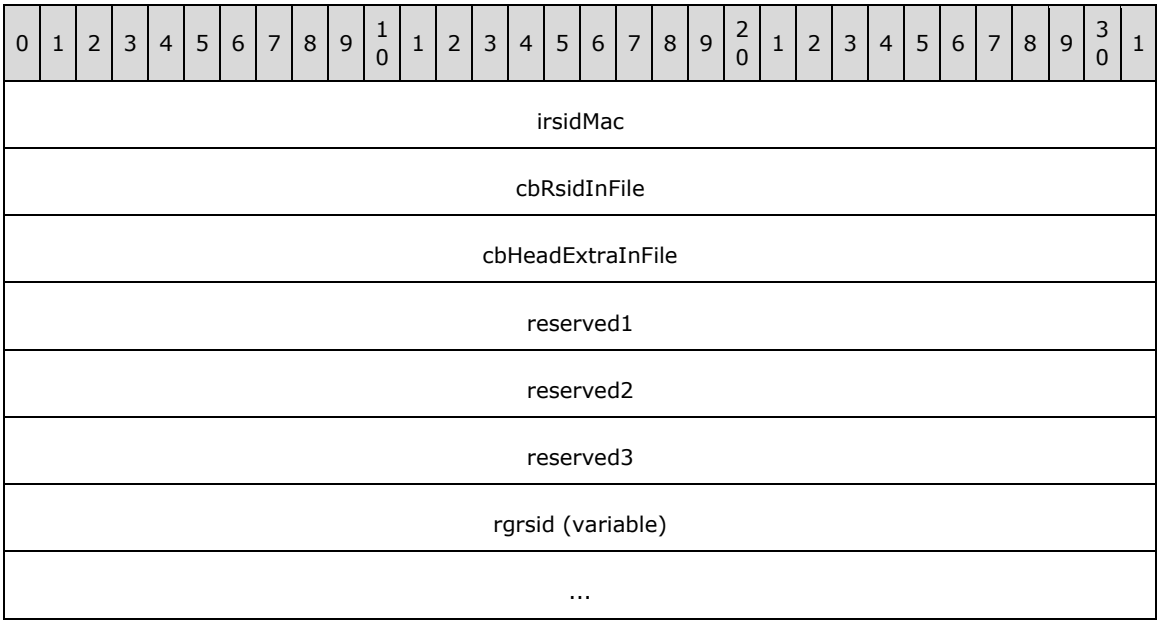
iMac (4 bytes): A signed integer that specifies the number of macro command descriptor structures, as specified by the **Mcd** structure, to follow this structure. This value MUST be greater than or equal to 0.

Copyright © 2025 Microsoft Corporation

rgmcd (variable): An array of **Mcd** structures. The number of structures that are contained in the array is specified by **iMac**.

2.9.203 PLRSID

The **PLRSID** structure is an array of revision-save identifiers (RSIDs), as specified in [ECMA-376] part 1, section 17.15.1.70.



irsidMac (4 bytes): An unsigned integer value that specifies the count of RSIDs that are contained in **rgrsid**.

cbRsidInFile (4 bytes): An unsigned integer value that specifies the size, in bytes, of an RSID. This value MUST be 4.

cbHeadExtraInFile (4 bytes): An unsigned integer value that MUST be 8.

reserved1 (4 bytes): An unsigned integer value that MUST be 229.

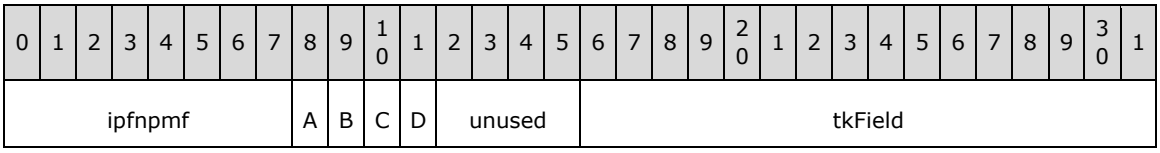
reserved2 (4 bytes): An unsigned **integer** value that MUST be less than "32". This value MUST be ignored.

reserved3 (4 bytes): This value is undefined and MUST be ignored.

rgrsid (variable): An array of RSID elements.

2.9.204 Pmfs

The **Pmfs** structure specifies the mail merge data source connection properties.



Copyright © 2025 Microsoft Corporation

tkRec	fnpi

ipfnpmf (8 bits): An unsigned integer value that specifies the type of data source for the mail merge. This MUST be one of the following values.

Value	Data Source
0xFF	None.
0x00	Data file.
0x01	Microsoft Access database.
0x02	Microsoft Excel file.
0x03	Microsoft Query database.
0x04	ODBC.
0x05	Office Data Source Object (ODSO).

A - fLinkToFnm (1 bit): If the data source is not a data file, this bit MUST be ignored. ,When the data source is a data file, this bit specifies whether the file MUST exist as specified in **fnpi**.

B - fLinkToConn (1 bit): Specifies whether an extra string is appended to the DDE initial connection string. This extra string is stored in the **SttbfRfs** structure in the **Pms** structure.

C - fNoPromptQT (1 bit): Specifies whether the user was already asked about whether to use Microsoft Query to edit ODBC.

D - fQuery (1 bit): Specifies whether the mail merge uses a query (such as "SELECT * FROM x") to obtain the data. If this value is set to zero, the mail merge reads the data file directly.

unused (4 bits): This field is undefined and MUST be ignored.

tkField (2 bytes): A signed integer that specifies the token to separate fields in the data file. If **ipfnpmf** is not 0x00 (data file), this value is undefined and MUST be ignored. Otherwise it MUST be one of the following tokens.

Value	Token
0x00	(none)
0x02	(enter)
0x06	(Tab)
0x0A	,
0x0B	.
0x0C	!
0x0D	#

Copyright © 2025 Microsoft Corporation

Value	Token
0x0E	$
0x0F	%
0x10	&
0x11	(
0x12	)
0x13	*
0x14	+
0x15	-
0x16	/
0x17	:
0x18	;
0x19	<
0x1A	=
0x1B	>
0x1C	?
0x1D	@
0x1E	[
0x1F	]
0x21	^
0x22	_
0x23	`
0x24	{
0x25	}
0x26	|
0x27	~
0x46	(field end)

Copyright © 2025 Microsoft Corporation

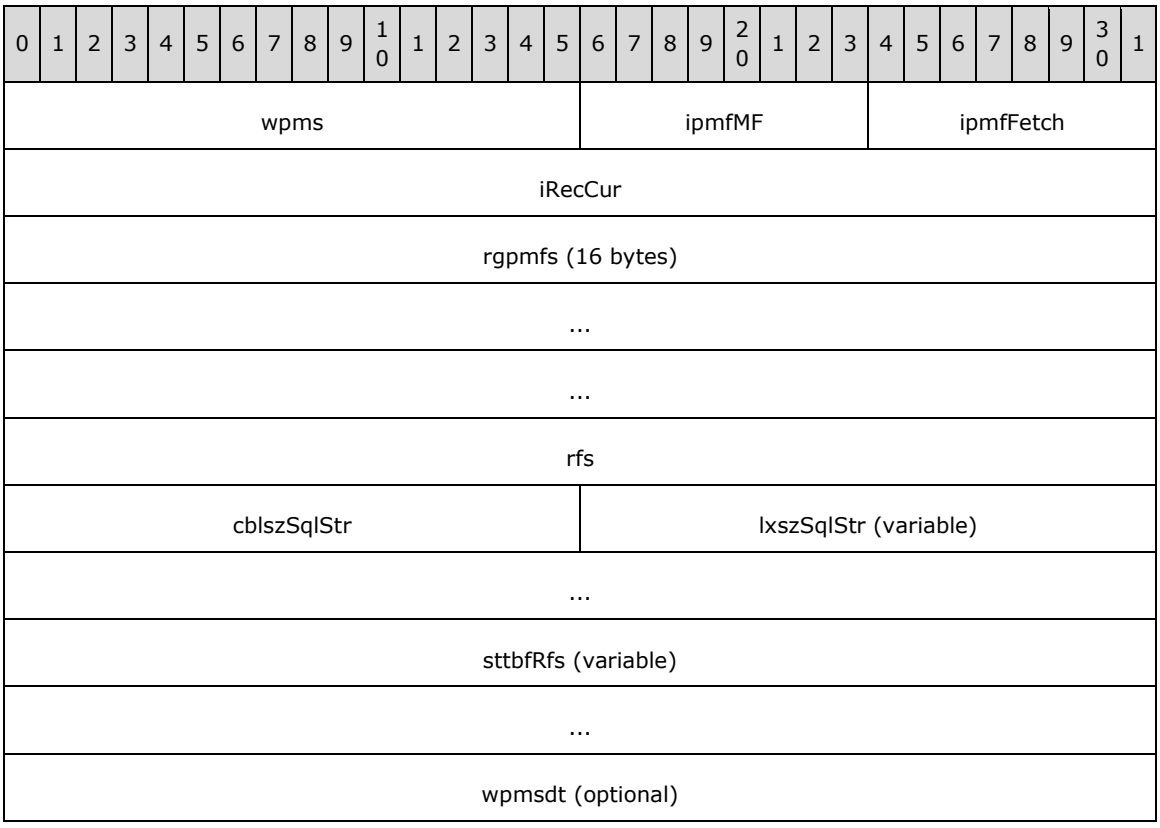
Value	Token
0x47	(table cell)
0x48	(table row)

tkRec (2 bytes): A signed integer that specifies the token to separate records in the data file. If **ipfnpmf** is not 0x00 (data file), this value is undefined and MUST be ignored. Otherwise, it MUST be one of the tokens shown in the table for **tkField**, MUST NOT be 0x00 (none) and MUST be different from **tkField**.

fnpi (2 bytes): An **FNPI** that specifies the type and identifier of a data file. The **fnpt** inside this **fnpi** MUST be 0x3 for mail merge type. The string in the **SttbFnm** structure that has an appended FNIF structure with an **fnpi** that is identical to this one is the file name of this data file for mail merge.

2.9.205 Pms

The **Pms** data structure contains the print merge or mail merge state information.



wpms (2 bytes): The mail merge state as a **Wpms**.

ipmfMF (1 byte): An unsigned integer that specifies the index in the array **rgpmfs** and MUST be 0 or 1. This value is used for the mail merge header field source from which the mail merge column names are obtained.

Copyright © 2025 Microsoft Corporation

ipmfFetch (1 byte): An unsigned integer that specifies the index in the array **rgpmfs** and MUST be 0 or 1. This value is used for the mail merge data fetch source from which the mail merge data is obtained.

iRecCur (4 bytes): An unsigned integer that specifies the index of the current mail merge record. This value MUST be between 0 and 0xFFFFFFF0 as the record index, or it MUST be 0xFFFFFFFF as a nil value.

rgpmfs (16 bytes): An array of two **Pmfs** elements.

rfs (4 bytes): The mail merge record filtering information. See **Rfs**.

cblszSqlStr (2 bytes): An unsigned integer that specifies the length, in bytes, of the string **lxszSqlStr**. Because **lxszSqlStr** is in **Unicode**, **cblszSqlStr** MUST be an even number. If **cblszSqlStr** is zero, **lxszSqlStr** does not exist; otherwise this value MUST be greater than 2 but MUST NOT exceed 512 bytes.

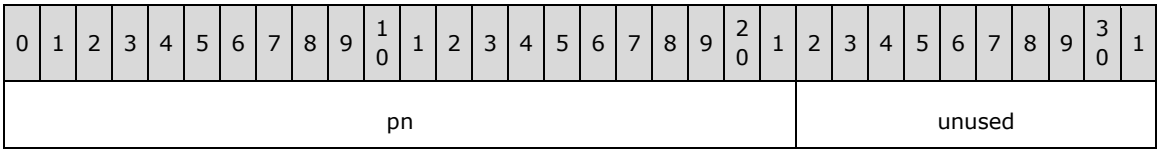
lxszSqlStr (variable): The null-terminated Unicode SQL Query string. For example, "SELECT * FROM [*myTable*] WHERE …", where *myTable* is the table name in the database that is connected. This field is not present if **cblszSqlStr** is zero.

sttbfRfs (variable): The string table, **STTB**, that contains the strings for mail merge connection and record filtering. See the **SttbfRfs** structure. **Pms.sttbfRfs** does not exist if **Pms.rfs.hsttbRfs** is zero. See the Rfs structure.

wpmsdt (4 bytes): The mail merge document type. See the **Wpmsdt** structure.

2.9.206 PnFkpChpx

The **PnFkpChpx** structure specifies the location in the WordDocument Stream of a **ChpxFkp** structure.

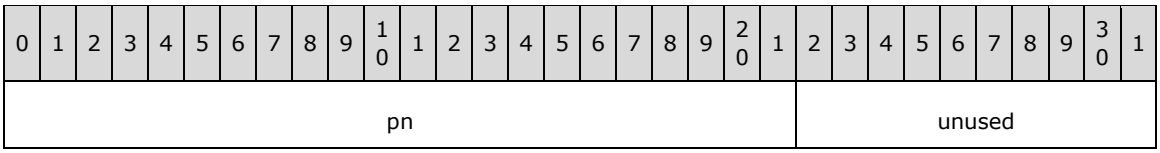


pn (22 bits): An unsigned integer value that specifies the offset in the WordDocument Stream of a **ChpxFkp** structure. The **ChpxFkp** structure begins at an offset of **pn** * 512.

unused (10 bits): This value is undefined and MUST be ignored.

2.9.207 PnFkpPapx

The **PnFkpPapx** structure specifies the offset of a PapxFkp in the WordDocument Stream.



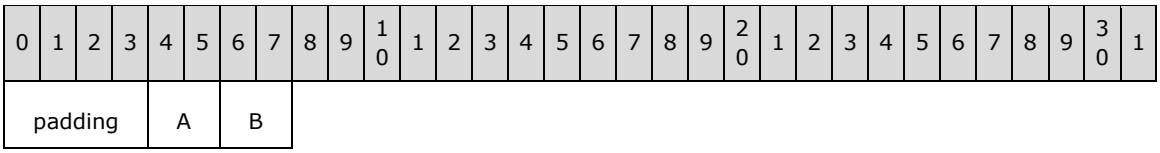
pn (22 bits): An unsigned integer that specifies the offset in the WordDocument Stream of a **PapxFkp** structure. The **PapxFkp** structure begins at an offset of **pn**×512.

unused (10 bits): This value is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

2.9.208 PositionCodeOperand

The **PositionCodeOperand** structure is an operand that specifies the location of an anchor point for an absolutely positioned table or frame.



padding (4 bits): This value MUST be 0 and MUST be ignored.

A - pcVert (2 bits): An unsigned integer that MUST be one of the following values.

Value	Meaning
0	The vertical position of the table or frame is relative to the top page margin.
1	The vertical position of the table or frame is relative to the top edge of the page.
2	The vertical position of the table or frame is relative to the paragraph bottom of the paragraph that precedes it.
3	None. The table or frame is not absolutely positioned.

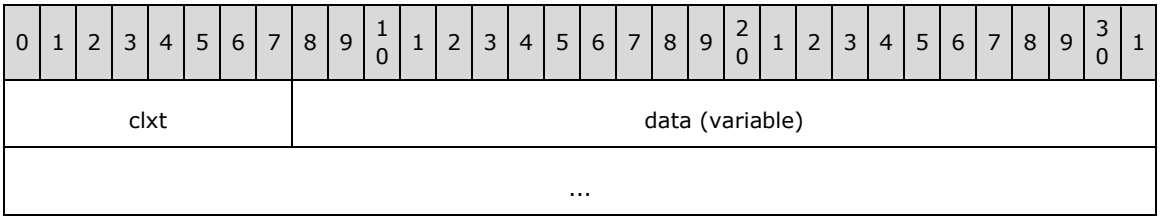
B - pcHorz (2 bits): An unsigned integer that MUST be one of the following values.

Value	Meaning
0	The horizontal position of the table or frame is relative to the left edge of the current column.
1	The horizontal position of the table or frame is relative to the left page margin.
2	The horizontal position of the table or frame is relative to the left edge of the page.
3	None. The table or frame is not absolutely positioned.

Note that all horizontal position measurements are made from the **physical left**.

2.9.209 Prc

The **Prc** structure specifies a set of properties for document content that is referenced by a **Pcd** structure.



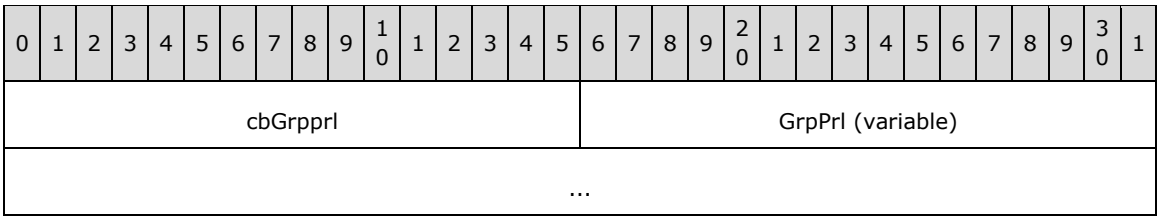
clxt (1 byte): This value MUST be 0x01.

data (variable): A PrcData that specifies a set of properties.

2.9.210 PrcData

The **PrcData** structure specifies an array of **Prl** elements and the size of the array.

Copyright © 2025 Microsoft Corporation

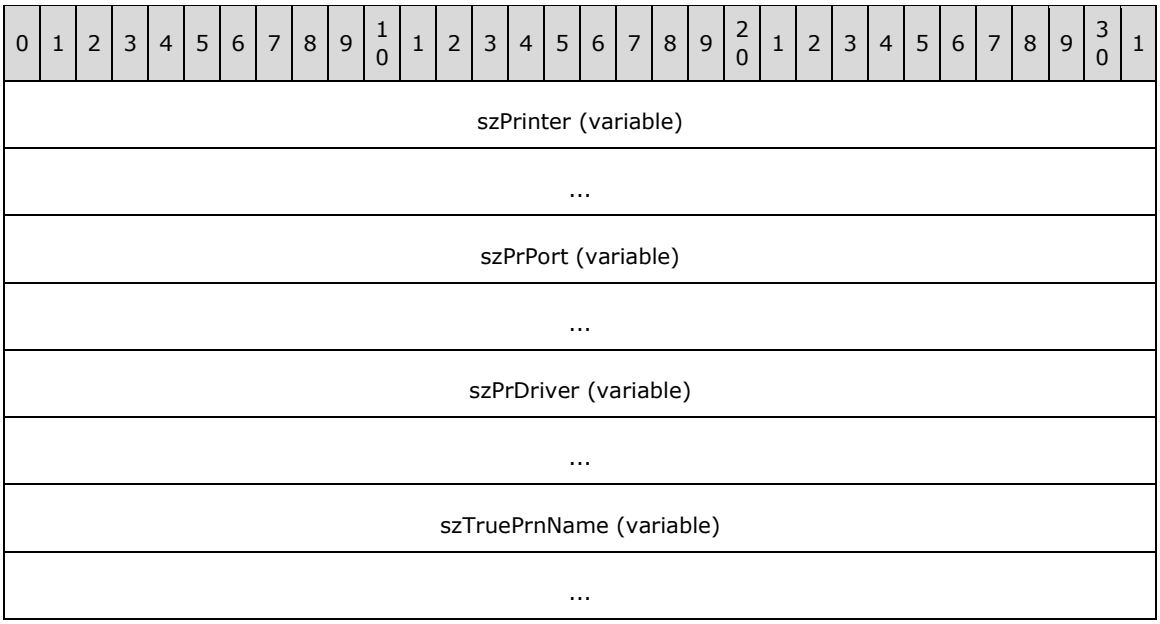


cbGrpprl (2 bytes): A signed integer that specifies the size of **GrpPrl**, in bytes. This value MUST be less than or equal to 0x3FA2.

GrpPrl (variable): An array of **Prl** elements. **GrpPrl** contains a whole number of **Prl** elements.

2.9.211 PrDrvr

The **PrDrvr** structure specifies printer driver information. It contains four null-terminated strings of ANSI characters that specify the printer name, the port, the driver, and the product name of the printer.



szPrinter (variable): A null-terminated string of ANSI characters that specifies the printer name that is used by the computer or the network.

szPrPort (variable): A null-terminated string of ANSI characters that specifies the printer port.

szPrDriver (variable): A null-terminated string of ANSI characters that specifies the printer driver.

szTruePrnName (variable): A null-terminated string of ANSI characters that specifies the product name from the printer manufacturer.

2.9.212 PrEnvLand

The **PrEnvLand** structure specifies print environment information in landscape mode, which is obtained from the printer as a binary block. This is unused and MUST be ignored.

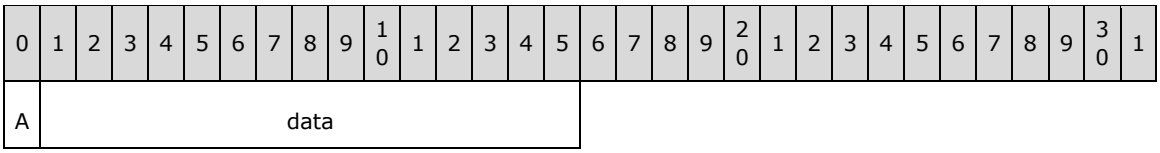
Copyright © 2025 Microsoft Corporation

2.9.213 PrEnvPort

The **PrEnvPort** structure specifies print environment information in portrait mode, which is obtained from the printer as a binary block. This is unused and MUST be ignored.

2.9.214 Prm

A **Prm** structure is either a **Prm0** structure or a **Prm1** structure, depending on the value of the **fComplex** bit.

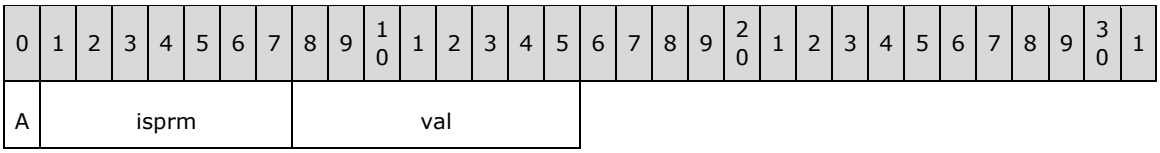


A - fComplex (1 bit): If **fComplex** is 1, this **Prm** is a **Prm1** structure. If **fComplex** is zero, this **Prm** is a **Prm0** structure.

data (15 bits): The interpretation of this field depends on the value of **fComplex**. If **fComplex** is zero, then **data** is the last 15 bits of a **Prm0** structure. If **fComplex** is 1, then **data** is the last 15 bits of a **Prm1** structure.

2.9.215 Prm0

The **Prm0** structure is a **Prm** that has an **fComplex** value of zero. It specifies a single **Sprm** and operand to apply to all document content that is referenced by a **Pcd**.



A - fComplex (1 bit): This value MUST be 0.

isprm (7 bits): An unsigned integer that specifies the Sprm to apply, according to the following table. The operand is specified by **val**.

Isprm	Sprm
0x00	sprmCLbcCRJ. If **val** is also zero, this Prm0 does not apply sprmCLbcCRJ with an operand of zero; instead, it has no effect.
0x04	sprmPIncLvl
0x05	sprmPJc
0x07	sprmPFKeep
0x08	sprmPFKeepFollow
0x09	sprmPFPageBreakBefore
0x0C	sprmPIlvl
0x0D	sprmPFMirrorIndents
0x0E	sprmPFNoLineNumb
0x0F	sprmPTtwo
0x18	sprmPFInTable
0x19	sprmPFTtp
0x1D	sprmPPc
0x25	sprmPWr

Copyright © 2025 Microsoft Corporation

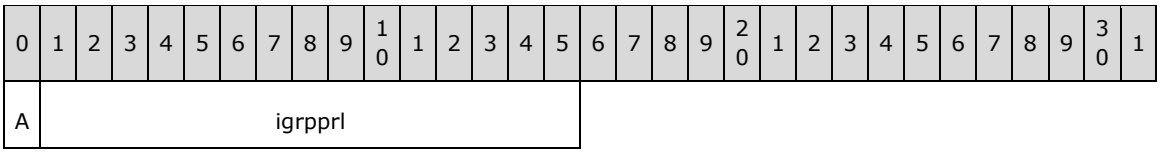
Isprm	Sprm
0x2C	sprmPFNoAutoHyph
0x32	sprmPFLocked
0x33	sprmPFWidowControl
0x35	sprmPFKinsoku
0x36	sprmPFWordWrap
0x37	sprmPFOverflowPunct
0x38	sprmPFTopLinePunct
0x39	sprmPFAutoSpaceDE
0x3A	sprmPFAutoSpaceDN
0x41	sprmCFRMarkDel
0x42	sprmCFRMarkIns
0x43	sprmCFFldVanish
0x47	sprmCFData
0x4B	sprmCFOle2
0x4D	sprmCHighlight
0x4E	sprmCFEmboss
0x4F	sprmCSfxText
0x50	sprmCFWebHidden
0x51	sprmCFSpecVanish
0x53	sprmCPlain
0x55	sprmCFBold
0x56	sprmCFItalic
0x57	sprmCFStrike
0x58	sprmCFOutline
0x59	sprmCFShadow
0x5A	sprmCFSmallCaps
0x5B	sprmCFCaps
0x5C	sprmCFVanish
0x5E	sprmCKul
0x62	sprmCIco
0x68	sprmCIss
0x73	sprmCFDStrike
0x74	sprmCFImprint
0x75	sprmCFSpec
0x76	sprmCFObj
0x78	sprmPOutLvl
0x7B	sprmCFSdtVanish
0x7C	sprmCNeedFontFixup
0x7E	sprmPFNumRMIns

val (8 bits): The operand for the **Sprm** that is specified by **isprm**.

2.9.216 Prm1

The **Prm1** structure is a **Prm** with an **fComplex** value of 1. It specifies properties for document content that is referenced by a **Pcd**.

Copyright © 2025 Microsoft Corporation

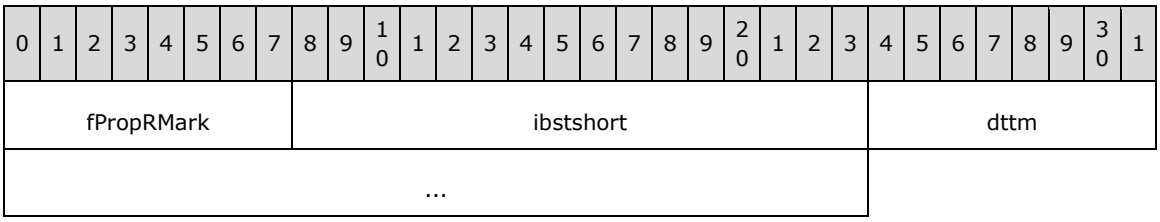


A - fComplex (1 bit): This value MUST be 1.

igrpprl (15 bits): An unsigned integer that specifies a zero-based index of a **Prc** in **Clx**.**RgPrc**. This value MUST be less than the number of **Prc** elements in **Clx.RgPrc**.

2.9.217 PropRMark

The **PropRMark** structure specifies information about a **property revision mark**.



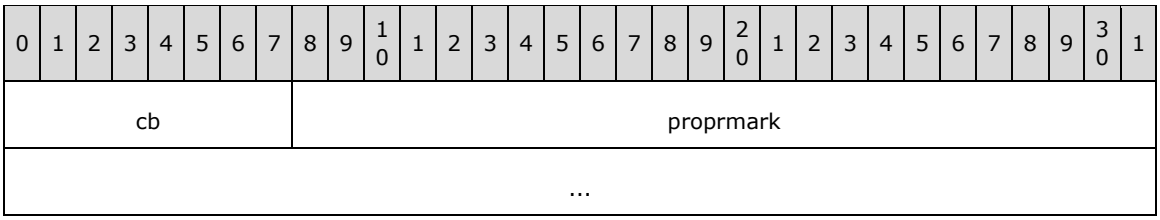
fPropRMark (1 byte): An unsigned integer that specifies if there is a property revision. This value is 1 if there is a property revision; otherwise, if there is no property revision, this value is 0.

ibstshort (2 bytes): A signed integer value that specifies the index into the **SttbfRMark** string table at which the name of the author of the revision is specified.

dttm (4 bytes): A **DTTM** structure that specifies the date and time at which the property revision was made.

2.9.218 PropRMarkOperand

The **PropRMarkOperand** structure is the operand to several **Sprm** structures that specify the properties of **property revision marks**.



cb (1 byte): An unsigned integer that specifies the size of this **PropRMarkOperand**, excluding the current byte. This value MUST be 7.

proprmark (7 bytes): A **PropRMark** structure that holds the properties of the property revision mark that is being specified.

2.9.219 ProtectionType

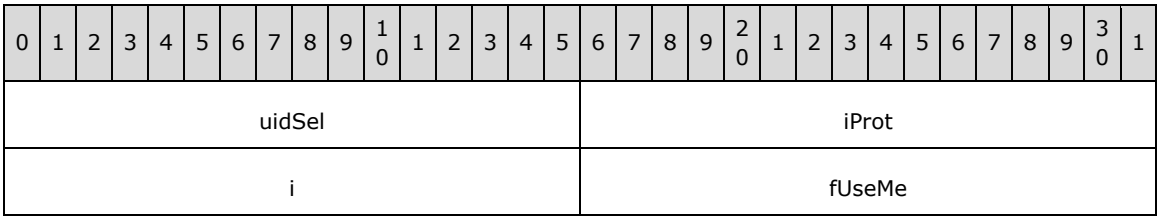
The **ProtectionType** enumeration identifies common types of editing protection for ranges of text in a document.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
iProtNone	0x0000	Allow all changes.
iProtReadWrite	0x0001	Allow the editing of the regions that are marked as editable in forms.
iProtRevision	0x0002	Allow the creation, deletion, and editing of annotations. For all other changes: Allow them, but track them with revision marks.
iProtComment	0x0003	Allow the creation, deletion, and editing of annotations, but allow no other changes.
iProtRead	0x0004	Allow no changes.

2.9.220 PRTI

The **PRTI** structure contains information about a span of text that is delimited by a **range-level protection bookmark** in the document.



uidSel (2 bytes): A UidSel that identifies the permitted editors for the text range that is associated with this PRTI.

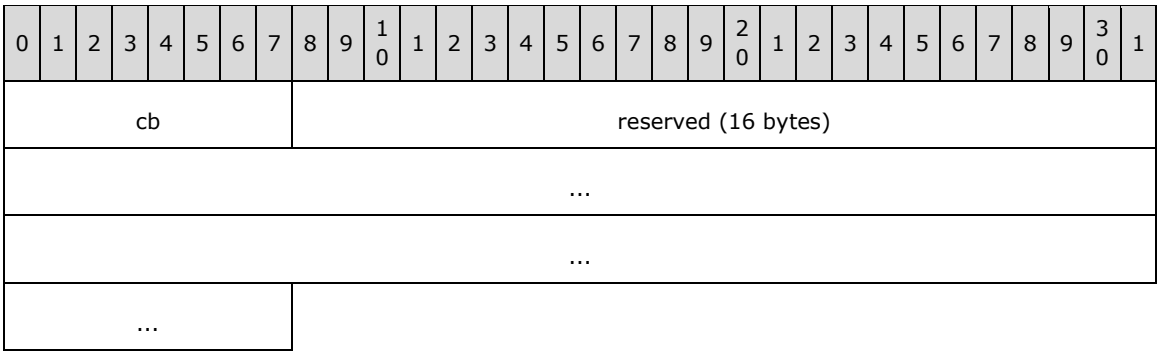
iProt (2 bytes): A ProtectionType that identifies the kind of protection for which exception is granted to the editors that are specified by **uidSel** within a span of text. The span of text is delimited by the **bookmark** that is associated with this PRTI. This MUST be iProtReadWrite.

i (2 bytes): This value is undefined and MUST be ignored.

fUseMe (2 bytes): This value is undefined and MUST be ignored.

2.9.221 PTIstdInfoOperand

The **PTIstdInfoOperand** structure is the operand for sprmPTIstdInfo, and MUST be ignored.



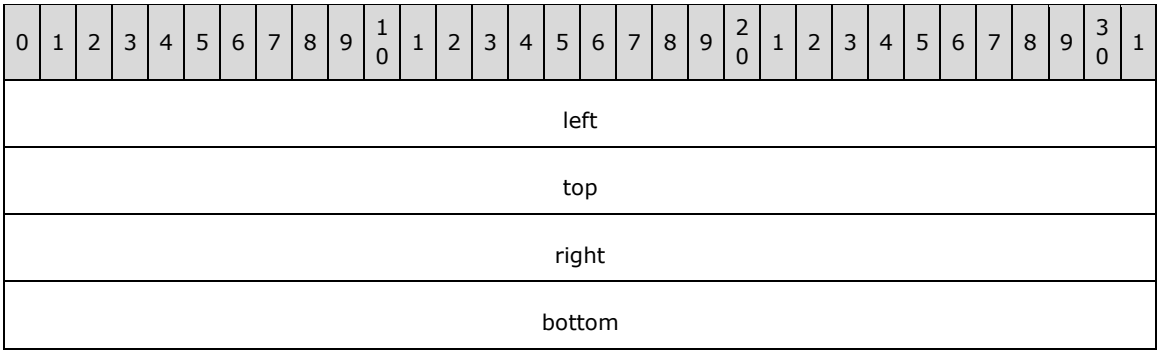
cb (1 byte): An unsigned integer value that specifies the size, in bytes, of this **PTIstdInfoOperand**, excluding the **cb** member. This value MUST be 16.

reserved (16 bytes): This value is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

2.9.222 Rca

The **Rca** structure is used to define the coordinates of a rectangular area in the document. Unless otherwise specified by the other structures that use this structure, the origin is at the top left of the page and the units are in **twips**.



left (4 bytes): An integer that specifies the X coordinate of the top left corner of the rectangle.

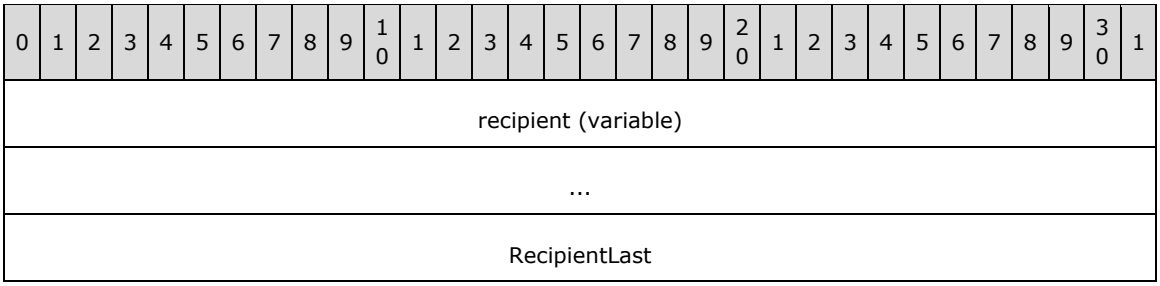
top (4 bytes): An integer that specifies the Y coordinate of the top left corner of the rectangle.

right (4 bytes): An integer that specifies the X coordinate of the bottom right corner of the rectangle.

bottom (4 bytes): An integer that specifies the X coordinate of the bottom right corner of the rectangle.

2.9.223 RecipientBase

The **RecipientBase** structure contains information about a mail merge recipient followed by a marker (**RecipientLast**) that specifies where the recipient information ends.



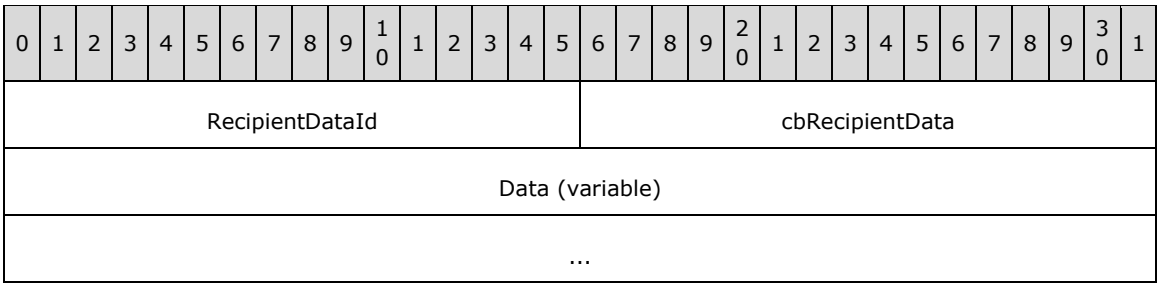
recipient (variable): An array of RecipientDataItem containing data that describes a mail merge recipient. Each **recipient** MUST have a RecipientDataItem with a **RecipientDataID** of 0x0003 or MUST have RecipientDataItem elements that have **RecipientDataID**s of 0x0002 and 0x0004.

RecipientLast (4 bytes): Contains a RecipientTerminator that specifies that there is no further data to read for the current **recipient**.

2.9.224 RecipientDataItem

The **RecipientDataItem** structure specifies information about a mail merge recipient. All the **RecipientDataItem** elements that pertain to a particular recipient are grouped together. The presence of a RecipientTerminator indicates that there is no further data about this recipient. **RecipientDataItem** elements that follow a RecipientTerminator relate to subsequent recipients.

Copyright © 2025 Microsoft Corporation



RecipientDataId (2 bytes): An unsigned integer value that specifies the type of a **RecipientDataItem**. This value MUST be 0x0001, 0x0002, 0x0003, or 0x0004.

cbRecipientData (2 bytes): An unsigned integer that specifies the size, in bytes, of the following **Data** element.

Data (variable): Contains the actual data for this **RecipientDataItem**. The meaning of the data depends on the preceding **RecipientDataId** and is described following.

RecipientDataId	Data
0x0001	An unsigned integer that specifies the status (included or excluded) of a recipient record. This value MUST be 0 (excluded) or 1 (included). If not present, this value defaults to 1.
0x0002	An unsigned integer that specifies the zero-based index of the data source column that uniquely identifies a recipient.

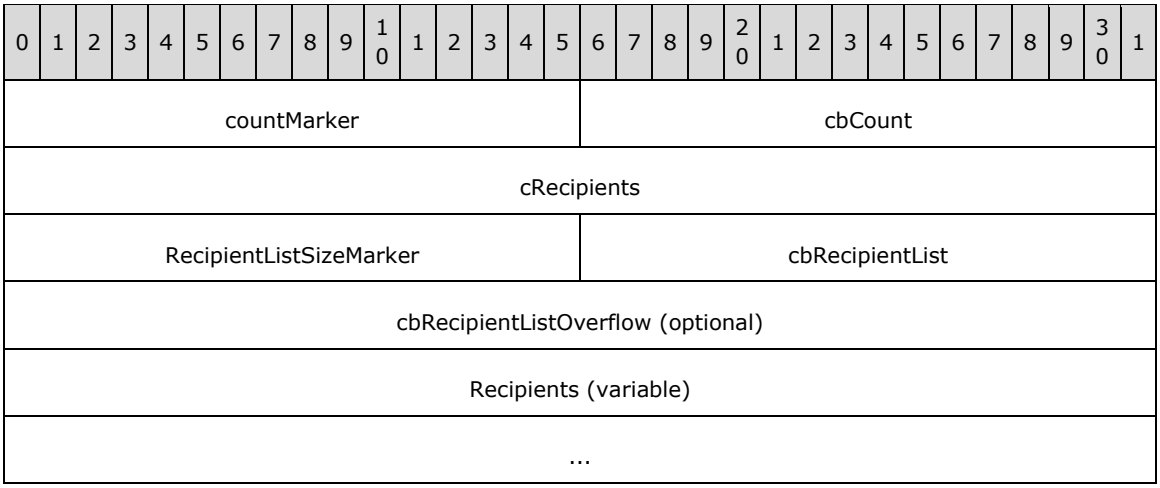
0x0003	An unsigned integer that specifies a hashed DWORD that uniquely identifies a **recipient** if there is no unique column in the data source. The hash value for a data source record is generated as follows:
	
	FUNCTION GetHash
	SET hashValue to 0x00000000
	FOR each column in the data source
	SET strColumn to the string value in the column
	SET hashValue to CALL AddStringToHash hashValue strColumn
	END FOR
	RETURN hashValue
	END FUNCTION
	
	FUNCTION AddStringToHash hashValue, unicodeString
	FOR each character in the unicodeString
	SET hashValue to CALL AddCharacterToHash hashValue character
	END FOR
	END FUNCTION
	
	FUNCTION AddCharacterToHash hashValue, unicodeCharacter
	SET tempCalc to 131 times hashValue plus unicodeCharacter
	IF tempCalc >= 4294967291
	SET tempCalc to tempCalc minus 4294967291
	END IF
	RETURN tempCalc
	END FUNCTION
	
	If the data source is Microsoft Outlook, the last column in the data source SHOULD NOT<232> be used in the preceding function **GetHash**.

0x0004	A Unicode string that specifies the contents of the data source column that uniquely identifies a recipient. The string is not null-terminated.

Copyright © 2025 Microsoft Corporation

2.9.225 RecipientInfo

The **RecipientInfo** structure specifies which recipients in the data source are excluded from the mail merge. It also provides data to uniquely identify each recipient in case the data source was altered after the last read operation.



countMarker (2 bytes): An unsigned integer that specifies that the count of recipients follows. This value MUST be 0.

cbCount (2 bytes): An unsigned integer that specifies the size, in bytes, of **cRecipients**. This value MUST be 0x0004.

cRecipients (4 bytes): An unsigned integer that specifies the number of elements in the **Recipients** array.

RecipientListSizeMarker (2 bytes): An unsigned integer that specifies that the size, in bytes, of the **Recipients** array follows. This value MUST be 0x0001.

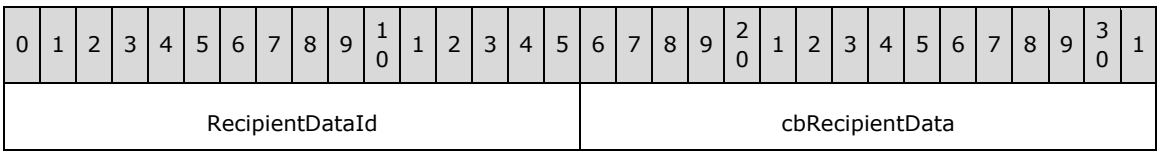
cbRecipientList (2 bytes): An unsigned integer that specifies the size, in bytes, of the **Recipients** array, or, if the size is greater than 0xFFFE, this value MUST be 0xFFFF.

cbRecipientListOverflow (4 bytes): An unsigned integer that specifies the size, in bytes, of the **Recipients** array. This value is present only if **cbRecipientList** is set to 0xFFFF.

Recipients (variable): An array of RecipientBase. An array that contains information about the recipients in the mail merge data source.

2.9.226 RecipientTerminator

The **RecipientTerminator** structure marks the end of the RecipientDataItem elements that pertain to a recipient.



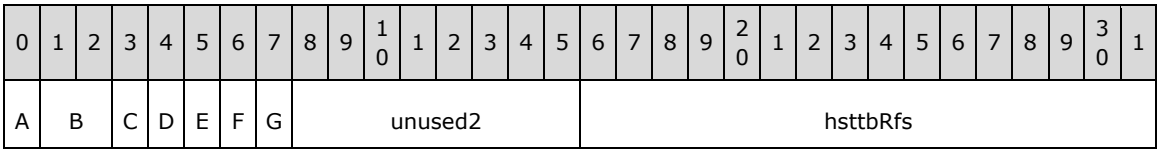
RecipientDataId (2 bytes): An unsigned integer value that specifies there is no further data to read for the current recipient. This value MUST be 0.

cbRecipientData (2 bytes): This value MUST be 0.

Copyright © 2025 Microsoft Corporation

2.9.227 Rfs

The **Rfs** structure specifies record filtering and the other mail merge properties.



A - fShowData (1 bit): Specifies whether the data are shown in the merged fields. If this value is set to zero, only the merged field names are shown.

B - grfChkErr (2 bits): An integer that specifies the settings for error checking and reporting. It MUST be one of the following values.

Value	Meaning
0	Simulate the merge and report errors in a new document.
1	Complete the merge and pause to report errors.
2	Complete the merge and report errors in a new document.

C - fManDocSetup (1 bit): Specifies whether the main document envelope or mailing labels are set up.

D - fMailAsText (1 bit): Specifies whether the e-mail message is in plain text format.

E - unused1 (1 bit): This bit is undefined and MUST be ignored.

F - fDefaultSQL (1 bit): Specifies whether the default SQL query string is "SELECT * FROM x".

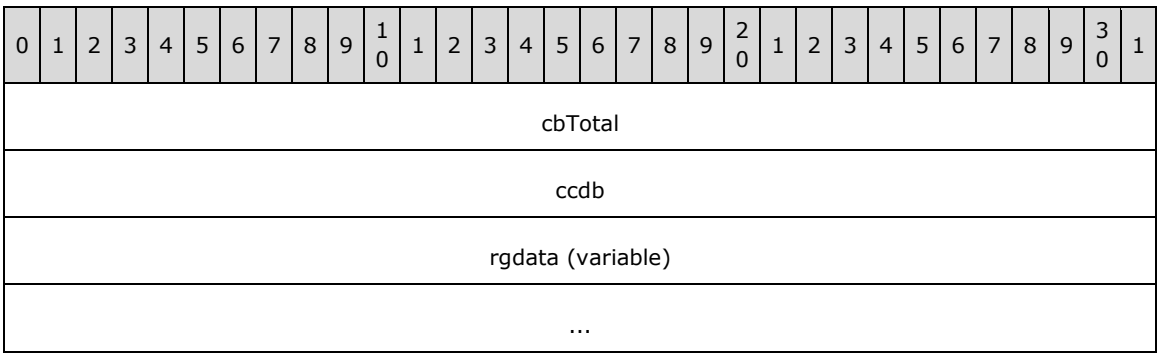
G - fMailAsHtml (1 bit): Specifies whether the e-mail message is in HTML format.

unused2 (8 bits): This field is undefined and MUST be ignored.

hsttbRfs (2 bytes): An unsigned integer that specifies whether **SttbfRfs** exists in **Pms**. If **SttbfRfs** does not exist in **Pms**, **hsttbRfs** MUST be zero. If **Pms** contains **SttbfRfs**, **hsttbRfs** MUST be nonzero (any nonzero value).

2.9.228 RgCdb

The **RgCdb** structure contains binary data for **grammar checker cookies** which are stored by **grammar checkers** that implement the **NLCheck** interface. The data for a grammar checker cookie is implementation-specific to the grammar checker that created the grammar checker cookie.



Copyright © 2025 Microsoft Corporation

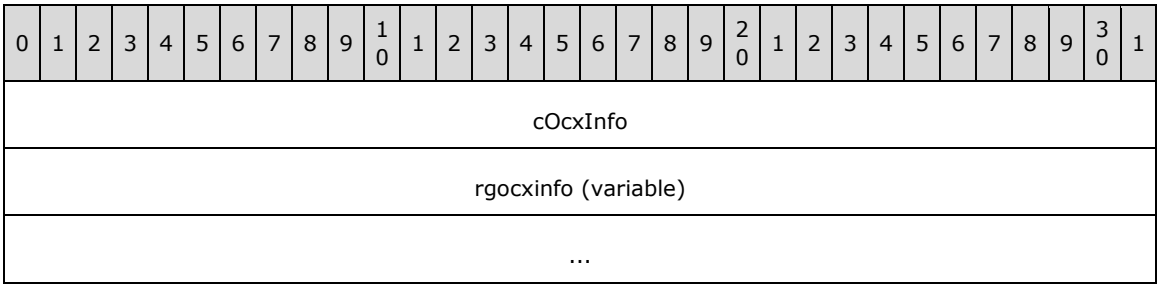
cbTotal (4 bytes): An unsigned integer that specifies the size of this **RgCdb**, including **cbTotal**, in bytes.

ccdb (4 bytes): An unsigned integer that specifies the number of **CDB** entries in **rgdata**.

rgdata (variable): An array of **CDB**. These entries are accessed by using the **icdb** field of **FCKS**.

2.9.229 RgxOcxInfo

The **RgxOcxInfo** structure is an array of **OcxInfo** structures.

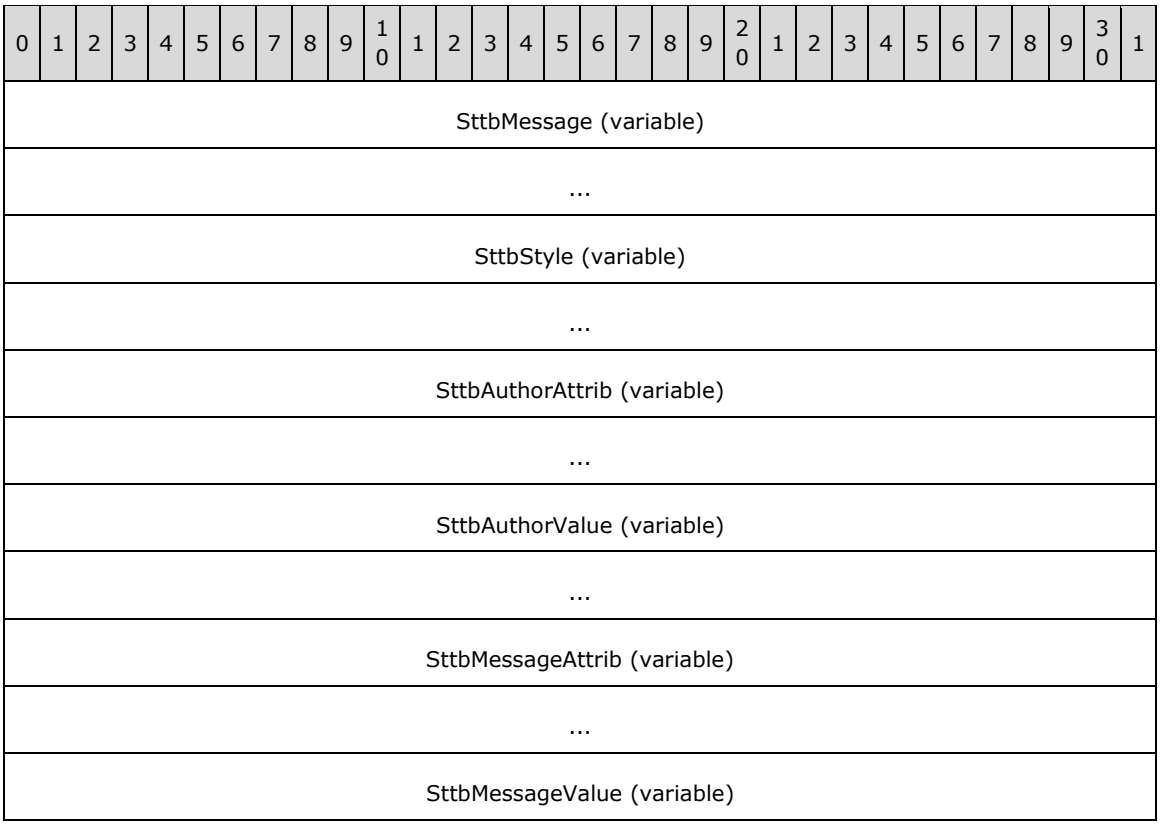


cOcxInfo (4 bytes): An unsigned integer that specifies the number of **OcxInfo** structures in **rgocxinfo**.

rgocxinfo (variable): An array of **OcxInfo** structures.

2.9.230 RmdThreading

The **RmdThreading** structure specifies data about e-mail messages and their authors.



Copyright © 2025 Microsoft Corporation

…

SttbMessage (variable): An **STTB** where each string specifies the **message identifier** for the corresponding author in the parallel **SttbfRMark**. The string is empty if the corresponding author is not the author of an e-mail message. The extra data that is appended to each string is an **MDP** that specifies the message display properties. If a string is empty, the extra data that is appended to it MUST be ignored.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (variable)																															
…																															
$ExtraData_0$ (8 bytes)																															
…																															
$cchData_1$ (2 bytes)																$Data_1$ (variable)															
…																															
$ExtraData_1$ (8 bytes)																															
…																															
…																$cchData_{cData-1}$ (2 bytes)															
$Data_{cData-1}$ (variable)																															
…																															
$ExtraData_{cData-1}$ (8 bytes)																															
…																															

SttbMessage is an **STTB** with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x0008.

SttbStyle (variable): An **STTB** where each string specifies the **personal style** of the corresponding author in the parallel **SttbfRMark**. The string is empty if the corresponding author does not have a personal style. There is no extra data appended to the strings of this **STTB**.

Copyright © 2025 Microsoft Corporation

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (variable)																															
…																															
$cchData_1$ (2 bytes)																$Data_1$ (variable)															
…																															
…																$cchData_{cData-1}$ (2 bytes)															
$Data_{cData-1}$ (variable)																															
…																															

SttbStyle is an **STTB** with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0.

SttbAuthorAttrib (variable): An **STTB** in which each string specifies an author attribute. The extra data appended to each string is a 16-bit signed integer that specifies a zero-based index of an author in the **SttbfRMark** to which this attribute is related. If a string is an empty string, the data that is appended to it MUST be ignored, and the corresponding value in the parallel **SttbAuthorValue** MUST be ignored. **SttbAuthorAttrib** SHOULD<233> be ignored.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (variable)																															
…																															
$ExtraData_0$ (2 bytes)																$cchData_1$ (2 bytes)															
$Data_1$ (variable)																															
…																															

Copyright © 2025 Microsoft Corporation

ExtraData$_1$ (2 bytes)	…
cchData$_{cData-1}$ (2 bytes)	Data$_{cData-1}$ (variable)
…	
ExtraData$_{cData-1}$ (2 bytes)	

SttbAuthorAttrib is an **STTB** with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x0002.

SttbAuthorValue (variable): An **STTB** where each string specifies the value of the corresponding author attribute in the parallel **SttbAuthorAttrib**. There is no extra data appended to the strings of this **STTB**. **SttbAuthorValue** SHOULD<234> be ignored.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																cchData$_0$ (2 bytes)															
Data$_0$ (variable)																															
…																															
cchData$_1$ (2 bytes)																Data$_1$ (variable)															
…																															
…																cchData$_{cData-1}$ (2 bytes)															
Data$_{cData-1}$ (variable)																															
…																															

SttbAuthorValue is an STTB with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0.

SttbMessageAttrib (variable): An **STTB** in which each string specifies a message attribute. The extra data that is appended to each string is a 16-bit signed integer that specifies a zero-based index of a message that this attribute pertains to in **SttbMessage**. If a string is an empty string, the data that is appended to it MUST be ignored, and the corresponding value in the parallel **SttbMessageValue** MUST be ignored. **SttbMessageAttrib** SHOULD<235> be ignored.

Copyright © 2025 Microsoft Corporation

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (variable)																															
…																															
$ExtraData_0$ (2 bytes)																$cchData_1$ (2 bytes)															
$Data_1$ (variable)																															
…																															
$ExtraData_1$ (2 bytes)																…															
$cchData_{cData-1}$ (2 bytes)																$Data_{cData-1}$ (variable)															
…																															
$ExtraData_{cData-1}$ (2 bytes)																															

SttbMessageAttrib is an **STTB** with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x0002.

SttbMessageValue (variable): An **STTB** in which each string specifies the value of the corresponding message attribute in the parallel **SttbMessageAttrib**. No extra data is appended to the strings of this **STTB**. **SttbMessageValue** SHOULD<236> be ignored.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (variable)																															
…																															
$cchData_1$ (2 bytes)																$Data_1$ (variable)															
…																															

Copyright © 2025 Microsoft Corporation

…	cchData$_{cData-1}$ (2 bytes)
Data$_{cData-1}$ (variable)	
…	

SttbMessageValue is an **STTB** with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0.

2.9.231 Rnc

The **Rnc** enumeration specifies whether and when the numbering for footnotes or endnotes restarts. The members of this enumeration are specified as the following 8-bit unsigned integer values.

Name	Value	Meaning
rncCont	0x00	Numbering is continuous throughout the whole document.
rncRstSect	0x01	Numbering restarts at the beginning of the **section**.
rncRstPage	0x02	Numbering restarts every page.

2.9.232 RouteSlip

The **RouteSlip** structure contains information about the routing slip of the document.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fRouted																fReturnOrig															
fTrackStatus																fDirty															
nProtect																iStage															
delOption																cRecip															
szSubject (variable)																															
...																															
szMessage (variable)																															
...																															
szStatus (variable)																															
...																															

Copyright © 2025 Microsoft Corporation

szTitle (variable)
...
rgRouteSlips (variable)
...

fRouted (2 bytes): A 16-bit Boolean value that specifies whether the document was sent out for review.

fReturnOrig (2 bytes): A 16-bit Boolean value that specifies whether the document is returned to the original sender after the review route is complete.

fTrackStatus (2 bytes): A 16-bit Boolean value that specifies whether status tracking e-mail is sent to the original sender.

fDirty (2 bytes): This value MUST be 0, and MUST be ignored.

nProtect (2 bytes): An unsigned integer value that specifies the kinds of changes allowed to the document being routed. This MUST be one of the values that are defined in RouteSlipProtectionEnum.

iStage (2 bytes): A 16-bit signed integer value that specifies the index of the current routing recipient. This value MUST be greater or equal to zero, and less than the value of **cRecip**.

delOption (2 bytes): A 16-bit signed integer value that specifies how the document is routed. This value MUST be 0 or 1. A value of 0 means the document is sent to reviewers in serial order. A value of 1 means the document is sent to all reviewers in parallel order.

cRecip (2 bytes): A 16-bit signed integer that specifies the number of recipients of the routing slip. This is the size of the **rgRouteSlips** array. This value MUST be greater than or equal to zero.

szSubject (variable): A string with a length prefix of 2 bytes containing ANSI characters that represent the subject to be mailed with the route slip. This string MUST be less than 256 characters in length. The string is encoded by using the system code page of the computer that saved the file.

szMessage (variable): A string with a length prefix of 2 bytes containing ANSI characters that represent the message body to be mailed with the route slip. This string MUST be less than 256 characters in length. The string is encoded by using the system code page of the computer that saved the file.

szStatus (variable): A string with a length prefix of 2 bytes containing ANSI characters that represent status information about the document to be mailed with the route slip. This string MUST be less than 256 characters in length. The string is encoded by using the system code page of the computer that saved the file.

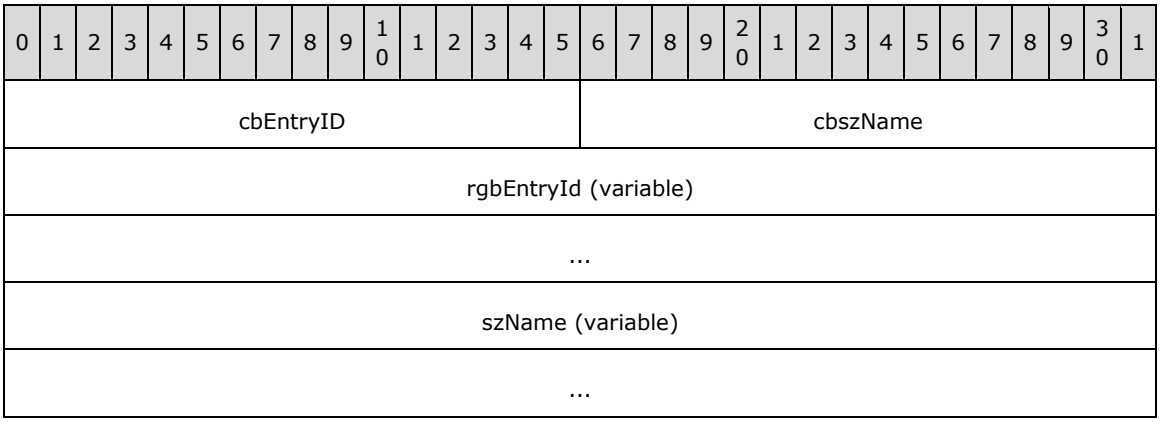
szTitle (variable): A string with a length prefix of 2 bytes containing ANSI characters that represent a title for the route slip. This string MUST be less than 256 characters long. The string is encoded by using the system code page of the computer that saved the file.

rgRouteSlips (variable): An array of **cRecip** **RouteSlipInfo** structures that contains all the routing slips.

Copyright © 2025 Microsoft Corporation

2.9.233 RouteSlipInfo

The **RouteSlipInfo** structure provides information about a single routing slip recipient.



cbEntryID (2 bytes): A 16-bit signed integer that specifies the number of bytes in **rgbEntryId.**

cbszName (2 bytes): A 16-bit signed integer that specifies the number of bytes in **szName.** This value MUST be greater than zero.

rgbEntryId (variable): An array of bytes that provide a unique identifier for this routing slip recipient.

szName (variable): A narrow string that specifies the name or e-mail alias of the routing slip recipient. The length of the string MUST be equal to **cbszName**. The string is encoded by using the operating system code page of the computer that last saved this file.

2.9.234 RouteSlipProtectionEnum

The **RouteSlipProtectionEnum** enumeration lists the possible protection levels for a document being routed.

Name	Value	Meaning
ProtectOff	0x0000	No protection.
ProtectRevMark	0x0001	Changes to the document can be neither accepted nor rejected, and change tracking cannot be turned off.
ProtectAnnot	0x0002	Users can insert comments into the document but cannot change the content of the document.
ProtectForm	0x0003	Users can make changes only in form fields or in unprotected sections of a document.

2.9.235 SBkcOperand

The **SBkcOperand** structure is the operand to **sprmSBkc**. This structure is an 8-bit unsigned integer that specifies the type of the **section break** that is being described.

Name	Value	Meaning
bkcContinuous	0x00	A continuous section break. The next **section** starts on the next line.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
bkcNewColumn	0x01	A new column section break. The next section starts in the next column.
bkcNewPage	0x02	A new page section break. The next section starts on the next page.
bkcEvenPage	0x03	An even page section break. The next section starts on an even page.
bkcOddPage	0x04	An odd page section break. The next section starts on an odd page.

2.9.236 SBOrientationOperand

The **SBOrientationOperand** structure is the operand to **sprmSBOrientation**. This structure is an 8-bit unsigned integer that specifies page orientation.

Name	Value	Meaning
dmOrientPortrait	0x01	Portrait orientation.
dmOrientLandscape	0x02	Landscape orientation.

2.9.237 SClmOperand

The **SClmOperand** structure provides an enumeration which specifies the type of document grid that is used for the **section**. This enumeration defines the following 16-bit unsigned integer values.

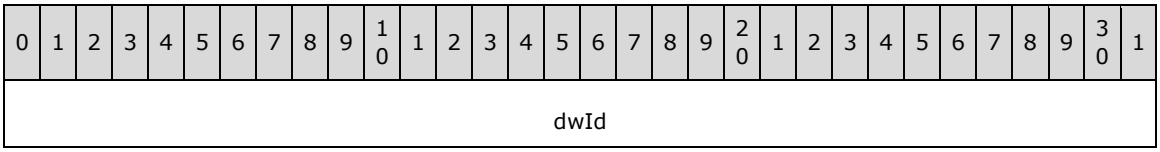
Name	Value	Meaning
clmUseDefault	0x0000	Specifies that document grid is disabled.
clmCharsAndLines	0x0001	Specifies a document grid that enforces both character spacing and line pitch. Line pitch is specified by **sprmSDyaLinePitch**; character spacing is specified by **sprmSDxtCharSpace**.
clmLinesOnly	0x0002	Specifies a document grid that enforces only line pitch. Line pitch is specified by **sprmSDyaLinePitch**.
clmEnforceGrid	0x0003	Specifies a document grid that enforces both character spacing and line pitch. Line pitch is specified by **sprmSDyaLinePitch**; character spacing is specified by **sprmSDxtCharSpace**. Each full-width character MUST occupy its own grid square.

2.9.238 SDmBinOperand

The **SDmBinOperand** enumeration is a 16-bit unsigned integer that specifies a paper source for a printer. The determination and interpretation of this value is implementation specific.

2.9.239 SDTI

The SDTI structure contains information about a **structured document tag bookmark** in the document.



Copyright © 2025 Microsoft Corporation

tiq
...
sdtt
cfsdap
cbPlaceholder
fsdaparray (variable)
...
xszPlaceholder (variable)
...

dwId (4 bytes): An unsigned integer that specifies a unique value which is used to reference the structured document tag bookmark associated with this **SDTI** structure. This value MUST be unique for all **SDTI** structures that are contained in a given **SttbfBkmkSdt**. This value MUST NOT be 0.

tiq (8 bytes): A **TIQ** that specifies further information about the structured document tag bookmark that is associated with this **SDTI** structure.

sdtt (4 bytes): An **SDTT** structure that specifies further information about the structured document tag bookmark that is associated with this **SDTI**. The **SDTT** structure MUST NOT be **sdttUnknown**.

cfsdap (4 bytes): An unsigned integer value that specifies the number of elements in **fsdaparray**.

cbPlaceholder (4 bytes): An unsigned integer that specifies the count of bytes, including the terminating NULL character, in **xszPlaceholder**.

fsdaparray (variable): An array of **FSDAP** structures, each of which specifies further information about the structured document tag bookmark that is associated with this **SDTI** structure.

xszPlaceholder (variable): A null-terminated sequence of **Unicode** characters that specifies the text to show when the **structured document tag** that is denoted by this structured document tag bookmark is empty and XML tag characters themselves are not being shown.

2.9.240 SDTT

The SDTT structure specifies the type of **structured document tag** that is represented by a **structured document tag bookmark** in the document.

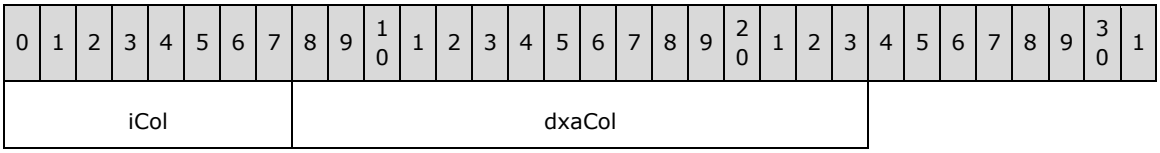
Name	Value	Meaning
sdttUnknown	0x00000000	The type of the tag is determined from the range it encloses.
sdttRegular	0x00000001	The tag encloses a range of characters.
sdttPara	0x00000002	The tag encloses a range of paragraphs.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
sdttCell	0x00000003	The tag encloses a range of cells in a table.
sdttRow	0x00000004	The tag encloses a range of rows in a table.

2.9.241 SDxaColSpacingOperand

The **SDxaColSpacingOperand** structure is the operand to **Sprm** structures that control column size and spacing.

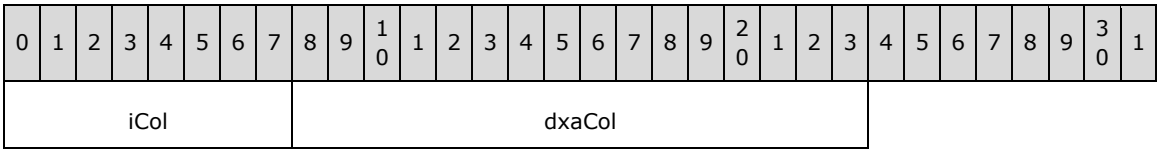


iCol (1 byte): An unsigned integer that specifies the zero-based index of the column that is being referenced by the **Sprm**. This value MUST be less than or equal to 43.

dxaCol (2 bytes): An **XAS_nonNeg** value that specifies the space after the column that is specified by **iCol**.

2.9.242 SDxaColWidthOperand

The **SDxaColWidthOperand** structure is the operand to **Sprm** structures that control column size and spacing.

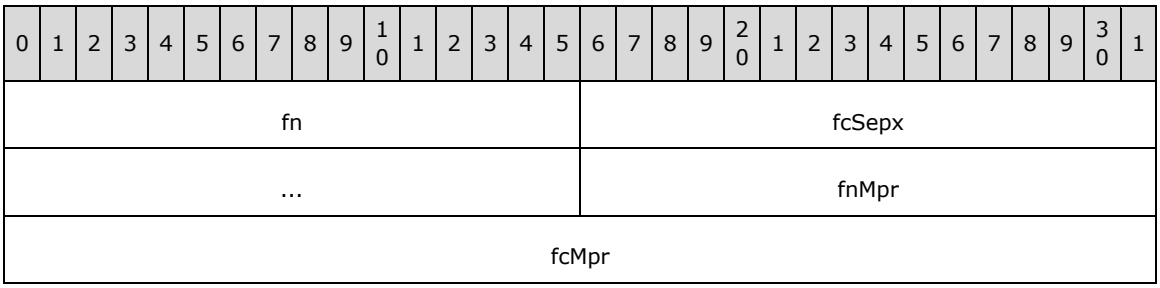


iCol (1 byte): An unsigned integer value that specifies the zero-based index of the column that is referenced by the **Sprm**. This value MUST be less than or equal to 43.

dxaCol (2 bytes): An **XAS_nonNeg** value that specifies the width of the column that is specified by **iCol**. This value MUST be at least 718.

2.9.243 Sed

The **Sed** structure specifies the location of the **section** properties.



fn (2 bytes): This value is undefined and MUST be ignored.

Copyright © 2025 Microsoft Corporation

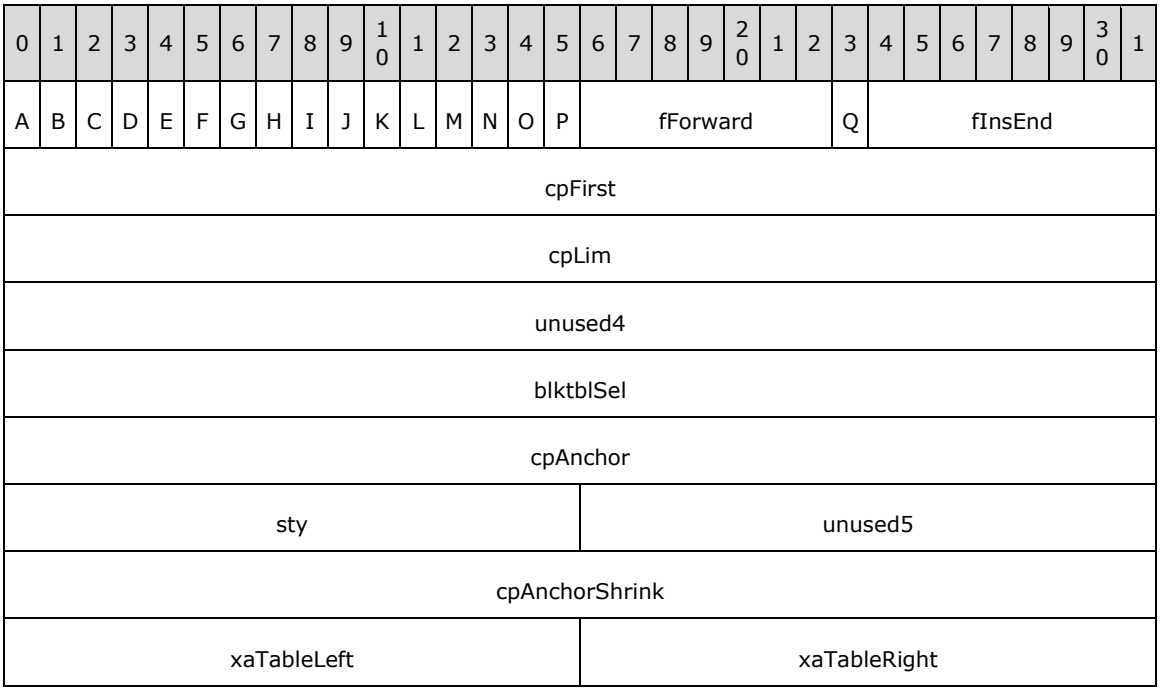
fcSepx (4 bytes): A signed integer value that specifies the position in the WordDocument Stream at which a **Sepx** structure is located.

fnMpr (2 bytes): This value is undefined and MUST be ignored.

fcMpr (4 bytes): This value is undefined and MUST be ignored.

2.9.244 Selsf

The **Selsf** structure specifies the last selection that was made to the document.



A - fRightward (1 bit): A bit that specifies whether the selection was made from the **physical left** to the **physical right**. If **fBlock** is 0, this bit is undefined and MUST be ignored.

B - unused1 (1 bit): This bit is undefined and MUST be ignored.

C - fWithinCell (1 bit): A bit that specifies that the selection is content within a table cell. This value MUST be 0 if the selection contains only whole table cells.

D - fTableAnchor (1 bit): If this bit is 1, then the selection began with either table content or table cells.

E - fTableSelNonShrink (1 bit): If this bit is 1, then the selection began with the use of the mouse to select the whole table cell and that the selection contains only whole table cells.

F - unused2 (1 bit): This bit is undefined and MUST be ignored.

G - fDiscontiguous (1 bit): If this bit is 1, then the selection was made of two or more ranges within the document. The **Selsf** structure describes only the most recent range that was selected.

H - fPrefix (1 bit): If this bit is 1, then the selection is a bullet in a bulleted list or a number in a numbered list.

I - fShape (1 bit): A bit that specifies that the selection is a shape or floating picture. This value MUST be 0 if the selection is a textbox or inline picture.

Copyright © 2025 Microsoft Corporation

J - fFrame (1 bit): A bit that specifies that the selection is a text frame. This value MUST be 0 if the selection is a textbox.

K - fColumn (1 bit): If this bit is 1, then the selection contains one or more whole table cells. This bit MUST be 0 if the selection was made strictly of whole table rows or the entire table.

L - fTable (1 bit): If this bit is 1, then the selection contains one or more whole table cells.

M - fGraphics (1 bit): A bit that specifies that the selection is an inline picture. This value MUST be 0 if the selection is a floating picture.

N - fBlock (1 bit): A bit that specifies that the selection was made of a rectangular block. If **fTable** is 0, the selection is a block of text and MUST NOT contain table content. If **fTable** is 1, the selection is a block of table cells; **fBlock** MUST be 0 if the table selection is restricted to whole table rows or is the entire table.

O - unused3 (1 bit): This bit is undefined and MUST be 0.

P - fIns (1 bit): A bit that specifies that the selection is an insertion point. If **fIns** is 1, **cpFirst** MUST equal **cpLim**.

fForward (7 bits): An unsigned integer that MUST be 0 or 1. This field specifies that the selection was made in a downward direction or towards the **logical right** if the value is 1.

Q - fPrefixW2007 (1 bit): A bit that SHOULD<237> be 0 and MUST be ignored.

fInsEnd (8 bits): An unsigned integer value that MUST be 0 or 1. If this value is 1, the selection is an insertion point at the end of the line, as opposed to at the beginning of the following line. If **fInsEnd** is 1, **fIns** MUST also be 1. If **fShape** is 1, **fInsEnd** is undefined and MUST be ignored. If the selection does not fall at a line break, fInsEnd MUST be ignored.

cpFirst (4 bytes): A signed integer that specifies the start point, in characters, of the selection range. This value MUST be at least 0, and MUST NOT exceed the end of the text piece. If the selection begins with whole table cells, **cpFirst** MUST be the location of the beginning of the row that contains the first selected cell. If the selection is a block selection of text, **cpFirst** MUST be the location of the beginning of the first line that contains selected characters.

cpLim (4 bytes): A signed integer that specifies the endpoint, in characters, of the selection range. This value MUST be at least 0, MUST be greater than or equal to **cpFirst**, and MUST NOT exceed the end of the document. If the selection ends with whole table cells, **cpLim** MUST be the location of the end of the row that contains the last selected cell. If the selection is a block selection of text, **cpLim** MUST be the location of the beginning of the last line that contains selected characters.

unused4 (4 bytes): Undefined and MUST be ignored.

blktblSel (4 bytes): Specifies a selection range. The interpretation of **blktblSel** depends on the values of **fTable** and **fBlock**, which are provided following.

fTable	fBlock	Interpretation
0	0	**blktblSel** is undefined and MUST be ignored.
0	1	**blktblSel** is a **BlockSel** and specifies the dimensions of a block selection.
1	0	**blktblSel** is a **TableSel** and specifies a row selection.

Copyright © 2025 Microsoft Corporation

fTable	fBlock	Interpretation
1	1	**blktblSel** is a **TableSel** and specifies a range of table cells.

cpAnchor (4 bytes): A signed integer that specifies the point, in characters, at which the selection initially began. This value MUST be greater than or equal to **cpFirst**. If the selection was automatically extended to include text before **cpAnchor**, **cpFirst** is less than **cpAnchor**. If the selection was not extended before the point where the selection began, **cpAnchor** is equal to **cpFirst**.

sty (2 bytes): A **Sty** structure that specifies the type of selection that was made.

unused5 (2 bytes): This field is undefined and MUST be ignored.

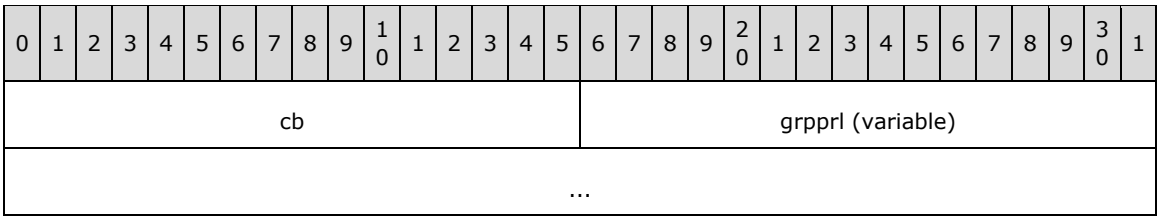
cpAnchorShrink (4 bytes): A signed integer that specifies the point, in characters, where a block selection began. If **fBlock** is 0 or **fTable** is 1, **cpAnchorShrink** is undefined and MUST be ignored.

xaTableLeft (2 bytes): A signed integer that specifies, in **twips**, the physical left edge of the first selected cell if the selection contains whole table cells. This value MUST be in the range of -31680 to 31680, inclusive. If the entire row is selected, **xaTableLeft** MUST be -31680. If the selection does not contain whole table cells, **xaTableLeft** is undefined and MUST be ignored.

xaTableRight (2 bytes): A signed integer that specifies, in twips, the physical right edge of the last selected cell if the selection contains whole table cells. This value MUST be in the range of -31680 to 31680, inclusive, and MUST be greater than or equal to **xaTableLeft**. If the entire row is selected, **xaTableRight** MUST be 31680. If the selection does not contain whole table cells, **xaTableRight** is undefined and MUST be ignored.

2.9.245 Sepx

The **Sepx** structure specifies an array of **Prl** structures and the size of the array.



cb (2 bytes): A signed integer that specifies the size of **grpprl**, in bytes.

grpprl (variable): An array of **Prl** structures that specify the properties of a **section**. This array MUST contain a whole number of **Prl** structures.

2.9.246 SFpcOperand

The **SFpcOperand** enumeration provides an 8-bit unsigned integer that specifies the positioning of the **section** footnote. **SFpcOperand** is the operand to **sprmSFpc**.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
fpcBottomPage	0x01	Footnotes are positioned at the bottom of the page.
fpcBeneathText	0x02	Footnotes are positioned beneath the text on the page.

2.9.247 Shd

The **Shd** structure specifies the colors and pattern that are used for background shading.

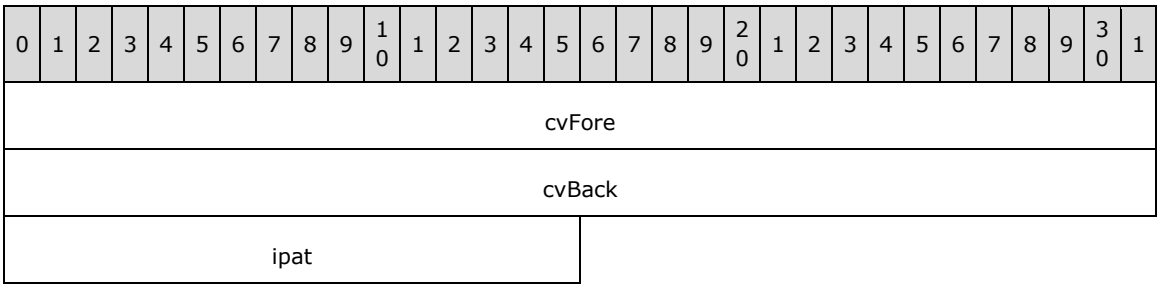
ShdAuto is a special value for **Shd** that specifies that no shading is applied and is defined as the following **Shd**.

Field	Value
cvFore	Field: Value Red: 0x00 Green: 0x00 Blue: 0x00 fAuto: 0xFF
cvBack	Field: Value Red: 0x00 Green: 0x00 Blue: 0x00 fAuto: 0xFF
ipat	ipatAuto

ShdNil is a special value for **Shd**. If **ShdNil** is used in a Table Style definition, **ShdNil** is ignored and the shading of the cell is not affected. If **ShdNil** is applied outside of a Table Style, **ShdNil** specifies that no shading is applied. **ShdNil** is defined as the following Shd.

Copyright © 2025 Microsoft Corporation

Field	Value
cvFore	Field: Red, Green, Blue, fAuto Value: 0xFF, 0xFF, 0xFF, 0xFF
cvBack	Field: Red, Green, Blue, fAuto Value: 0xFF, 0xFF, 0xFF, 0xFF
ipat	ipatAuto



cvFore (4 bytes): A **COLORREF** that specifies the foreground color of **ipat**.

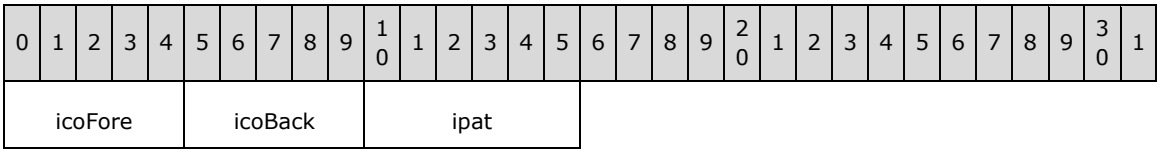
cvBack (4 bytes): A **COLORREF** that specifies the background color of **ipat**.

ipat (2 bytes): An **Ipat** that specifies the pattern used for shading.

2.9.248 Shd80

The **Shd80** structure specifies the colors and pattern that are used for background shading. As an exception to the constraints that are specified by **Ico** and **Ipat**, a **Shd80** can be set to **Shd80Nil** and specifies that no shading is applied. **Shd80Nil** is defined as the following **Shd80**.

Field	Value
icoFore	0x1F
icoBack	0x1F
ipat	0x3F



icoFore (5 bits): An **Ico** that specifies the foreground color of **ipat**.

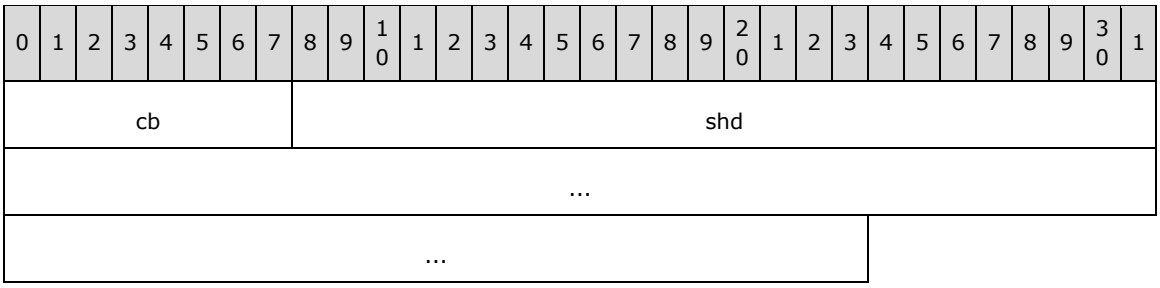
icoBack (5 bits): An **Ico** that specifies the background color of **ipat**.

ipat (6 bits): An **Ipat** that specifies the pattern used for shading.

Copyright © 2025 Microsoft Corporation

2.9.249 SHDOperand

The **SHDOperand** structure is an operand that is used by several **Sprm** structures to specify the background shading to be applied.



cb (1 byte): An unsigned integer that specifies the size of this operand in bytes, not including **cb**. This value MUST be 10.

shd (10 bytes): A **Shd** structure that specifies the background shading that is applied.

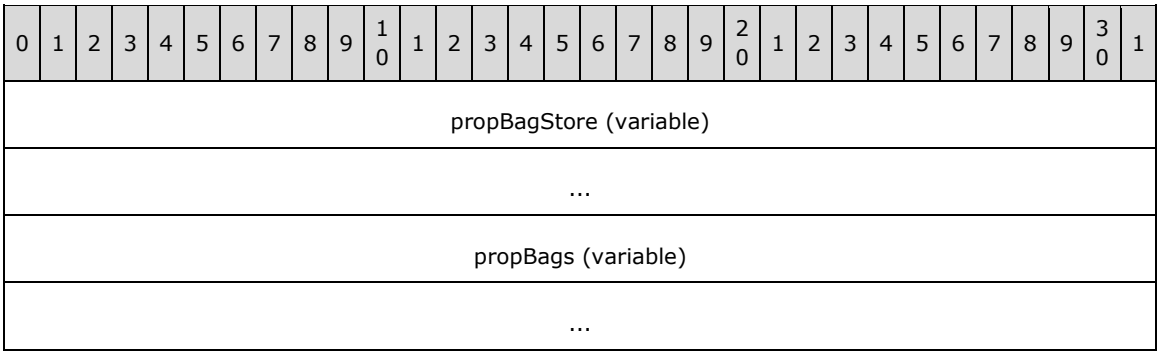
2.9.250 SLncOperand

The **SLncOperand** enumeration is the operand to **sprmSLnc**. This structure is an 8-bit unsigned integer that specifies the line numbering mode for the **section**.

Name	Value	Meaning
lncPerPage	0x00	Line numbers restart every page.
lncRestart	0x01	Line numbers restart at the beginning of the section.
lncContinue	0x02	Line numbers continue from the preceding section, or start at 1 if this is the first section of the document.

2.9.251 SmartTagData

The **SmartTagData** structure stores information about all the **smart tags** in the document. The location of each smart tag is specified by the **fcPlcfBkfFactoid** and **lcbPlcfBkfFactoid** members of the **FibRgFcLcb2002**.



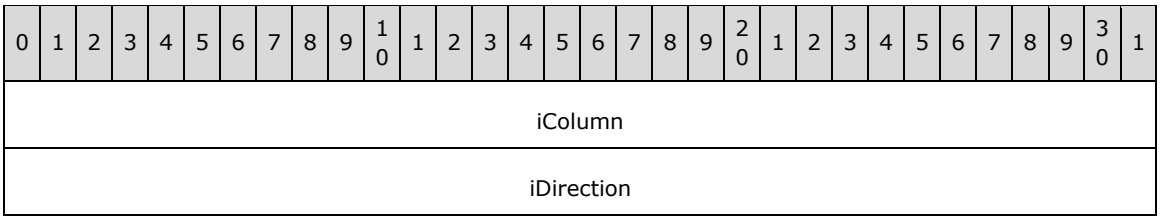
propBagStore (variable): A **PropertyBagStore**, as specified in [MS-OSHARED] section 2.3.4.1.

propBags (variable): An array of **PropertyBag** structures, as specified in [MS-OSHARED] section 2.3.4.3. The size of this array, in bytes, is determined by subtracting the size of **propBagStore** from the **lcbFactoidData** member of FibRgFcLcb2002.

Copyright © 2025 Microsoft Corporation

2.9.252 SortColumnAndDirection

The **SortColumnAndDirection** structure specifies the sort order and the column by which the list of mail merge recipients is sorted.

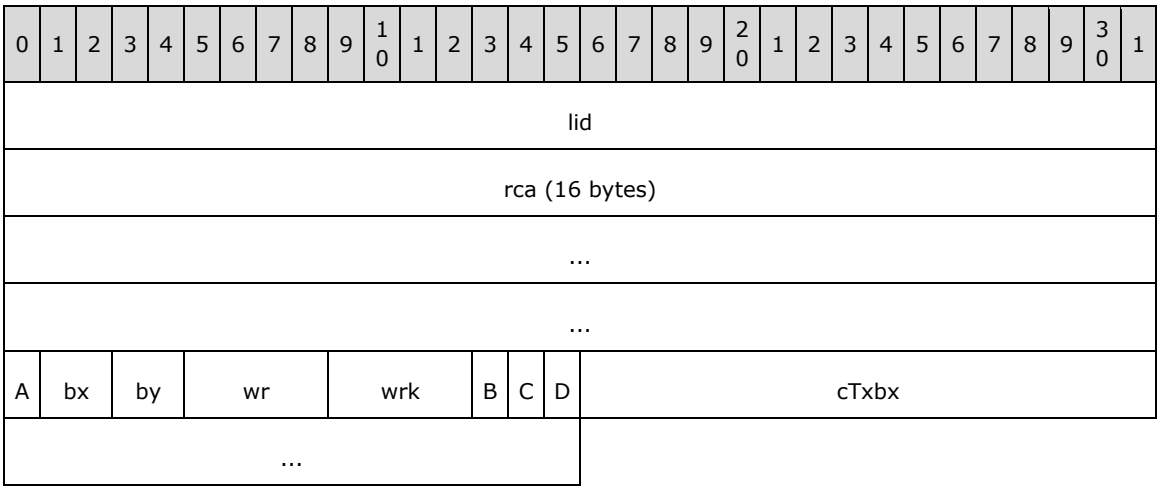


iColumn (4 bytes): An unsigned integer that specifies the zero-based index of the database column to which this filter applies. This value MUST be greater than or equal to zero and MUST be less than or equal to 254.

iDirection (4 bytes): An unsigned integer that specifies the sort order to be used when sorting the associated column. The value MUST be zero (ascending) or 1 (descending).

2.9.253 Spa

The **Spa** structure specifies information about the shapes and drawings that the document contains.



lid (4 bytes): An integer that specifies the identifier of a shape that is contained in the **OfficeArtDggContainer** structure. This value corresponds to the **spid** field of an **OfficeArtFSP** structure that specifies the data for this shape. **OfficeArtDggContainer** and **OfficeArtFSP** are specified in [MS-ODRAW] sections 2.2.12 and 2.2.40, respectively.

rca (16 bytes): An **Rca** structure that specifies the rectangle where the drawing exists. The coordinates of **rca** are in **twips**.

A - fHdr (1 bit): This bit is undefined and MUST be ignored.

bx (2 bits): An unsigned integer that specifies the horizontal position of the origin that is used to calculate the **rca**. This MUST be one of the following values.

Copyright © 2025 Microsoft Corporation

Value	Meaning
0	Anchored at the leading margin of the page.
1	Anchored at the leading edge of the page.
2	Anchored at the leading edge of the column.

by (2 bits): An unsigned integer that specifies the vertical position of the origin that is used to calculate the **rca**. This MUST be one of the following values.

Value	Meaning
0	Anchored at the top margin of the page.
1	Anchored at the top edge of the page.
2	Anchored at the top edge of the paragraph.

wr (4 bits): An unsigned integer that specifies the style of text wrapping around this shape. This MUST be one of the following values.

Value	Meaning
0	Wrap text around the object.
1	No text wrapping around the object. No text appears on either side of the shape (top and bottom wrapping).
2	Wrap text around an absolutely positioned object (square wrapping).
3	Display as if the shape is not there. The shape appears either in front of or behind the text, based on **fBelowText.**
4	Wrap text tightly around this shape, following its contour only on the left and right sides (tight wrapping).
5	Wrap text tightly around this shape, following its contour on all sides (through wrapping).

Copyright © 2025 Microsoft Corporation

wrk (4 bits): An unsigned integer that specifies the details of the text wrapping around this shape. This field MUST be ignored when **wr** is 1 or 3. This MUST be one of the following values.

Value	Meaning
0	Allow text wrapping on both sides of the shape.
1	Allow text wrapping only on the left side of the shape.
2	Allow text wrapping only on the right side of the shape.
3	Allow text wrapping only on the largest side of the shape.

B - fRcaSimple (1 bit): MUST be zero.

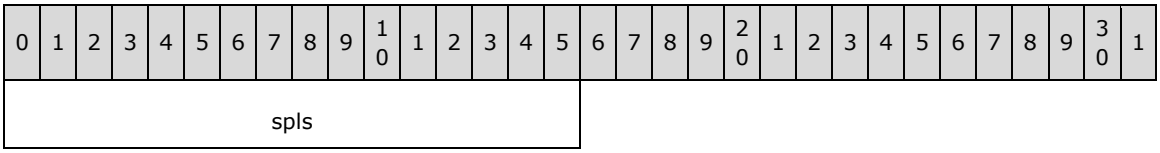
C - fBelowText (1 bit): An unsigned integer that specifies whether this shape is behind the text. A value of 1 specifies that the shape appears behind the paragraph. A value of 0 specifies that the shape appears in front of the text and obscures it. If **wr** is not 3, this field MUST be ignored.

D - fAnchorLock (1 bit): An unsigned integer that specifies whether the anchor of the shape is locked to its current paragraph.

cTxbx (4 bytes): This value is undefined and MUST be ignored.

2.9.254 SpellingSpls

The **SpellingSpls** is an **SPLS** structure that specifies the state of the spell-checker over a range of text. Some states that are possible in a generic **SPLS** are not allowed in a **SpellingSpls** structure.



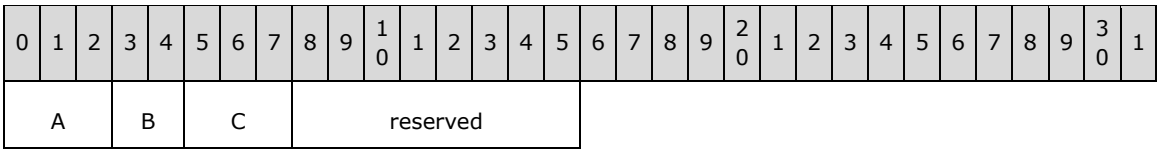
spls (2 bytes): An SPLS structure. The **spls.fExtend** and **spls.fTypo** fields are not used and MUST be zero. The **spls.splf** field MUST be one of the following:

- splfMaybeDirty
- splfDirty
- splfEdit
- splfForeign
- splfClean
- splfRepeatWord
- splfUnknownWord

Copyright © 2025 Microsoft Corporation

2.9.255 SPgbPropOperand

The **SPgbPropOperand** structure is the operand to **sprmSPgbProp**. It specifies the properties of a **page border**.



A - pgbApplyTo (3 bits): A value from the **PgbApplyTo** enumeration that specifies to what pages the border applies.

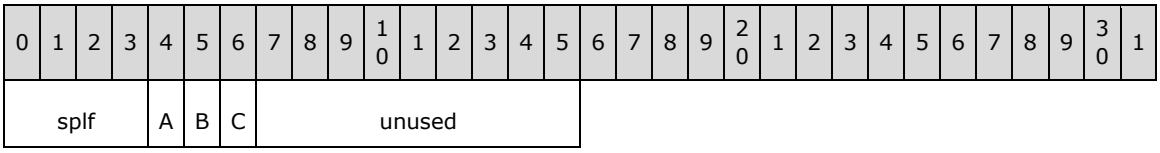
B - pgbPageDepth (2 bits): A value from the **PgbPageDepth** enumeration controlling the "depth" of the border—for example, whether it is above or below other page elements.

C - pgbOffsetFrom (3 bits): A value from the **PgbOffsetFrom** enumeration that specifies from where the offset of the border is measured.

reserved (1 byte): This value MUST be 0.

2.9.256 SPLS

The **SPLS** structure specifies the current state of a range of text with regard to one of the language checking features such as the spell-checker, grammar-checker, language auto-detection, or **smart tag recognizer**.



splf (4 bits): This MUST be one of the following values.

Name	Value	Meaning
splfPending	0x1	Specifies that the text range is currently undergoing checking in another thread. Used only within the **PlcfFactoid** structure. On load, this is converted to **splfDirty**.
splfMaybeDirty	0x2	Specifies that the text range was edited, and could be re-scanned. Having text ranges in the document with this value does not, by itself, cause a new scan. This value MUST only be used in the header document.
splfDirty	0x3	Specifies that the text range was created or changed since the last scan, and that a new scan is needed to evaluate it. Additionally, the **PlcfGram** structure SHOULD<238> use this value for all grammatical errors, in which case **fError** is set to 1.
splfEdit	0x4	Specifies that the text range has been created or changed, and that the user is still editing in the vicinity. A scan is not needed for this text range until the user can be assumed to be finished making the edits.
splfForeign	0x5	Specifies that the text range is a foreign language or phrase. When used by the language auto-detection, the language was explicitly set and no auto-detection is necessary. When used by the spell-checker or grammar-checker, the text range is not subject to further checking.
splfClean	0x7	Specifies that the text range was checked and contains no errors or other special states.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
splfNoLAD	0x8	Specifies that the text range is to be skipped by language auto-detection. Used only within **Plcflad**.
splfErrorMin	0xA	Specifies that the text range contains an error.
splfRepeatWord	0xB	Specifies that the text range contains a word or phrase that duplicates a preceding word or phrase. It is an error.
splfUnknownWord	0xC	Specifies that the text range contains a word that is unknown to the language checker. It is an error.

A - fError (1 bit): The range is an error. This bit MUST be set when the **splf** value is **splfErrorMin**, **splfRepeatWord**, or **splfUnknownWord**. It can also be set when the **splf** value is **splfDirty** or **splfEdit**, which both indicate that the range is currently an error but is still subject to further checking. This bit MUST NOT be set for any other **splf** value.

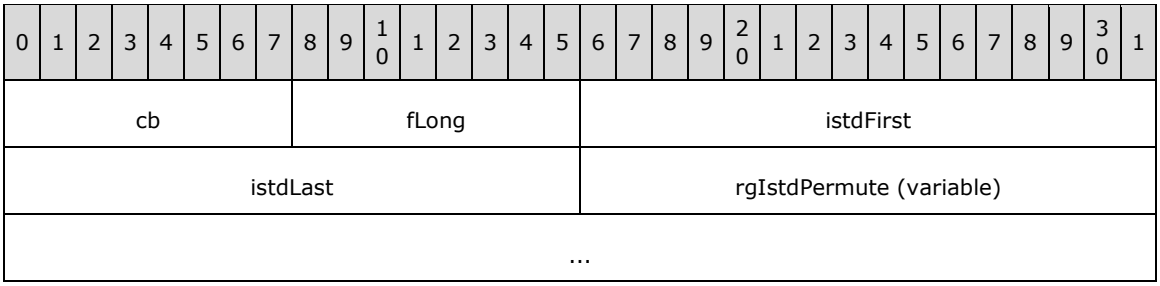
B - fExtend (1 bit): The range is an error. When rechecked, the surrounding text is also rechecked.

C - fTypo (1 bit): The range is a spelling error that was caught by a grammar-checker.

unused (9 bits): This field is not used. This value MUST be 0.

2.9.257 SPPOperand

The **SPPOperand** structure specifies a potential change in the current style as specified by an **istd** value. A given **istd** is affected only if it is within the **istdFirst** and **istdLast** bounds (inclusive). If the **istd** is affected, the new **istd** is **rgIstdPermute**[istd – **istdFirst**].



cb (1 byte): An unsigned 8-bit integer that specifies the size, in bytes, of this **SPPOperand** structure, excluding the **cb** member.

fLong (1 byte): This value MUST be 0 and MUST be ignored.

istdFirst (2 bytes): An unsigned 16-bit integer that specifies the first **istd** to which this change applies.

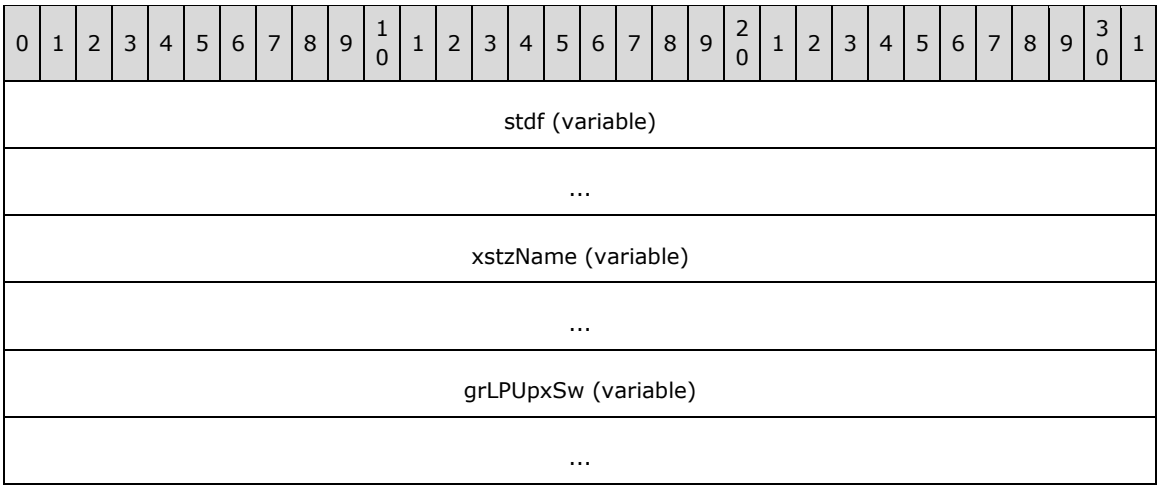
istdLast (2 bytes): An unsigned 16-bit integer that specifies the last **istd** to which this change applies. This value MUST be greater than or equal to **istdFirst**.

rgIstdPermute (variable): An array of unsigned 16-bit integers that specifies an array of remapped **istd** values. The count of elements MUST be equal to **istdLast** – **istdFirst** + 1.

2.9.258 STD

The **STD** structure specifies a style definition.

Copyright © 2025 Microsoft Corporation



stdf (variable): An **Stdf** that specifies basic information about the style.

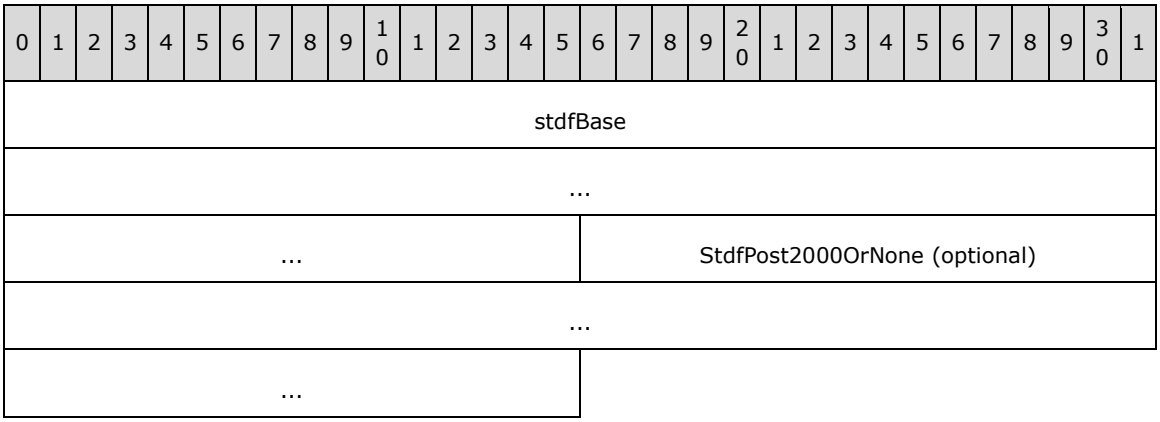
xstzName (variable): An **Xstz** structure that specifies the primary style name followed by any alternate names (aliases), with meaning as specified in [ECMA-376] part 1, section 17.7.4.9 (name) and [ECMA-376] part 1, section 17.7.4.1 (aliases). The primary style name and any alternate style names are combined into one string, with a comma character (U+002C) separating the primary style name and any alternate style names. If there are no alternate style names, the trailing comma is omitted.

Each name, whether primary or alternate, MUST NOT be empty and MUST be unique within all names in the stylesheet.

grLPUpxSw (variable): A **GrLPUpxSw** structure that specifies the formatting for the style.

2.9.259 Stdf

The **Stdf** structure specifies general information about the style.



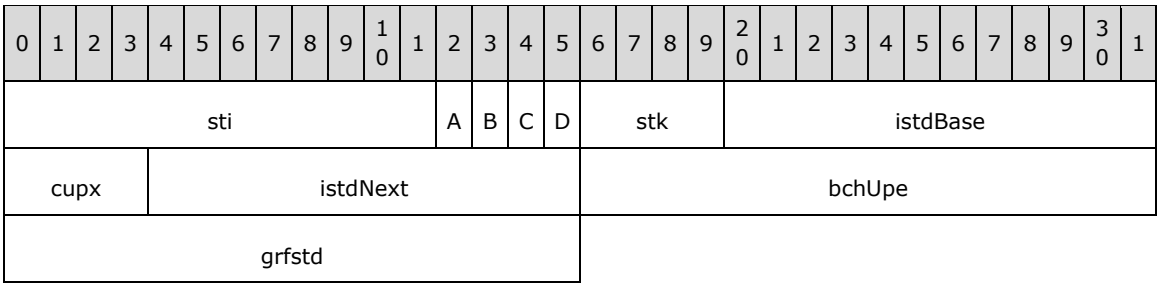
stdfBase (10 bytes): An **StdfBase** structure that specifies general information about the style.

StdfPost2000OrNone (8 bytes): An **StdfPost2000OrNone** that specifies general information about the style.

2.9.260 StdfBase

The **StdfBase** structure specifies general information about a style.

Copyright © 2025 Microsoft Corporation



sti (12 bits): An unsigned integer that specifies the invariant style identifier for application-defined styles, or 0x0FFE for user-defined styles.

The **sti** identifies which styles in the stylesheet correspond to which application-defined styles. An application-defined style can have different names in different languages, but it MUST have the same **sti** value regardless of language. The **sti** values correspond to the "Index within Built-in Styles" table column that is specified in [ECMA-376] part 1, section 17.7.4.9 (name).

A - fScratch (1 bit): This bit is undefined and MUST be ignored.

B - fInvalHeight (1 bit): Specifies whether the paragraph height information in the **fcPlcfPhe** field of FibRgFcLcb97, for any paragraphs having this paragraph style, MUST be ignored. SHOULD<239> be 0.

C - fHasUpe (1 bit): This bit is undefined and MUST be ignored.

D - fMassCopy (1 bit): This bit is undefined and MUST be ignored.

stk (4 bits): An unsigned integer that specifies the type of this style, which corresponds to the "type" attribute of the style element as specified in [ECMA-376] part 1, section 17.7.4.17 (Style Definition). This MUST be one of the following values:

Value	Meaning
1	Paragraph style, as specified by the "paragraph" value in [ECMA-376] part 1, section 17.18.83 (ST_StyleType).
2	Character style, as specified by the "character" value in [ECMA-376] part 1, section 17.18.83 (ST_StyleType).
3	Table style, as specified by the "table" value in [ECMA-376] part 1, section 17.18.83 (ST_StyleType).
4	Numbering style, as specified by the "numbering" value in [ECMA-376] part 1, section 17.18.83 (ST_StyleType).

istdBase (12 bits): An unsigned integer that specifies the **istd** (see the **rglpstd** array in the **STSH** structure) of the parent style from which this style inherits in the style inheritance tree, or 0x0FFF if this style does not inherit from any other style in the current document. The meaning of the parent style is specified in the basedOn element in [ECMA-376] part 1, section 17.7.4.3. However, the style reference in that specification is a style identifier rather than an **istd**, and an **istdBase** value of 0x0FFF corresponds to omitting the basedOn element.

Copyright © 2025 Microsoft Corporation

The **istdBase** value MUST be an index that refers to a valid non-empty style in the array of style definitions. The **istdBase** value MUST NOT be the same as the **istd** of the current style and MUST NOT cause a loop in the style inheritance tree.

cupx (4 bits): An unsigned integer that specifies the count of formatting sets inside the structure, specified to style type, that is contained in the **GrLPUpxSw**.

Each type of style contains a different structure within **GrLPUpxSw**, as shown in the following table. The **cupx** value specifies the count of structures within the structure that is contained in the **GrLPUpxSw**. For each type of style, the **cupx** MUST be equal to the values that are shown in the table, depending on whether the style is revision-marked (in a revision-marked style the **fHasOriginalStyle** value in **StdfPost2000** is 1; in a non-revision-marked style the value is 0.)

Table and numbering styles MUST NOT be revision-marked.

stk value	GrLPUpxSw contains	cupx for non-revision-marked style	cupx for revision-marked-style
1 (paragraph)	StkParaGRLPUPX	2	3
2 (character)	StkCharGRLPUPX	1	2
3 (table)	StkTableGRLPUPX	3	N/A
4 (numbering)	StkListGRLPUPX	1	N/A

istdNext (12 bits): An unsigned integer that specifies the **istd** (see **rglpstd** in **STSH**) of the style which is automatically applied to a new paragraph created following a paragraph with the current style, as specified in more detail in [ECMA-376] part 1, section 17.7.4.10 (next). However, the style reference in that specification is a style identifier rather than an **istd**.

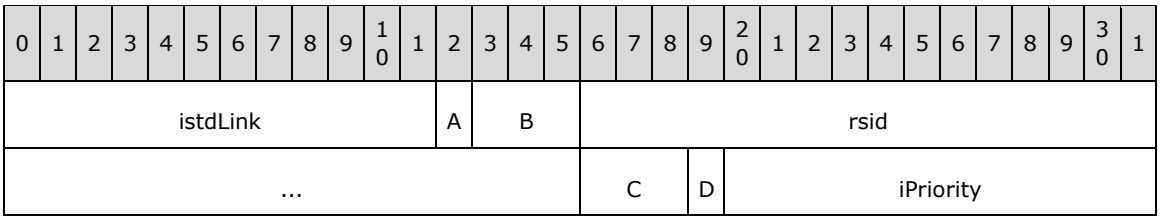
The **istdNext** value MUST be an index that refers to a valid non-empty style in the array of style definitions.

bchUpe (2 bytes): An unsigned integer that specifies the size, in bytes, of **std** in **LPStd**. This value MUST be equal to **cbStd** in **LPStd**.

grfstd (2 bytes): A **GRFSTD** that specifies miscellaneous style properties.

2.9.261 StdfPost2000

The **StdfPost2000** structure specifies general information about a style.



istdLink (12 bits): An unsigned integer that specifies the **istd** of the style that is linked to this one, or 0x0000 if this style is not linked to any other style in the document. The meaning of a linked style is as specified in [ECMA-376] part 1, section 17.7.4.6 (link). However, the style reference in that specification is a style identifier rather than an **istd**, and an **istdLink** value of 0x0000 corresponds to omitting the link element.

The **istdLink** value MUST be an index that refers to a valid non-empty style in the array of style definitions, or 0x0000.

Copyright © 2025 Microsoft Corporation

A - fHasOriginalStyle (1 bit): Specifies whether the style is revision-marked. A revision-marked style stores the pre-revision formatting in addition to the current formatting. If this bit is set to 1, the **cupx** member of **StdfBase** MUST include the formatting sets that specify that pre-revision formatting.

B - fSpare (3 bits): This value MUST be 0 and MUST be ignored.

rsid (4 bytes): An unsigned integer that specifies the revision save identifier of the session when this style definition was last modified, as specified in [ECMA-376] part 1, section 17.7.4.15 (rsid).

C - iftcHtml (3 bits): This field is undefined and MUST be ignored.

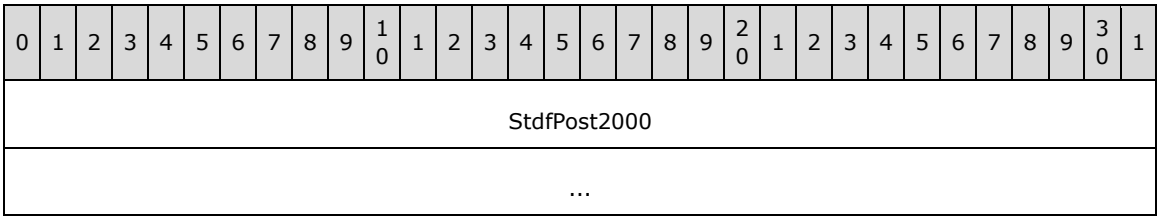
D - unused (1 bit): This value MUST be 0 and MUST be ignored.

iPriority (12 bits): An unsigned integer that specifies the priority value that is assigned to this style and that is used when ordering the styles by priority in the user interface, as specified in [ECMA-376] part 1, section 17.7.4.19 (uiPriority).

This MUST be a value between 0x0000 and 0x0063, inclusive.

2.9.262 StdfPost2000OrNone

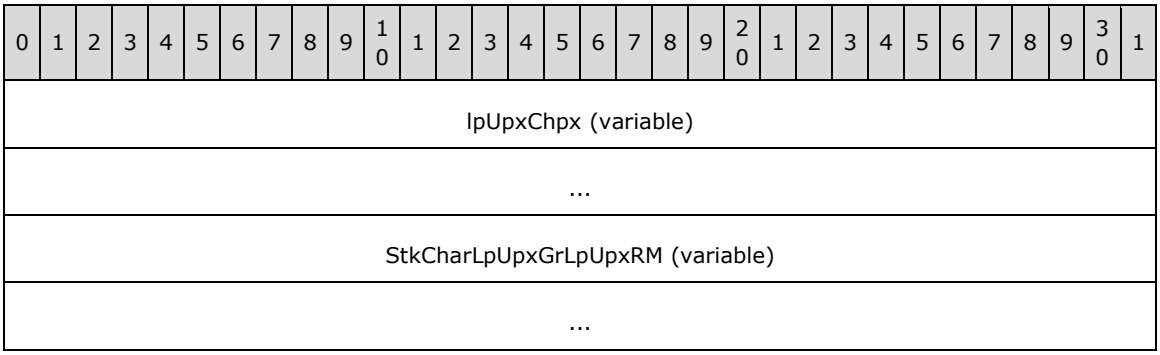
The **StdfPost2000OrNone** structure specifies general information about a style.



StdfPost2000 (8 bytes): An **StdfPost2000** structure that specifies general information about the style. This field is optional; **Stshif**.**cbSTDBaseInFile** defines whether it is included or not.

2.9.263 StkCharGRLPUPX

The **StkCharGRLPUPX** structure specifies the formatting properties for a character style. All members of **StkCharGRLPUPX** are optional, but those that are present MUST appear in the order that is specified in the following table. Additionally, the number of members that are present MUST match the **cupx** member of **StdfBase** for the style.



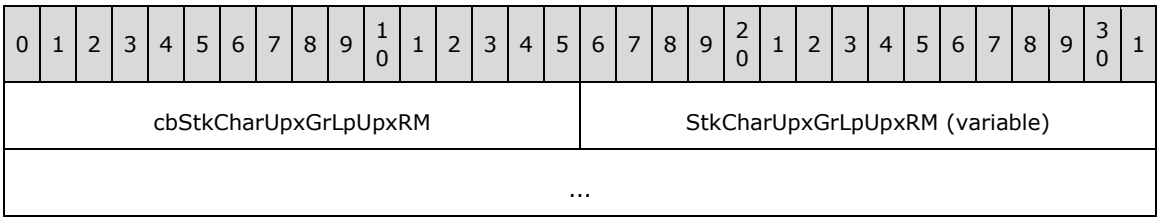
lpUpxChpx (variable): A **LPUpxChpx** that specifies the character formatting properties for the style.

Copyright © 2025 Microsoft Corporation

StkCharLpUpxGrLpUpxRM (variable): A **StkCharLPUpxGrLPUpxRM** that specifies the revision-marking information and formatting for the style.

2.9.264 StkCharLPUpxGrLPUpxRM

The **StkCharLPUpxGrLPUpxRM** structure specifies revision-marking information and formatting for character styles. The structure is padded to be an even length. The length in **cbStkCharUpxGrLpUpxRM** MUST include this padding.

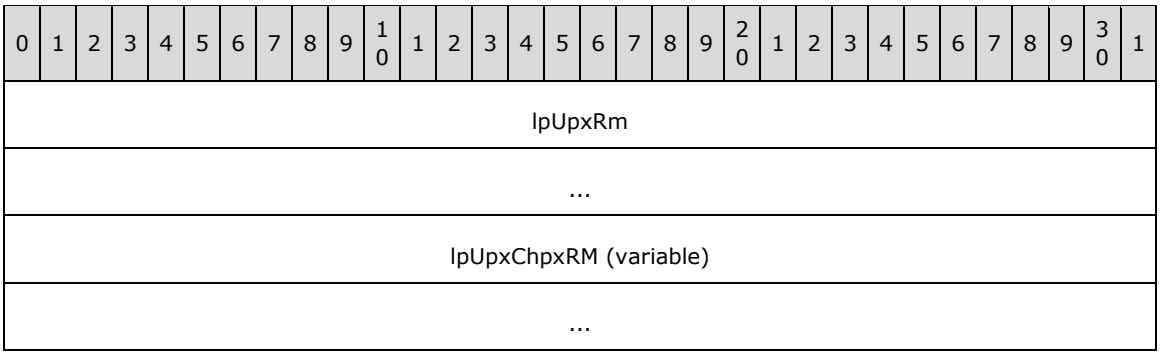


cbStkCharUpxGrLpUpxRM (2 bytes): An unsigned 16-bit integer that specifies the size, in bytes, of **StkCharUpxGrLpUpxRM**. This field MUST include padding if it is needed to make **StkCharLPUpxGrLPUpxRM** an even length.

StkCharUpxGrLpUpxRM (variable): A StkCharUpxGrLPUpxRM that specifies revision-marking information and formatting.

2.9.265 StkCharUpxGrLPUpxRM

The **StkCharUpxGrLPUpxRM** structure specifies revision-marking information and formatting for character styles.



lpUpxRm (8 bytes): An **LPUpxRm** structure that specifies the revision-marking information for the style.

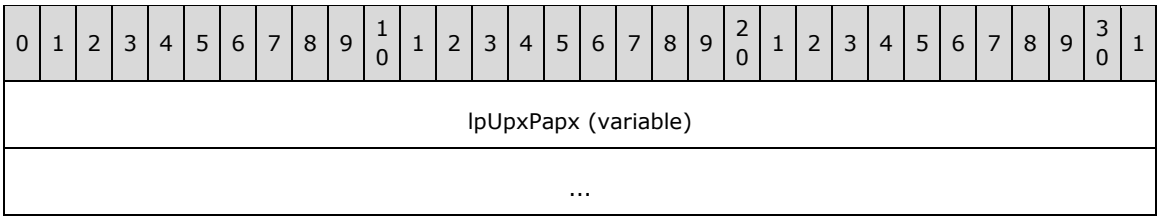
lpUpxChpxRM (variable): An **LPUpxChpxRM** that specifies the character formatting properties for the revision-marked style formatting.

2.9.266 StkListGRLPUPX

The **StkListGRLPUPX** structure specifies formatting properties for a numbering style.

Each set of formatting properties is a length-prefixed variable-length structure. All members of **StkListGRLPUPX** are optional, but those that are present MUST appear in the order that is specified in the following table. Additionally, the number of members that are present MUST match the **cupx** member of **StdfBase** for the style.

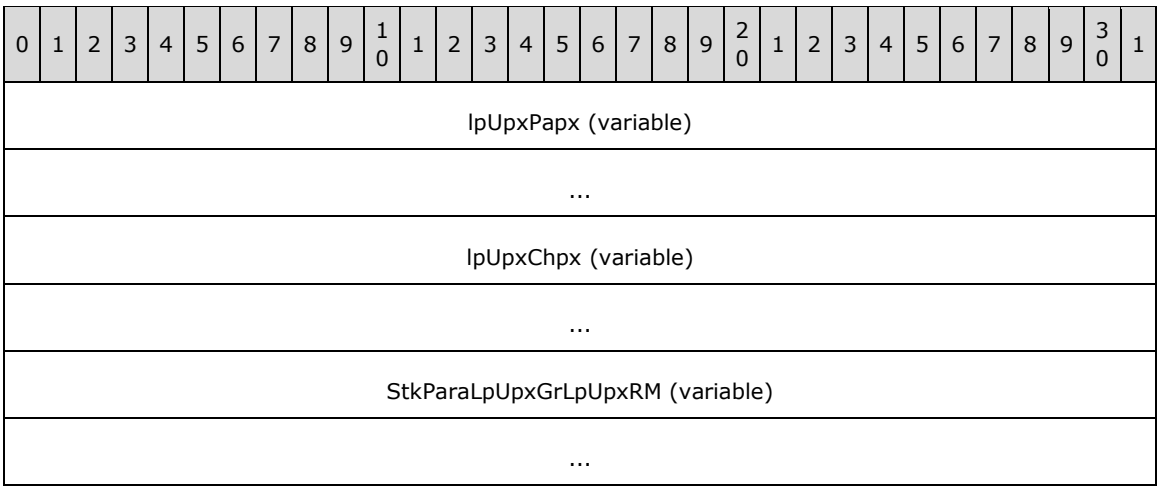
Copyright © 2025 Microsoft Corporation



lpUpxPapx (variable): An **LPUpxPapx** that specifies the paragraph formatting properties for the style.

2.9.267 StkParaGRLPUPX

The **StkParaGRLPUPX** structure that specifies the formatting properties for a paragraph style. All members of **StkParaGRLPUPX** are optional, but those that are present MUST appear in the order that is specified in the following table. Additionally, the number of members that are present MUST match the **cupx** member of **StdfBase** for the style.



lpUpxPapx (variable): A **LPUpxPapx** that specifies the paragraph formatting properties for the style.

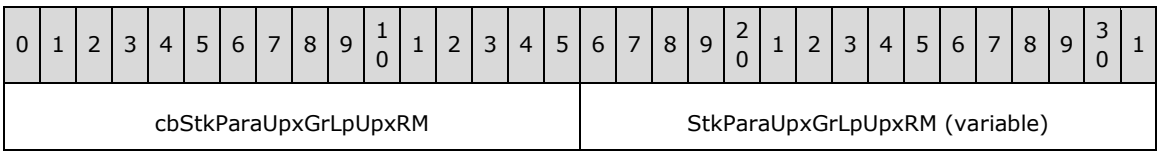
lpUpxChpx (variable): A **LPUpxChpx** that specifies the character formatting properties for the style.

StkParaLpUpxGrLpUpxRM (variable): A **StkParaLPUpxGrLPUpxRM** that specifies the revision-marking information and formatting for the style.

2.9.268 StkParaLPUpxGrLPUpxRM

The **StkParaLPUpxGrLPUpxRM** structure specifies revision-marking information and formatting for paragraph styles. This structure is length-prefixed and of variable length.

The structure is padded to be an even length.



Copyright © 2025 Microsoft Corporation

...

cbStkParaUpxGrLpUpxRM (2 bytes): An unsigned 16-bit integer that specifies the size, in bytes, of **StkParaUpxGrLpUpxRM**, including the padding.

StkParaUpxGrLpUpxRM (variable): An **StkParaUpxGrLPUpxRM** structure that specifies revision-marking information and formatting.

2.9.269 StkParaUpxGrLPUpxRM

The **StkParaUpxGrLPUpxRM** structure specifies style revision-marking and formatting for paragraph styles.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
lpUpxRm																															
...																															
lpUpxPapxRM (variable)																															
...																															
lpUpxChpxRM (variable)																															
...																															

lpUpxRm (8 bytes): An **LPUpxRm** structure that specifies the revision-marking information for the style.

lpUpxPapxRM (variable): An **LPUpxPapxRM** structure that specifies the paragraph formatting properties for the revision-marked style formatting.

lpUpxChpxRM (variable): An **LPUpxChpxRM** structure that specifies the character formatting properties for the revision-marked style formatting.

2.9.270 StkTableGRLPUPX

The **StkTableGRLPUPX** structure specifies the formatting properties for a table style. This structure is variable in length. All members of **StkTableGRLPUPX** are optional, but those members that are present MUST appear in the order that is specified in the following table. Additionally, the number of members that are present MUST match the **cupx** member of StdfBase for the style.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
lpUpxTapx (variable)																															
...																															
lpUpxPapx (variable)																															

Copyright © 2025 Microsoft Corporation

...
lpUpxChpx (variable)
...

lpUpxTapx (variable): An **LPUpxTapx** that specifies the table formatting properties for the style.

lpUpxPapx (variable): An **LPUpxPapx** that specifies the paragraph formatting properties for the style.

lpUpxChpx (variable): An **LPUpxChpx** that specifies the character formatting properties for the style.

2.9.271 STSH

The **STSH** structure specifies the stylesheet for a document. The stylesheet describes the styles that are available within a document as well as their formatting.

An **istd** is an index into **rglpstd** that is used to reference a particular style definition. The **istd** values are used internally within the stylesheet, such as in the **istdBase** member of the **StdfBase** structure, as well as externally outside the stylesheet, such as in **Sprm** structures such as **sprmPIstd**. An **istd** value MUST be greater than or equal to 0x0000 and less than 0x0FFE.

Each **FIB** MUST contain a stylesheet.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
lpstshi (variable)																															
...																															
rglpstd (variable)																															
...																															

lpstshi (variable): An LPStshi that specifies information about the stylesheet.

rglpstd (variable): An array of **LPStd** that specifies the style definitions.

The beginning of the **rglpstd** array is reserved for specific "fixed-index" application-defined styles. A particular fixed-index, application-defined style has the same istd value in every stylesheet. The **rglpstd** MUST contain an **LPStd** for each of these fixed-index styles and the order MUST match the order in the following table.

istd	sti of application-defined style (see sti in StdfBase)
0	0
1	1
2	2
3	3
4	4
5	5
6	6

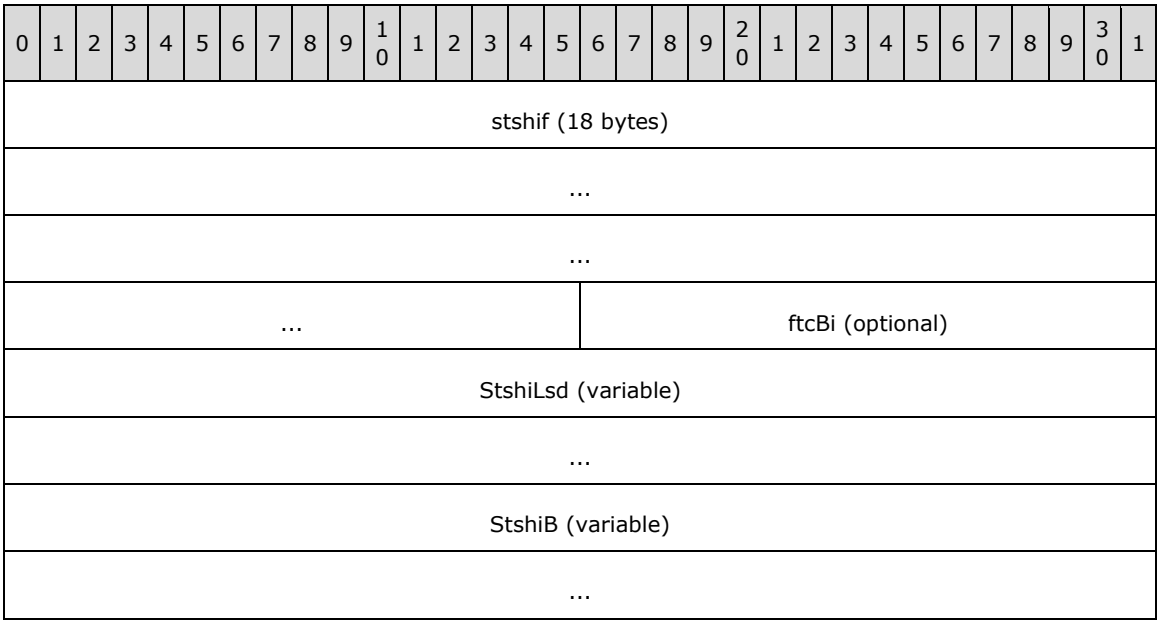
Copyright © 2025 Microsoft Corporation

istd	sti of application-defined style (see sti in StdfBase)
7	7
8	8
9	9
10	65
11	105
12	107
13	*Reserved for future use*
14	*Reserved for future use*

A style is "empty" if the **cbStd** member of the **LPStd** is 0. The fixed-index styles from **istd** 0 to 12 MAY<240> be empty, while those from istd 13 to 14 MUST be empty.

2.9.272 STSHI

The **STSHI** structure specifies general stylesheet and related information.



stshif (18 bytes): An **Stshif** that specifies general stylesheet information.

ftcBi (2 bytes): A signed integer that specifies an operand value for the **sprmCFtcBi** for default document formatting, as defined in the section Determining Formatting Properties.

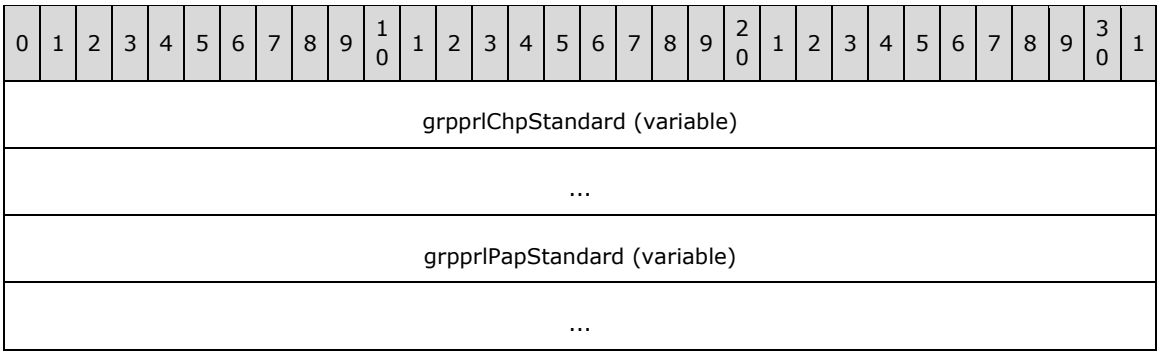
StshiLsd (variable): An **StshiLsd** that specifies latent style data.

StshiB (variable): An **STSHIB**. This MUST be ignored.

2.9.273 STSHIB

The **STSHIB** structure has no effect and MUST be ignored.

Copyright © 2025 Microsoft Corporation

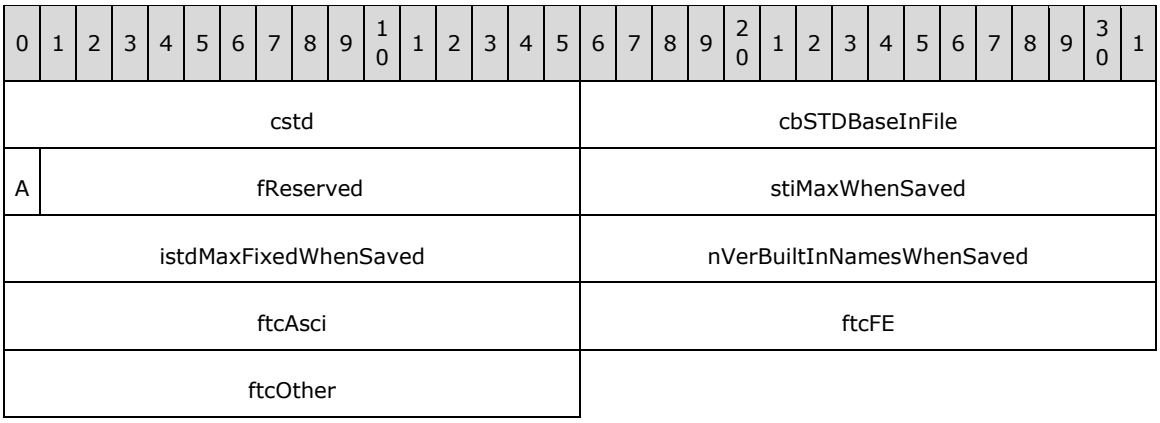


grpprlChpStandard (variable): An **LPStshiGrpPrl** that MUST be ignored.

grpprlPapStandard (variable): An **LPStshiGrpPrl** that MUST be ignored.

2.9.274 Stshif

The **Stshif** structure specifies general stylesheet information.



cstd (2 bytes): An unsigned integer that specifies the count of elements in **STSH.rglpstd**. This value MUST be equal to or greater than 0x000F, and MUST be less than 0x0FFE.

cbSTDBaseInFile (2 bytes): An unsigned integer that specifies the size, in bytes, of the **Stdf** structure. The **Stdf** structure contains an **StdfBase** structure that is followed by a **StdfPost2000OrNone** structure which contains an optional **StdfPost2000** structure. This value MUST be 0x000A when the **Stdf** structure does not contain an **StdfPost2000** structure and MUST be 0x0012 when the **Stdf** structure does contain an **StdfPost2000** structure.

A - fStdStylenamesWritten (1 bit): This value MUST be 1 and MUST be ignored.

fReserved (15 bits): This value MUST be 0 and MUST be ignored.

stiMaxWhenSaved (2 bytes): An unsigned integer that specifies a value that is 1 larger than the largest **StdfBase.sti** index of any application-defined style. This SHOULD<241> be equal to the largest **sti** index that is defined in the application, incremented by 1.

istdMaxFixedWhenSaved (2 bytes): An unsigned integer that specifies the count of elements at the start of **STSH.rglpstd** that are reserved for fixed-index application-defined styles. This value MUST be 0x000F.

nVerBuiltInNamesWhenSaved (2 bytes): An unsigned integer that specifies the version number of the style names as defined by the application that writes the file. This value SHOULD<242> be 0.

Copyright © 2025 Microsoft Corporation

ftcAsci (2 bytes): A signed integer that specifies an operand value for the **sprmCRgFtc0** for default document formatting, as defined in the section Determining Formatting Properties.

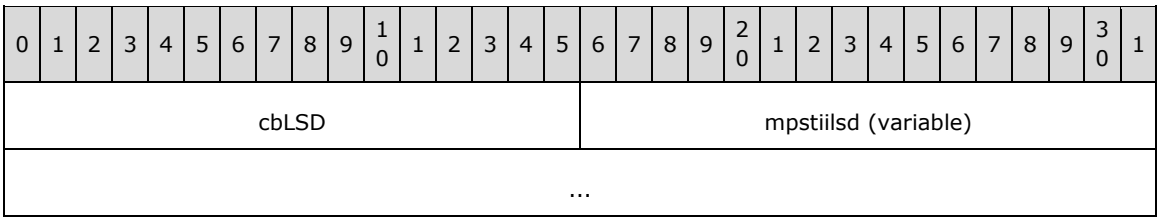
ftcFE (2 bytes): A signed integer that specifies an operand value for the sprmCRgFtc1 for default document formatting, as defined in the section Determining Formatting Properties.

ftcOther (2 bytes): A signed integer that specifies an operand value for the sprmCRgFtc2 for default document formatting, as defined in the section Determining Formatting Properties.

2.9.275 StshiLsd

The **StshiLsd** structure specifies latent style data for application-defined styles. Application-defined styles are considered to be latent if they have an **LPStd** that is 0x0000 in **STSH.rglpstd** or if they have no corresponding **LPStd** in **STSH.rglpstd**. (For example, if an application has a built-in definition for a "Heading 1" style but that style is not currently defined in a document stylesheet, that style is considered latent.) Latent style data specifies a default set of behavior properties to be used when latent styles are first created.

The index into **mpstiilsd** is the **sti** value (in the **StdfBase** structure) of the application-defined style to which it applies. An **LSD** structure MUST be provided for every application-defined style with **sti** values from 0 to one less than **stiMaxWhenSaved** (in the **Stshif** structure), regardless of whether those application-defined styles are currently latent or not.

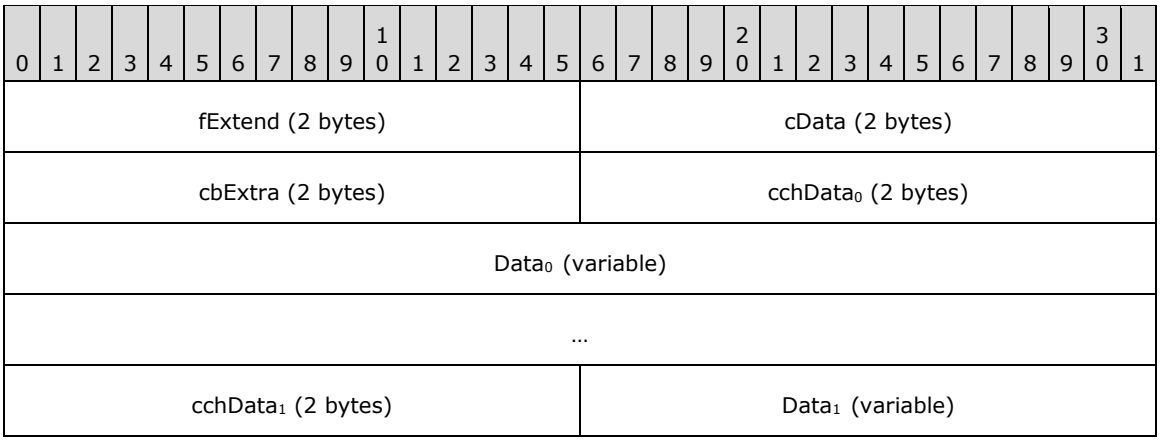


cbLSD (2 bytes): An unsigned 16-bit integer that specifies the size in bytes of the **LSD** structure. This value MUST be 4.

mpstiilsd (variable): An array of **LSD** structures that specifies the latent style data for application-defined styles. The count of elements MUST be equal to the **stiMaxWhenSaved** member of the **Stshif** structure.

2.9.276 SttbfAssoc

The **SttbfAssoc** structure is an **STTB** that contains strings which are associated with this document.



Copyright © 2025 Microsoft Corporation

…	
…	cchData$_{cData-1}$ (2 bytes)
Data$_{cData-1}$ (variable)	
…	

This **STTB** MUST contain 18 strings. No extra data is appended to the strings of this **STTB**. Unless otherwise noted, each string in this **STTB** MUST contain no more than 255 characters. The indexes and meanings of these strings are as follows.

Index	Meaning
0x00	Unused. MUST be ignored.
0x01	The path of the associated **document template**, if it is not the default **Normal template**.
0x02	The title of the document. This MUST be ignored if title information, as specified in [MS-OLEPS] section 3.1.2, exists in the Summary Information Stream.
0x03	The subject of the document. This MUST be ignored if subject information, as specified in [MS-OLEPS] section 3.1.3, exists in the Summary Information Stream.
0x04	Key words associated with the document. This MUST be ignored if key word information, as specified in [MS-OLEPS] section 3.1.5, exists in the Summary Information Stream.
0x05	Unused. This index MUST be ignored.
0x06	The author of the document. This index MUST be ignored if author information, as specified in [MS-OLEPS] section 3.1.4, exists in the Summary Information Stream.
0x07	The user who last revised the document. This index MUST be ignored if last author information, as specified in [MS-OLEPS] section 3.1.8, exists in the Summary Information Stream.
0x08	The path of the associated **mail merge data source**.
0x09	The path of the associated **mail merge header document**.
0x0A	Unused. This index MUST be ignored.
0x0B	Unused. This index MUST be ignored.
0x0C	Unused. This index MUST be ignored.
0x0D	Unused. This index MUST be ignored.
0x0E	Unused. This index MUST be ignored.
0x0F	Unused. This index MUST be ignored.
0x10	Unused. This index MUST be ignored.
0x11	The **write-reservation password** of the document. This value MUST NOT exceed 15 characters in length.

The **SttbfAssoc** structure is an **STTB** structure that has the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cData (2 bytes): This value MUST be 0x0012.

cbExtra (2 bytes): This value MUST be 0.

2.9.277 SttbfAtnBkmk

The **SttbfAtnBkmk** structure is an **STTB** whose strings are all of zero length. The **cData** field size of this **STTB** is two bytes. Although this **STTB** contains no strings, it is an extended **STTB**, meaning that its **cchData** field size is two bytes. The extra data that is appended to each string of this **STTB** is an

Copyright © 2025 Microsoft Corporation

ATNBE which contains information about an **annotation bookmark** in the document. In a document, the number of annotation bookmarks MUST NOT exceed 0x3FFB.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$ExtraData_0$ (10 bytes)																															
...																															
...																$cchData_1$ (2 bytes)															
$ExtraData_1$ (10 bytes)																															
...																															
...																...															
$cchData_{cData-1}$ (2 bytes)																$ExtraData_{cData-1}$ (10 bytes)															
...																															
...																															

The **SttbfAtnBkmk** structure is an **STTB** structure that has the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0xA.

cData (2 bytes): This value MUST NOT exceed 0x3FFC.

cchData (2 bytes): This value MUST be 0.

2.9.278 SttbfAutoCaption

The **SttbfAutoCaption** structure is an **STTB** that contains **AutoCaption** information. Each string is the **ProgID** of an **OLE object** that, when inserted into the document, automatically has a caption inserted with it. The extra data which is appended to each string is an unsigned 16-bit integer that specifies a zero-based index into **SttbfCaption**. The data at that index defines the **caption** that is inserted.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															

Copyright © 2025 Microsoft Corporation

$Data_0$ (variable)
…
$ExtraData_0$ (2 bytes) | $cchData_1$ (2 bytes)
$Data_1$ (variable)
…
$ExtraData_1$ (2 bytes) | …
$cchData_{cData-1}$ (2 bytes) | $Data_{cData-1}$ (variable)
…
$ExtraData_{cData-1}$ (2 bytes)

The **SttbfAutoCaption** structure is an **STTB** structure that has following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x0002.

2.9.279 SttbfBkmk

The **SttbfBkmk** structure is an **STTB** structure whose strings specify the names of **bookmarks** in the document. The **cData** field size of this **STTB** structure is 2 bytes. The strings of this **STTB** contain extended (2-byte) characters, and there is no extra data appended to them—in other words, it is equivalent to an **SttbfBkmkBPRepairs** structure. The names in this table that begin with the Unicode character 0x005F correspond to hidden bookmarks. The strings in this table MUST be greater than 0 and less than 40 characters in length. The strings in this table MUST be unique, and there MUST NOT be more than 0x3FFB of them.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (variable)																															
…																															
$cchData_1$ (2 bytes)																$Data_1$ (variable)															
…																															

Copyright © 2025 Microsoft Corporation

…	cchData$_{cData-1}$ (2 bytes)
Data$_{cData-1}$ (variable)	
…	

The **SttbfBkmk** structure is an **STTB** structure with the following additional restrictions on its field values:

fExtend (2 bytes): MUST be 0xFFFF.

cData (2 bytes): MUST NOT exceed 0x3FFC.

cbExtra (2 bytes): MUST be 0.

cchData (2 bytes): MUST NOT exceed 40.

Data (variable): For the purpose of achieving the correct definition of "skip character", the following constraints MUST be evaluated using delayed evaluation and examination of characters in a string MUST take place in first-to-last order. Delayed evaluation requires that each constraint not be read until the result of that constraint is needed. For example, application of the following algorithm to the string "Abc" will never require reading of the constraints defining a single byte Katakana character.

To be a valid member of SttbfBkmk, all characters in the string that are not preceded by a skip character SHOULD<243> meet all of the following constraints:

- Is the first character of the name and satisfies all of the following constraints:
  - Is not **Unicode** character 0x3000.
  - Is not a double-byte digit, meaning that it is between 0xFF10 and 0xFF19, inclusive.
  - Is one of the following:
    - An alpha character, as defined later.
    - The hidden **bookmark** character, 0x005F.
    - A single-byte Katakana character, meaning that it is between 0xFF61 and 0xFF9F, inclusive.
    - A far-east, double-byte text character as defined later.
- Is not the first character of the name and satisfies all of the following constraints:
  - Is not Unicode character 0x3000.
  - Is one of the following:
    - An East Asian, double-byte text character as defined later.
    - An alpha character as defined later.
    - A digit character as defined later.
    - The hidden **bookmark** character, 0x005F.
    - A single-byte Katakana character, meaning it is between 0xFF61 and 0xFF9F, inclusive.

Copyright © 2025 Microsoft Corporation

A digit character is defined as that which satisfies both of the following constraints:

- Is not 0xFFFF.
- Satisfies one of the following constraints:
  - Is between 0x0030 and 0x0039, inclusive.
  - Is between 0xFF10 and 0xFF19, inclusive.
  - Is between 0x0E50 and 0x0E59, inclusive.
  - Is between 0x0966 and 0x096F, inclusive.
  - Is between 0x0F18 and 0x0F19, inclusive.
  - Is between 0x0F20 and 0x0F33, inclusive.
  - Is between 0x0F3E and 0x0F3F, inclusive.
  - Is between 0x0ED0 and 0x0ED9, inclusive.
  - Is between 0x17E0 and 0x17F9, inclusive.

A bidirectional alpha character is defined as a character that satisfies one of the following constraints:

- Is 0x067E or 0x0686 or 0x0698 or 0x06AF or 0x05C4.
- Is between 0x0621 and 0x0652, inclusive.
- Is between 0x05D0And 0x05EA, inclusive.
- Is between 0x05B0 and 0x05B9, inclusive.
- Is between 0x05BB and 0x05C2, inclusive.
- Is between 0x05F0 and 0x05F2, inclusive.
- Is between 0x0591 and 0x05A1, inclusive.
- Is between 0x05A3 and 0x05AF, inclusive.
- Is between 0x0710 and 0x072C, inclusive.
- Is between 0x0730 and 0x073F, inclusive.
- Is any linguistic character in a right-to-left alphabet.

An alpha character is defined as that which satisfies one of the following constraints:

- Is between 'a' and 'z', inclusive.
- Is between 'A' and 'Z', inclusive.
- Is an uppercase or lowercase character in a left-to-right, non-East Asian alphabet.
- Is a Hangul compatibility Jamo, meaning between 0x3131 and 0x318E, inclusive.
- Is a Hangul Jamo, meaning between 0xAC00 and 0xD7A3, inclusive.
- Is a Kanji character, meaning that it is 0x3005 or 0x3007 or between 0x4E00 and 0x9FFF, inclusive, or the Unicode sub-range of the character is either CJK Compatibility Ideographs or CJK Unified Ideographs Extension A.

Copyright © 2025 Microsoft Corporation

- Is not a character that satisfies the definition of a digit given earlier, and satisfies one of the following constraints:
  - Is not 0x1780 and the top 2 bytes of the character are 0x900, 0xE00, 0xF00 or 0x1700 and satisfies one of the following constraints:
    - Is between 0x901 and 0x939, inclusive.
    - Is 0x93D.
    - Is between 0x93E and 0x94D, inclusive.
    - Is between 0x950 and 0x963, inclusive.
    - Is between 0x966 and 0x96F, inclusive.
    - Is between 0x0E01 and 0x0E2E, inclusive.
    - Is between 0x0E30 and 0x0E3A, inclusive.
    - Is between 0x0E40 and 0x0E4C, inclusive.
    - Is between 0x0E50 and 0x0E59, inclusive.
    - Is between 0x0E5A and 0x0E5B, inclusive.
    - Is between 0x0E80 and 0x0ECD, inclusive.
    - Is between 0x0EDC and 0x0EDD, inclusive.
    - Is between 0x0F00 and 0x0F07, inclusive.
    - Is between 0x0F15 and 0x0F17, inclusive.
    - Is between 0x0F1A and 0x0F1F, inclusive.
    - Is between 0x0F34 and 0x0F3D, inclusive.
    - Is between 0x0F40 and 0x0FCF, inclusive.
    - Is between 0x1780 and 0x17DD, inclusive.
  - Satisfies all of the following:
    - The top 2 bytes of the character are not 0x900, 0xE00, 0xF00 or 0x1700.
    - Is a Unicode 3 South Asian character—meaning that it is less than or equal to 0x900 and satisfies one of the following:
    - Is less than or equal to 0x109F.
    - Is between 0x1780 and 0x19FF, inclusive.
    - Is any linguistic character in a left-to-right, non-East Asian language.
- Satisfies the definition of bidirectional alpha character that was given earlier.
- Is a Vietnamese tonemark, meaning it is one of the following: 0x0300, 0x0301, 0x0303, 0x0309, or 0x0323.
- Is a low surrogate character, meaning that it is between 0xDC00 and 0xDFFF, inclusive.
- Is a high surrogate character, meaning that it is between 0xD840 and 0xD869, inclusive.

Copyright © 2025 Microsoft Corporation

- Is between 0xA000 and 0xA4C6, inclusive.

  An East Asian double-byte text character is defined as that which satisfies one of the following constraints:

- Is between 0x3000 and 0x4DB5, inclusive.
- Is between 0x1100 and 0x11F9, inclusive.
- Is between 0xAC00 and 0xD7A3, inclusive.
- Is between 0x4E00 and 0x9FFF, inclusive.
- Is between 0xE815 and 0xE864, inclusive.
- Is between 0xF900 and 0xFAFF, inclusive.
- Is between 0xFE30 and 0xFE4F, inclusive.
- Is between 0xFF00 and 0xFF5F, inclusive.
- Is between 0xE000 and 0xE7FF, inclusive.
- Is between 0x2460 and 0x24FF, inclusive.
- Is between 0x0080 and satisfies both of the following constraints:
  - Is a high surrogate character, meaning it is between 0xD800 and 0xDBFF, inclusive. If this constraint is reached and satisfied during delayed evaluation of the conditions upon strings in SttbfBkmk, then it is a skip character.
  - Is between 0xD840 and 0xD869, inclusive.
- Is greater than or equal to 0x0080 and satisfies all of the following constraints:
  - Not a high or low surrogate character, where a low surrogate character is defined as between 0xDC00 and 0xDFFF, inclusive.
  - Can be expressed as a multibyte character string in an East Asian code page.

2.9.280 SttbfBkmkBPRepairs

The **SttbfBkmkBPRepairs** structure is an **STTB** structure whose strings specify the descriptions of **repair bookmarks** in the document. The **cData** field size of this **STTB** structure is 2 bytes. The strings of this STTB structure contain extended (two-byte) characters, and there is no extra data appended to them—in other words, it is equivalent to an SttbfBkmk. The strings of this table are not null-terminated. In a document, the number of repair bookmarks MUST NOT exceed 0x7FF0.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (variable)																															
…																															

Copyright © 2025 Microsoft Corporation

$cchData_1$ (2 bytes)	$Data_1$ (variable)
…	
…	$cchData_{cData-1}$ (2 bytes)
$Data_{cData-1}$ (variable)	
…	

The **SttbfBkmkBPRepairs** structure is an **STTB** structure with the following additional restrictions on its field values:

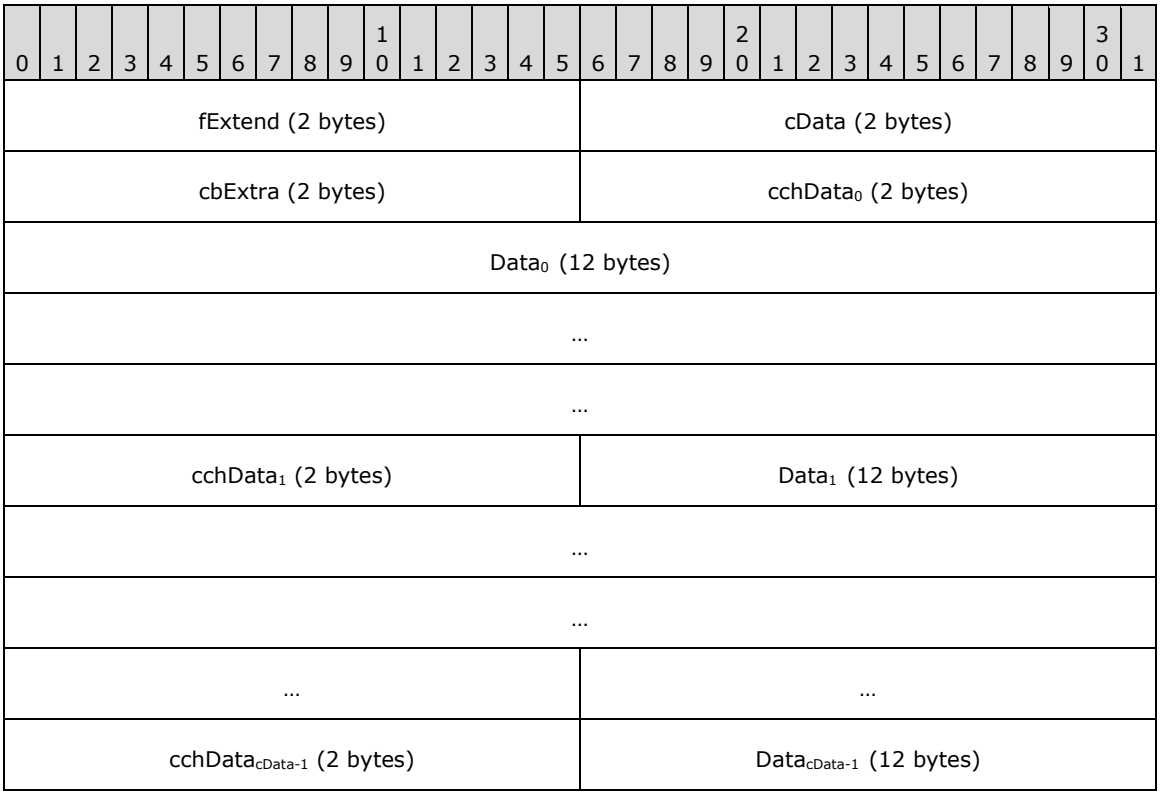
fExtend (2 bytes): This value MUST be 0xFFFF.

cData (2 bytes): This value MUST NOT exceed 0x7FF0.

cbExtra (2 bytes): This value MUST be 0.

2.9.281 SttbfBkmkFactoid

The **SttbfBkmkFactoid** structure is an **STTB** whose strings are **FACTOIDINFO** structures, each of which contains information about a **smart tag bookmark** in the document. The **cData** field size of this **STTB** is 2 bytes. This **STTB** is an extended **STTB**, meaning that its **cchData** field size is 2 bytes. There is no extra data appended to the strings of this **STTB**. In a document, the number of smart tag bookmarks MUST NOT exceed 0x7FF0.



Copyright © 2025 Microsoft Corporation

…
…
…

The **SttbfBkmkFactoid** structure is an STTB structure that has the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cData (2 bytes): This value MUST NOT exceed 0x7FF0.

cbExtra (2 bytes): This value MUST be 0.

cchData (2 bytes): This value MUST be 0x6.

2.9.282 SttbfBkmkFcc

The **SttbfBkmkFcc** structure is an **STTB** whose strings are **DPCID** structures. Each DPCID contains information about a **format consistency-checker bookmark** in the document. The **cData** field size of this **STTB** is 2 bytes. This **STTB** is an extended **STTB**, which means that its **cchData** field size is 2 bytes. There is no extra data appended to the strings of this **STTB**. In a document, the number of format consistency-checker bookmark elements MUST NOT exceed 0x7FF0.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (20 bytes)																															
…																															
…																															
…																															
…																															
$cchData_1$ (2 bytes)																$Data_1$ (20 bytes)															
…																															
…																															
…																															

Copyright © 2025 Microsoft Corporation

…	
…	…
cchData$_{cData-1}$ (2 bytes)	Data$_{cData-1}$ (20 bytes)
…	
…	
…	
…	
…	

The **SttbfBkmkFcc** structure is an **STTB** structure that has the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cData (2 bytes): This value MUST NOT exceed 0x7FF0.

cbExtra (2 bytes): This value MUST be 0.

cchData (2 bytes): This value MUST be 0xA.

2.9.283 SttbfBkmkProt

The **SttbfBkmkProt** structure is an STTB whose strings are all of length zero. The **cData** field of this STTB is four bytes. Although this STTB contains no strings, it is an extended STTB, which means that its **cchData** fields are two bytes in size. The extra data that is appended to each string of this STTB is a PRTI which contains information about the **range-level protection bookmarks** in the document. In a document, the number of range-level protection bookmarks MUST NOT exceed 0x00007FF0.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
fExtend (2 bytes)																cData (4 bytes)															
…																cbExtra (2 bytes)															
cchData$_0$ (2 bytes)																ExtraData$_0$ (8 bytes)															
…																															
…																cchData$_1$ (2 bytes)															
ExtraData$_1$ (8 bytes)																															

Copyright © 2025 Microsoft Corporation

…	
…	cchData$_{cData-1}$ (2 bytes)
ExtraData$_{cData-1}$ (8 bytes)	
…	

The **SttbfBkmkProt** structure is an STTB structure that has the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cData (4 bytes): This value MUST NOT exceed 0x00007FF0.

cbExtra (2 bytes): This value MUST be 0x8.

cchData (2 bytes): This value MUST be 0.

ExtraData (8 bytes): A PRTI.

2.9.284 SttbfBkmkSdt

The **SttbfBkmkSdt** structure is an **STTB** whose strings are **SDTI** structures, each of which contains information about a **structured document tag bookmark** in the document. The **cData** field size of this **STTB** is 4 bytes. This **STTB** is an extended **STTB**, which means that its **cchData** field size is 2 bytes. There is no extra data appended to the strings of this **STTB**. In a document, the number of structured document tag bookmarks MUST NOT exceed 0x7FFFFFFF.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (4 bytes)															
…																cbExtra (2 bytes)															
cchData$_0$ (2 bytes)																Data$_0$ (variable)															
…																															
cchData$_1$ (2 bytes)																Data$_1$ (variable)															
…																															
…																cchData$_{cData-1}$ (2 bytes)															
Data$_{cData-1}$ (variable)																															
…																															

Copyright © 2025 Microsoft Corporation

The **SttbfBkmkSdt** structure is an **STTB** structure that has the following additional constraints on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cData (4 bytes): This value MUST NOT exceed 0x7FFFFFFF.

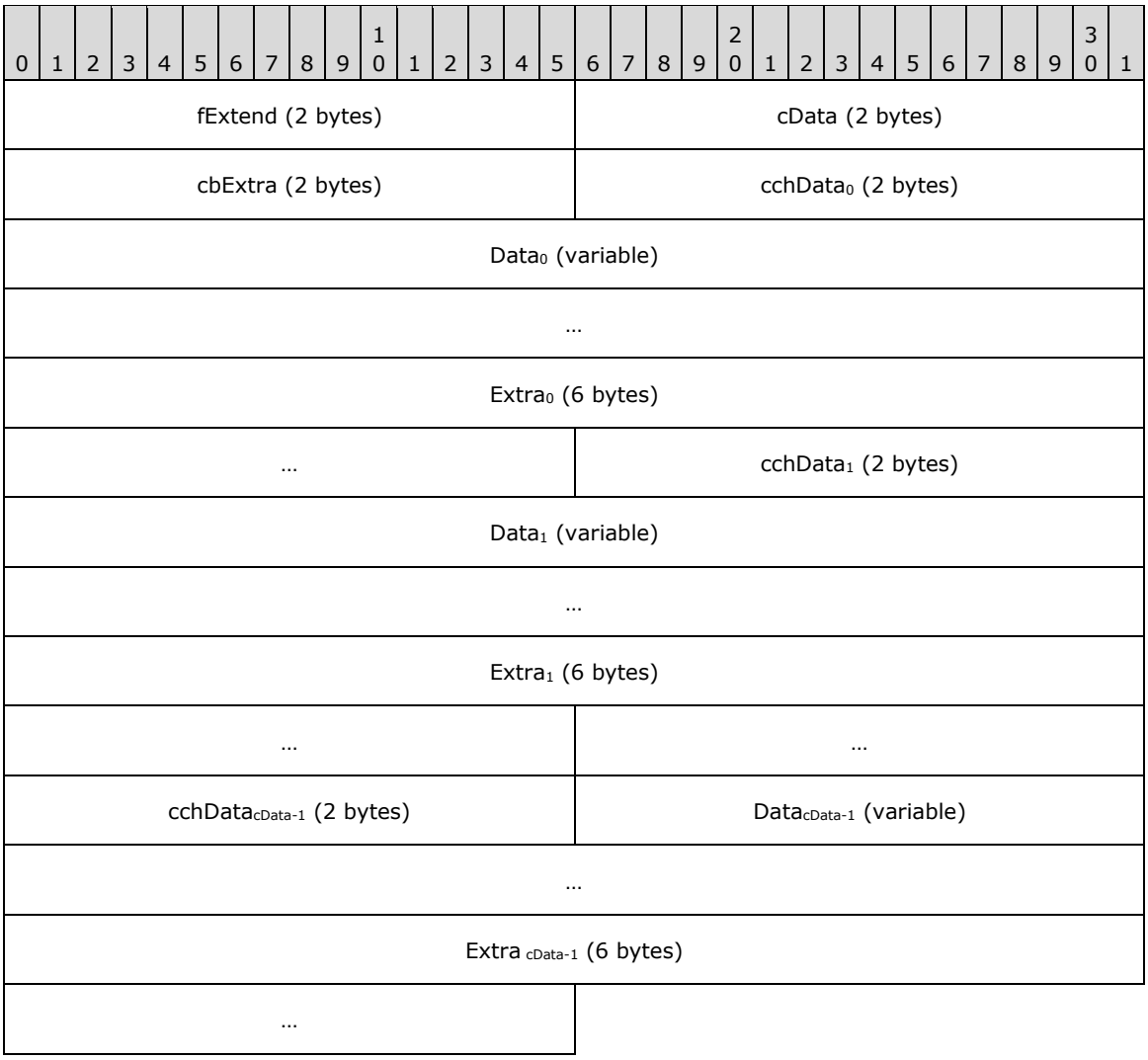
cbExtra (2 bytes): This value MUST be 0.

cchData (2 bytes): This value MUST be 0x000C.

Data (variable): An **SDTI**. The size of this field is 2 * **cchData** bytes, incremented by the value of the **cbPlaceholder** of this **SDTI** plus the size, in bytes, of the **fsdaparray** of this **SDTI**.

2.9.285 SttbfCaption

The **SttbfCaption** structure is an **STTB** structure that defines **caption**s. Each string in this **STTB** structure is the label of a caption, and MUST have less than or equal to 40 characters. The extra data appended to each string is a **CAPI** structure that specifies addition information about the caption.



Copyright © 2025 Microsoft Corporation

The **SttbfCaption** structure is an **STTB** structure that has the following additional restrictions on its field values:

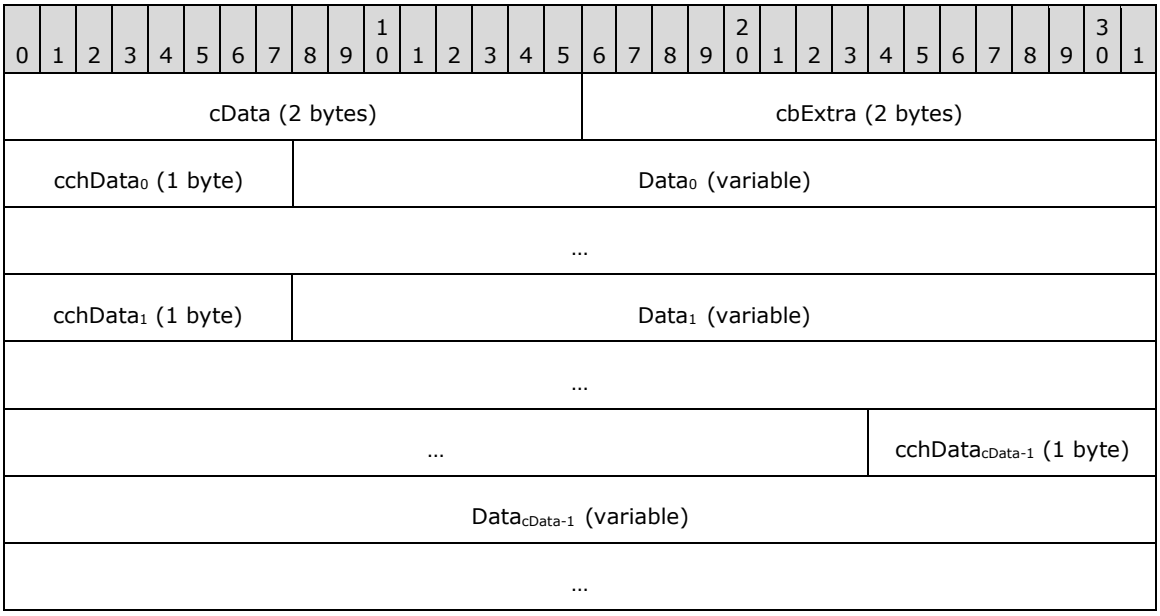
fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x0006.

cchData (2 bytes): This value MUST be less than or equal to 40.

2.9.286 SttbfFfn

The **SttbfFfn** structure is an **STTB** whose strings are **FFN** records that specify details of system fonts. Each font that is used in the document MUST have an entry in this list. There is no extra data appended to the strings of this **STTB**. Each **FFN** MUST be completely and accurately filled out with attributes that match the corresponding system font. This table MAY<244> contain fonts that are not referenced in the document.



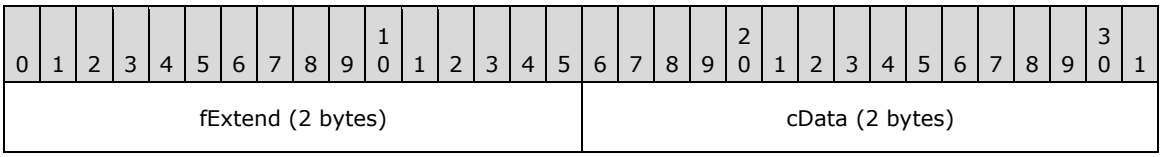
The **SttbfFfn** structure is a non-extended character **STTB** that has the following additional restrictions on its field values:

cData (2 bytes): This value MUST NOT exceed 0x7FF0.

cbExtra (2 bytes): This value MUST be 0.

2.9.287 SttbfGlsy

The **SttbfGlsy** structure is an STTB structure in which the strings specify the names of the **AutoText** and **rich text AutoCorrect** items that are defined in this document. These names correspond to their respective entries in the parallel PlcfGlsy. Each string in this STTB MUST have no more than 32 characters. The extra data that is appended to each string of this STTB is a LEGOXTR_V11, which specifies additional data about the item with which the string is associated.



Copyright © 2025 Microsoft Corporation

cbExtra (2 bytes)	$cchData_0$ (2 bytes)
$Data_0$ (variable)	
…	
$Extra_0$ (4 bytes)	
$cchData_1$ (2 bytes)	$Data_1$ (variable)
…	
$Extra_1$ (4 bytes)	
…	$cchData_{cData-1}$ (2 bytes)
$Data_{cData-1}$ (variable)	
…	
Extra $_{cData-1}$ (4 bytes)	

The **SttbfGlsy** structure is an **STTB** with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x0004.

cchData (2 bytes): This value MUST be less than or equal to 32.

2.9.288 SttbFnm

The **SttbFnm** structure is an **STTB** structure in which the strings specify the file names of the external files that are referenced by this document. Each file name contains the full path of the file, including the name and extension of the file. The extra data that is appended to each string of this **STTB** is an **FNIF** which contains additional information about the path. **fnpi.fnpd** MUST be unique for all **FNIF** structures in this **STTB** structure that share the same **fnpi.fnpt**. Because **fnpi** is unique for all **FNIF** structures in this **STTB** structure, **FNPI** structures can be used by other structures to reference the file names in this **STTB** structure.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (variable)																															
…																															

Copyright © 2025 Microsoft Corporation

Extra$_0$ (8 bytes)	
…	
cchData$_1$ (2 bytes)	Data$_1$ (variable)
…	
Extra$_1$ (8 bytes)	
…	
…	cchData$_{cData-1}$ (2 bytes)
Data$_{cData-1}$ (variable)	
…	
Extra $_{cData-1}$ (8 bytes)	
…	

The **SttbFnm** structure is an **STTB** with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x0008.

2.9.289 SttbfRfs

The **SttbfRfs** structure is an **STTB** structure that contains the strings for a mail merge. This structure SHOULD<245> contain 5 strings, and MUST contain at least 4 strings, as shown in the following table. There is no extra data appended to the strings of this **STTB**.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																cchData$_0$ (2 bytes)															
Data$_0$ (variable)																															
…																															
cchData$_1$ (2 bytes)																Data$_1$ (variable)															
…																															

Copyright © 2025 Microsoft Corporation

cchData$_2$ (2 bytes)	Data$_2$ (variable)
…	
cchData$_3$ (2 bytes)	Data$_3$ (variable)
…	
cchData$_4$ (2 bytes)	Data$_4$ (variable)
…	

The **SttbfRfs** structure is an **STTB** structure that has the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cData (2 bytes): This value SHOULD<246> be 0x0005, and MUST be either 0x0005 or 0x0004.

cbExtra (2 bytes): This value MUST be 0x0000.

cchData$_{0-4}$ (2 bytes): An unsigned integer that specifies the count of characters in the corresponding **Data** fields. This value MUST be less than 256.

Data$_0$ (variable): A **Unicode** string that specifies the **connection string** to the mail merge data source. This string MUST be identical to the string with **id**=0x0000 inside ODSOPropertyBase, if neither of these two strings is empty.

Data$_1$ (variable): A Unicode string that specifies the connection string to the source for the field names of the mail merge data. This string MUST be empty if the field names are from the same data source as **Data$_0$**.

Data$_2$ (variable): A Unicode string that specifies the e-mail subject line if the mail merge is for e-mail.

Data$_3$ (variable): A Unicode string that specifies the name of the data column that contains either e-mail addresses, if the mail merge is for e-mail, or fax numbers, if the mail merge is for fax.

Data$_4$ (variable): This value MUST be ignored.

2.9.290 SttbfRMark

The **SttbfRMark** structure is an **STTB** structure where the strings specify the names of the authors of the revision marks, comments, and e-mail messages in the document. There is no extra data appended to the strings of this **STTB**. The first entry MUST be "Unknown".

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																cchData$_0$ (2 bytes)															
Data$_0$ (variable)																															

Copyright © 2025 Microsoft Corporation

...	
...	
...	
cchData$_1$ (2 bytes)	Data$_1$ (variable)
...	
...	
...	
...	
cchData$_{cData-1}$ (2 bytes)	Data$_{cData-1}$ (variable)
...	
...	
...	

The **SttbfRMark** structure is an **STTB** with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0.

2.9.291 SttbGlsyStyle

The **SttbGlsyStyle** structure is an **STTB** structure in which the strings specify the names of the **style**s used by the **AutoText** and **rich text AutoCorrect** items that are defined in the parallel **SttbfGlsy**. The extra data that is appended to each string in this **STTB** is an unsigned 8-bit integer that specifies how many items use the style indicated by the string and that MUST be less than or equal to 0x32.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																cchData$_0$ (2 bytes)															
Data$_0$ (variable)																															
...																															
Extra$_0$ (1 byte)								cchData$_1$ (2 bytes)																						Data$_1$ (variable)			

Copyright © 2025 Microsoft Corporation

…	
Extra$_1$ (1 byte)	…
cchData$_{cData-1}$ (2 bytes)	Data$_{cData-1}$ (variable)
…	
Extra $_{cData-1}$ (1 byte)	

The **SttbGlsyStyle** structure is an **STTB** with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x0001.

2.9.292 SttbListNames

The **SttbListNames** structure is an **STTB** structure whose strings are the names used by the **LISTNUM field**, as specified by **LISTNUM** in flt, for each of the **LSTF** structures in the document. There is no extra data appended to the strings of this **STTB** structure. This **STTB** is parallel to **PlfLst.rgLstf**. If this **STTB** has more entries than **PlfLst.rgLstf**, the extra entries in this **STTB** MUST be ignored. If a list does not have a name, its corresponding string is an empty string. All non-empty strings in this **STTB** structure MUST be unique. Each string in this **STTB** structure MUST contain no more than 255 characters.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																cchData$_0$ (2 bytes)															
Data$_0$ (variable)																															
…																															
cchData$_1$ (2 bytes)																Data$_1$ (variable)															
…																															
…																cchData$_{cData-1}$ (2 bytes)															
Data$_{cData-1}$ (variable)																															
…																															

The **SttbListNames** structure is an **STTB** structure that has the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

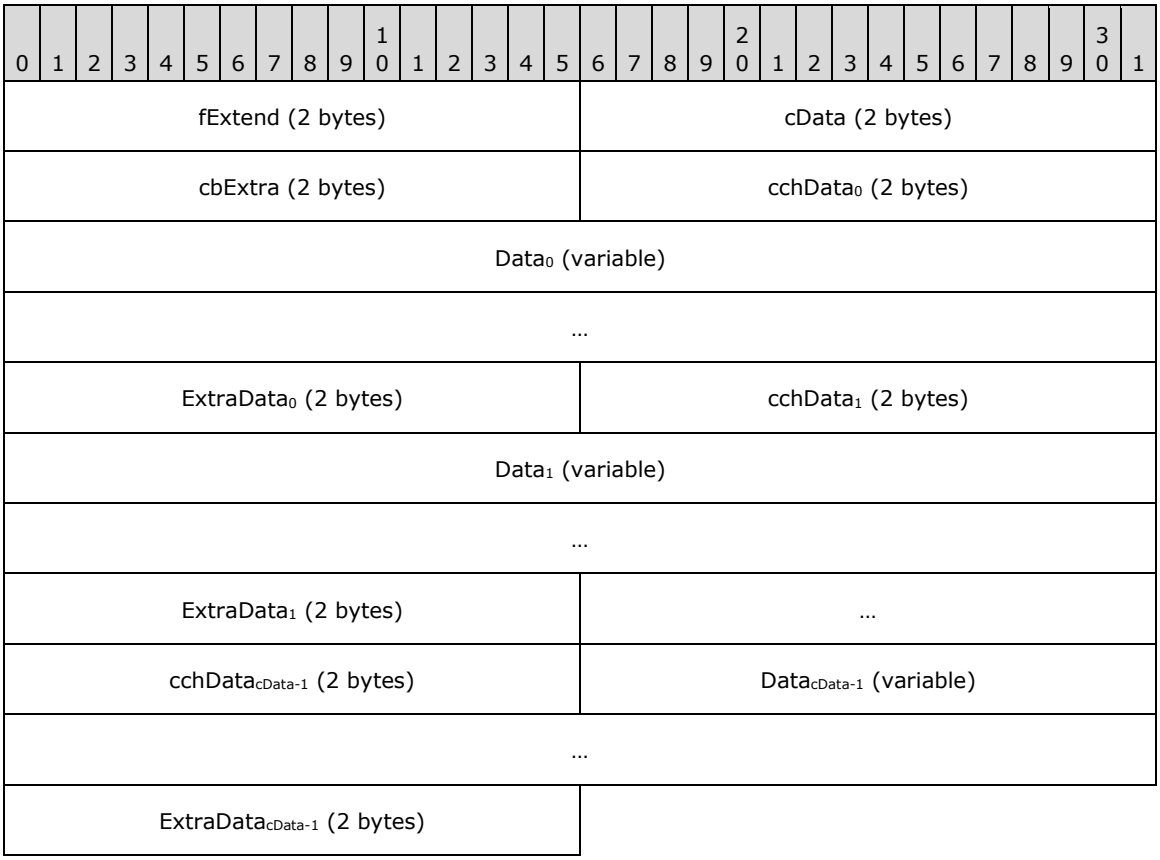
Copyright © 2025 Microsoft Corporation

cbExtra (2 bytes): This value MUST be 0x0000.

cchData (2 bytes): This value MUST be less than or equal to 0x00FF.

2.9.293 SttbProtUser

The **SttbProtUser** structure is an **STTB** structure in which the strings specify the usernames of users who have different roles with respect to a protected range of content in the document.



Each string is either the name of a mapped Windows user or group account that MUST be in the form "DOMAIN\NAME" or a valid e-mail address as defined in [RFC2822]. Each string in this **STTB** MUST be unique, and MUST have less than or equal to 255 characters. The extra data that is appended to each string of this **STTB** is a signed 16-bit integer that specifies the role for the username and MUST be one of the following values.

Value	Meaning
0x0000	There is no role specified for the user name.
0xFFFC	The username specifies an owner.
0xFFFB	The username specifies an editor.

The **SttbProtUser** structure is an **STTB** structure that has the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x0002.

cchData (2 bytes): This value MUST be less than or equal to 0x00FF.

Copyright © 2025 Microsoft Corporation

2.9.294 SttbRgtplc

The **SttbRgtplc** structure is an **STTB** structure in which each string specifies the bullet/numbering formats for a hybrid bulleted/numbered multi-level list. Because such a list can have a maximum of 9 levels, each string, if not empty, is in fact an array of 9 32-bit **Tplc** elements. The first element in each array specifies the format of the outermost level in the **hybrid list**.

SttbRgtplc is used parallel to **PlfLst** to specify the list formatting details. The index of each string inside SttbRgtplc corresponds to the **LSTF** of the same index inside **PlfLst**, with each **Tplc** mapped to the corresponding **LVL** inside the **LSTF**.

If the **fHybrid** member of the LSTF corresponding to a string in SttbRgtplc is 1, then that string in **SttbRgtplc** is not used and thus can be empty. In that case, the **cchData** of that string in the following table can be zero.

There is no extra data appended to the strings of this **STTB**.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (0 or 36 bytes)																															
…																															
$cchData_1$ (2 bytes)																$Data_1$ (0 or 36 bytes)															
…																															
…																$cchData_{cData-1}$ (2 bytes)															
$Data_{cData-1}$ (0 or 36 bytes)																															
…																															

The **SttbRgtplc** structure is an **STTB** with the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cData (2 bytes): This value MUST NOT exceed 0x7FF0.

cbExtra (2 bytes): This value MUST be 0.

cchData (2 bytes): This value MUST be either 0x0 or 0x12.

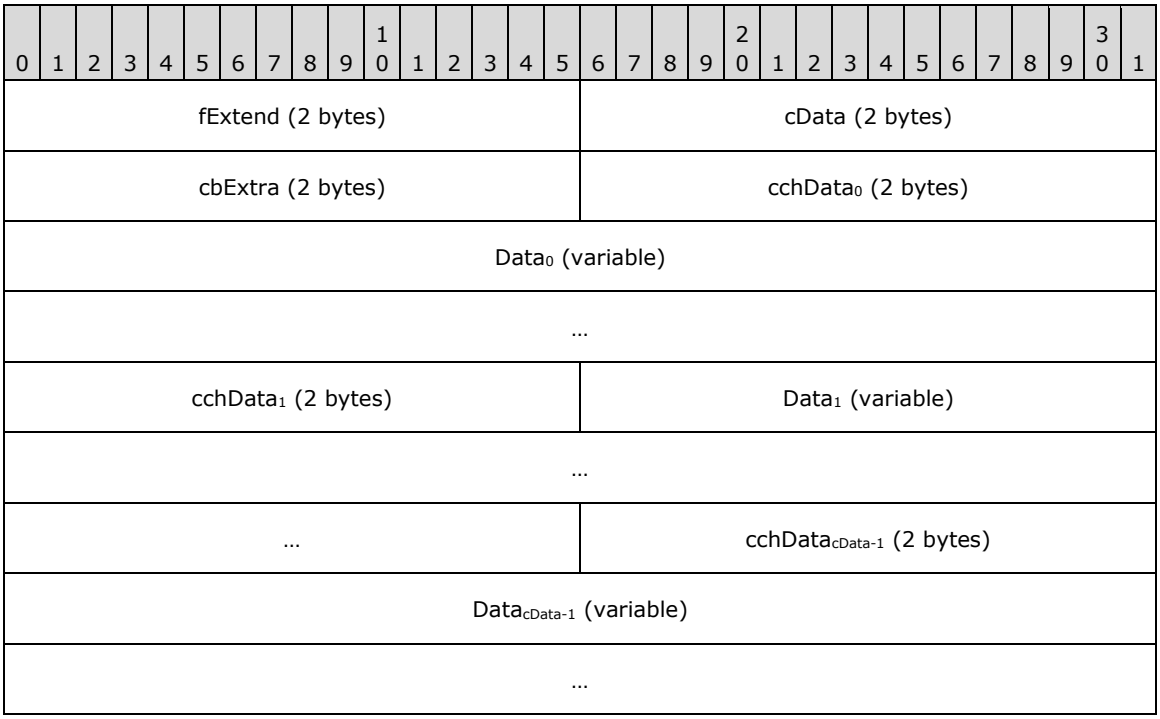
Data (0 or 36 bytes): An array that contains 9 **Tplc** elements. This does not exist if **cchData** is 0x0.

2.9.295 SttbSavedBy

The **SttbSavedBy** structure is an **STTB** structure that specifies the save history of this document. The strings in the **STTB** structure are arranged in pairs: A string that specifies the name of the author who saved the document, followed by a string that specifies the path and name of the saved file. These

Copyright © 2025 Microsoft Corporation

pairs are in order from the earliest saved file to the latest saved file. This **STTB** structure MUST have an even number of strings, and MUST have less than or equal to 20 strings. There is no extra data appended to the strings of this **STTB**.



The **SttbSavedBy** structure is an **STTB** structure that has the following additional restrictions on its field values:

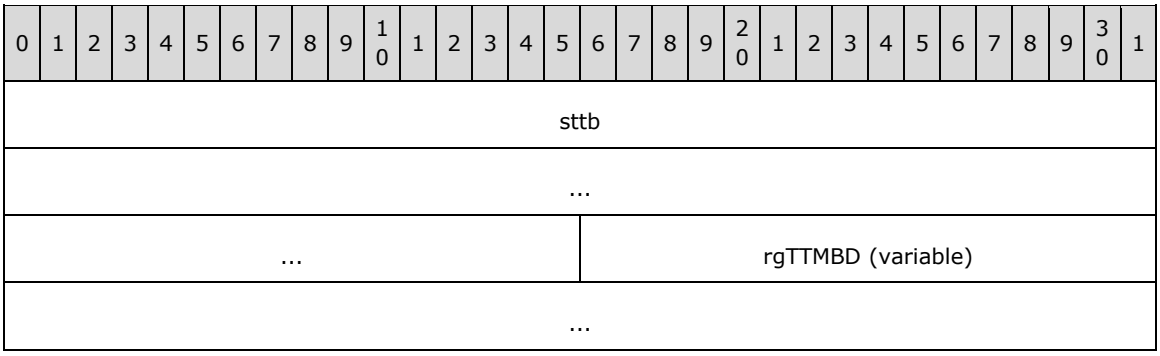
fExtend (2 bytes): This value MUST be 0xFFFF.

cData (2 bytes): This value MUST be even and MUST be less than or equal to 0x0014.

cbExtra (2 bytes): This value MUST be 0x0000.

2.9.296 SttbTtmbd

The **SttbTtmbd** structure contains the list of **TrueType font**s that are embedded in the document.



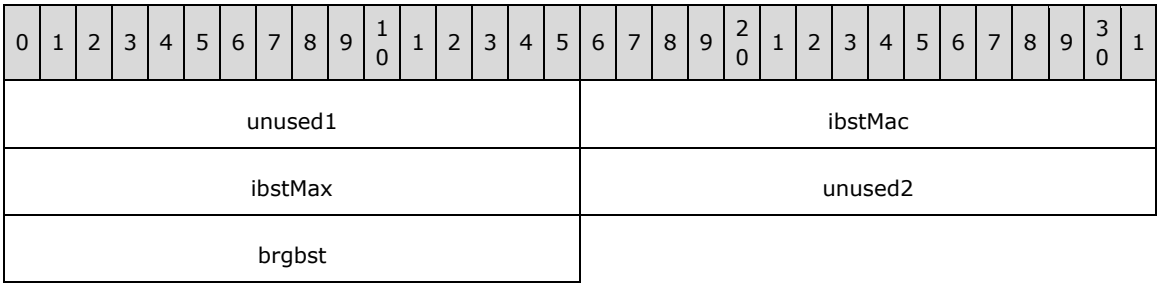
sttb (10 bytes): An **SttbW6** structure that specifies the number of TrueType fonts that are embedded in the document.

Copyright © 2025 Microsoft Corporation

rgTTMBD (variable): An array of **Ttmbd** elements. The number of elements is equal to **sttb.ibstMac** and MUST NOT exceed 64.

2.9.297 SttbW6

The **SttbW6** structure specifies the count of **TrueType font**s that are embedded in the document.



unused1 (2 bytes): This value MUST be 0 and MUST be ignored.

ibstMac (2 bytes): A signed integer that specifies the count of **Ttmbd** in **SttbTtmbd**.**rgTTMBD** (the number of TrueType fonts embedded in the document). This value MUST be nonnegative and MUST NOT exceed 64.

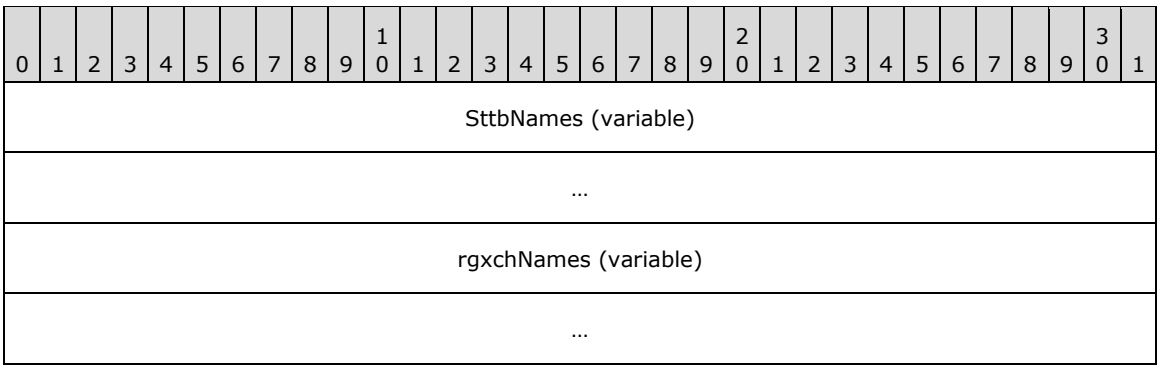
ibstMax (2 bytes): A signed integer that specifies the maximum number of embedded TrueType fonts that are supported by the document. This value MUST be 64.

unused2 (2 bytes): This value MUST be 0 and MUST be ignored.

brgbst (2 bytes): An unsigned integer that specifies the offset from the location of the **SttbW6** structure where **SttbTtmbd.rgTTMBD** begins. This value SHOULD<247> be 10 (the size of the **SttbW6** structure).

2.9.298 StwUser

The **StwUser** structure specifies the names and values of the user-defined variables that are stored in the document.



SttbNames (variable): An extended-character **STTB** that specifies the names of the variables. Each string in this **STTB** specifies the name of a variable. The extra data appended to each string in this **STTB** is a 4-byte unsigned integer that MUST be ignored. Each string in this **STTB** MUST be unique. The name **"Sign",** if it exists, SHOULD<248> specify the **VBA digital signature** variable. The name **"SigAgile",** if it exists, SHOULD<249> specify the VBA digital signature variable.

Copyright © 2025 Microsoft Corporation

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
fExtend (2 bytes)																cData (2 bytes)															
cbExtra (2 bytes)																$cchData_0$ (2 bytes)															
$Data_0$ (variable)																															
…																															
$Extra_0$ (4 bytes)																															
$cchData_1$ (2 bytes)																$Data_1$ (variable)															
…																															
$Extra_1$ (4 bytes)																															
…																$cchData_{cData-1}$ (2 bytes)															
$Data_{cData-1}$ (variable)																															
…																															
Extra $_{cData-1}$ (4 bytes)																															

The **SttbNames** structure is an **STTB** structure that has the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x0004.

rgxchValues (variable): An array of **Xst** elements. This array is parallel to **SttbNames**. Each string in this array specifies the value of the variable that is named by the corresponding string in **SttbNames**. The value that corresponds to the **"Sign"** name string in **SttbNames**, if it exists, SHOULD<250> be a special value that specifies the VBA digital signature of the document. The bytes of this value, including the count prefix, specify a **WordSigBlob** structure, as specified in [MS-OSHARED] section 2.3.2.3. The **WordSignBlob** MUST have the **contentInfo** field of the **SignedData** structure ([MS-OSHARED] section 2.3.2.4.1) as an **SpcIndirectDataContent** structure ([MS-OSHARED] section 2.3.2.4.3.1). The value that corresponds to the **"SigAgile"** name string in **SttbNames**, if exists, SHOULD<251> be a special value that specifies the VBA digital signature of the document. The bytes of this value, including the count prefix, specify a **WordSigBlob** structure, as specified in [MS-OSHARED] section 2.3.2.3. The **WordSignBlob** MUST have the **contentInfo** field of the **SignedData** structure ([MS-OSHARED] section 2.3.2.4.1) as an **SpcIndirectDataContentV2** structure ([MS-OSHARED] section 2.3.2.4.3.2).

2.9.299 Sty

The **Sty** structure is used by the **Selsf** structure and specifies the type of the selection that was made.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
styNil	0x0000	The selection type is undefined and is determined from the **Selsf** structure.
styChar	0x0001	The selection is one or more characters, an inline picture, or a text frame.
styWord	0x0002	The selection is one or more whole words.
stySent	0x0003	The selection is a sentence.
styPara	0x0004	The selection is a paragraph or a table cell.
styLine	0x0005	The selection is one or more whole lines of text.
styCol	0x000C	The selection is one or more whole table cells.
styRow	0x000D	The selection is one or more table rows.
styColAll	0x000E	The selection is one or more table columns.
styWholeTable	0x000F	The selection is a whole table.
styPrefix	0x001B	The selection is a bullet or numbering character in a bulleted or numbered list.

2.9.300 TabJC

The **TabJC** enumeration provides a 3-bit unsigned integer that specifies the type of alignment which is applied to the text that is entered at this tab stop. This MUST be one of the following values.

Name	Value	Meaning
jcLeft	0x0	Left justification.
jcCenter	0x1	Center justification.
jcRight	0x2	Right justification.
jcDecimal	0x3	Specifies that the current tab stop results in a location in the document at which all following text is aligned around the first decimal separator in the following text runs. If there is no decimal separator, text is aligned around the implicit decimal separator after the last digit of the first numeric value that appears in the following text. All text runs before the first decimal character appear before the tab stop; all text runs after it appear after the tab stop location.
jcBar	0x4	Specifies that the current tab is a **bar tab**.
jcList	0x6	Specifies that the current tab is a **list tab**.

2.9.301 TabLC

The **TabLC** enumeration is a 3-bit unsigned integer that specifies the characters that are used to fill in the space which is created by a tab that ends at a custom tab stop. This MUST be one of the following values.

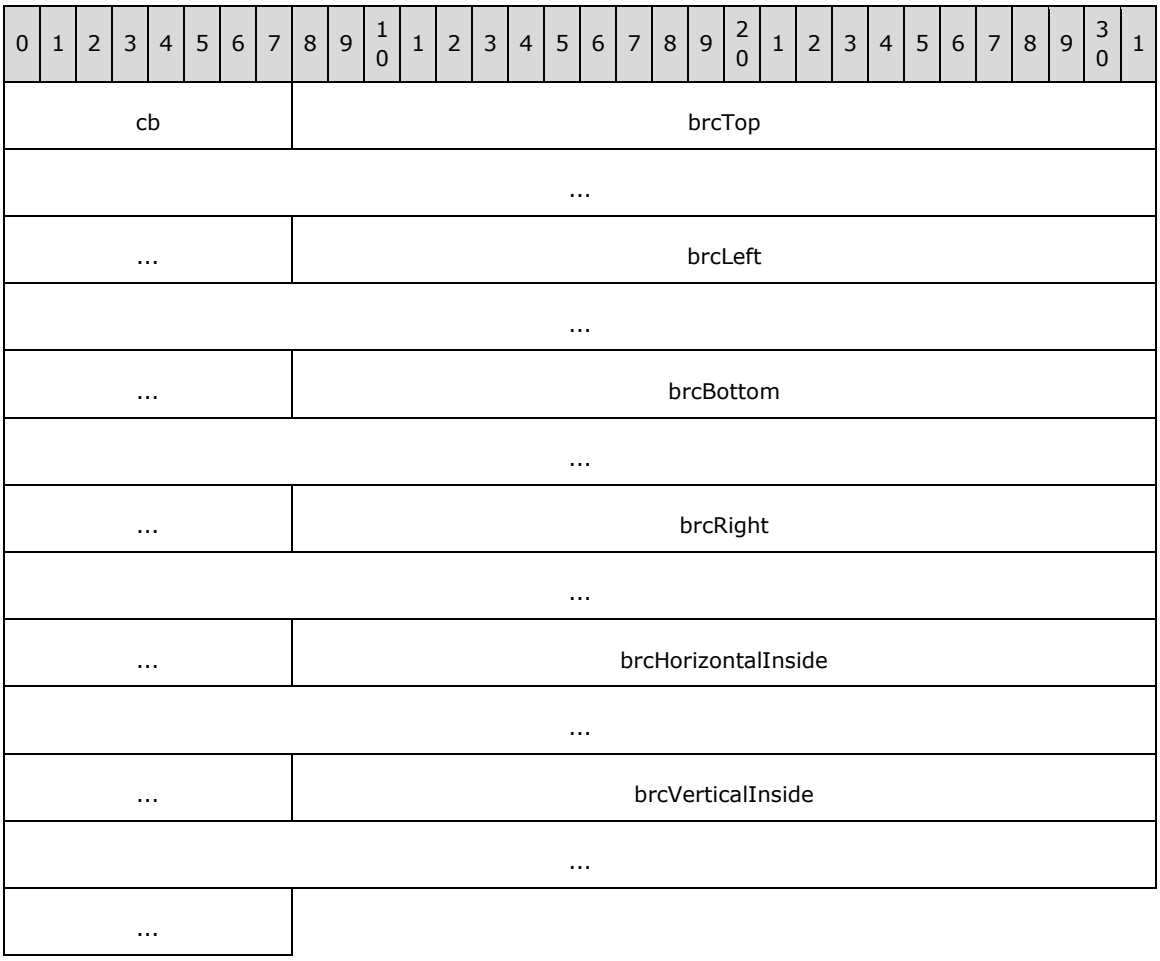
Name	Value	Meaning
tlcNone	0x0	No leader.
tlcDot	0x1	Dot leader.
tlcHyphen	0x2	Dashed leader.

Copyright © 2025 Microsoft Corporation

Name	Value	Meaning
tlcUnderscore	0x3	Underscore leader.
tlcHeavy	0x4	Same as tlcUnderscore.
tlcMiddleDot	0x5	Centered dot leader.
tlcDefault	0x7	Same as tlcNone.

2.9.302 TableBordersOperand

The **TableBordersOperand** structure specifies a set of borders for a table row.



cb (1 byte): An unsigned integer that specifies the size, in bytes, of this **TableBordersOperand** structure, not including this byte. This value MUST be 0x30.

brcTop (8 bytes): A **Brc** structure that specifies the top border of the row, if it is the first row in the table.

brcLeft (8 bytes): A **Brc** structure that specifies the **logical left** border of the row.

brcBottom (8 bytes): A **Brc** structure that specifies the bottom border of the row, if it is the last row in the table.

Copyright © 2025 Microsoft Corporation

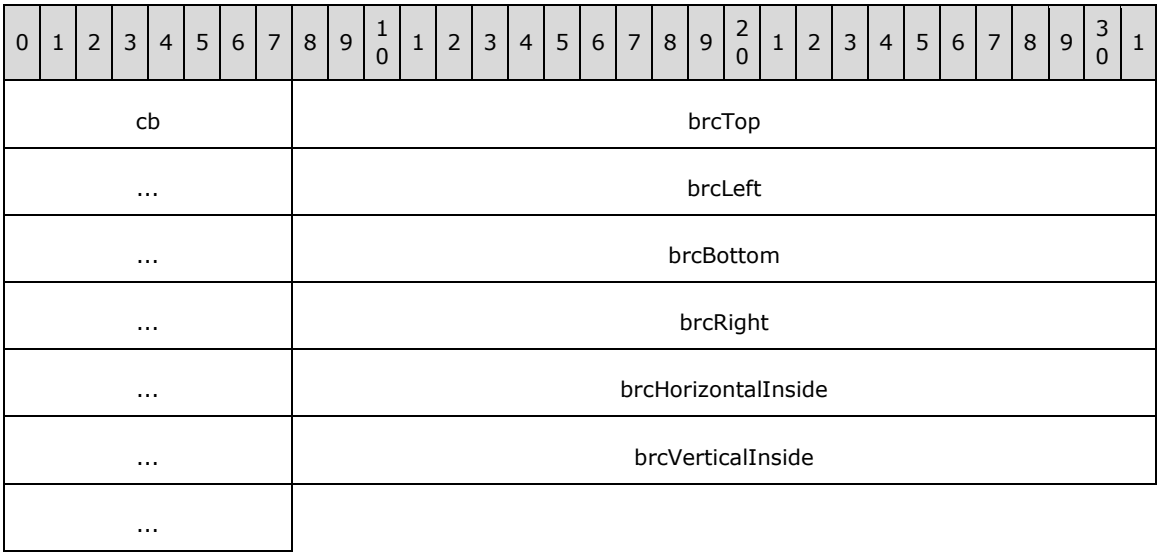
brcRight (8 bytes): A **Brc** structure that specifies the **logical right** border of the row.

brcHorizontalInside (8 bytes): A **Brc** structure that specifies the horizontal border between the row and the preceding and succeeding rows.

brcVerticalInside (8 bytes): A **Brc** structure that specifies the vertical border between the cells in the row.

2.9.303 TableBordersOperand80

The **TableBordersOperand80** structure is an operand that specifies the borders which are applied to a row of table cells.



cb (1 byte): An unsigned integer that specifies the size of this operand, not including this byte. This value MUST be 0x18.

brcTop (4 bytes): A **Brc80MayBeNil** structure that specifies the top border of the row, if it is the first row in the table.

brcLeft (4 bytes): A **Brc80MayBeNil** structure that specifies the **logical left** border of the row.

brcBottom (4 bytes): A **Brc80MayBeNil** structure that specifies the bottom border of the row, if it is the last row in the table.

brcRight (4 bytes): A **Brc80MayBeNil** structure that specifies the **logical right** border of the row.

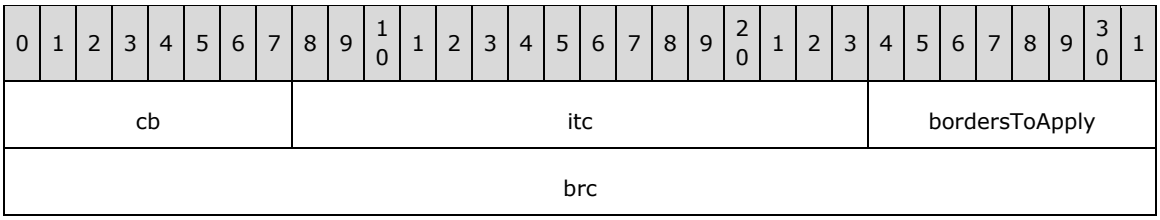
brcHorizontalInside (4 bytes): A **Brc80MayBeNil** structure that specifies the horizontal border between cells in this table row and those in the preceding or succeeding table rows.

brcVerticalInside (4 bytes): A **Brc80MayBeNil** structure that specifies the vertical border between neighboring cells of this table row.

2.9.304 TableBrc80Operand

The **TableBrc80Operand** structure is an operand that specifies borders for a range of cells in a table row.

Copyright © 2025 Microsoft Corporation



cb (1 byte): An unsigned integer that specifies the size, in bytes, of the remainder of this structure. The value MUST be 7.

itc (2 bytes): An **ItcFirstLim** structure that specifies the range of cell columns to apply the border type format.

bordersToApply (1 byte): An unsigned integer that specifies which borders are affected. The value MUST be the result of the bitwise OR of any subset of the following values that specifies an edge to be formatted:

0x01: Top border.

0x02: Logical left border.

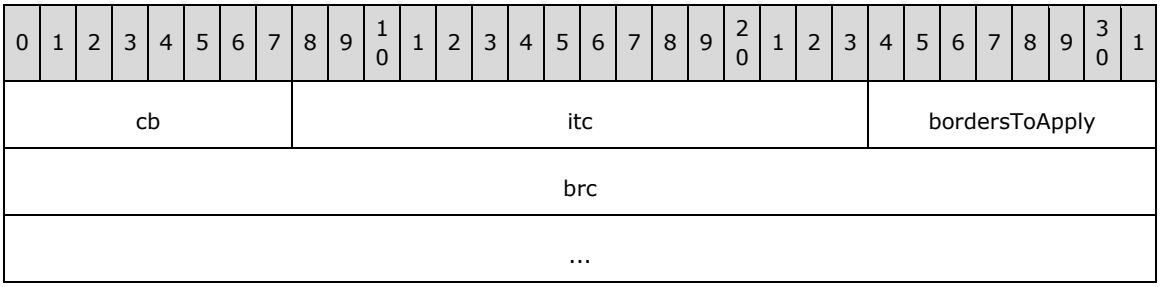
0x04: Bottom border.

0x08: Logical right border.

brc (4 bytes): A **Brc80MayBeNil** structure that specifies the border type that is applied to the edges which are indicated by **bordersToApply**.

2.9.305 TableBrcOperand

The **TableBrcOperand** structure is an operand that specifies borders for a range of cells in a table row.



cb (1 byte): An unsigned integer that specifies the size, in bytes, of the remainder of this structure. This value MUST be 11.

itc (2 bytes): An **ItcFirstLim** structure that specifies the range of cell columns to which the border type format is applied.

bordersToApply (1 byte): An unsigned integer that specifies which borders are affected. The value MUST be the result of the bitwise OR of any subset of the following values that specifies an edge to be formatted:

0x01: Top border.

0x02: Logical left border.

0x04: Bottom border.

Copyright © 2025 Microsoft Corporation

0x08: Logical right border.

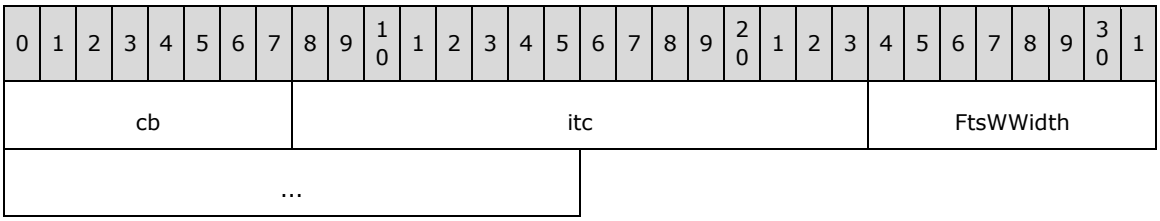
0x10: Border line from top left to bottom right.

0x20: Border line from top right to bottom left.

brc (8 bytes): A **BrcMayBeNil** structure that specifies the border type that is applied to the edges which are indicated by **bordersToApply**.

2.9.306 TableCellWidthOperand

The **TableCellWidthOperand** structure is an operand that is used by the **sprmTCellWidth** value to specify the width of one or more table cells.



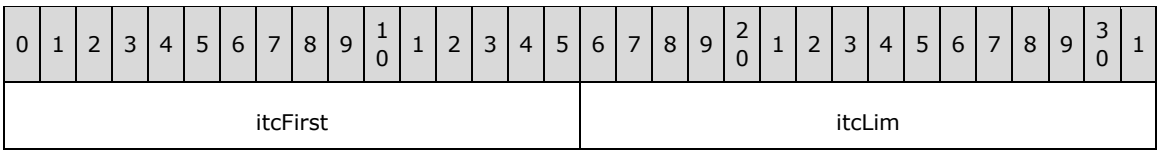
cb (1 byte): An unsigned integer that specifies the size of this operand in bytes, not including **cb**. The value of **cb** MUST be 5.

itc (2 bytes): An **ItcFirstLim** that specifies the cells to which this **TableCellWidthOperand** structure applies.

FtsWWidth (3 bytes): An **FtsWWidth_TablePart** that specifies the width of cells **itc.itcFirst** through **itc.itcLim** – 1.

2.9.307 TableSel

The **TableSel** structure is used by **Selsf** to specify the range of cells in a table cell block selection. **Selsf.fTable** MUST be 1. If **Selsf.fBlock** is zero, the selection is one or more table rows; otherwise, the selection is a range of cells. If **Selsf.fBlock** is 1 and the selection includes rows with differing cell counts, the **TableSel** is interpreted based on the first row in the selection.



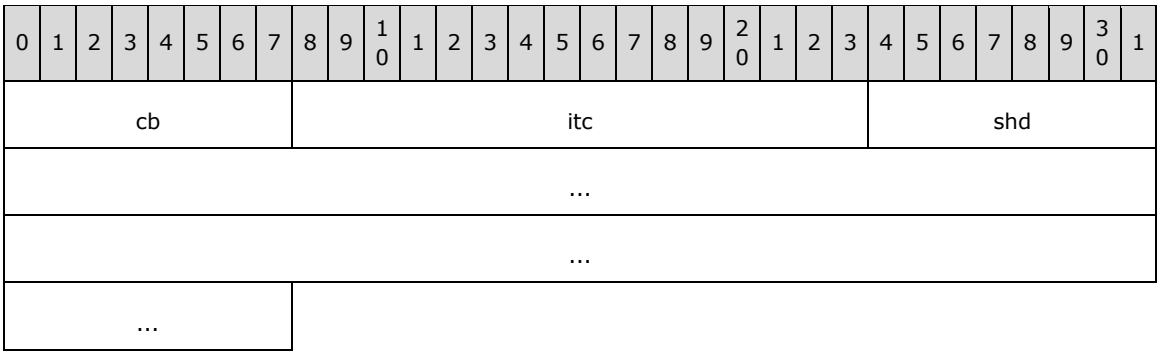
itcFirst (2 bytes): An integer that specifies the first cell that is included in the selection. Cell indices are zero-based. **itcFirst** MUST be at least zero, SHOULD NOT<252> exceed the number of cells in the row, and MUST NOT exceed 63. If **itcFirst** is greater than or equal to the number of cells in the row, the selection begins at the **end of row mark**. If **Selsf.fBlock** is zero, **itcFirst** MUST be zero.

itcLim (2 bytes): An integer that specifies the cell at which the selection ends, exclusive. Cell indices are zero-based. If the selection includes the last cell in the row, the **itcLim** value is the number of cells in the row. If the selection includes the end of row mark, **itcLim** is equal to the number of cells in the row incremented by 1. The **itcLim** value SHOULD<253> be greater than the **itcFirst** value and MUST NOT exceed 64. If **Selsf.fBlock** is zero, then **itcLim** MUST be 64. If the **itcLim** value is 64, the entire **Selsf** MAY<254> be ignored.

Copyright © 2025 Microsoft Corporation

2.9.308 TableShadeOperand

The **TableShadeOperand** structure specifies a range of cells in a table row and the background shading to apply to those cells.



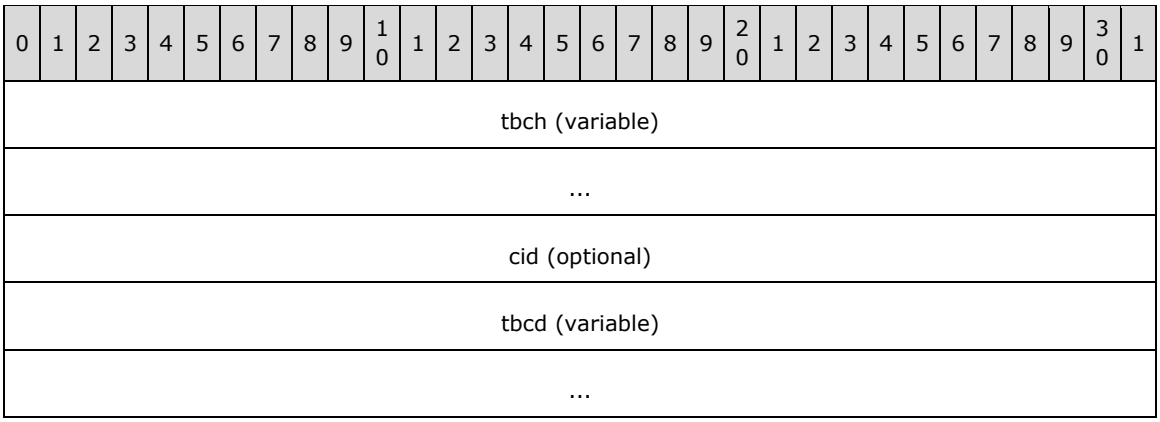
cb (1 byte): Specifies the byte count of the remainder of this structure. The value MUST be 12.

itc (2 bytes): An **ItcFirstLim** that specifies a cell range in the table row.

shd (10 bytes): A **Shd** structure that specifies the background shading that is applied.

2.9.309 TBC

The **TBC** structure specifies a **toolbar control**.



tbch (variable): A structure of type **TBCHeader**, as specified in [MS-OSHARED], that contains toolbar control information.

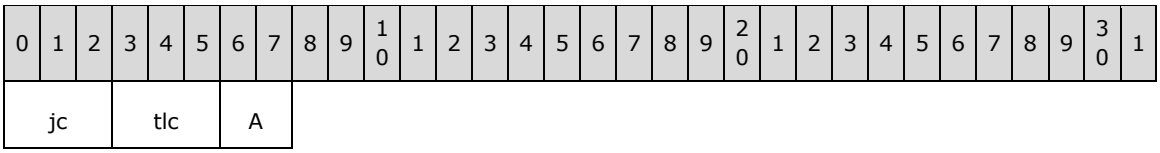
cid (4 bytes): A structure of type **Cid** that specifies the command identifier for this toolbar control. This MUST only exist if **tbch.tcid** is not equal to 0x0001 and is not equal to 0x1051. Toolbar controls MUST have only **Cid** structures whose **Cmt** values are equal to 0x0001 or 0x0003.

tbcd (variable): A structure of type **TBCData**, as specified in [MS-OSHARED], that contains toolbar control data. This MUST exist if **tbch.tct** is not equal to 0x16. This MUST NOT exist if **tbch.tct** is equal to 0x016.

2.9.310 TBD

A **TBD** structure specifies the alignment type and the leader type for a custom tab stop.

Copyright © 2025 Microsoft Corporation



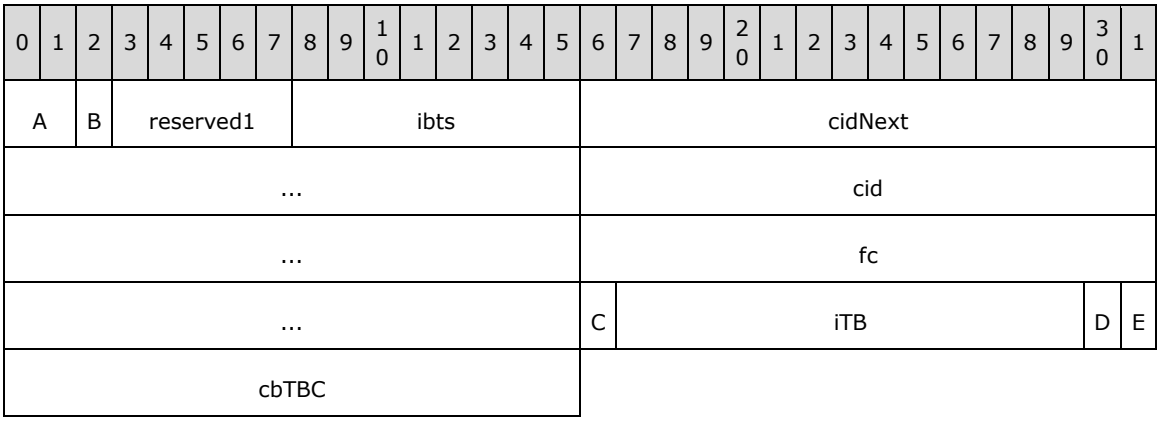
jc (3 bits): A **TabJC** value that specifies the alignment (justification) type for the current custom tab stop.

tlc (3 bits): A **TabLC** value that specifies the leader type for the current custom tab stop. The value MUST be ignored if **jc** is equal to 0x4 (jcBar).

A - UNUSED (2 bits): This field MUST be ignored.

2.9.311 TBDelta

The **TBDelta** structure specifies a **toolbar delta**. When the toolbar delta involves adding or modifying a **toolbar control**, the affected toolbar control is stored in the **rtbdc** array of the **CTBWRAPPER** structure that contains the **rCustomizations** array, that contains the **Customization** structure that contains the **customizationData** array, that contains this structure.



A - dopr (2 bits): These bits specify the type of toolbar delta operation. This MUST be one of the following values.

Value (Binary value in parenthesis)	Meaning
0x00 (00)	Change a toolbar control.
0x01 (01)	Insert a toolbar control.
0x02 (10)	Modify a toolbar control.

B - fAtEnd (1 bit): A bit that specifies if the toolbar control that is associated with this **TBDelta** was inserted at the end of the **toolbar** at the time the toolbar delta was created. A value of 1 specifies that the toolbar control that is associated with this **TBDelta** was inserted at the end of the toolbar. This bit MUST be 0 if **dopr** is not equal to 1.

reserved1 (5 bits): This value MUST be 0 and MUST be ignored.

ibts (8 bits): An unsigned integer that specifies the zero-based index of the toolbar control that was associated with this **TBDelta** in the toolbar at the time that the toolbar delta was created. It is possible for more than one **TBDelta** structure that affects the same toolbar to have the same

Copyright © 2025 Microsoft Corporation

value in the **ibts** field. This is because this field specifies the index of the toolbar control that was associated with the **TBDelta** in the toolbar at the time the toolbar delta was created.

cidNext (4 bytes): A signed integer value. This value MUST be one of the following.

Condition	Value of cidNext
dopr equals 1 and **fAtEnd** equals 1	0xFFFFFFFF
dopr equals 1, **fAtEnd** equals 0, and the toolbar control after the inserted toolbar control associated to this **TBDelta** at the time the **TBDelta** was created is not a **custom toolbar control**	A structure of type **Cid** that specifies the command identifier, at the time the toolbar delta was created, for the toolbar control after the inserted toolbar control associated to this **TBDelta**. Toolbar controls MUST only have **Cid** structures that have **Cmt** values equal to 0x0001 or 0x0003.
dopr equals 1, **fAtEnd** equals 0, and the toolbar control after the inserted toolbar control associated to this **TBDelta** at the time the **TBDelta** was created is a custom toolbar control	0x00001EF9
dopr equals 0	**cidNext** equals **cid**.
dopr equals 2 and the toolbar control after the deleted toolbar control that was associated with this **TBDelta** at the time the **TBDelta** was created is not a custom toolbar control	A structure of type **Cid** that specifies the command identifier at the time that the toolbar delta was created for the toolbar control after the deleted toolbar control was associated with this **TBDelta**. Toolbar controls MUST only have **Cid** structures that have **Cmt** values equal to 0x0001 or 0x0003.
dopr equals 2 and the toolbar control after the deleted toolbar control associated to this **TBDelta** at the time the **TBDelta** was created is a custom toolbar control	0x00001EF9

cid (4 bytes): A structure of type **Cid** that specifies the command identifier for the toolbar control that is associated with this **TBDelta**. Toolbar controls MUST only have **Cid** structures that have **Cmt** values equal to 0x0001 or 0x0003.

fc (4 bytes): An unsigned integer that specifies the file offset in the **Table Stream** where the toolbar control that is associated with this **TBDelta** is stored. This value MUST be 0x00000000 if **fOnDisk** is not equal to 1.

C - fOnDisk (1 bit): A bit that specifies if a toolbar control that is associated with this **TBDelta** was written to the file. A value of 1 specifies that a toolbar control that is associated with this **TBDelta** was written to the file. This value MUST be 1 if **dopr** is equal to 0 or 1.

iTB (13 bits): This field MUST be used only when the toolbar control that is associated with this **TBDelta** is a custom toolbar control that drops a custom **menu toolbar**. This is an unsigned integer that specifies the index to the **Customization** structure, contained in the **rCustomizations** array, that also contains the **Customization** that contains the **customizationData** array that contains this structure, that contains the **CTB** structure that specifies the custom menu toolbar dropped by the toolbar control associated to this **TBDelta**. This value MUST be 0 if the toolbar control that is associated with this **TBDelta** is not a custom toolbar control that drops a custom menu toolbar. This value MUST be greater than or equal to 0 and SHOULD<255> be less than the value of the **cCust** field of the **CTBWRAPPER** structure that contains the **rCustomizations** array, that contains the **Customization** structure, that contains the **customizationData** array that contains this structure.

D - reserved2 (1 bit): This value MUST be 0 and MUST be ignored.

E - fDead (1 bit): A bit that specifies if the toolbar control that is associated with this **TBDelta** does not drop a menu toolbar. A value of 1 specifies that the toolbar control that is associated with this **TBDelta** does not drop a custom menu toolbar. This value MUST be 0 if the toolbar control that is

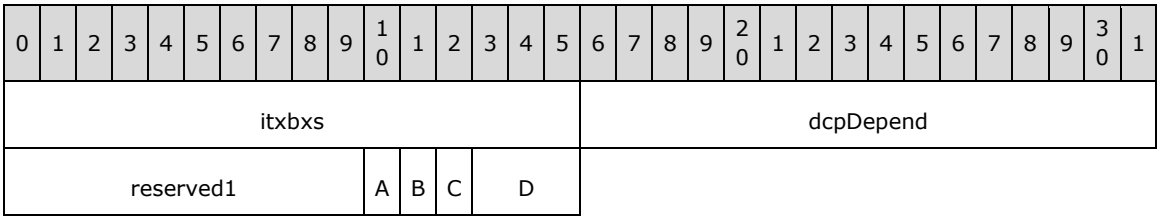
Copyright © 2025 Microsoft Corporation

associated with this **TBDelta** is not a custom toolbar control that drops a custom menu toolbar or if **dopr** is not equal to 1.

cbTBC (2 bytes): An unsigned integer that specifies the size, in bytes, of the toolbar control that is associated with this **TBDelta**. This field MUST only be used when **fOnDisk** equals 1. If **fOnDisk** is equal to 0, this value MUST be 0x0000.

2.9.312 Tbkd

The **Tbkd** structure is used by the **PlcfTxbxBkd** and **PlcfTxbxHdrBkd** structures to associate ranges of text from the Textboxes Document and the Header Textboxes Document with **FTXBXS** objects. Ranges of text from the Textboxes Document are associated with **FTXBXS** objects from **PlcftxbxTxt**; ranges of text from the Header Textboxes Document are associated with **FTXBXS** objects from **PlcfHdrtxbxTxt**.



itxbxs (2 bytes): A signed integer that specifies the index of an **FTXBXS** object within the **PlcftxbxTxt** structure or the **PlcfHdrtxbxTxt** structure. The text range of this **Tbkd** object MUST be the same as the text range of the **FTXBXS** object, or else it MUST be a subset of that range. When the **FTXBXS** object specifies a chain of linked textboxes, the text range of each component textbox MUST be represented by its own **Tbkd** object and a discrete text range.

In all but the last **Tbkd** object, **itxbxs** MUST be a valid **FTXBXS** index. The final **Tbkd** is not associated with any **FTXBXS** object. The itxbxs value for the final **Tbkd** MUST be ignored.

dcpDepend (2 bytes): Specifies version-specific information about the quantity of text that was processed. This makes it possible to identify the end of the corresponding text range. This value SHOULD<256> be zero and SHOULD<257> be ignored.

reserved1 (10 bits): This value MUST be 0 and MUST be ignored.

A - fMarkDelete (1 bit): This value MUST be 0 and MUST be ignored.

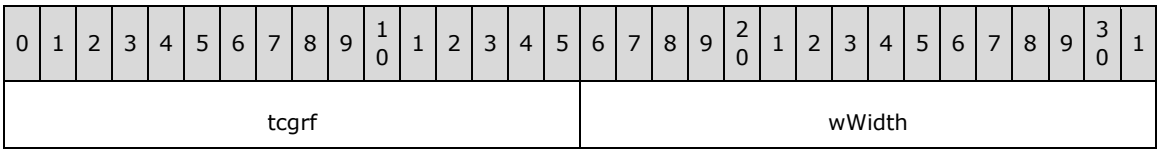
B - fUnk (1 bit): Specifies version-specific information that flags the text range which corresponds to this **Tbkd** as not being used by a textbox. This value SHOULD<258> be zero and SHOULD<259> be ignored.

C - fTextOverflow (1 bit): Specifies version-specific information about whether the text that is associated with a textbox exceeds the amount that fits into the associated shape. This value SHOULD<260> be zero and SHOULD<261> be ignored.

D - reserved2 (3 bits): This value MUST be 0 and MUST be ignored.

2.9.313 TC80

The **TC80** structure specifies the border and other formatting for a single cell in a table.



Copyright © 2025 Microsoft Corporation

brcTop
brcLeft
brcBottom
brcRight

tcgrf (2 bytes): A **TCGRF** that specifies table cell formatting.

wWidth (2 bytes): An integer that specifies the preferred width of the cell. The width includes cell margins, but does not include cell spacing. This value MUST be a non-negative number.

The unit of measurement depends on **tcgrf.ftsWidth**. If **tcgrf.ftsWidth** is set to **ftsPercent**, the value is a fraction of the width of the entire table.

brcTop (4 bytes): A **Brc80MayBeNil** structure that specifies the border to be used on the top side of the table cell.

brcLeft (4 bytes): A **Brc80MayBeNil** structure that specifies the border to be used on the **logical left** side of the table cell.

brcBottom (4 bytes): A **Brc80MayBeNil** that specifies the border to be used on the bottom side of the table cell.

brcRight (4 bytes): A **Brc80MayBeNil** that specifies the border to be used on the **logical right** side of the table cell.

2.9.314 TCellBrcTypeOperand

A **TCellBrcTypeOperand** structure specifies an array of border types for table cells.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
cb								rgBrcType (variable)																							
...																															

cb (1 byte): cb (1 byte): An unsigned integer that specifies the size, in bytes, of **rgBrcType**. This value MUST be evenly divisible by four.

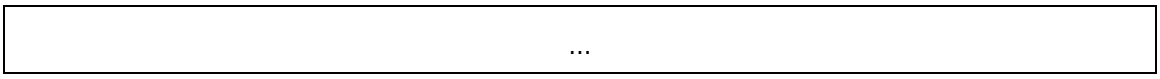
rgBrcType (variable): An array of **BrcType** that specifies border types for a set of table cells. Each cell corresponds to four bytes. Every four bytes specify the border types of the top, left, bottom and right borders, in that order.

2.9.315 Tcg

The **Tcg** structure specifies command-related customizations.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
nTcgVer								tcg (variable)																							

Copyright © 2025 Microsoft Corporation

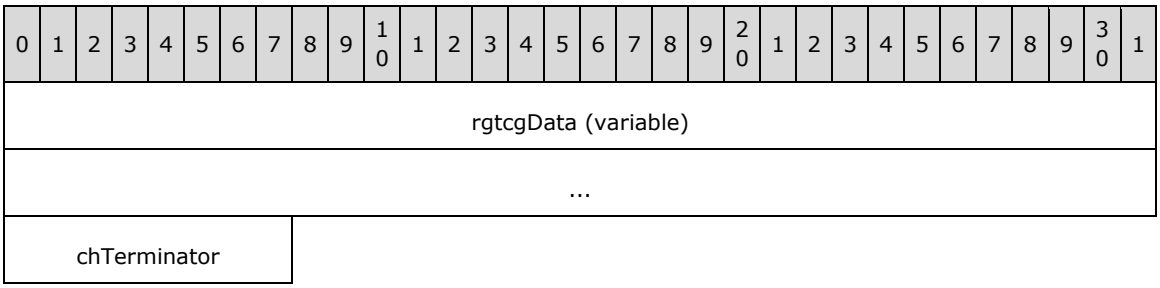


nTcgVer (1 byte): An unsigned integer that specifies the version of the structure. This value MUST be 255.

tcg (variable): A **Tcg255** structure, as specified following.

2.9.316 Tcg255

The **Tcg255** structure contains a sequence of structures that specify command-related customizations. The type of each structure is specified by its first byte with a special value that acts as a terminator.



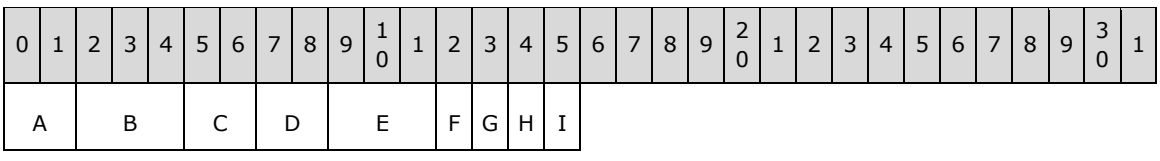
rgtcgData (variable): A sequence of structures. Each structure is identified by its first byte, as follows.

First byte	Structure
0x01	A **PlfMcd** structure that specifies macro commands.
0x02	A **PlfAcd** structure that specifies **allocated commands**.
0x03	A **PlfKme** structure that contains key map entries (**Kme**). Each key map entry MUST specify at least a primary key code, and the entries MUST be unique with regards to the key codes they specify.
0x04	A **PlfKme** structure that contains key map entries (**Kme**). Unlike when the first byte is equal to 3, there are no restrictions on the **Kme**.**kcm or Kme**.**kcm2** of each entry. If a keyboard key map entry does not specify at least a primary key code, that entry MUST be ignored. If two or more entries specify the same key codes, all except the first such entry MUST be ignored.
0x10	A **TcgSttbf** structure whose string table contains macro names and allocated command arguments.
0x11	A **MacroNames** structure that contains macro names.
0x12	A **CTBWRAPPER** structure that specifies toolbar customizations.

chTerminator (1 byte): An unsigned integer that specifies a terminator for the sequence. This value MUST be 0x40.

2.9.317 TCGRF

A **TCGRF** structure specifies the text layout and cell merge properties for a single cell in a table.



Copyright © 2025 Microsoft Corporation

A - horzMerge (2 bits): A value that specifies how this cell merges horizontally with the neighboring cells in the same row. This value MUST be one of the following.

Value	Meaning
0	The cell is not merged with the cells on either side of it.
1	The cell is one of a set of horizontally merged cells. It contributes its layout region to the set and its own contents are not rendered.
2, 3	The cell is the first cell in a set of horizontally merged cells. The contents and formatting of this cell extend into any consecutive cells following it that are designated as part of the merged set.

B - textFlow (3 bits): A value from the **TextFlow** enumeration that specifies rotation settings for the text in the cell.

C - vertMerge (2 bits): A value from the **VerticalMergeFlag** enumeration that specifies how this cell merges vertically with the cells above or below it.

D - vertAlign (2 bits): A value from the **VerticalAlign** enumeration that specifies how contents inside this cell are aligned.

E - ftsWidth (3 bits): An **Fts** value that specifies the unit of measurement for the **wWidth** field in the **TC80** structure.

F - fFitText (1 bit): Specifies whether the contents of the cell are to be stretched out such that the full cell width is used.

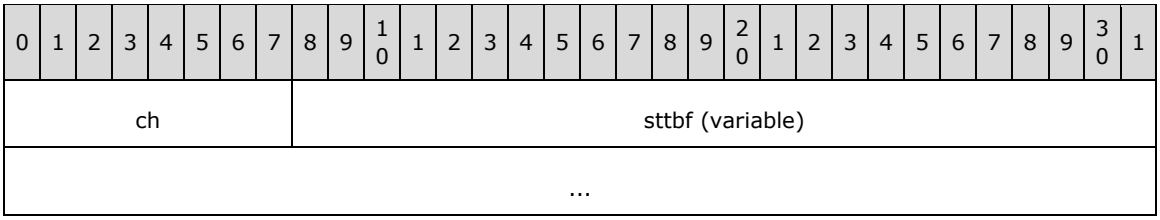
G - fNoWrap (1 bit): When set, specifies that the preferred layout of the contents of this cell is as a single line and that cell widths can be adjusted to accommodate long lines. This preference is ignored when the preferred width of this cell is set to **ftsDxa**.

H - fHideMark (1 bit): When set, specifies that this cell is rendered with no height if all cells in the row are empty.

I - fUnused (1 bit): This bit MUST be ignored.

2.9.318 TcgSttbf

The **TcgSttbf** structure specifies the command string table that is used to store the names of macros and the arguments to the **allocated commands**. This structure is used in the sequence of structures that specify command-related customizations in **Tcg255**.



ch (1 byte): This value MUST be 16.

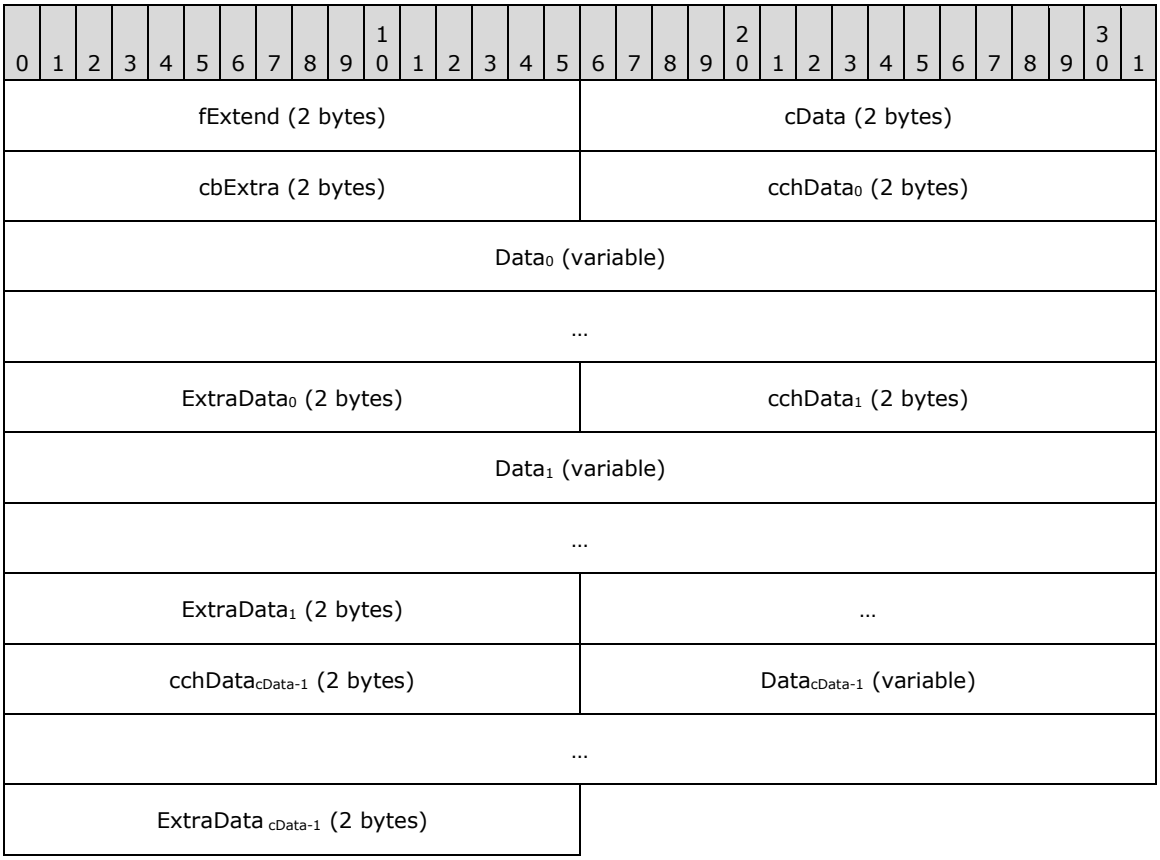
sttbf (variable): A **TcgSttbfCore** structure, as described following.

2.9.319 TcgSttbfCore

The **TcgSttbfCore** structure is an **STTB** structure whose strings are used by the **Acd** and **Mcd** structures. The **cData** field of this **STTB** structure is two bytes. This is an extended **STTB** structure,

Copyright © 2025 Microsoft Corporation

which means that its **cchData** fields are 2 bytes in size. The extra data that is appended to each string of this **STTB** is an unsigned 16-bit integer that specifies the number of references that other structures have to that string.



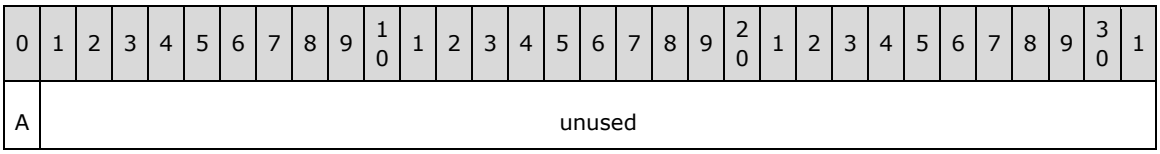
The **TcgSttbfCore** structure is an **STTB** that has the following additional restrictions on its field values:

fExtend (2 bytes): This value MUST be 0xFFFF.

cbExtra (2 bytes): This value MUST be 0x2.

2.9.320 Tch

The **Tch** structure is used by **PlcfTch** and specifies table character information for the CP range.



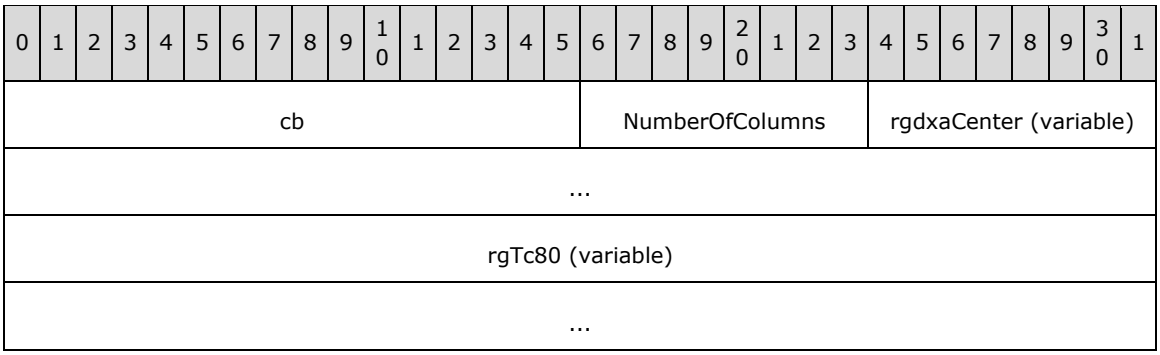
A - fUnk (1 bit): A bit that specifies that the table character cache for the CP range is unknown. If **fUnk** is set, **unused** MUST be ignored.

unused (31 bits): A bit field that specifies information for the CP range. This value SHOULD<262> be zero and SHOULD be ignored.

Copyright © 2025 Microsoft Corporation

2.9.321 TDefTableOperand

The **TDefTableOperand** structure is the operand that is used by the sprmTDefTable value. It specifies the initial layout of the columns in the current table row.



cb (2 bytes): An unsigned integer that specifies the number of bytes that are used by the remainder of this structure, incremented by 1.

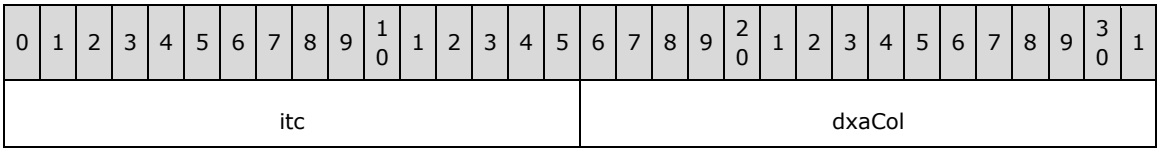
NumberOfColumns (1 byte): An integer that specifies the number of columns in this table. The number MUST be at least zero, and MUST NOT exceed 63.

rgdxaCenter (variable): An array of **XAS**. There MUST be exactly one **XAS** value in this array for every column specified in **NumberOfColumns**, incremented by 1. The first entry specifies the horizontal position of the **logical left** edge of the table, as indented from the logical left page margin. The remaining entries specify the horizontal positions of the **logical right** edges of each cell progressing logical right across the row. More specifically, the positions for all edges between cells are the midpoints of the inter-cell spacing. The first and last entries specify the positions of the outer edges of the table, including all cell spacing. The values in the array MUST be in non-decreasing order.

rgTc80 (variable): An array of **TC80** that specifies the default formatting for a cell in the table. Each TC80 in the array corresponds to the equivalent column in the table. If there are fewer TC80s than columns, the remaining columns are formatted with the default TC80 formatting. If there are more TC80s than columns, the excess TC80s MUST be ignored.

2.9.322 TDxaColOperand

The **TDxaColOperand** structure is used by the sprmTDxaCol value and specifies a range of table cells and the width of each cell.



itc (2 bytes): An **ItcFirstLim** structure that specifies which cells this column width applies to.

dxaCol (2 bytes): An **XAS_nonNeg** value that specifies the width of each of the columns, measured in **twips**. The width of a column is the measurement from the midpoint of the cell spacing before it to the midpoint of the cell spacing after it. For the first and last columns in a row, the width additionally includes the remainder of the cell spacing out to the outer border of the table.

Copyright © 2025 Microsoft Corporation

2.9.323 TextFlow

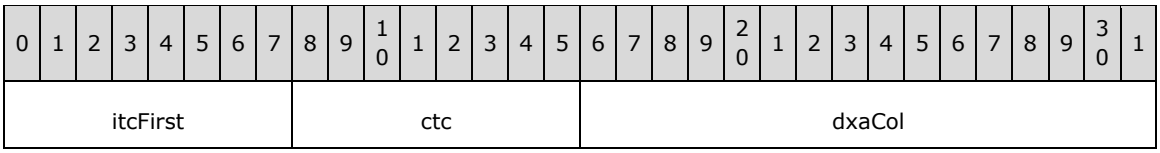
The **TextFlow** enumeration specifies the rotation settings for a block of text and for the individual **East Asian characters** in each line of the block.

A **TextFlow** value MUST be one of the following.

Name	Value	Meaning
grpfTFlrtb	0x0000	Specifies the standard vertical text arrangement. The text block is not rotated. Multiple lines are arranged top to bottom. The characters on a line are laid out left to right.
grpfTFtbrl	0x0001	Specifies a 90-degree clockwise rotation of the standard vertical text block. The lines in the block are vertical and arranged right to left. The characters on a line are rotated 90 degrees in a clockwise direction and laid out top to bottom.
grpfTFbtlr	0x0003	Specifies a 90 degree, counter-clockwise rotation of the standard vertical text block. The lines in the block are vertical and arranged left to right. The characters on a line are rotated 90 degrees in a counter-clockwise direction and laid out bottom to top.
grpfTFlrtbv	0x0004	Specifies the same line layout as **grpfTFlrtb**, however, each East Asian character is rotated 90 degrees in a counter-clockwise direction. All other text is not rotated.
grpfTFtbrlv	0x0005	Specifies the same rotated line layout as **grpfTFtbrl**, however, each East Asian character is rotated 90-degrees in a counter-clockwise direction within the block, canceling out the rotation in **grpfTFtbrl**. All other text is left with the rotation found in **grpfTFtbrl**.

2.9.324 TInsertOperand

The **TInsertOperand** structure is an operand that is used by the **sprmTInsert** value and specifies a range of default table cell definitions to add to a table row.



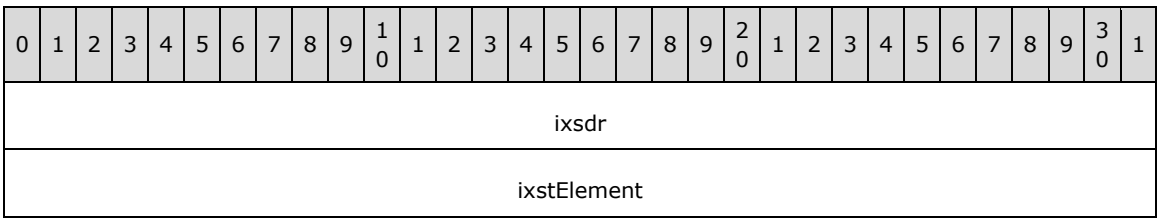
itcFirst (1 byte): An unsigned integer that specifies the zero-based index of the first new table cell definition.

ctc (1 byte): An unsigned integer that specifies the number of new table cells. This value MUST be greater than zero. Table rows MUST NOT have more than 63 cells after the insertion.

dxaCol (2 bytes): An **XAS_nonNeg** value that specifies the width of each of the new cells. The total width of the table after inserting the new cells MUST NOT exceed 31680 **twips**.

2.9.325 TIQ

The **TIQ** structure specifies information about a **structured document tag** node, or an attribute on a structured document tag node, in the document.



Copyright © 2025 Microsoft Corporation

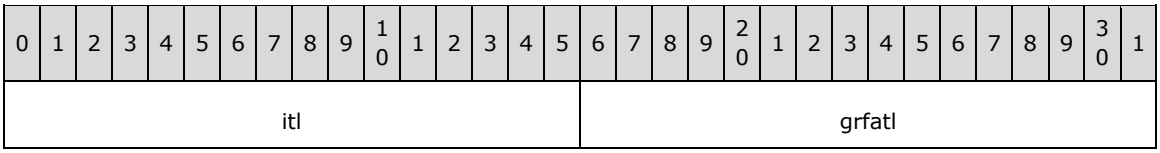
ixsdr (4 bytes): An unsigned integer that specifies the **STTB** which is the namespace of the structured document tag node or attribute that is represented by the structure containing this **TIQ** structure. This value MUST be less than 0x7FFFFFFF. This **STTB** can be found by using the following algorithm:

1. The structure that contains this **TIQ** is contained in an **SttbfBkmkSdt** which is located at the offset specified by the **fcSttbfBkmkSdt** field of **FibRgFcLcb2003** structure.
2. The **fcHplxsdr** field of that section **FibRgFcLcb2003** structure specifies the location of an **Hplxsdr**.
3. **ixsdr** is the zero-based index of an **XSDR** within the **rgXSDR** array of that **Hplxsdr**.
4. If this TIQ is a field of an **FSDAP** structure, the string table that is specified by this **ixsdr** is the **SttbElements** field of the **XSDR** in step 3. If this **TIQ** is a field of an **SDTI** structure, the string table is the **SttbAttributes** field.

ixstElement (4 bytes): An integer that specifies a zero-based index into the STTB namespace that is denoted by **ixsdr**. The string that is found at offset **ixstElement** is the name of the structured document tag node or attribute associated with the structure containing this TIQ.

2.9.326 TLP

The **TLP** structure specifies the **table style** options for the current table.



itl (2 bytes): A signed integer that MAY<263> specify the index of a predefined table auto-format. Formats applied through auto-formatting are distributed to all of the affected rows and cells, and can be changed independently of this value. As such, the value that is found here does not specify any formatting for the table as it exists now. The purpose of this data is to aid in the re-application of the auto-format in the future.

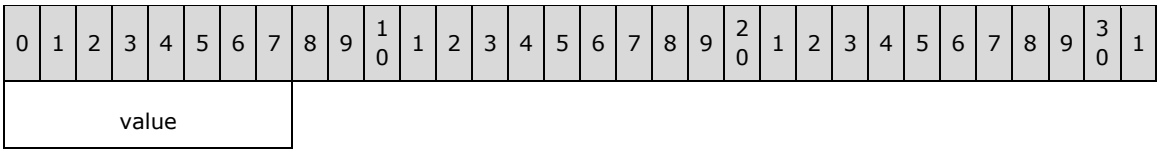
The list of auto-formats is application specific. The special values for **itl** are as follows.

Name	Value	Meaning
itlNil	-1	No predefined table auto-format was applied to this table row.
itlNone	0	A predefined table auto-format where all border, shading, font, and best fit formats are the defaults.

grfatl (2 bytes): A bit field of **Fatl** flags that SHOULD<264> specify which optional formats are in effect from the table style or table auto-format applied to the table.

2.9.327 ToggleOperand

The **ToggleOperand** structure is an operand to an **SPRM** whose **spra** is 0 and whose **sgc** is 2. It modifies a Boolean character property.



value (1 byte): An unsigned integer which MUST be one of the following values.

Copyright © 2025 Microsoft Corporation

Value	Meaning
0x00	The Boolean property is set to 0, which means the property is turned OFF.
0x01	The Boolean property is set to 1, which means the property is turned ON.
0x80	The Boolean property is set to match the value of the property in the current style that is applied to the text.
0x81	The Boolean property is set to the opposite of the value of the property in the current style that is applied to the text.

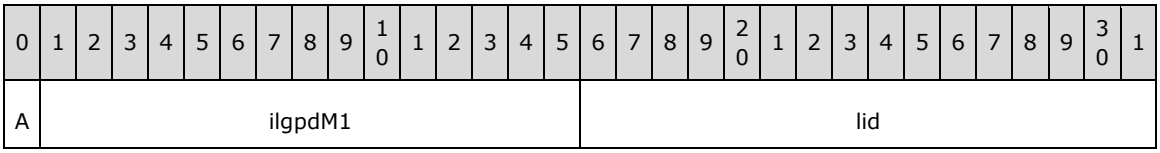
2.9.328 Tplc

The **Tplc** structure is a 32-bit unsigned integer that specifies the format of one level of a bulleted or numbered list.

If the first bit (lowest bit) is 1, **Tplc** specifies an application built-in format, as specified in **TplcBuildIn**. If the first bit is 0, Tplc specifies a user-defined format, as specified in **TplcUser**. See **SttbRgtplc** for more details about how Tplcs are mapped to **LVL**s inside **LSTF**.

2.9.329 TplcBuildIn

The **TplcBuildIn** structure is a **Tplc** structure that specifies an application predefined format for the bulleted or numbered list.



A - fBuildIn (1 bit): This value MUST be 1.

ilgpdM1 (15 bits): An unsigned integer that specifies the predefined bulleted or numbered format and that MUST be one of the values from the following table. The precise rendering of the bulleted or numbering format is application-dependent.

Value	Bullet/numbering format
0x7FFF	(none)
0x0000	•
0x0001	○
0x0002	▪
0x0003	❑
0x0004	❖
0x0005	➢
0x0006	➢

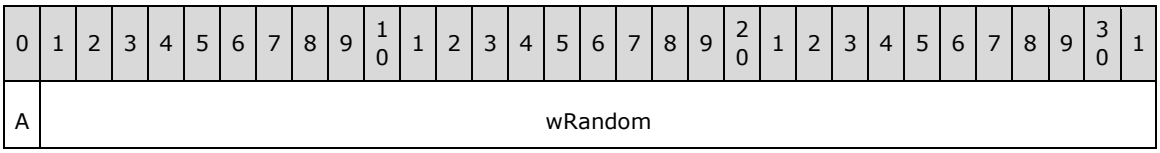
Copyright © 2025 Microsoft Corporation

Value	Bullet/numbering format
0x0007	1.
0x0008	1)
0x0009	I.
0x000A	A.
0x000B	a)
0x000C	a.
0x000D	i.

lid (2 bytes): A **LID** that specifies the language identifier for the bullet or number.

2.9.330 TplcUser

The **TplcUser** structure is a **Tplc** value that specifies a user-defined bulleted or numbered format. It MUST correspond to an **LSTF** structure (see **LSTF**.tplc) or it MUST correspond to an individual **LVL** structure. This **LVL** structure MUST correspond to an **LSTF** structure in the **PlfLst** structure. The **LSTF** and **LVL** structures contain the detailed format specification. See the **SttbRgtplc** structure for more details about how **Tplc** values are mapped to **LVL** structures inside the **LSTF** structure.

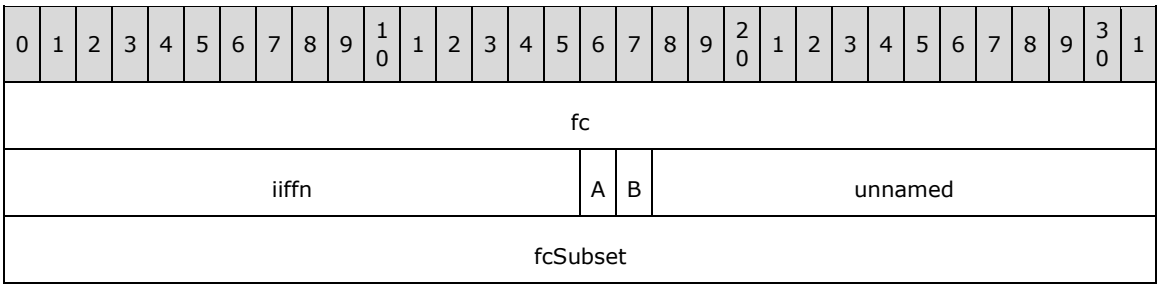


A - fBuildIn (1 bit): This value MUST be 0.

wRandom (31 bits): An unsigned random integer assigned by the application. Any unsigned integer is valid as long as it is unique for each user-defined bulleted or numbered format.

2.9.331 Ttmbd

The **Ttmbd** structure specifies information about an embedded **TrueType font**, including where to locate the font in the document.



Copyright © 2025 Microsoft Corporation

fc (4 bytes): An unsigned integer value that specifies an offset into the WordDocument Stream where the embedded TrueType font is stored. This value MUST be nonzero. The font data that is stored at this offset is written as specified in [Embed-Open-Type-Format].

iiffn (2 bytes): A signed integer value that specifies an index into the **SttbfFfn** string table stored at **FibRgFcLcb97**.**fcSttbffffn**. This value MUST be a non-negative number.

A - fBold (1 bit): Specifies whether the font is bold.

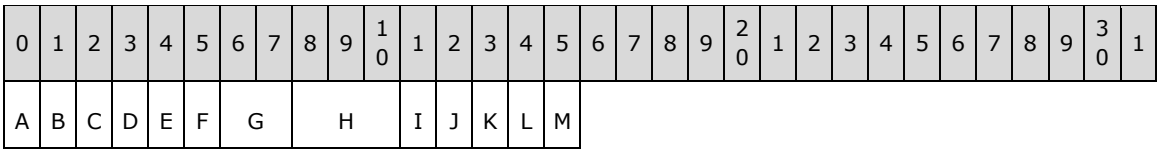
B - fItalic (1 bit): Specifies whether the font is italic.

unnamed (14 bits): Undefined and MUST be ignored.

fcSubset (4 bytes): If entire fonts are embedded in the document, **fcSubset** MUST be 0xFFFFFFFF. If only the characters that are used by the document are embedded in the document, **fcSubset** is an unsigned integer that specifies the order in which fonts are first used. The first font to be used in the document has an **fcSubset** value that is equal to zero; all subsequent fonts are incremented by 1 in order of first use. **fcSubset** MUST be incremented for all fonts that are used in the document, including fonts that are not embedded in the document. This value MUST NOT exceed the total number of fonts used in the document.

2.9.332 UFEL

The **UFEL** structure specifies layout information for text in East Asian languages. See also [ECMA-376] part 1, section 17.3.2.10 eastAsianLayout paragraph property.



A - fTNY (1 bit): A bit that specifies if the text displays horizontally within vertical text, or vertically within horizontal text. The text is rendered with a 90-degree rotation to the left from all other contents of the containing line, while keeping the text on the same line as all other text in the paragraph.

B - fWarichu (1 bit): A bit that specifies that the text displays on a single line by creating two sub-lines within the regular line, and laying out this text equally between those sub-lines.

C - fKumimoji (1 bit): This value MUST be 0 and MUST be ignored.

D - fRuby (1 bit): This value MUST be 0 and MUST be ignored.

E - fLSFitText (1 bit): The value MUST be 0 and MUST be ignored.

F - fVRuby (1 bit): This value MUST be 0 and MUST be ignored.

G - spare1 (2 bits): This value MUST be ignored.

H - iWarichuBracket (3 bits): An unsigned integer that specifies whether the two sub-lines within one line are enclosed within a pair of brackets when displayed, and the type of brackets that are displayed. If **fWarichu** is equal to 0x0, this value MUST be ignored.

The **iWarichuBracket** value MUST be one of the following.

Copyright © 2025 Microsoft Corporation

Value	Meaning
0x0	No brackets
0x1	Round brackets, "()"
0x2	Square brackets, "[]"
0x3	Angle brackets, "<>"
0x4	Curly brackets, "{}"

I - fWarichuNoOpenBracket (1 bit): This value MUST be 0 and MUST be ignored.

J - fTNYCompress (1 bit): A bit that specifies whether other **Sprm** structures were applied that cause the text to be scaled to fit within the existing line. A value of 0x1 means that other **Sprm** structures were applied. A value of 0x0 means that they were not.

K - fTNYFetchTxm (1 bit): This value MUST be 0 and MUST be ignored.

L - fCellFitText (1 bit): This value MUST be 0 and MUST be ignored.

M - spare2 (1 bit): This value MUST be ignored.

2.9.333 UID

The **UID** enumeration identifies common user types.

Name	Value	Meaning
uidNone	0x0000	No users.
uidCurrent	0xFFFA	The current user.
uidEditors	0xFFFB	Editors of the document.
uidOwners	0xFFFC	Owners of the document.
uidContributors	0xFFFD	Contributors to the document.
uidAdministrators	0xFFFE	Members of the administrator group on the computer.
uidEveryone	0xFFFF	All users.

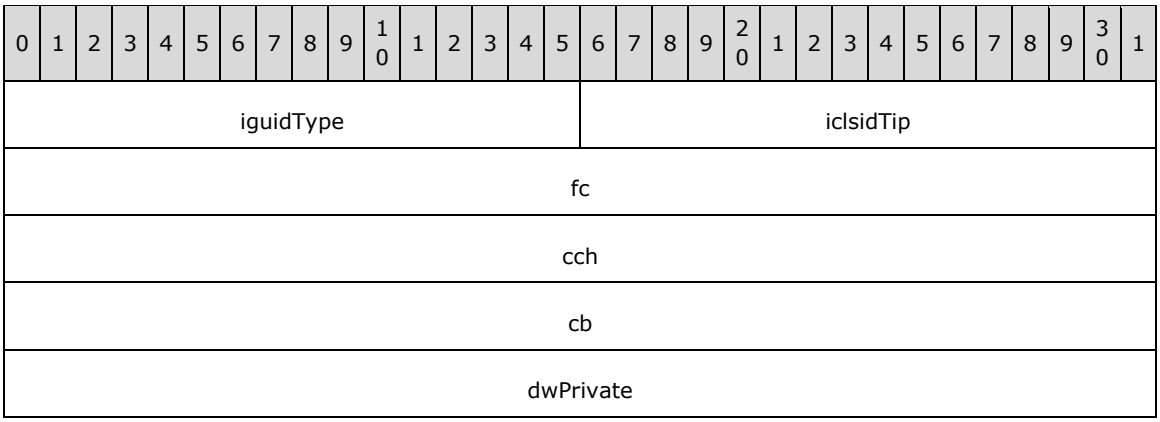
2.9.334 UidSel

The **UidSel** structure is a 2-byte integer that identifies a user or group of users for the purpose of specifying **range-level protection** information about the given users. If the integer is greater than zero, it MUST be a 1-based index into the **SttbProtUser** at an offset of **FibRgFcLcb2003.fcSttbProtUser** in the Table Stream. Otherwise, it is a **UID** type that MUST be one of the uidEveryone, uidEditors, or uidOwners values.

2.9.335 UIM

The **UIM** structure contains data that was provided by the Microsoft Windows Text Services Framework, a service provided by Microsoft Windows that enables the application to receive input from different input sources, such as handwriting.

Copyright © 2025 Microsoft Corporation



iguidType (2 bytes): A signed integer value that specifies an index into the GUIDs that are listed in **PlfguidUim**.**rgguidUim**. This value MUST be greater than or equal to 0 and less than **PlfguidUim**.**iMac**. The referenced GUID specifies the Text Services category of the service that provided this data.

iclsidTip (2 bytes): A signed integer value that specifies an index into the GUIDs that are listed in PlfguidUim.**rgguidUim**. This value MUST be greater than or equal to 0 and less than PlfguidUim.**iMac**. The referenced GUID specifies the **CLSID** of the service that provided this data.

fc (4 bytes): A signed integer that specifies an offset into the Table Stream. The data that is provided by the service which is identified by **iguidType** and **iclsidTip** begins at this offset. The size of this data, in bytes, is specified by **cb**. The meaning of this data is determined by the service that provided it.

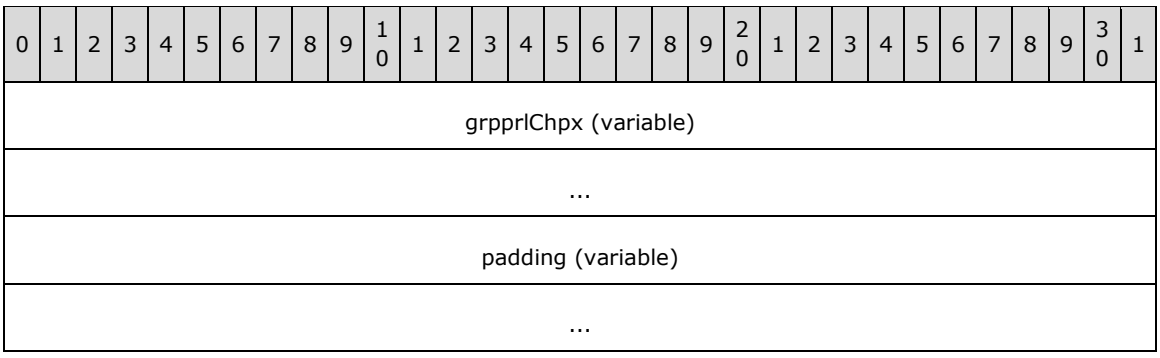
cch (4 bytes): A signed integer that specifies the size of text, in count of characters, which starts at the corresponding CP in the **plcfUim** value of the main document.

cb (4 bytes): An unsigned integer that specifies the size, in bytes, of the data at offset **fc** in the Table Stream.

dwPrivate (4 bytes): An unsigned integer that specifies the private data that is generated by the service which is identified by **iguidType** and **iclsidTip**.

2.9.336 UpxChpx

The **UpxChpx** structure specifies the character formatting properties that differ from the parent style as defined by **StdfBase**.**istdBase**.



grpprlChpx (variable): An array of **Prl** elements that specifies character formatting properties.

Copyright © 2025 Microsoft Corporation

This array MUST contain only character Sprm structures. However, this array MUST NOT contain any **Sprm** structure that specifies a property that is preserved across the application of the **sprmCIstd** value. Finally, this array MUST NOT contain any of the following:

1. sprmCFSpecVanish
2. sprmCIstd
3. sprmCIstdPermute
4. sprmCPlain
5. sprmCMajority
6. sprmCDispFldRMark
7. sprmCIdslRMarkDel
8. sprmCLbcCRJ
9. sprmCPbiIBullet
10. sprmCPbiGrf

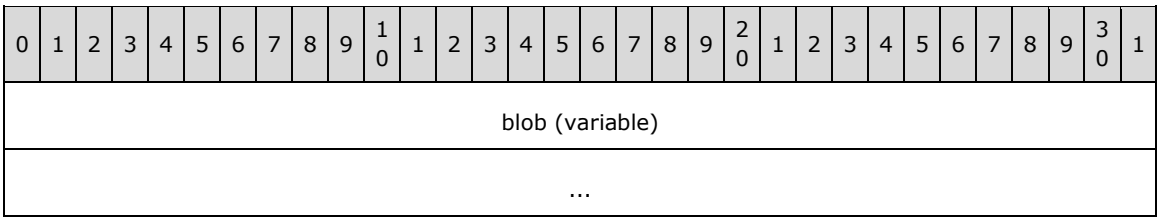
Additionally, character, paragraph, and list styles MUST NOT contain the sprmCCnf value.

padding (variable): A **UPXPadding** structure that specifies the padding that is required to make the **UpxChpx** structure an even length.

2.9.337 UPXPadding

The **UPXPadding** structure specifies the padding that is used to pad the **UpxPapx**, **UpxChpx**, or **UpxTapx** structure if any of them are an odd number of bytes in length. The number of bytes that are required MUST be written as a zero value.

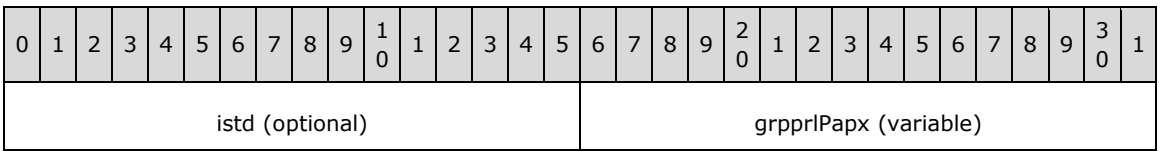
The **UpxPapx**, **UpxChpx**, and **UpxTapx** structures MUST be written as an even length, even if their contents are an odd length.



blob (variable): A structure that specifies any padding that is required to pad structures of an odd number of bytes in length so that they end on an even-byte boundary. It has a size of 1 byte if padding is needed, and 0 bytes if no padding is needed.

2.9.338 UpxPapx

The **UpxPapx** structure specifies the paragraph formatting properties that differ from the parent style, as defined by **StdfBase**.**istdBase**.



Copyright © 2025 Microsoft Corporation

...
padding (variable)
...

istd (2 bytes): An unsigned integer that specifies the **istd** value of the paragraph style. The **istd** value MUST be equal to the current style.

grpprlPapx (variable): An array of **Prl** elements that specify paragraph formatting properties. This array MUST contain only paragraph **Sprm** structures.

List styles MUST contain only the **sprmPIlfo** value.

Paragraph and table styles MUST NOT contain any **Sprm** structure that specifies a property that is preserved across the application of the **sprmPIstd** value. Additionally, paragraph and table styles MUST NOT contain any of the following:

- **sprmPIstd**
- **sprmPIstdPermute**
- **sprmPIncLvl**
- **sprmPNest80**
- **sprmPChgTabs**
- **sprmPDcs**
- **sprmPHugePapx**
- **sprmPFInnerTtp**
- **sprmPFOpenTch**
- **sprmPNest**
- **sprmPFNoAllowOverlap**
- **sprmPIstdListPermute**
- **sprmPTableProps**
- **sprmPTIstdInfo**

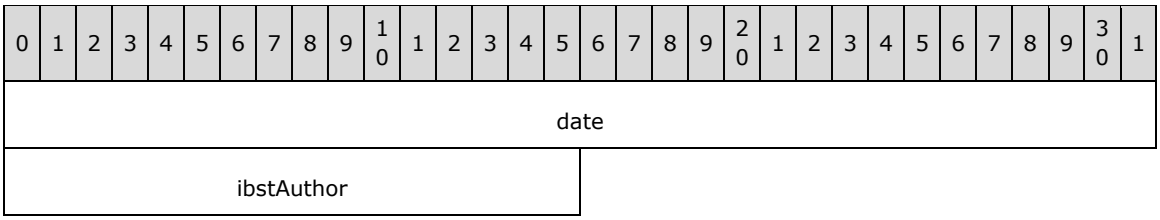
Additionally, paragraph styles MUST NOT contain sprmPCnf.

padding (variable): A **UPXPadding** value that specifies the padding that is required to make the **UpxPapx** structure an even length.

2.9.339 UpxRm

The **UpxRm** structure specifies that the style was revision-marked, and the date and author of the revision. A revision-marked style contains a set of formatting properties that specify the formatting of the style at the time that the style was modified for revision-marking.

Copyright © 2025 Microsoft Corporation

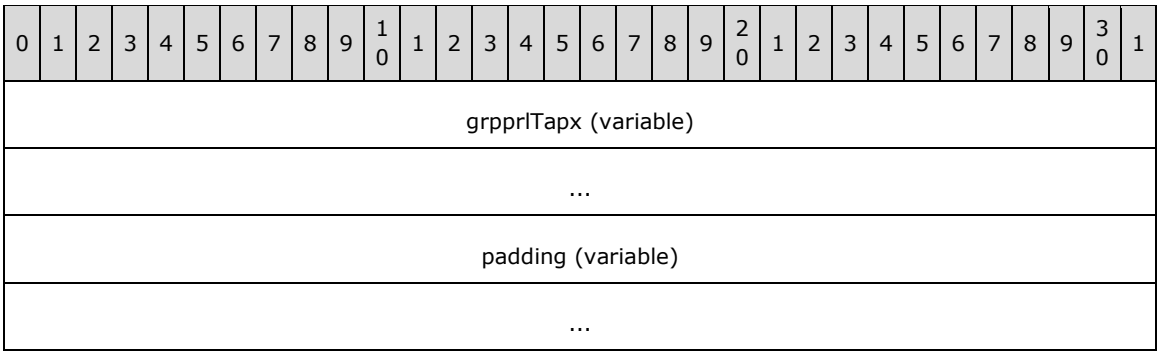


date (4 bytes): A DTTM that specifies the date and time at which this style revision occurred.

ibstAuthor (2 bytes): A signed integer that specifies the index location of the string in the **SttbfRMark** string table that describes the author who modified the style.

2.9.340 UpxTapx

The **UpxTapx** structure specifies the table formatting properties that differ from the parent style, as defined by the **StdfBase**.**istdBase** value.



grpprlTapx (variable): An array of **Prl** elements that specify table formatting properties. This array MUST contain only table **Sprm** structures.

Any **sprmTIstd** value that is contained in the array MUST be ignored.

This array MUST NOT contain the **sprmTWidthBefore** value, except when specifying the table formatting properties for the **table style** with an **istd** of 0x000B, which MUST contain a **sprmTWidthBefore** value with an **FtsWWidth_TablePart** operand that specifies a **ftsWidth** of ftsDxa (0x03) and a **wWidth** of zero.

Additionally, this array MUST NOT contain any **Sprm** structure that specifies a property that is preserved across the application of the **sprmTIstd** value.

Finally, this array MUST NOT contain any of the following:

1. sprmTDxaLeft
2. sprmTDefTable
3. sprmTDefTableShd80
4. sprmTDefTableShd3rd
5. sprmTDefTableShd
6. sprmTDefTableShd2nd
7. sprmTWidthAfter

Copyright © 2025 Microsoft Corporation

8. sprmTFKeepFollow
9. sprmTBrcTopCv
10. sprmTBrcLeftCv
11. sprmTBrcBottomCv
12. sprmTBrcRightCv
13. sprmTSetBrc80
14. sprmTInsert
15. sprmTDelete
16. sprmTDxaCol
17. sprmTMerge
18. sprmTSplit
19. sprmTTextFlow
20. sprmTVertMerge
21. sprmTVertAlign
22. sprmTSetBrc
23. sprmTCellPadding
24. sprmTCellWidth
25. sprmTFitText
26. sprmTFCellNoWrap
27. sprmTCellFHideMark
28. sprmTSetShdTable
29. sprmTCellBrcType
30. sprmTFBiDi90
31. sprmTFNoAllowOverlap
32. sprmTIpgp
33. sprmTDefTableShdRaw
34. sprmTDefTableShdRaw2nd
35. sprmTDefTableShdRaw3rd
36. sprmTCellBrcTopStyle (except within a sprmTCnf)
37. sprmTCellBrcBottomStyle (except within a sprmTCnf)
38. sprmTCellBrcLeftStyle (except within a sprmTCnf)
39. sprmTCellBrcRightStyle (except within a sprmTCnf)

Copyright © 2025 Microsoft Corporation

40. sprmTCellBrcInsideHStyle (except within a sprmTCnf)

41. sprmTCellBrcInsideVStyle (except within a sprmTCnf)

padding (variable): A **UPXPadding** value that specifies the padding that is required to make **UpxTapx** an even length.

2.9.341 VerticalAlign

The **VerticalAlign** enumeration specifies the vertical alignment of content within table cells.

Name	Value	Meaning
vaTop	0x00	Specifies that content is vertically aligned to the top of the cell.
vaCenter	0x01	Specifies that content is vertically aligned to the center of the cell.
vaBottom	0x02	Specifies that content is vertically aligned to the bottom of the cell.

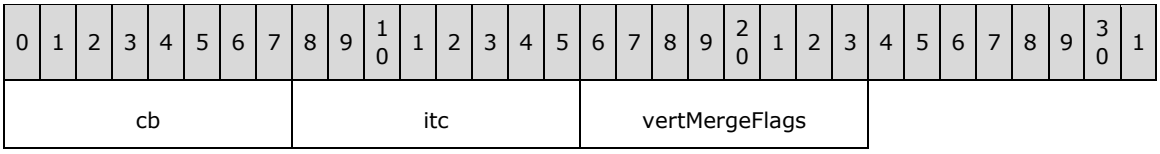
2.9.342 VerticalMergeFlag

The **VerticalMergeFlag** enumeration provides a 2-bit value that specifies whether a table cell is merged with the cells above or below it. This MUST be one of the following values.

Name	Value	Meaning
fvmClear	0x00	The cell is not merged with cells above or below it. This is the default behavior.
fvmMerge	0x01	The cell is one of a set of vertically merged cells. It contributes its layout region to the set and its own contents are not rendered.
fvmRestart	0x03	The cell is the first cell in a set of vertically merged cells. The contents and formatting of this cell extend down into any consecutive cells below it that are set to the **fvmMerge** value.

2.9.343 VertMergeOperand

The **VertMergeOperand** structure is an operand that specifies the merge behavior of a cell in a table row with the equivalent cells in the rows immediately above or below it.



cb (1 byte): An integer value that specifies the byte count of the remainder of this structure. This value MUST be 2.

itc (1 byte): An integer that specifies the index of a cell in the table row. The first cell has an index of zero. All cells in the row are counted, even if they are vertically merged with cells above or below them. This value MUST be a valid index of a cell in the table row.

vertMergeFlags (1 byte): A value from the **VerticalMergeFlag** enumeration that specifies whether this cell is vertically merged with the cells above or below it.

Copyright © 2025 Microsoft Corporation

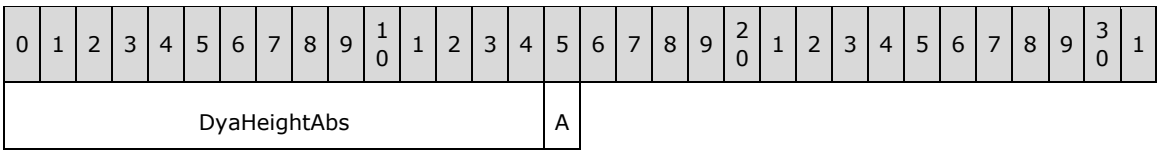
2.9.344 Vjc

The **Vjc** enumeration provides an 8-bit unsigned integer that specifies the vertical alignment of text.

Name	Value	Meaning
vjcTop	0x00	Top
vjcCenter	0x01	Centered
vjcBoth	0x02	Justified
vjcBottom	0x03	Bottom

2.9.345 WHeightAbs

The **WHeightAbs** structure specifies the frame height.

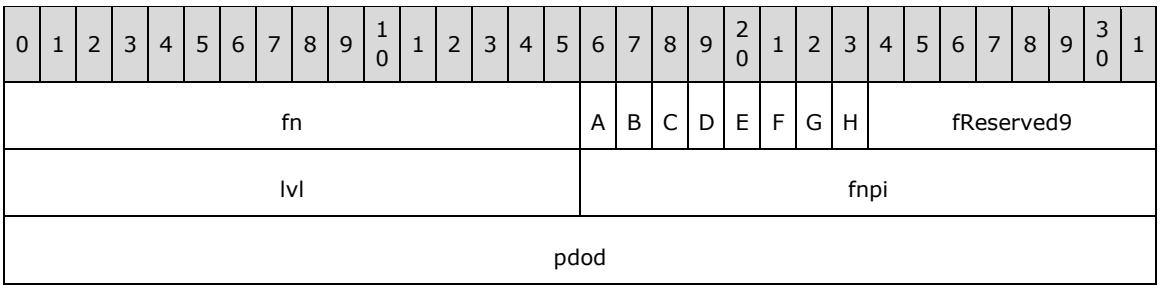


DyaHeightAbs (15 bits): A **YAS_nonNeg** value that specifies frame height. If this value is 0x0000, the frame height is automatically determined based on the height of its contents.

A - fMinHeight (1 bit): A bit that specifies whether **DyaHeightAbs** specifies minimum frame height. The **DyaHeightAbs** MUST NOT be 0x0000 when **fMinHeight** is set.

2.9.346 WKB

The **WKB** structure describes a **subdocument**.



fn (2 bytes): This value MUST be 0.

A - fReserved1 (1 bit): This value MUST be 0.

B - fReserved2 (1 bit): This value MUST be 0.

C - fReserved3 (1 bit): This value is undefined and MUST be ignored.

D - fReserved4 (1 bit): This value MUST be 0.

E - fReserved5 (1 bit): This value MUST be 0.

F - fReserved6 (1 bit): This value MUST be 1.

G - fReserved7 (1 bit): This value MUST be 0.

Copyright © 2025 Microsoft Corporation

H - fReserved8 (1 bit): This value is undefined and MUST be ignored.

fReserved9 (1 byte): This value MUST be 0.

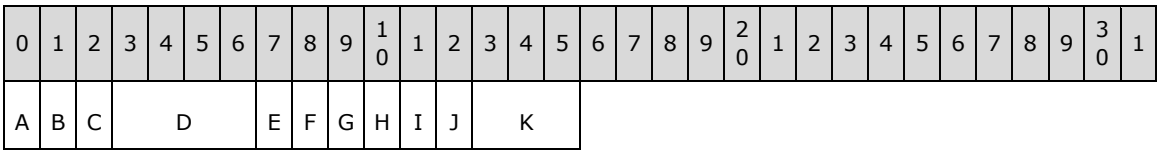
lvl (2 bytes): This value MUST be 0x0002.

fnpi (2 bytes): An **FNPI** structure that specifies the type and identifier of a file name. The string that is contained in the **SttbFnm** structure and that is appended by an **FNIF** structure that has an **fnpi** which is identical to this one, is the file name of the file that this **WKB** references.

pdod (4 bytes): This value is unused and MUST be zero.

2.9.347 Wpms

The **Wpms** structure specifies the current state of the mail merge.



A - wpmsMainDoc (1 bit): Specifies whether the main document was selected for the mail merge.

B - wpmsDF (1 bit): Specifies whether the data source was selected for the mail merge.

C - wpmsHF (1 bit): Specifies whether the mail merge obtains the merge field names from a header file.

D - wpmsType (4 bits): An unsigned integer that specifies the document type of the mail merge. This value MUST be one of the following.

Value	Meaning
0x0	No mail merge.
0x1	Letters.
0x2	Labels.
0x4	Envelopes.
0x8	Catalog or directory.

E - unused1 (1 bit): This bit is undefined and MUST be ignored.

F - wpmsAuto (1 bit): Specifies whether this is an automatic label or envelope mail merge.

G - unused2 (1 bit): This value MUST be 0 and MUST be ignored.

H - wpmsSuppression (1 bit): Specifies whether the blank lines in the data files MUST be suppressed.

I - wpmsRecSelect (1 bit): Specifies whether record selection is enabled.

J - unused3 (1 bit): This value MUST be 0 and MUST be ignored.

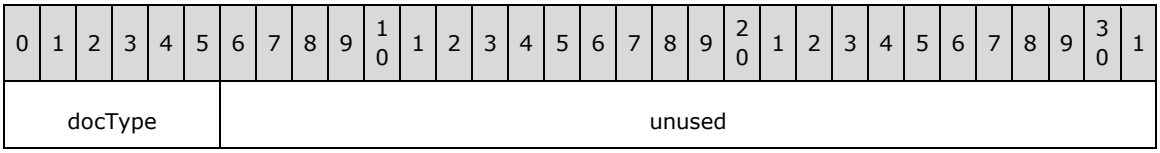
K - wpmsDest (3 bits): An unsigned integer that specifies the destination of the mail merge. This MUST be one of the following values.

Copyright © 2025 Microsoft Corporation

Value	Meaning
0x0	None
0x1	Printer
0x2	E-mail
0x4	Fax

2.9.348 Wpmsdt

A **Wpmsdt** structure specifies the document type of the mail merge.



docType (6 bits): An unsigned integer that specifies the document type of the mail merge. This MUST be one of the following values.

Value	Meaning
0x00	No mail merge.
0x01	Letters.
0x02	Labels.
0x04	Envelopes.
0x08	Catalog or directory.
0x10	E-mail messages.
0x20	Fax.

unused (26 bits): This field is undefined and MUST be ignored.

2.9.349 XAS

The **XAS** value is a 16-bit signed integer that specifies horizontal distance in **twips**. This value MUST be greater than or equal to -31680 and less than or equal to 31680.

2.9.350 XAS_nonNeg

The **XAS_nonNeg** value is a 16-bit unsigned integer that specifies horizontal distance, in **twips**. This value MUST be less than or equal to 31680.

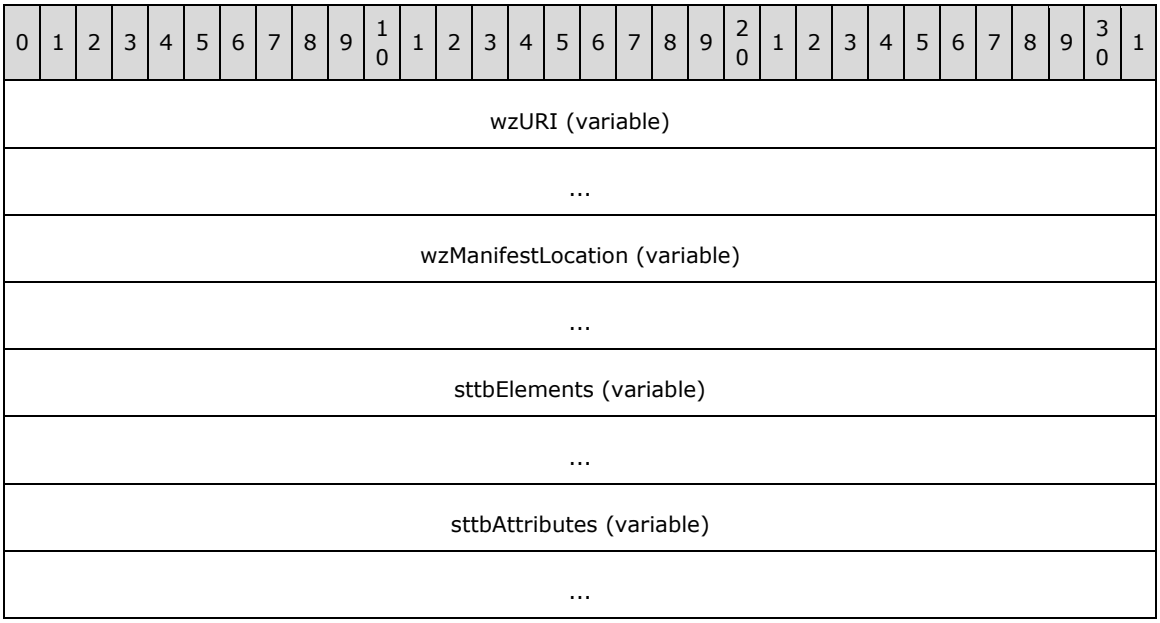
2.9.351 XAS_plusOne

The **XAS_plusOne** value is a 16-bit signed integer that specifies the horizontal distance, in **twips**, after the stored value is decremented by 1. This value MUST be greater than or equal to -31679 and less than or equal to 31681.

Copyright © 2025 Microsoft Corporation

2.9.352 XSDR

The **XSDR** structure specifies a single reference to an XML schema definition.



wzURI (variable): A **Unicode** string that indicates the **URI** of this schema definition. The string is length-prefixed with a 16-bit integer and is not null-terminated.

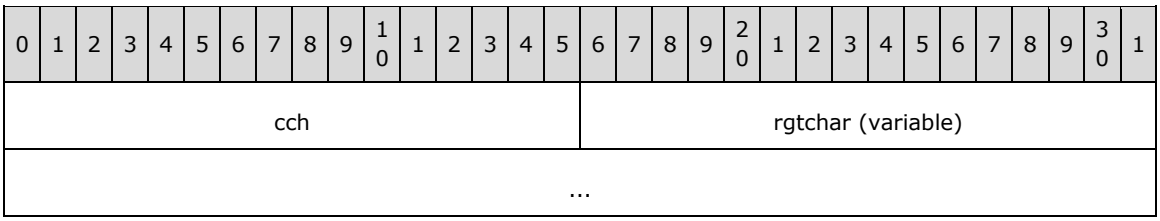
wzManifestLocation (variable): A Unicode string that is length-prefixed with a 16-bit integer and is not null-terminated. If this schema definition was loaded through an XML expansion pack, **wzManifestLocation** is the URI of the expansion pack **manifest**. If it was not loaded through an expansion pack, the string is empty.

sttbElements (variable): An **STTB** structure that contains all the elements within this XML schema. This structure uses a 4-byte **cData**.

sttbAttributes (variable): An **STTB** structure that contains all the attributes within this XML schema. This structure uses a 4-byte **cData**.

2.9.353 Xst

The **Xst** structure is a string. The string is prepended by its length and is not null-terminated.



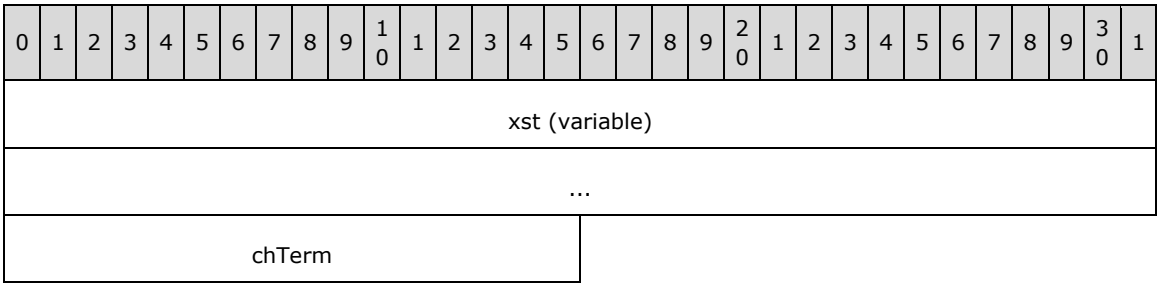
cch (2 bytes): An unsigned integer that specifies the number of characters that are contained in the **rgtchar** array.

rgtchar (variable): An array of 16-bit **Unicode** characters that make up a string.

Copyright © 2025 Microsoft Corporation

2.9.354 Xstz

The **Xstz** structure is a string. The string is prepended by its length and is null-terminated.



xst (variable): An **Xst** structure that is prepended with a value which specifies the length of the string.

chTerm (2 bytes): A null-terminating character. This value MUST be 0.

2.9.355 YAS

The **YAS** value is a 16-bit signed integer that specifies vertical distance in **twips**. This value MUST be greater than or equal to -31680 and less than or equal to 31680.

2.9.356 YAS_nonNeg

The **YAS_nonNeg** value is a 16-bit unsigned integer that specifies vertical distance, in **twips**. This value MUST be less than or equal to 31680.

2.9.357 YAS_plusOne

The **YAS_plusOne** value is a 16-bit signed integer that specifies vertical distance, in **twips**, after the stored value is decremented by 1. This value MUST be greater than or equal to -31679 and less than or equal to 31681.

Copyright © 2025 Microsoft Corporation

3 Structure Examples

This section contains examples of some of the most commonly used data structures in MS-DOC files. The examples are based on common computational tasks.

Section 3.1 provides examples of the data structures that are used to find the text of the document.

Section 3.2 provides examples of the data structures that are used to find the properties of a **section**, including page size and margins.

Section 3.3 provides examples of the data structures that are used to determine the ranges of **bookmarks**.

Sections 3.4 and 3.5 provide examples of the data structures that are used to determine direct character and paragraph formatting.

Section 3.6 provides an example of the use of the sprmTInsert value to define a table row and of the use of an **ItcFirstLim** structure to specify a range of cells to which a **Sprm** structure applies.

Finally, Section 3.7 provides an example of the data structures that are used to determine the formatting and **number text** of a list.

3.1 Example of a Clx

The following is an example of a **Clx**. This structure demonstrates the mapping between CP elements and the location of text in the file. See section 2.4.1, Retrieving Text.

Offset	Size	Structure	Value
0000009A	02E8	FibRgFcLcb97 **- rgFcLcb97**	
0000009A	0108	... (omitted for brevity) **-**	
000001A2	0004	**- fcClx**	0x000001F8
000001A6	0004	**- lcbClx**	0x0000002D
000001AA	01D8	... (omitted for brevity) **-**	

Figure 3: Portions of the FibRgFcLcb97 structure, with emphasis on fcClx and lcbClx

As with all Word Binary files, this file has a **Fib** at an offset of zero in the WordDocument Stream. The preceding figure shows a portion of the **FibRgFcLcb97** that is contained in that Fib. The **FibRgFcLcb97** is very large. Most fields have been omitted here, for brevity.

fcClx: 0x000001F8 specifies the offset, in bytes, of a location in the Table Stream. A **Clx** begins at this offset.

lcbClx: 0x0000002D specifies the size, in bytes, of the **Clx** at offset 0x000001F8 in the Table Stream.

The following shows the top level of the **Clx** at offset 0x000001F8 in the Table Stream.

Offset	Size	Structure	Value
000001F8	002D	Clx **- Clx**	
000001F8	0000	RgPrc **- RgPrc**	
000001F8	002D	Pcdt **- Pcdt**	

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
000001F8	0001	BYTE **- clxt**	0x02
000001F9	0004	ULONG **- lcb**	0x00000028
000001FD	0028	PlcPcd **- PlcPcd**	

Figure 4: A Clx structure

RgPrc: This optional member is not present in this **Clx** structure. Because the first byte of this **Clx** structure is 0x02, the **Clx** begins with a **Pcdt** structure and does not contain an array of **Prc** structures.

Pcdt.clxt: 0x02 specifies that this is a **Pcdt** structure, as opposed to a **Prc** structure.

Pcdt.lcb: 0x00000028 specifies the size, in bytes, of **PlcPcd**. A **PlcPcd** is a **Plc** structure whose data members are **Pcd** structures. A **Pcd** is 8 bytes in size, so this **PlcPcd** consists of three **Pcd** structures and four CP elements.

The following shows the top-level expansion of the **PlcPcd** that is contained in this **Clx**. The **Pcd** structures, which are expanded in later tables, specify the locations of text in the file.

Offset	Size	Structure	Value
000001FD	0028	PlcPcd **- PlcPcd**	
000001FD	0004	LONG **- cp[0]**	0x00000000
00000201	0004	LONG **- cp[1]**	0x00000006
00000205	0004	LONG **- cp[2]**	0x0000000D
00000209	0004	LONG **- cp[3]**	0x0000000E
0000020D	0008	Pcd **- pcd[0]**	
00000215	0008	Pcd **- pcd[1]**	
0000021D	0008	Pcd **- pcd[2]**	

Figure 5: The top-level expansion of a PlcPcd

cp[0]: 0x00000000 specifies that **pcd[0]** applies to text starting at CP zero. Because **cp[1]** is 0x0000006, **pcd[0]** applies to CP values zero through 5, inclusive.

cp[1]: 0x00000006 specifies that **pcd[1]** applies to text starting at CP 0x00000006. Because **cp[2]** is 0x0000000D, **pcd[1]** applies to CP values 0x00000006 through 0x0000000C, inclusive.

cp[2]: 0x0000000D specifies that **pcd[2]** applies to text starting at CP 0x0000000D. Because **cp[3]** is 0x0000000E, **pcd[2]** applies only to CP value 0x0000000D.

cp[3]: 0x0000000E specifies that the last CP value to which **pcd[2]** applies is 0x0000000D.

pcd[0]: Specifies the location of text for CP values zero through 5, inclusive. This structure is expanded in the following table.

pcd[1]: Specifies the location of text for CP values 0x00000006 through 0x0000000C, inclusive. This structure is expanded following.

pcd[2]: Specifies the location of text for CP value 0x0000000D. This structure is expanded following.

Copyright © 2025 Microsoft Corporation

The following table shows the expansion of **pcd[0]**. This structure specifies the location of the text at CP zero through 5, inclusive.

Offset	Size	Structure	Value
0000020D	0008	Pcd **- pcd**	
0000020D	1 bit	USHORT **- fNoParaLast**	0x1
0000020D	1 bit	USHORT **- fR1 (ignored)**	0x0
0000020D	1 bit	USHORT **- fDirty (ignored)**	0x0
0000020D	13 bits	USHORT **- fR2 (ignored)**	0x0006
0000020F	0004	FcCompressed **- fc**	
0000020F	30 bits	ULONG **- fc**	0x00000C22
0000020F	1 bit	ULONG **- fCompressed**	0x0
0000020F	1 bit	ULONG **- r1 (ignored)**	0x0
00000213	0002	Prm0 **- prm0**	
00000213	1 bit	USHORT **- fComplex**	0x0
00000213	7 bits	USHORT **- isprm**	0x00
00000213	8 bits	USHORT **- val**	0x00

Figure 6: The expansion of pcd[0]

fNoParaLast: 0x1 specifies that the text that is referenced by this **Pcd** structure does not contain any paragraph marks.

fc.fc: 0x00000C22 specifies the offset, in bytes, in the WordDocument Stream where the text at CP zero begins. Because **cp[1]** is 0x00000006, there are 6 characters of text at this offset.

fc.fCompressed: 0x0 specifies that the text at offset **fc.fc** in the WordDocument Stream consists of 16-bit Unicode characters.

prm0.fComplex: 0x0 specifies that this is a **Prm0** structure, as opposed to a **Prm1** structure.

prm0.isprm: 0x00 specifies that **sprmCLbcCRJ** is applied to the range of CPs that are referenced by this **Pcd** structure. However, an **isprm** of 0x0000, combined with a **val** of 0x0000, is a special case that specifies that the CPs that are referenced by this **Pcd** have no additional formatting from their **Pcd** structure

prm0.val: 0x00, combined with **isprm** 0x0000, specifies that the CPs that are referenced by this **Pcd** have no additional formatting from their **Pcd**.

The following shows the expansion of **pcd[1]**. This structure specifies the location of the text at CP 0x0000006 through 0x0000000C, inclusive.

Offset	Size	Structure	Value
00000215	0008	Pcd **- pcd**	
00000215	1 bit	USHORT **- fNoParaLast**	0x0
00000215	1 bit	USHORT **- fR1 (ignored)**	0x0

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
00000215	1 bit	USHORT **- fDirty (ignored)**	0x0
00000215	13 bits	USHORT **- fR2 (ignored)**	0x0006
00000217	0004	FcCompressed **- fc**	
00000217	30 bits	ULONG **- fc**	0x00000800
00000217	1 bit	ULONG **- fCompressed**	0x1
00000217	1 bit	ULONG **- r1 (ignored)**	0x0
0000021B	0002	Prm0 **- prm0**	
0000021B	1 bit	USHORT **- fComplex**	0x0
0000021B	7 bits	USHORT **- isprm**	0x00
0000021B	8 bits	USHORT **- val**	0x00

Figure 7: Expansion of pcd[1]

fNoParaLast: 0x0 specifies that the text that is referenced by this **Pcd** might contain a paragraph mark. A value of 0x0001 specifies that there is no paragraph mark. A value of 0x0000 specifies that the referenced text might or might not contain a paragraph mark.

fc.fc: 0x00000800 specifies the offset, in bytes, in the WordDocument Stream where the text at CP 0x00000006 begins. Because **fCompressed** is 1, the actual offset is **fc**/2, or 0x00000400. Because **cp[2]** is 0x0000000D, there are 7 characters at this offset.

fc.fCompressed: 0x1 specifies that the text at offset **fc**/2 consists of 8-bit ANSI characters, except for the values that are listed in the table in the specification of FcCompressed (section 2.9.73).

prm0.fComplex: 0x0 specifies that this is a **Prm0** structure, as opposed to a **Prm1** structure.

prm0.isprm: 0x00 specifies that sprmCLbcCRJ is applied to the range of CPs that are referenced by this **Pcd**. However, an **isprm** of 0x0000, combined with a **val** of 0x0000, is a special case that specifies that the CPs referenced by this **Pcd** have no additional formatting from their **Pcd**.

prm0.val: 0x00, combined with **isprm** 0x0000, specifies that the CPs that are referenced by this **Pcd** structure have no additional formatting from their **Pcd**.

The following shows the expansion of **pcd[2]**. This structure specifies the location of the text at CP 0x0000000D.

Offset	Size	Structure	Value
0000021D	0008	Pcd **- pcd**	
0000021D	1 bit	USHORT **- fNoParaLast**	0x0
0000021D	1 bit	USHORT **- fR1 (ignored)**	0x0
0000021D	1 bit	USHORT **- fDirty (ignored)**	0x0
0000021D	13 bits	USHORT **- fR2 (ignored)**	0x0006
0000021F	0004	FcCompressed **- fc**	
0000021F	30 bits	ULONG **- fc**	0x0000080E

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
0000021F	1 bit	ULONG **- fCompressed**	0x1
0000021F	1 bit	ULONG **- r1 (ignored)**	0x0
00000223	0002	Prm0 **- prm0**	
00000223	1 bit	USHORT **- fComplex**	0x0
00000223	7 bits	USHORT **- isprm**	0x00
00000223	8 bits	USHORT **- val**	0x00

Figure 8: Expansion of pcd[2]

fNoParaLast: 0x0 specifies that the text that is referenced by this **Pcd** might contain a paragraph mark. A value of 0x0001 specifies that there is no paragraph mark. A value of 0x0000 indicates that a paragraph mark might, or might not, be contained in the referenced text.

fc.fc: 0x0000080E specifies the offset, in bytes, in the WordDocument Stream where the text at CP 0x0000000D begins. Because **fCompressed** is 1, the actual offset is **fc**/2, or 0x00000407. Because **cp[3]** is 0x0000000E, there is 1 character at this offset.

fc.fCompressed: 0x1 specifies that the text at offset **fc**/2 consists of 8-bit ANSI characters, except for the values that are listed in the table in the specification of **FcCompressed** (section 2.9.73).

prm0.fComplex: 0x0 specifies that this is a **Prm0** structure, as opposed to a **Prm1** structure.

prm0.isprm: 0x00 specifies that sprmCLbcCRJ is applied to the range of CPs that are referenced by this **Pcd**. However, an **isprm** of 0x0000, combined with a **val** of 0x0000, is a special case that specifies that the CPs that are referenced by this **Pcd** have no additional formatting from their **Pcd**.

prm0.val: 0x00, combined with **isprm** 0x0000, specifies that the CPs that are referenced by this **Pcd** have no additional formatting from their **Pcd**.

The following shows the Unicode text at offset 0x00000C22 in the WordDocument Stream. This is an array of two-byte characters. This array is not null-terminated.

Offset	Size	Structure	Value
00000C22	000C	USHORT array **- text**	
00000C22	0002	USHORT **- text[0]**	0x0048
00000C24	0002	USHORT **- text[1]**	0x0065
00000C26	0002	USHORT **- text[2]**	0x006C
00000C28	0002	USHORT **- text[3]**	0x006C
00000C2A	0002	USHORT **- text[4]**	0x006F
00000C2C	0002	USHORT **- text[5]**	0x0020

Figure 9: The text at offset 0x00000C22 in the Table Stream

text[0]: 0x0048 Unicode 'H'.

text[1]: 0x0065 Unicode 'e'.

text[2]: 0x006C Unicode 'l'.

Copyright © 2025 Microsoft Corporation

text[3]: 0x006C Unicode 'l'.

text[4]: 0x006F Unicode 'o'.

text[5]: 0x0020 Unicode space.

The following shows the ANSI text at offset 0x00000400 in the WordDocument Stream. This is an array of single byte characters. This array is not null-terminated.

Offset	Size	Structure	Value
00000400	0007	BYTE array **- text**	
00000400	0001	BYTE **- text[0]**	0x57
00000401	0001	BYTE **- text[1]**	0x6F
00000402	0001	BYTE **- text[2]**	0x72
00000403	0001	BYTE **- text[3]**	0x6C
00000404	0001	BYTE **- text[4]**	0x64
00000405	0001	BYTE **- text[5]**	0x2E
00000406	0001	BYTE **- text[6]**	0x0D

Figure 10: The text at offset 0x00000400 in the WordDocument Stream

text[0]: 0x57 ANSI 'W'.

text[1]: 0x6F ANSI 'o'.

text[2]: 0x72 ANSI 'r'.

text[3]: 0x6C ANSI 'l'.

text[4]: 0x64 ANSI 'd'.

text[5]: 0x2E ANSI period ('.').

text[6]: 0x0D ANSI paragraph mark.

The following structure shows the ANSI text at offset 0x00000407 in the WordDocument Stream. This is an array of single byte characters. This array is not null-terminated.

Offset	Size	Structure	Value
00000407	0001	BYTE array **- text**	
00000407	0001	BYTE **- text[0]**	0x0D

Figure 11: The text at offset 0x00000407 in the WordDocument Stream

text[0]: 0x0D ANSI paragraph mark.

The complete text of this document is therefore, "Hello World", followed by a period and two paragraph marks.

Copyright © 2025 Microsoft Corporation

3.2 Example of a section

A document that is created by using this specification is divided into **sections**. Each section can store unique page-level formatting such as page size and orientation, in addition to other features such as headers and footers. A document contains at least 1 section.

PlcfSed contains information about how the document is divided into sections, as well as the properties of each section. The following is an example of a **PlcfSed** that was taken from a small document with two sections.

To find the **PlcfSed**, start in the FibRgFcLcb97.

Offset	Size	Structure	Value
0000009A	02E8	FibRgFcLcb97 **- rgFcLcb97**	
0000009A	0030	... (omitted for brevity) **-**	
000000CA	0004	ULONG **- fcPlcfSed**	0x000012D5
000000CE	0004	ULONG **- lcbPlcfSed**	0x00000024
000000D2	02B0	... (omitted for brevity) **-**	

Figure 12: Portions of the FibRgFcLcb97 structure, highlighting fc/lcbPlcfSed

The FibRgFcLcb97 structure is very large. Most fields have been omitted here for brevity.

fcPlcfSed: 0x000012D5 specifies that the **PlcfSed** structure begins at byte 0x12D5 in the Table Stream.

lcbPlcfSed: 0x00000024 specifies that the **PlcfSed** structure is 36 bytes long. Because each **Sed** structure is 12 bytes, the **PlcfSed** structure contains exactly three CPs and two **Sed** structures.

Using the offset and length that are specified by **fcPlcfSed** and **lcbPlcfSed**, read the **PlcfSed** structure, shown following.

Offset	Size	Structure	Value
000012D5	0024	PlcfSed **- PlcfSed**	
000012D5	0004	LONG **- cp[0]**	0x00000000
000012D9	0004	LONG **- cp[1]**	0x0000000B
000012DD	0004	LONG **- cp[2]**	0x00000016
000012E1	000C	Sed **- sed[0]**	
000012E1	0002	SHORT **- fn**	0x000D
000012E3	0004	ULONG **- fcSepx**	0x00000E00
000012E7	0002	SHORT **- fnMpr**	0x0000
000012E9	0004	ULONG **- fcMpr**	0xFFFFFFFF
000012ED	000C	Sed **- sed[1]**	
000012ED	0002	SHORT **- fn**	0x000D
000012EF	0004	ULONG **- fcSepx**	0x00000E2E

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
000012F3	0002	SHORT **- fnMpr**	0x0004
000012F5	0004	ULONG **- fcMpr**	0xFFFFFFFF

Figure 13: The PlcfSed structure that is referenced by fcPlcfSed and lcbPlcfSed in the FibRgFcLcb97 structure

This **PlcfSed** structure is 36 bytes long. Because each **Sed** structure is 12 bytes, this **PlcfSed** structure contains exactly 3 CPs and 2 **Sed** structures and from that information it can be determined that there are 2 sections.

cp[0]: 0x00000000 specifies that the text for the first section begins at position 0 in the main document.

cp[1]: 0x0000000B specifies that the text for the second section begins at position 11 in the main document. The last character in the first section is at position 10, and has a **Unicode** value of 0x0C.

cp[2]: 0x00000016 specifies that the remainder of this document is in the second section. The character position 21 does not need to be 0x0C, because no more sections follow it.

sed[0]: The **Sed** structure for the text range from **cp[0]** to **cp[1]**.

sed[0].fcSepx: 0x00000E00 specifies that the properties for the section are found at position 0x0E00 in the WordDocument Stream.

sed[0].fnMpr: 0x0000, **sed[0].fcMpr:** 0xFFFFFFFF, and **sed[0].fn:** 0x000D are ignored.

sed[1]: The **Sed** structure for the text range from **cp[1]** to **cp[2]**. Its **fcSepx** field specifies that the properties for the second section are a **Sepx** structure that begins at byte 0x00000E2E in the WordDocument Stream.

The details for **sed[1]** are very similar to **sed[0]**. They have been omitted for brevity.

Use the offset specified in **sed[0].fcSepx** to find the **Sepx** structure that contains the properties of the first section.

Offset	Size	Structure	Value
00000E00	002E	Sepx **- Sepx**	
00000E00	0002	USHORT **- cb**	0x002C
00000E02	002C	GrpPrlSepx **- grpprl**	
00000E02	0004	Prl **- prl[0]**	
00000E02	0002	Sprm **- sprmSDyaLinePitch**	0x9031
00000E04	0002	SHORT **- operand**	0x0168
00000E06	0004	Prl **- prl[1]**	
00000E06	0002	Sprm **- sprmSXaPage**	0xB01F
00000E08	0002	USHORT **- operand**	0x2FD0
00000E0A	0004	Prl **- prl[2]**	
00000E0A	0002	Sprm **- sprmSYaPage**	0xB020

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
00000E0C	0002	USHORT **- operand**	0x3DE0
00000E0E	0004	Prl **- prl[3]**	
00000E0E	0002	Sprm **- sprmSDxaLeft**	0xB021
00000E10	0002	USHORT **- operand**	0x05A0
00000E12	0004	Prl **- prl[4]**	
00000E12	0002	Sprm **- sprmSDxaRight**	0xB022
00000E14	0002	USHORT **- operand**	0x05A0
00000E16	0004	Prl **- prl[5]**	
00000E16	0002	Sprm **- sprmSDyaTop**	0x9023
00000E18	0002	SHORT **- operand**	0x05A0
00000E1A	0004	Prl **- prl[6]**	
00000E1A	0002	Sprm **- sprmSDyaBottom**	0x9024
00000E1C	0002	SHORT **- operand**	0x05A0
00000E1E	0004	Prl **- prl[7]**	
00000E1E	0002	Sprm **- sprmSDzaGutter**	0xB025
00000E20	0002	USHORT **- operand**	0x0000
00000E22	0004	Prl **- prl[8]**	
00000E22	0002	Sprm **- sprmSDyaHdrTop**	0xB017
00000E24	0002	USHORT **- operand**	0x02D0
00000E26	0004	Prl **- prl[9]**	
00000E26	0002	Sprm **- sprmSDyaHdrBottom**	0xB018
00000E28	0002	USHORT **- operand**	0x02D0
00000E2A	0004	Prl **- prl[10]**	
00000E2A	0002	Sprm **- sprmSDxaColumns**	0x900C
00000E2C	0002	SHORT **- operand**	0x02D0

Figure 14: The Sepx structure that is referenced by sed[0].fcSepx

cb: 0x002C specifies that there are a total of 44 bytes (not counting this cb) of properties that apply to section 1. Given only this information, it cannot be determined how many properties this represents, because property sizes vary from property to property.

grpprl.prl[0]: The first property. All properties contain a sprm to identify them and an operand which contains the property value.

grpprl.prl[0].sprmSDyaLinePitch: 0x9031 specifies that this is the section property sprmSDyaLinePitch and that the operand is two bytes.

Copyright © 2025 Microsoft Corporation

grpprl.prl[0].operand: 0x0168 specifies that the line height of the **document grid** in section 1 is 360 **twips** (0.25 inches)

grpprl.prl[1].sprmSXaPage: 0xB01F specifies that this is the section property sprmSXaPage and that the operand is two bytes.

grpprl.prl[1].operand: 0x2FD0 specifies that the page width for pages in section 1 is 12,240 twips (8.5 inches).

grpprl.prl[2].sprmSYaPage: 0xB020 specifies that this is the section property sprmSYaPage and that the operand is two bytes.

grpprl.prl[2].operand: 0x3DE0 specifies that the page height for pages in section 1 is 15,840 twips (11 inches).

grpprl.prl[3].sprmSDxaLeft: 0xB021 specifies that this is the section property sprmSDxaLeft and that the operand is two bytes.

grpprl.prl[3].operand: 0x05A0 specifies that the left margin for pages in section 1 is 1440 twips (1 inch) wide.

grpprl.prl[4].sprmSDxaRight: 0xB022 specifies that this is the section property sprmSDxaRight and that the operand is two bytes.

grpprl.prl[4].operand: 0x05A0 specifies that the right margin for pages in section 1 is 1440 twips (1 inch) wide.

grpprl.prl[5].sprmSDyaTop: 0x9023 specifies that this is the section property sprmSDyaTop and that the operand is two bytes.

grpprl.prl[5].operand: 0x05A0 specifies that the top margin for pages in section 1 is 1440 twips (1 inch) high.

grpprl.prl[6].sprmSDyaBottom: 0x9024 specifies that this is the section property sprmSDyaBottom and that the operand is two bytes.

grpprl.prl[6].operand: 0x05A0 specifies that the bottom margin for pages in section 1 is 1440 twips (1 inch) high.

grpprl.prl[7].sprmSDzaGutter: 0xB025 specifies that this is the section property sprmSDzaGutter and that the operand is two bytes.

grpprl.prl[7].operand: 0x0000 specifies that the **gutter margin** for pages in section 1 is 0 twips (0 inches) wide.

grpprl.prl[8].sprmSDyaHdrTop: 0xB017 specifies that this is the section property sprmSDyaHdrTop and that the operand is two bytes.

grpprl.prl[8].operand: 0x02D0 specifies that headers for pages in section 1 are positioned 720 twips (0.5 inches) from the top edge of the page.

grpprl.prl[9].sprmSDyaHdrBottom: 0xB018 specifies that this is the section property sprmSDyaHdrBottom and that the operand is two bytes.

grpprl.prl[9].operand: 0x02D0 specifies that footers for pages in section 1 are positioned 720 twips (0.5 inches) from the bottom edge of the page.

grpprl.prl[10].sprmSDxaColumns: 0x900C specifies that this is the section property sprmSDxaColumns and that the operand is two bytes.

Copyright © 2025 Microsoft Corporation

grpprl.prl[10].operand: 0x02D0 specifies that the spacing between columns, if there are multiple columns in section 1, is 720 twips (0.5 inches) wide.

Sed[0].fcSpex contains only some of the properties that apply to the **Sepx** structure. Properties that are not contained in **sed[0].fcSpex** take on their respective default values.

3.3 Example of a Bookmark

The following is an example of a standard **bookmark**. This structure demonstrates the mapping between the name of a bookmark, the CP of the first character of the bookmark, and the CP of the first character beyond the end of the bookmark.

Offset	Size	Structure	Value
0000009A	02E8	FibRgFcLcb97 **- rgFcLcb97**	
0000009A	0108	... (omitted for brevity) **-**	
00000142	0004	**- fcSttbfBkmk**	0x0000146B
00000146	0004	**- lcbSttbfBkmk**	0x0000004E
0000014A	0004	**- fcPlcfBkf**	0x000014B9
0000014E	0004	**- lcbPlcfBkf**	0x0000001C
00000152	0004	**- fcPlcfBkl**	0x000014D5
00000156	0004	**- lcbPlcfBkl**	0x00000010
000001AA	01D8	... (omitted for brevity) **-**	

Figure 15: Portions of the FibRgFcLcb97 structure, highlighting the three fc/lcb pairs for standard bookmarks

As with all MS-DOC files, this file has a **Fib** structure at offset zero in the WordDocument Stream. The preceding table shows a portion of the **FibRgFcLcb97** structure that is contained in that **Fib**. The **FibRgFcLcb97** structure is very large. Most fields have been omitted here, for brevity.

fcSttbfBkmk: 0x0000146B specifies the offset, in bytes, of a location in the Table Stream. An **SttbfBkmk** that contains the names of standard bookmarks in the document begins at this offset.

lcbSttbfBkmk: 0x0000004E specifies the size, in bytes, of the **SttbfBkmk** structure at offset 0x0000146B in the Table Stream.

fcPlcfBkf: 0x000014B9 specifies the offset, in bytes, of a location in the Table Stream. A **Plcfbkf** structure that contains information about standard bookmarks in the document begins at this offset. This **Plcfbkf** structure is parallel to the **SttbfBkmk** structure at offset **fcSttbfBkmk** in the Table Stream. Each data element in the **Plcfbkf** structure specifies information about the bookmark that is associated with the element which is located at the same offset in that **SttbfBkmk** structure.

lcbPlcfBkf: 0x0000001C specifies the size, in bytes, of the **Plcfbkf** structure at offset **fcPlcfBkf**.

fcPlcfBkl: 0x000014D5 specifies the offset, in bytes, of a location in the Table Stream. A **Plcfbkl** structure that contains information about standard bookmarks in the document begins at this offset. Each data element in the **Plcfbkl** structure is associated in a one-to-one correlation with a data element in the **Plcfbkf** structure at offset **fcPlcfBkf**.

lcbPlcfBkl: 0x00000010 specifies the size, in bytes, of the **Plcfbkl** structure at offset **fcPlcfBkl**.

Copyright © 2025 Microsoft Corporation

The following table shows the expansion of the **SttbfBkmk** structure at offset 0x0000146B in the Table Stream.

Offset	Size	Structure	Value
0000146B	004E	SttbfBkmk **- sttbfBkmk**	
0000146B	0002	USHORT **- fExtend**	0xFFFF
0000146D	0002	USHORT **- cData**	0x0003
0000146F	0002	USHORT **- cbExtra**	0x0000
00001471	0002	USHORT **- cchString[0]**	0x000B
00001473	0016	**- string[0]**	BookmarkThr
00001489	0002	USHORT **- cchString[1]**	0x000B
0000148B	0016	**- string[1]**	BookmarkTwo
000014A1	0002	USHORT **- cchString[2]**	0x000B
000014A3	0016	**- string[2]**	BookmarkOne

Figure 16: The expansion of an SttbfBkmk

fExtend: 0xFFFF specifies that the **string** fields in this **STTB** contain extended (2-byte) characters.

cData: 0x0003 specifies that this string table contains three elements.

cbExtra: 0x0000 specifies that there is no extra data appended to the **string** fields in this table.

cchString[0]: 0x000B specifies the count of characters in **string[0]**.

string[0]: BookmarkThr specifies the name of a bookmark (1) in the file.

cchString[1]: 0x000B specifies the count of characters in **string[1]**.

string[1]: BookmarkTwo specifies the name of a bookmark (1) in the file.

cchString[2]: 0x000B specifies the count of characters in **string[2]**.

string[2]: BookmarkOne specifies the name of a bookmark (1) in the file.

The following table shows the top-level expansion of the **Plcfbkf** at offset 0x000014B9 in the Table Stream. Each CP in the **Plcfbkf** specifies the location of the start of a bookmark in the document. Each **FBKF** specifies further information about the bookmark starting at the corresponding CP. The **FBKF** structures are expanded in later figures.

Offset	Size	Structure	Value
000014B9	001C	Plcfbkf **- PlcfBkf**	
000014B9	0004	LONG **- cp[0]**	0x00000000
000014BD	0004	LONG **- cp[1]**	0x0000000D
000014C1	0004	LONG **- cp[2]**	0x00000011
000014C5	0004	LONG **- cp[3]**	0x00000021
000014C9	0004	FBKF **- fbkf[0]**	

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
000014CD	0004	FBKF **- fbkf[1]**	
000014D1	0004	FBKF **- fbkf[2]**	

Figure 17: The top-level expansion of a Plcfbkf

cp[0]: 0x00000000 specifies the character position of the beginning of the bookmark associated with **fbkf[0]**. The same bookmark is associated with **string[0]** in the **SttbfBkmk** at offset **fcSttbfBkmk** in the Table Stream, so its name is "BookmarkThr".

cp[1]: 0x0000000D specifies the character position of the beginning of the bookmark associated with **fbkf[1]**. The same bookmark is associated with **string[1]** in the **SttbfBkmk** at offset **fcSttbfBkmk** in the Table Stream, so its name is "BookmarkTwo".

cp[2]: 0x00000011 specifies the character position of the beginning of the bookmark associated with **fbkf[2]**. The same bookmark is associated with **string[2]** in the SttbfBkmk at offset **fcSttbfBkmk** in the Table Stream, so its name is "BookmarkOne".

cp[3]: 0x00000021 specifies the value one greater than the largest value that a CP marking the start or end of a standard bookmark is allowed to have, which is one beyond the character position of the end of all document parts.

fbkf[0]: This value specifies further information about the bookmark named "BookmarkThr", whose range begins at CP 0x00000000. This structure is expanded in the following table.

fbkf[1]: This value specifies further information about the bookmark named "BookmarkTwo", whose range begins at CP 0x0000000D. This structure is expanded later.

fbkf[2]: This value specifies further information about the bookmark named "BookmarkOne", whose range begins at CP 0x00000011. This structure is expanded later.

The following table shows the expansion of **fbkf[0]** in the **Plcfbkf** structure at offset 0x000014B9 in the Table Stream.

Offset	Size	Structure	Value
000014C9	0004	FBKF **- fbkf**	
000014C9	0002	USHORT **- ibkl**	0x0002
000014CB	0002	BKC **- bkc**	
000014CB	0002	USHORT **- itcFirst**	0x0000
000014CD	0002	USHORT **- fPub**	0x0000
000014CF	0002	USHORT **- itcLim**	0x0000
000014D1	0002	USHORT **- fNative**	0x0000
000014D3	0002	USHORT **- fCol**	0x0000

Figure 18: Expansion of fbkf[0]

ibkl: A value of 0x0002 specifies a zero-based index into the **Plcfbkl** structure at offset 0x000014D5 in the Table Stream. The entry found at said index specifies the location of the end of the bookmark named "BookmarkThr".

bkc.itcFirst: A value of 0x0000 is ignored, because the value of the **fCol** value that belongs to this **BKC** structure is 0.

Copyright © 2025 Microsoft Corporation

bkc.fPub: A value of 0x0000 is ignored.

bkc.itcLim: A value of 0x0000 is ignored, because the value of the **fCol** value that belongs to this **BKC** structure is 0.

bkc.fNative: 0x0000 specifies that an application is expected to include the bookmark named "BookmarkThr" when saving its file as RTF (**Rich text** Format), HTML, or XML.

bkc.fCol: This value is 0x0000 because some of the text that is spanned by the bookmark named "BookmarkThr" is not inside a table, so the lowest table nesting depth within the span of text that is defined by its character positions is 0.

The following table shows the expansion of **fbkf[1]** in the **Plcfbkf** structure at offset 0x000014B9 in the Table Stream.

Offset	Size	Structure	Value
000014CD	0004	FBKF **- fbkf**	
000014CD	0002	USHORT **- ibkl**	0x0000
000014CF	0002	BKC **- bkc**	
000014CF	0002	USHORT **- itcFirst**	0x0001
000014D1	0002	USHORT **- fPub**	0x0000
000014D3	0002	USHORT **- itcLim**	0x0002
000014D5	0002	USHORT **- fNative**	0x0000
000014D7	0002	USHORT **- fCol**	0x0001

Figure 19: Expansion of fbkf[1]

ibkl: 0x0000 specifies a zero-based index into the **Plcfbkl** structure at offset 0x000014D5 in the Table Stream. The entry found at the index specifies the location of the end of the bookmark named "BookmarkTwo".

bkc.itcFirst: A value of 0x0001 specifies the zero-based index of the table column that is the start of the table column range associated with the bookmark named "BookmarkTwo".

bkc.fPub: A value of 0x0000 is ignored.

bkc.itcLim: A value of 0x0002 specifies that the zero-based index of the first column beyond the end of the table column range associated with the bookmark named "BookmarkTwo".

bkc.fNative: A value of 0x0000 specifies that an application is expected to include the bookmark named "BookmarkTwo" when saving its file as RTF (Rich text Format), HTML, or XML.

bkc.fCol: This value is 0x0001 because both of the following conditions hold:

- All of the text that is spanned by the bookmark named "BookmarkTwo" is inside a table, so the lowest table nesting depth within the span of text that is defined by its character positions is greater than 0.
- The span of text that is defined by the character positions of that bookmark contains a table cell mark from that table and nothing outside that table.

The following table shows the expansion of **fbkf[2]** in the **Plcfbkf** structure at offset 0x000014B9 in the Table Stream.

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
000014D1	0004	FBKF **- fbkf**	
000014D1	0002	USHORT **- ibkl**	0x0001
000014D3	0002	BKC **- bkc**	
000014D3	0002	USHORT **- itcFirst**	0x0000
000014D5	0002	USHORT **- fPub**	0x0000
000014D7	0002	USHORT **- itcLim**	0x0000
000014D9	0002	USHORT **- fNative**	0x0000
000014DB	0002	USHORT **- fCol**	0x0000

Figure 20: Expansion of fbkf[2]

ibkl: A value of 0x0001 specifies a zero-based index into the **Plcfbkl** structure at offset 0x000014D5 in the Table Stream. The entry found at the index specifies the location of the end of the bookmark named "BookmarkOne".

bkc.itcFirst: A value of 0x0000 is ignored, because the value of the **fCol** that belongs to this **BKC** is 0.

bkc.fPub: A value of 0x0000 is ignored.

bkc.itcLim: A value of 0x0000 is ignored, because the value of the **fCol** that belongs to this **BKC** is 0.

bkc.fNative: A value of 0x0000 specifies that an application is expected to include the bookmark named "BookmarkOne" when saving its file as RTF (Rich text Format), HTML, or XML.

bkc.fCol: This value is 0x0000 because some of the text spanned by the bookmark named "BookmarkOne" is not inside a table, so the lowest table nesting depth within the span of text defined by its character positions is 0.

The following table shows the top-level expansion of the **Plcfbkl** structure at offset 0x000014D5 in the Table Stream. Each CP in the **Plcfbkl** structure specifies the location of the end of a bookmark (1) in the document.

Offset	Size	Structure	Value
000014D5	0010	Plcfbkl **- plcfBkl**	
000014D5	0004	LONG **- cp[0]**	0x00000016
000014D9	0004	LONG **- cp[1]**	0x0000001B
000014DD	0004	LONG **- cp[2]**	0x0000001E
000014E1	0004	LONG **- cp[3]**	0x00000021

Figure 21: The expansion of a Plcfbkl

cp[0]: A value of 0x00000016 specifies the character position that is 1 beyond the end of the bookmark associated with **fbkf[1]** in the **Plcfbkf** structure at offset **fcPlcfBkf** in the Table Stream, whose name is "BookmarkTwo". This CP is known to be associated with **fbkf[1]** because **fbkf[1].ibkl** is 0.

cp[1]: A value of 0x0000001B specifies the character position that is 1 beyond the end of the bookmark associated with **fbkf[2]** in the **Plcfbkf** structure at offset **fcPlcfBkf** in the Table

Copyright © 2025 Microsoft Corporation

Stream, whose name is "BookmarkOne". This CP is known to be associated with **fbkf[2]** because **fbkf[2].ibkl** is 1.

cp[2]: A value of 0x0000001E specifies the character position that is 1 beyond the end of the bookmark associated with **fbkf[0]** in the **Plcfbkf** structure at offset **fcPlcfBkf** in the Table Stream, whose name is "BookmarkThr". This CP is known to be associated with **fbkf[0]** because **fbkf[0].ibkl** is 2.

cp[3]: A value of 0x00000021 specifies a value that is 1 greater than the largest value that a CP marking the start or end of a standard bookmark is allowed to have, which is 1 beyond the character position of the end of all document parts.

3.4 Example of a PlcBteChpx

The following is an example of a **PlcBteChpx** structure. It demonstrates how to apply character formatting properties to text in a document. See section 2.4.6.2, Direct Character Formatting.

Offset	Size	Structure	Value
0000009A	02E8	FibRgFcLcb97 **- rgFcLcb97**	
0000009A	0060	... (omitted for brevity) **-**	
000000FA	0004	**- fcPlcfBteChpx**	0x000000D6
000000FE	0004	**- lcbPlcfBteChpx**	0x0000000C
00000102	0280	... (omitted for brevity) **-**	

Figure 22: Portions of the FibRgFcLcb97 structure, highlighting fc/lcbPlcfBteChpx

The FibRgFcLcb97 structure is very large. Most fields have been omitted here for brevity.

fcPlcfBteChpx: A value of 0x000000D6 specifies the offset, in bytes, of a location in the Table Stream. A **PlcBteChpx** structure begins at this offset.

lcbPlcfBteChpx: A value of 0x0000000C specifies the size, in bytes, of the PlcBteChpx at offset 0x000000D6 in the Table Stream. Because each **PnFkpChpx** structure is four bytes, this **PlcBteChpx** structure contains exactly two CPs and one **PnFkpChpx** structures.

The following table shows the top level of the **PlcBteChpx** at offset 0x000000D6 in the Table Stream.

Offset	Size	Structure	Value
000000D6	000C	PlcBteChpx **- PlcBteChpx**	
000000D6	0004	LONG **- fc[0]**	0x00000400
000000DA	0004	LONG **- fc[1]**	0x00000411
000000DE	0004	PnFkpChpx **- pn[0]**	
000000DE	22 bits	LONG **- pn**	0x000003
000000DE	10 bits	LONG **- unused**	0x000

Figure 23: A PlcBteChpx

fc[0]: 0x00000400 specifies the offset in the WordDocument Stream where a text range begins. This is the first and only text range that is specified; this is evident because there are only two FCs.

Copyright © 2025 Microsoft Corporation

fc[1]: 0x00000411 specifies the offset in the WordDocument Stream immediately after the end of the text range. Because the text is 8-bit ANSI (see Section 2.4.1, Retrieving Text) the end of the text range is 0x410. If this document had more than one text range, 0x00000411 would also specify the start of the next text range.

pn[0].pn: 0x00000003 specifies the offset in the WordDocument Stream of the **ChpxFkp** structure that is applied to the text range. This **ChpxFkp** structure is referred to as **chpxfkp[0]**. The **chpxfkp[0]** element begins at offset 3 * 512 = 1536 = 0x00000600. See the following table for the expansion of **chpxfkp[0]**.

pn[0].unused: Undefined and ignored.

The following table shows the expansion of **chpxfkp[0]**, which specifies the character formatting properties for the first and only text range in the document.

Offset	Size	Structure	Value
00000600	0200	ChpxFkp **- chpxfkp[0]**	
00000600	0010	Array of ULONG **- rgfc**	
00000600	0004	ULONG **- rgfc[0]**	0x00000400
00000604	0004	ULONG **- rgfc[1]**	0x00000407
00000608	0004	ULONG **- rgfc[2]**	0x00000410
0000060C	0004	ULONG **- rgfc[3]**	0x00000411
00000610	0003	Array of BYTE **- rgb**	
00000610	0001	BYTE **- rgb[0]**	0xFA
00000611	0001	BYTE **- rgb[1]**	0xF8
00000612	0001	BYTE **- rgb[2]**	0x00
000007FF	0001	BYTE **- crun**	0x03

Figure 24: Expansion of chpxfkp[0]

rgfc.rgfc[0]: A value of 0x00000400 specifies the offset in the WordDocument Stream at which the first text run in the text range begins. This text run ends at 0x00000406, immediately before the start of the next run, and includes the text "Orange".

rgfc.rgfc[1]: A value of 0x00000407 specifies the offset in the WordDocument Stream at which the second text run in the text range begins. This text run ends at 0x0000040F, immediately before the start of the next run, and includes the text "Underline".

rgfc.rgfc[2]: A value of 0x00000410 specifies the offset in the WordDocument Stream at which the third text run in the text range begins. This text run ends at 0x00000410, and is therefore a single character, which is a paragraph marker.

rgfc.rgfc[3]: A value of 0x00000411 specifies the offset in the WordDocument Stream immediately after the end of the third text run in the text range.

rgb.rgb[0]: A value of 0xFA specifies the offset of the Chpx for the first text run, referred to as **chpx[0]** (see its expansion later). The **chpx[0]** element is 2 * 0xFA = 0x1F4 bytes from the start of **chpxfkp[0]**, or 0x600 + 0x1F4 = 0x7F4 bytes from the start of the Table Stream.

Copyright © 2025 Microsoft Corporation

rgb.rgb[1]: A value of 0xF8 specifies the offset of the Chpx for the second text run, referred to as **chpx[1]** (see its expansion later). The **chpx[1]** element is 2 * 0xF8 = 0x1F0 bytes from the start of **chpxfkp[0]**, or 0x600 + 0x1F0 = 0x7F0 bytes from the start of the Table Stream.

rgb.rgb[2]: A value of 0x00 specifies that there are no character properties associated with the third text run.

crun: A value of 0x03 specifies the number of runs in this text range. This is equal to the number of elements in **rgb**, and is 1 less than the number of elements in **rgfc**.

The following table shows the expansion of the **chpx[0]** element, which specifies the character property information for the first text run of the text range.

Offset	Size	Structure	Value
000007F4	000A	Chpx **- chpx[0]**	
000007F4	0001	BYTE **- cb**	0x09
000007F5	0009	Array of Prl **- GrpPrl**	
000007F5	0003	Prl **- GrpPrl[0]**	
000007F8	0006	Prl **- GrpPrl[1]**	

Figure 25: Expansion of chpx[0]

cb: A value of 0x09 specifies that **GrpPrl** is 9 bytes long.

GrpPrl: The array of properties being applied.

GrpPrl.GrpPrl[0]: The first property that is being applied. See the **chpx[0].GrpPrl.GrpPrl[0]** element that is described later in this document.

GrpPrl.GrpPrl[1]: The second property that is being applied. See the **chpx[0].GrpPrl.GrpPrl[1]** element that is described later in this document. The fact that there are no more bytes left in the **GrpPrl** element after this property is read indicates that there are no more properties.

The **chpx[0]** element contains some of the properties that apply to the first run of the text range. The properties that are not contained in **chpx[0]** take on their respective default values.

The following table shows the expansion of the **chpx[0].GrpPrl.GrpPrl[0]** element, which is the first property that is applied to the first text run ("Orange "). It applies a color to the text.

Offset	Size	Structure	Value
000007F5	0003	Prl **- chpx[0].GrpPrl.GrpPrl[0]**	
000007F5	0002	Sprm **- sprm**	
000007F5	9 bits	USHORT **- ispmd**	0x042
000007F5	1 bit	USHORT **- fSpec**	0x1
000007F5	3 bits	USHORT **- sgc**	0x2
000007F5	3 bits	USHORT **- spra**	0x1
000007F7	0001	Ico **- operand**	
000007F7	0001	BYTE **- value**	0x07

Figure 26: Expansion of chpx[0].GrpPrl.GrpPrl[0]

Copyright © 2025 Microsoft Corporation

sprm: The property being modified.

sprm.ispmd: If **ispmd** is equal to 0x0042 and **fSpec** is equal to 0x0001, this property has a value of sprmCIco.

sprm.sgc: A value of 0x2 specifies that this is a character property.

sprm.spra: A value of 0x1 specifies that **operand** is 1 byte long.

operand: The property value, which is an RGB color value that is expressed by an **Ico** structure.

operand.value: A value of 0x07 specifies that the text color will be represented using RGB (0xFF, 0xFF, 0x00) values.

The following table shows the expansion of the **chpx[0].GrpPrl.GrpPrl[1]** element, which is the second property that is applied to the first text run ("Orange"). It also applies a color to the text. Because this property occurs after the occurrence of sprmCIco, the color it specifies takes precedence.

Offset	Size	Structure	Value
000007F8	0006	Prl **- chpx[0].GrpPrl.GrpPrl[1]**	
000007F8	0002	Sprm **- sprm**	
000007F8	9 bits	USHORT **- ispmd**	0x070
000007F8	1 bit	USHORT **- fSpec**	0x0
000007F8	3 bits	USHORT **- sgc**	0x2
000007F8	3 bits	USHORT **- spra**	0x3
000007FA	0004	COLORREF **- operand**	
000007FA	0001	BYTE **- red**	0xFF
000007FB	0001	BYTE **- green**	0x99
000007FC	0001	BYTE **- blue**	0x00
000007FD	0001	BYTE **- fAuto**	0x00

Figure 27: Expansion of chpx[0].GrpPrl.GrpPrl[1]

sprm: The property that is being modified.

sprm.ispmd: If **ispmd** is equal to 0x0070 and **fSpec** is equal to 0x0000, the value of this property is sprmCCv.

sprm.sgc: A value of 0x2 specifies that this is a character property.

sprm.spra: A value of 0x3 specifies that **operand** is four bytes long.

operand: The property value, which is an RGB color value that is expressed by a COLORREF.

operand.red: A value of 0xFF specifies the red component of the RGB value.

operand.green: A value of 0x99 specifies the green component of the RGB value.

operand.blue: A value of 0x00 specifies the blue component of the RGB value.

operand.fAuto: A value of 0x00 specifies that the RGB value will be used as specified.

Copyright © 2025 Microsoft Corporation

The following table shows the expansion of the **chpx[1]** element, which specifies the character property information for the second text run of the text range ("Underline").

Offset	Size	Structure	Value
000007F0	0004	Chpx **- chpx[1]**	
000007F0	0001	BYTE **- cb**	0x03
000007F1	0003	GrpPrlChpx **- GrpPrl**	
000007F1	0003	Prl **- GrpPrl[0]**	

Figure 28: Expansion of chpx[1]

cb: A value of 0x03 specifies that **GrpPrl** is 3 bytes long.

GrpPrl: The array of properties that is being applied.

GrpPrl.GrpPrl[0]: The first and only property that is being applied. See the **chpx[1].GrpPrl.GrpPrl[0]** element in the following table.

The **chpx[1]** element contains only some of the properties that apply to the second run of the text range. The properties that are not contained in the **chpx[1]** element take on their respective default values.

The following table shows the expansion of the **chpx[1].GrpPrl.GrpPrl[0]** value, which is the first and only property that is applied to the second text run.

Offset	Size	Structure	Value
000007F1	0003	Prl **- chpx[1].GrpPrl.GrpPrl[0]**	
000007F1	0002	Sprm **- sprm**	
000007F1	9 bits	USHORT **- ispmd**	0x03E
000007F1	1 bit	USHORT **- fSpec**	0x1
000007F1	3 bits	USHORT **- sgc**	0x2
000007F1	3 bits	USHORT **- spra**	0x1
000007F3	0001	Kul **- operand**	0x01

Figure 29: Expansion of chpx[1].GrpPrl.GrpPrl[0]

sprm: The property that is being modified.

sprm.ispmd: If **ispmd** is equal to 0x003E and **fSpec** is equal to 0x0001, the value of this property is sprmCKul.

sprm.sgc: A value of 0x2 specifies that this is a character property.

sprm.spra: A value of 0x1 specifies that **operand** is 1 byte long.

operand: A value of 0x01 specifies that the text has a single underline.

3.5 Example of a PlcBtePapx

The following is an example of a **PlcBtePapx**. This example demonstrates how to apply paragraph properties to text in a document. See section 2.4.6.1, Direct Paragraph Formatting.

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
0000009A	02E8	FibRgFcLcb97 **- rgFcLcb97**	
0000009A	0060	... (omitted for brevity) **-**	
00000102	0004	**- fcPlcfBtePapx**	0x0000010E
00000106	0004	**- lcbPlcfBtePapx**	0x0000000C
0000010A	0278	... (omitted for brevity) **-**	

Figure 30: Portions of the FibRgFcLcb97 structure, highlighting fc/lcbPlcfBtePapx

The FibRgFcLcb97 structure is very large. Most fields are omitted for reasons of brevity.

fcPlcfBtePapx: A value of 0x0000010E specifies the offset, in bytes, of a location in the Table Stream. A **PlcBtePapx** structure begins at this offset.

lcbPlcfBtePapx: A value of 0x0000000C specifies the size, in bytes, of the **PlcBtePapx** at offset 0x0000010E in the Table Stream. Because each **PnFkpPapx** structure is 4 bytes, this **PlcBtePapx** structure contains exactly two CPs and one **PnFkpPapx**.

The following table shows the top level of the **PlcBtePapx** structure at offset 0x0000010E in the Table Stream.

Offset	Size	Structure	Value
0000010E	000C	PlcBtePapx **- PlcBtePapx**	
0000010E	0004	LONG **- fc[0]**	0x00000400
00000112	0004	LONG **- fc[1]**	0x0000040B
00000116	0004	PnFkpPapx **- pn[0]**	
00000116	22 bits	LONG **- pn**	0x000004
00000116	10 bits	LONG **- unused**	0x000

Figure 31: A PlcBtePapx

fc[0]: A value of 0x00000400 specifies the offset in the WordDocument Stream at which a text range begins. The fact that there are only two FCs indicates that this is the first and only text range that is specified.

fc[1]: A value of 0x0000040B specifies the offset in the WordDocument Stream immediately after the end of the text range. Because the text is 8-bit ANSI (see Section 2.4.1, Retrieving Text), the end of the text range is 0x40A. If this document had more than one text range, 0x0000040B would also specify the start of the next text range.

pn[0].pn: A value of 0x00000004 specifies the offset in the WordDocument Stream of the **PapxFkp** structure that is applied to any paragraph in the document which ends within the text range. This **PapxFkp** element is referred to as **papxfkp[0]**. The **papxfkp[0]** element begins at offset 4 * 512 = 2048 = 0x00000800. See the following table for the expansion of the **papxfkp[0]** element.

pn[0].unused: This value is undefined and ignored.

The following table shows the expansion of the **papxfkp[0]** element, which specifies the paragraph formatting properties for all paragraphs ending in the first and only text range in the document. In this example all paragraphs in the document start and end within this text range.

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
00000800	0200	PapxFkp **- papxfkp[0]**	
00000800	0010	Array of ULONG **- rgfc**	
00000800	0004	ULONG **- rgfc[0]**	0x00000400
00000804	0004	ULONG **- rgfc[1]**	0x00000405
00000808	0004	ULONG **- rgfc[2]**	0x0000040A
0000080C	0004	ULONG **- rgfc[3]**	0x0000040B
00000810	0027	Array of BxPap **- rgbx**	
00000810	000D	BxPap **- rgbx[0]**	
00000810	0001	BYTE **- bOffset**	0xFA
00000811	000C	FixedBlob **- reserved**	00 00 00 00 00 00 00 00 00 00 00 00
0000081D	000D	BxPap **- rgbx[1]**	
0000081D	0001	BYTE **- bOffset**	0xF6
0000081E	000C	FixedBlob **- reserved**	00 00 00 00 00 00 00 00 00 00 00 00
0000082A	000D	BxPap **- rgbx[2]**	
0000082A	0001	BYTE **- bOffset**	0xF4
0000082B	000C	FixedBlob **- reserved**	00 00 00 00 00 00 00 00 00 00 00 00
000009FF	0001	BYTE **- cpara**	0x03

Figure 32: Expansion of papxfkp[0]

rgfc.rgfc[0]: A value of 0x00000400 specifies the offset in the WordDocument Stream at which the first paragraph in the text range begins. This paragraph ends at offset 0x00000404, and spans the text "Test" followed by a newline character.

rgfc.rgfc[1]: A value of 0x00000405 specifies the offset in the WordDocument Stream at which the second paragraph in the text range begins. This paragraph ends at 0x00000409, immediately before the start of the next paragraph, and includes the text "Test" followed by a newline character.

rgfc.rgfc[2]: A value of 0x0000040A specifies the offset in the WordDocument Stream at which the third paragraph in the text range begins. This paragraph ends at 0x0000040A, and is therefore a single character, which is a newline character.

rgfc.rgfc[3]: A value of 0x0000040B specifies the offset in the WordDocument Stream immediately after where the third paragraph in the text range ends.

rgbx.rgbx[0].bOffset: A value of 0xFA specifies the offset of the **PapxInFkp** structure for the first paragraph, referred to as **papxinfkp[0]** (see the following table for its expansion). The **papxinfkp[0]** element is 2 * 0xFA = 0x1F4 bytes from the start of the **papxfkp[0]** element, or 0x800 + 0x1F4 = 0x9F4 bytes from the start of the Table Stream.

rgbx.rgbx[0].reserved: This value is ignored.

rgbx.rgbx[1].bOffset: A value of 0xF6 specifies the offset of the **PapxInFkp** for the second paragraph, referred to as **papxinfkp[1]** (see its expansion later). The **papxinfkp[1]** element is 2

Copyright © 2025 Microsoft Corporation

* 0xF6 = 0x1EC bytes from the start of the **papxfkp[1]** element, or 0x800 + 0x1EC = 0x9EC bytes from the start of the Table Stream.

rgbx.rgbx[1].reserved: This value is ignored.

rgbx.rgbx[2].bOffset: A value of 0xF4 specifies the offset of the PapxInFkp for the first paragraph, referred to as **papxinfkp[2]** (see the following expansion of this element). The **papxinfkp[2]** element is 2 * 0xF4 = 0x1E8 bytes from the start of the **papxfkp[2]** element, or 0x800 + 0x1E8 = 0x9E8 bytes from the start of the Table Stream.

rgbx.rgbx[2].reserved: This value is ignored.

cpara: A value of 0x03 specifies the number of paragraphs in this text range. This is equal to the number of elements in **papxfkp[0].rgbx**, and 1 less than the number of elements in **papxfkp[0].rgfc**.

The following table shows the expansion of the **papxinfkp[0]** element, which specifies the paragraph property information for the first paragraph of the text range.

Offset	Size	Structure	Value
000009F4	000A	PapxInFkp **- papxinfkp[0]**	
000009F4	0001	BYTE **- cb**	0x00
000009F5	0001	BYTE **- cb'**	0x04
000009F6	0008	GrpPrlAndIstd **- GrpPrl**	
000009F6	0002	USHORT **- istd**	0x0000
000009F8	0003	Prl **- GrpPrl[0]**	
000009FB	0003	Prl **- GrpPrl[1]**	

Figure 33: Expansion of papxinfkp[0]

cb: A value of 0x00 specifies that size of **GrpPrl** is determined by **cb'**.

cb': A value of 0x04 specifies that there are 2 * 4 = 8 bytes in **GrpPrl**.

GrpPrl.istd: A value of 0x0000 specifies that the Normal style will be applied to this paragraph. See Section 2.4.6.5, Determining Properties of a Style.

GrpPrl.GrpPrl[0]: The first property that is being applied. See the **papxinfkp[0].GrpPrl.GrpPrl[0]** element in the following table.

GrpPrl.GrpPrl[1]: The second property that is being applied. See the **papxinfkp[0].GrpPrl.GrpPrl[1]** element that follows.

The **papxinfkp[0]** element contains only some of the properties that apply to the first paragraph of the text range. The properties that are not contained in the **papxinfkp[0]** element take on their respective default values.

The following table shows the expansion of the **papxinfkp[0].GrpPrl.GrpPrl[0]** element, which is the first property that is applied to the first paragraph ("Test" followed by a newline character). This element specifies that the paragraph will be center-justified.

Offset	Size	Structure	Value
000009F8	0003	Prl **- papxinfkp[0].GrpPrl.GrpPrl[0]**	

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
000009F8	0002	Sprm **- sprm**	
000009F8	9 bits	USHORT **- ispmd**	0x003
000009F8	1 bit	USHORT **- fSpec**	0x0
000009F8	3 bits	USHORT **- sgc**	0x1
000009F8	3 bits	USHORT **- spra**	0x1
000009FA	0001	BYTE **- operand**	0x01

Figure 34: Expansion of papxinfkp[0].GrpPrl.GrpPrl[0]

sprm: The property that is being modified.

sprm.ispmd: If **ispmd** is equal to 0x0003 and **fSpec** is equal to 0x0000, this property is sprmPJc80.

sprm.sgc: A value of 0x1 specifies that this is a paragraph property.

sprm.spra: A value of 0x1 specifies that **operand** is 1 byte long.

operand: The property value, which is an unsigned integer specifying the paragraph justification. A value of 0x1 specifies that the paragraph will be center-justified.

The following table shows the expansion of **papxinfkp[0].GrpPrl.GrpPrl[1],** which is the second property that is applied to the first paragraph ("Test" followed by a newline character). This value specifies that the paragraph will be center-justified. Because this property occurs after the occurrence of sprmPJc80, the justification that it specifies takes precedence. In this case they both specify center justification, so the paragraph justification is unchanged.

Offset	Size	Structure	Value
000009FB	0003	Prl **- papxinfkp[0].GrpPrl.prl[1]**	
000009FB	0002	Sprm **- sprm**	
000009FB	9 bits	USHORT **- ispmd**	0x061
000009FB	1 bit	USHORT **- fSpec**	0x0
000009FB	3 bits	USHORT **- sgc**	0x1
000009FB	3 bits	USHORT **- spra**	0x1
000009FD	0001	BYTE **- operand**	0x01

Figure 35: Expansion of papxinfkp[0].GrpPrl.GrpPrl[1]

sprm: The property that is being modified.

sprm.ispmd: If **ispmd** is equal to 0x0061 and **fSpec** is equal to 0x0000, this property is sprmPJc.

sprm.sgc: A value of 0x1 specifies that this is a paragraph property, which is appropriate because **fcPlcfBtePapx** specifies paragraph properties.

sprm.spra: A value of 0x1 specifies that **operand** is 1 byte long.

operand: The property value, which is an unsigned integer that specifies the paragraph justification. A value of 0x01 specifies that the paragraph will be center-justified.

Copyright © 2025 Microsoft Corporation

The following table shows the expansion of the **papxinfkp[1]** element, which specifies the paragraph property information for the second paragraph of the text range.

Offset	Size	Structure	Value
000009EC	0008	PapxInFkp **- papxinfkp[1]**	
000009EC	0001	BYTE **- cb**	0x00
000009ED	0001	BYTE **- cb'**	0x03
000009EE	0006	GrpPrlAndIstd **- GrpPrl**	
000009EE	0002	USHORT **- istd**	0x0000
000009F0	0004	Prl **- GrpPrl[0]**	

Figure 36: Expansion of papxinfkp[1]

cb: A value of 0x00 specifies that the size of **GrpPrl** is determined by **cb'**.

cb': A value of 0x03 specifies that there are 2 * 3 = 6 bytes in **GrpPrl**.

GrpPrl.istd: A value of 0x0000 specifies that the Normal style will be applied to this paragraph. See section 2.4.6.5, Determining Properties of a Style.

GrpPrl.GrpPrl[0]: The first and only property that is being applied. See **papxinfkp[1].GrpPrl.GrpPrl[0]** in the following table.

papxinfkp[1] contains only some of the properties that apply to the second paragraph of the text range. The properties that are not contained in **papxinfkp[1]** take on their respective default values.

The following table shows the expansion of the **papxinfkp[1].GrpPrl.GrpPrl[0]** element, which is the first property that is applied to the second paragraph ("Test" followed by a newline character). It specifies that there are 0x0168 **twips** of vertical space before this paragraph.

Offset	Size	Structure	Value
000009F0	0004	Prl **- papxinfkp[1].GrpPrl.GrpPrl[0]**	
000009F0	0002	Sprm **- sprm**	
000009F0	9 bits	USHORT **- ispmd**	0x013
000009F0	1 bit	USHORT **- fSpec**	0x0
000009F0	3 bits	USHORT **- sgc**	0x1
000009F0	3 bits	USHORT **- spra**	0x5
000009F2	0002	USHORT **- operand**	0x0168

Figure 37: Expansion of papxinfkp[1].GrpPrl.GrpPrl[0]

sprm: The property that is being modified.

sprm.ispmd: If **ispmd** is equal to 0x0013 and **fSpec** is equal to 0x0000, this property is sprmPDyaBefore.

sprm.sgc: A value of 0x1 specifies that this is a paragraph property, which is appropriate because **fcPlcfBtePapx** specifies paragraph properties.

sprm.spra: A value of 0x5 specifies that **operand** is two bytes long.

Copyright © 2025 Microsoft Corporation

operand: The property value, which is an unsigned integer that specifies the number of twips of vertical space before this paragraph. A value of 0x0168 specifies there SHOULD be 0x0168 twips of vertical space before this paragraph.

The following table shows the expansion of **papxinfkp[2]**, which specifies the paragraph property information for the third paragraph of the text range.

Offset	Size	Structure	Value
000009E8	0004	PapxInFkp **- papxinfkp[2]**	
000009E8	0001	BYTE **- cb**	0x00
000009E9	0001	BYTE **- cb'**	0x01
000009E9	0002	GrpPrlAndIstd **- GrpPrl**	
000009EA	0002	USHORT **- istd**	0x0000

Figure 38: Expansion of papxinfkp[2]

cb: A value of 0x00 specifies that the size of **GrpPrl** is determined by **cb'**.

cb': A value of 0x01 specifies that there are 2 * 1 = 2 bytes in **GrpPrl**. The **GrpPrl.istd** element takes up two bytes; this means that **GrpPrl** has no **Prl** elements.

GrpPrl.istd: A value of 0x0000 specifies that the Normal style will be applied to this paragraph. See section 2.4.6.5, Determining Properties of a Style.

Because **papxinfkp[2]** contains no properties, all properties for the third paragraph of the text range take on their respective default values.

3.6 Example of Table Row Properties

This example assumes that the application has found a table terminating paragraph mark by following the algorithm in section 2.4.5, Determining Row Boundaries, or through some other means such as sequentially retrieving characters. The application has located the direct paragraph formatting for this paragraph mark by using the algorithm in section 2.4.6.1, Direct Paragraph Formatting. The following table shows the first **Prl** (section 2.2.5.2) of the direct formatting.

Offset	Size	Structure	Value
00000D05	0006	Prl **- prl**	
00000D05	0002	Sprm **- sprm**	
00000D05	9 bits	USHORT **- ispmd**	0x06B
00000D05	1 bit	USHORT **- fSpec**	0x0
00000D05	3 bits	USHORT **- sgc**	0x1
00000D05	3 bits	USHORT **- spra**	0x3
00000D07	0004	LONG **- operand**	0x00000000

Figure 39: The first Prl of the direct formatting

sprm.ispmd: If this value is 0x06B and **fSpec** is set to 0x0, this is sprmPTableProps.

sprm.sgc: A value of 0x1 specifies that **sprm** modifies a paragraph property.

Copyright © 2025 Microsoft Corporation

sprm.spra: A value of 0x3 specifies that **operand** is 4 bytes in size.

operand: A value of 0x00000000 specifies the byte offset in the Data Stream where a **PrcData** begins.

This example assumes that the application can process sprmPTableProps. It therefore ignores the rest of the array of **Prl** that contains the sprmPTableProps and instead processes the **PrcData** at offset zero of the Data Stream.

The following table shows the **PrcData** at offset zero of the Data Stream.

Offset	Size	Structure	Value
00000000	004C	PrcData **- PrcData**	
00000000	0002	SHORT **- cbGrpprl**	0x004A
00000002	004A	Array of Prl **- GrpPrl**	

Figure 40: A PrcData element that contains table row property modifiers

cbGrpprl: A value of 0x004A specifies the size, in bytes, of **GrpPrl**. Because **Prl** elements are variably sized, this does not give any information about the number of **Prl** elements that are contained in **GrpPrl** other than the fact that there is at least one **Prl** element.

GrpPrl: An array of **Prl**, expanded in the following figures.

The following table shows the first **Prl** element that is contained in **GrpPrl**.

Offset	Size	Structure	Value
00000002	0003	Prl **- GrpPrl[0]**	
00000002	0002	Sprm **- sprm**	
00000002	9 bits	USHORT **- ispmd**	0x016
00000002	1 bit	USHORT **- fSpec**	0x0
00000002	3 bits	USHORT **- sgc**	0x1
00000002	3 bits	USHORT **- spra**	0x1
00000004	0001	BYTE **- operand**	0x01

Figure 41: The first Prl in GrpPrl

sprm.ispmd: If this value is 0x016 and **fSpec** is set to 0x0, this is sprmPFInTable.

sprm.sgc: A value of 0x1 specifies that **sprm** modifies a paragraph property.

sprm.spra: A value of 0x1 specifies that **operand** is 1 byte in size.

operand: A value of 0x01 specifies that this paragraph is in a table.

The **GrpPrl[0]** element is 3 bytes in size, leaving 0x47 bytes for the rest of **GrpPrl**.

The following table shows the second **Prl** that is contained in **GrpPrl**.

Offset	Size	Structure	Value
00000005	0003	Prl **- GrpPrl[1]**	

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
00000005	0002	Sprm **- sprm**	
00000005	9 bits	USHORT **- ispmd**	0x017
00000005	1 bit	USHORT **- fSpec**	0x0
00000005	3 bits	USHORT **- sgc**	0x1
00000005	3 bits	USHORT **- spra**	0x1
00000007	0001	BYTE **- operand**	0x01

Figure 42: The second Prl in GrpPrl

sprm.ispmd: If this value is 0x017 and **fSpec** is equal to 0x0, this is sprmPFTtp.

sprm.sgc: A value of 0x1 specifies that this **Sprm** modifies a paragraph property.

sprm.spra: A value of 0x1 specifies that **operand** is one byte in size.

operand: A value of 0x01 specifies that the paragraph mark is a table terminating paragraph mark. **SprmPFTtp** is only valid at table a table depth of 1. Nested tables use sprmPFInnerTtp.

The **GrpPrl[1]** element is 3 bytes in size, leaving 0x44 bytes for the rest of the **GrpPrl** element.

The following table shows the third **Prl** element in **GrpPrl**.

Offset	Size	Structure	Value
00000008	0006	Prl **- GrpPrl[2]**	
00000008	0002	Sprm **- sprm**	
00000008	9 bits	USHORT **- ispmd**	0x049
00000008	1 bit	USHORT **- fSpec**	0x1
00000008	3 bits	USHORT **- sgc**	0x1
00000008	3 bits	USHORT **- spra**	0x3
0000000A	0004	LONG **- operand**	0x00000001

Figure 43: The third Prl in GrpPrl

sprm.ispmd: If this value is 0x049 and **fSpec** is set to 0x1, this is sprmPItap.

sprm.sgc: A value of 0x1 specifies that **sprm** modifies a paragraph property.

sprm.spra: A value of 0x3 specifies that **operand** is 4 bytes in size.

operand: A value of 0x00000001 specifies that the table depth of this table row is 1. This table is not nested in another table.

The **GrpPrl[2]** element is 6 bytes in size, leaving 0x3E bytes for the rest of **GrpPrl**.

The following table shows the fourth **Prl** in **GrpPrl**.

Offset	Size	Structure	Value
0000000E	0004	Prl **- GrpPrl[3]**	

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
0000000E	0002	Sprm **- sprm**	
0000000E	9 bits	USHORT **- ispmd**	0x001
0000000E	1 bit	USHORT **- fSpec**	0x1
0000000E	3 bits	USHORT **- sgc**	0x5
0000000E	3 bits	USHORT **- spra**	0x4
00000010	0002	SHORT **- operand**	0x0000

Figure 44: The fourth Prl in GrpPrl

sprm.ispmd: If this value is 0x001 and **fSpec** is set to 0x1, this is sprmTDxaLeft.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

sprm.spra: A value of 0x4 specifies that **operand** is two bytes in size.

operand: A value of 0x0000 specifies that the horizontal origin of the table is the **logical left** margin. This is further modified in **GrpPrl[10]**.

The **GrpPrl[3]** element is 4 bytes in size, leaving 0x3A bytes for the rest of the **GrpPrl** element.

The following table shows the fifth **Prl** in **GrpPrl**.

Offset	Size	Structure	Value
00000012	0006	Prl **- GrpPrl[4]**	
00000012	0002	Sprm **- sprm**	
00000012	9 bits	USHORT **- ispmd**	0x021
00000012	1 bit	USHORT **- fSpec**	0x1
00000012	3 bits	USHORT **- sgc**	0x5
00000012	3 bits	USHORT **- spra**	0x3
00000014	0004	TInsertOperand **- operand**	
00000014	0001	BYTE **- itcFirst**	0x00
00000015	0001	BYTE **- ctc**	0x02
00000016	0002	USHORT **- dxaCol**	0x0168

Figure 45: The fifth Prl in GrpPrl

sprm.ispmd: If this value is 0x021 and **fSpec** is set to 0x1, this is sprmTInsert.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

sprm.spra: A value of 0x3 specifies that **operand** is 4 bytes in size.

operand.itcFirst: A value of 0x00 specifies the zero-based index of the first cell definition to be inserted. Because no cells are defined, 0x00 is the only valid value for **itcFirst**.

operand.ctc: A value of 0x02 specifies the number of cell definitions to insert. This row has two cells.

Copyright © 2025 Microsoft Corporation

operand.dxaCol: A value of 0x0168 specifies that each of the newly inserted cells is 0x0168 **twips** wide.

The **GrpPrl[4]** element is 6 bytes in size, leaving 0x34 for the rest of the **GrpPrl** element.

The following table shows the sixth **Prl** in the **GrpPrl** element.

Offset	Size	Structure	Value
00000018	0008	Prl **- GrpPrl[5]**	
00000018	0002	Sprm **- sprm**	
00000018	9 bits	USHORT **- ispmd**	0x035
00000018	1 bit	USHORT **- fSpec**	0x1
00000018	3 bits	USHORT **- sgc**	0x5
00000018	3 bits	USHORT **- spra**	0x6
0000001A	0006	TableCellWidthOperand **- operand**	
0000001A	0001	BYTE **- cb**	0x05
0000001B	0002	ItcFirstLim **- itc**	
0000001B	0001	SHORT **- itcFirst**	0x00
0000001B	0001	SHORT **- itcLim**	0x01
0000001D	0003	FtsWWidth_TablePart **- FtsWWidth**	
0000001D	0001	Fts **- ftsWidth**	0x03
0000001E	0002	SHORT **- wWidth**	0x114C

Figure 46: The sixth Prl in GrpPrl

sprm.ispmd: If this value is 0x035 and **fSpec** is set to 0x1, this is sprmTCellWidth.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

sprm.spra: A value of 0x6 specifies that the first byte of **operand** specifies the size of the rest of **operand**.

operand.cb: A value of 0x05 specifies that **operand** is 5 bytes in size, not including **operand.cb**.

operand.itc.itcFirst: A value of 0x0000 specifies the first zero-based index of the first cell to which **FtsWWidth** applies.

operand.itc.itcLim: A value of 0x0001 specifies the zero-based index of the first cell outside the range of cells to which **FtsWWidth** applies. **FtsWWidth** thus only applies to the first cell in the row.

operand.FtsWWidth.ftsWidth: A value of 0x03 specifies that **wWidth** is a measurement in twips.

operand.FtsWWidth.wWidth: A value of 0x114C specifies the preferred width of the first cell of the row, in twips.

The **GrpPrl[5]** element is 8 bytes in size, leaving 0x2C bytes for the rest of GrpPrl.

The following table shows the seventh **Prl** element in **GrpPrl**.

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
00000020	0008	Prl **- GrpPrl[6]**	
00000020	0002	Sprm **- sprm**	
00000020	9 bits	USHORT **- ispmd**	0x035
00000020	1 bit	USHORT **- fSpec**	0x1
00000020	3 bits	USHORT **- sgc**	0x5
00000020	3 bits	USHORT **- spra**	0x6
00000022	0006	TableCellWidthOperand **- operand**	
00000022	0001	BYTE **- cb**	0x05
00000023	0002	ItcFirstLim **- itc**	
00000023	0001	SHORT **- itcFirst**	0x01
00000023	0001	SHORT **- itcLim**	0x02
00000025	0003	FtsWWidth_TablePart **- FtsWWidth**	
00000025	0001	Fts **- ftsWidth**	0x03
00000026	0002	SHORT **- wWidth**	0x114C

Figure 47: The seventh Prl in GrpPrl.

sprm.ispmd: If this value is 0x035 and **fSpec** is set to 0x0001, this is sprmTCellWidth.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

sprm.spra: A value of 0x6 specifies that the first byte of **operand** specifies the size of the rest of **operand**.

operand.cb: A value of 0x05 specifies that **operand** is 5 bytes in size, not including **operand.cb**.

operand.itc.itcFirst: A value of 0x0001 specifies the first zero-based index of the first cell to which **FtsWWidth** applies.

operand.itc.itcLim: A value of 0x0002 specifies the zero-based index of the first cell outside the range of cells to which **FtsWWidth** applies. This means that the **FtsWWidth** value applies only to the second cell in the row.

operand.FtsWWidth.ftsWidth: A value of 0x03 specifies that **wWidth** is a measurement in twips.

operand.FtsWWidth.wWidth: A value of 0x114C specifies the preferred width of the second cell of the row, in twips.

The **GrpPrl[6]** element is 8 bytes in size, leaving 0x24 for the rest of the **GrpPrl** element.

The following table shows the eighth **Prl** in the **GrpPrl** element.

Offset	Size	Structure	Value
00000028	0006	Prl **- GrpPrl[7]**	
00000028	0002	Sprm **- sprm**	

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
00000028	9 bits	USHORT **- ispmd**	0x023
00000028	1 bit	USHORT **- fSpec**	0x1
00000028	3 bits	USHORT **- sgc**	0x5
00000028	3 bits	USHORT **- spra**	0x3
0000002A	0004	TDxaColOperand **- operand**	
0000002A	0002	ItcFirstLim **- itc**	
0000002A	0001	SHORT **- itcFirst**	0x00
0000002A	0001	SHORT **- itcLim**	0x02
0000002C	0002	SHORT **- dxaCol**	0x114C

Figure 48: The eighth Prl in GrpPrl

sprm.ispmd: If this value is 0x023 and **fSpec** is set to 0x1, this is sprmTDxaCol.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

sprm.spra: A value of 0x3 specifies that **operand** is 4 bytes in size.

operand.itc.itcFirst: A value of 0x0000 specifies the first zero-based index of the first cell to which **dxaCol** applies.

operand.itc.itcLim: A value of 0x0002 specifies the zero-based index of the first cell outside the range of cells to which **dxaCol** applies. This means that the **DxaCol** value applies to both cells in the row.

operand.dxaCol: A value of 0x114C specifies the width of each cell, in twips. This value overrides the widths that are specified in the **GrpPrl[4]** element.

The **GrpPrl[7]** element is 6 bytes in size, leaving 0x1E bytes for the rest of **GrpPrl**.

The following table shows the ninth **Prl** in the **GrpPrl** element.

Offset	Size	Structure	Value
0000002E	0004	Prl **- GrpPrl[8]**	
0000002E	0002	Sprm **- sprm**	
0000002E	9 bits	USHORT **- ispmd**	0x03A
0000002E	1 bit	USHORT **- fSpec**	0x1
0000002E	3 bits	USHORT **- sgc**	0x5
0000002E	3 bits	USHORT **- spra**	0x2
00000030	0002	USHORT **- operand**	0x000F

Figure 49: The ninth Prl in GrpPrl

sprm.ispmd: If this value is 0x03A and **fSpec** is set to 0x1, this is sprmTIstd.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

Copyright © 2025 Microsoft Corporation

sprm.spra: A value of 0x2 specifies that **operand** is two bytes in size.

operand: A value of 0x000F specifies the **istd** of this table. To find the properties that are specified by this style, an application would implement the algorithm from section 2.4.6.5, Determining Properties of a Style. This is outside the scope of this example.

The **GrpPrl[8]** element is 4 bytes in size, leaving 0x1A bytes for the rest of **GrpPrl**.

The following table shows the tenth **Prl** in **GrpPrl**.

Offset	Size	Structure	Value
00000032	0004	Prl **- GrpPrl[9]**	
00000032	0002	Sprm **- sprm**	
00000032	9 bits	USHORT **- ispmd**	0x002
00000032	1 bit	USHORT **- fSpec**	0x1
00000032	3 bits	USHORT **- sgc**	0x5
00000032	3 bits	USHORT **- spra**	0x4
00000034	0002	SHORT **- operand**	0x006C

Figure 50: The tenth Prl in GrpPrl

sprm.ispmd: If this value is 0x002 and **fSpec** is set to 0x0001, this is sprmTDxaGapHalf.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

sprm.spra: A value of 0x4 specifies that operand is two bytes in size.

operand: A value of 0x006C specifies the distance, in twips, from the logical left margin to the logical left origin of this row.

The **GrpPrl[9]** element is 4 bytes in size, leaving 0x16 bytes for the rest of **GrpPrl**.

The following table shows the eleventh **Prl** in **GrpPrl**.

Offset	Size	Structure	Value
00000036	0006	Prl **- GrpPrl[10]**	
00000036	0002	Sprm **- sprm**	
00000036	9 bits	USHORT **- ispmd**	0x00A
00000036	1 bit	USHORT **- fSpec**	0x0
00000036	3 bits	USHORT **- sgc**	0x5
00000036	3 bits	USHORT **- spra**	0x3
00000038	0004	TLP **- operand**	
00000038	0002	SHORT **- itl**	0x0000
0000003A	0002	Fatl **- grfatl**	
0000003A	1 bit	USHORT **- fatlBorders**	0x0

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
0000003A	1 bit	USHORT **- fatlShading**	0x0
0000003A	1 bit	USHORT **- fatlFont**	0x0
0000003A	1 bit	USHORT **- fatlColor**	0x0
0000003A	1 bit	USHORT **- fatlBestFit**	0x0
0000003A	1 bit	USHORT **- fatlHdrRows**	0x1
0000003A	1 bit	USHORT **- fatlLastRow**	0x1
0000003A	1 bit	USHORT **- fatlHdrCols**	0x1
0000003A	1 bit	USHORT **- fatlLastCol**	0x1
0000003A	1 bit	USHORT **- fatlNoRowBands**	0x0
0000003A	1 bit	USHORT **- fatlNoColBands**	0x0
0000003A	5 bits	USHORT **- padding (ignored)**	0x00

Figure 51: The eleventh Prl in GrpPrl

sprm.ispmd: If this value is 0x0A and **fSpec** is set to 0x0, this is sprmTTlp.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

sprm.spra: A value of 0x3 specifies that **operand** is 4 bytes in size.

operand.itl: A value of 0x0000 specifies that either a table autoformat has not been applied to this table or that the last time that a table autoformat was applied to this table, all border, shading, font, and best fit formats were reset to the default values. The user could have applied other properties since the last table autoformat.

operand.grfatl.fatlBorders: A value of 0x0 specifies that either a table autoformat has not been applied to this table or that borders were not applied as part of the last table autoformat.

operand.grfatl.fatlShading: A value of 0x0 specifies that either a table autoformat has never been applied to this table or that shading was not applied as part of the last table autoformat.

operand.grfatl.fatlFont: A value of 0x0 specifies that either a table autoformat has not been applied to this table or that a font was not applied as part of the last table autoformat.

operand.grfatl.fatlColor: A value of 0x0 specifies that either a table autoformat has not been applied to this table, or that the monochrome variant of the last table autoformat was used, or that the last table autoformat did not have separate color and monochrome variant.

operand.grfatl.fatlBestFit: A value of 0x0 specifies that either a table autoformat has not been applied to this table or that the table columns were not resized to fit their contents as part of the last table autoformat.

operand.grfatl.fatlHdrRows: A value of 0x1 specifies that the first row of this table receives special formatting if the table style specifies special formatting for them. Special formatting is specified by any or all of sprmCCnf, sprmPCnf, and sprmTCnf in the style definition.

operand.grfatl.fatlLastRow: A value of 0x1 specifies that the last row of this table receives special formatting if the table style specifies special formatting for them. Special formatting is specified by any or all of sprmCCnf, sprmPCnf, and sprmTCnf in the style definition.

Copyright © 2025 Microsoft Corporation

operand.grfatl.fatlHdrCols: A value of 0x1 specifies that the logical left column of this table receives special formatting if the table style specifies special formatting for them. Special formatting is specified by any or all of sprmCCnf, sprmPCnf, and sprmTCnf in the style definition.

operand.grfatl.fatlLastCol: A value of 0x1 specifies that the **logical right** column of this table receives special formatting if the table style specifies special formatting for them. Special formatting is specified by any or all of sprmCCnf, sprmPCnf, and sprmTCnf in the style definition.

operand.grfatl.fatlNoRowBands: 0x0 specifies that the rows in odd-numbered row bands receive special formatting if the table style specifies special formatting for them. Special formatting is specified by any or all of sprmCCnf, sprmPCnf, and sprmTCnf in the style definition. The number of rows in a row band is specified by sprmTCHorzBands in the style definition.

operand.grfatl.fatlNoColBands: 0x0 specifies that the rows in odd-numbered column bands receive special formatting if the table style specifies special formatting for them. Special formatting is specified by any or all of sprmCCnf, sprmPCnf, and sprmTCnf in the style definition. The number of columns in a column band is specified by sprmTCVertBands in the style definition.

The **GrpPrl[10]** element is 6 bytes in size, leaving 0x10 bytes for the rest of **GrpPrl**.

The following table shows the twelfth **Prl** in **GrpPrl**.

Offset	Size	Structure	Value
0000003C	0005	Prl **- GrpPrl[11]**	
0000003C	0002	Sprm **- sprm**	
0000003C	9 bits	USHORT **- ispmd**	0x014
0000003C	1 bit	USHORT **- fSpec**	0x1
0000003C	3 bits	USHORT **- sgc**	0x5
0000003C	3 bits	USHORT **- spra**	0x7
0000003E	0003	FtsWWidth Table **- operand**	
0000003E	0001	Fts **- ftsWidth**	0x01
0000003F	0002	SHORT **- wWidth**	0x0000

Figure 52: The twelfth Prl in GrpPrl

sprm.ispmd: If this value is 0x014 and **fSpec** is set to 0x01, this is sprmTTableWidth.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

sprm.spra: A value of 0x7 specifies that operand is 3 bytes in size.

operand.ftsWidth: A value of 0x01 specifies that the table has no preferred width.

operand.wWidth: A value of 0x0000 is ignored.

The **GrpPrl[11]** element is 5 bytes in size, leaving 0x0B for the rest of **GrpPrl**.

The following table shows the thirteenth **Prl** in **GrpPrl**.

Offset	Size	Structure	Value
00000041	0003	Prl **- GrpPrl[12]**	

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
00000041	0002	Sprm **- sprm**	
00000041	9 bits	USHORT **- ispmd**	0x015
00000041	1 bit	USHORT **- fSpec**	0x1
00000041	3 bits	USHORT **- sgc**	0x5
00000041	3 bits	USHORT **- spra**	0x1
00000043	0001	BYTE **- operand**	0x01

Figure 53: The thirteenth Prl in GrpPrl

sprm.ispmd: A value of 0x015 and **fSpec** 0x1 specifies that this is sprmTFAutoFit.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

sprm.spra: A value of 0x1 specifies that **operand** is 1 byte in size.

operand: A value of 0x01 specifies that the columns are to be resized to fit the contents.

The **GrpPrl[12]** element is 3 bytes in size, leaving 0x08 for the rest of **GrpPrl**.

The following table shows the fourteenth **Prl** in **GrpPrl**.

Offset	Size	Structure	Value
00000044	0008	Prl **- GrpPrl[13]**	
00000044	0002	Sprm **- sprm**	
00000044	9 bits	USHORT **- ispmd**	0x035
00000044	1 bit	USHORT **- fSpec**	0x1
00000044	3 bits	USHORT **- sgc**	0x5
00000044	3 bits	USHORT **- spra**	0x6
00000046	0006	TableCellWidthOperand **- operand**	
00000046	0001	BYTE **- cb**	0x05
00000047	0002	ItcFirstLim **- itc**	
00000047	0001	SHORT **- itcFirst**	0x00
00000047	0001	SHORT **- itcLim**	0x02
00000049	0003	FtsWWidth_TablePart **- FtsWWidth**	
00000049	0001	Fts **- ftsWidth**	0x03
0000004A	0002	SHORT **- wWidth**	0x114C

Figure 54: The fourteenth Prl in GrpPrl

sprm.ispmd: If this value is 0x035 and **fSpec** is set to 0x1, this is sprmTCellWidth.

sprm.sgc: A value of 0x5 specifies that **sprm** modifies a table property.

Copyright © 2025 Microsoft Corporation

sprm.spra: A value of 0x6 specifies that the first byte of **operand** specifies the size of the rest of **operand**.

operand.cb: A value of 0x05 specifies the size of **operand**, not including **operand.cb**.

operand.itc.itcFirst: A value of 0x0000 specifies the first zero-based index of the first cell to which **FtsWWidth** applies.

operand.itc.itcLim: A value of 0x0002 specifies the zero-based index of the first cell outside the range of cells to which **FtsWWidth** applies. This means that the **FtsWWidth** value applies to both cells in the row.

operand.FtsWWidth.ftsWidth: A value of 0x03 specifies that **wWidth** is a measurement in twips.

operand.FtsWWidth.wWidth: A value of 0x114C specifies the preferred width of each cell, in twips. This value overrides the widths that are specified in **GrpPrl[5]** and **GrpPrl[6]**.

The **GrpPrl[13]** element is 8 bytes in size, consuming all remaining space in **GrpPrl**.

3.7 Example of a List

The following is an example of a list. It demonstrates how LFO structures, LSTF structures, and LVL structures define the list formatting of a paragraph. See Determining List Formatting for information about how a paragraph is related to these structures.

Offset	Size	Structure	Value
0000009A	02E8	FibRgFcLcb97 **- rgFcLcb97**	
0000009A	0108	... (omitted for brevity)	
000002E2	0004	ULONG **- fcPlfLst**	0x00000536
000002E6	0004	ULONG **- lcbPlfLst**	0x0000001E
000002EA	0004	ULONG **- fcPlfLfo**	0x000007E1
000002EE	0004	ULONG **- lcbPlfLfo**	0x00000018
000002F2	01D8	... (omitted for brevity)	

Figure 55: Portions of the FibRgFcLcb97 structure, highlighting the two fc/lcb pairs used for lists

As with all Word Binary files, this file has a Fib at offset zero in the WordDocument Stream. The preceding table shows a portion of the FibRgFcLcb97 that is contained in that **Fib**. The FibRgFcLcb97 is very large. Most fields have been omitted here for brevity.

fcPlfLst: A value of 0x00000536 specifies the offset, in bytes, of a location in the Table Stream. A **PlfLst** containing list formatting information begins at this offset. An array of **LVL** structures is directly appended to the **PlfLst**. The offset, in bytes, of the array of **LVL** structures in the Table Stream is equal to **fcPlfLst + lcbPlfLst**, which in this case is 0x00000554.

lcbPlfLst: A value of 0x0000001E specifies the size, in bytes, of the **PlfLst** at offset 0x00000536 in the Table Stream. This does not account for the size of the array of **LVL** structures that is appended to the **PlfLst**. The size of the array of **LVL**s cannot be determined without reading each **LVL**, as each **LVL** is of a variable size that can only be determined by reading that **LVL**. The number of **LVL** structures in the array, however, is equal to ((number of **LSTF**s in **PlfLst** such that **lstf.fSimpleList** is equal to 1) + (number of **LSTF**s in **PlfLst** such that **lstf.fSimpleList** is equal to zero) * 9), which in this case is 9.

Copyright © 2025 Microsoft Corporation

fcPlfLfo: A value of 0x000007E1 specifies the offset, in bytes, of a location in the Table Stream. A **PlfLfo** containing list format override information begins at this offset.

lcbPlfLfo: A value of 0x00000018 specifies the size, in bytes, of the PlfLfo at offset 0x000007E1 in the Table Stream.

The following table shows the expansion of the **PlfLst** at offset 0x00000536 in the Table Stream.

Offset	Size	Structure	Value
00000536	001E	PlfLst **- PlfLst**	
00000536	0002	SHORT **- cLst**	0x0001
00000538	001C	Array of LSTF **- rgLstf**	
00000538	001C	LSTF **- lstf[0]**	

Figure 56: Expansion of a PlfLst

In this particular example, there is only one list definition stored in the document, so **rgLstf** contains only one **LSTF**. It is common for **rgLstf** to contain multiple **LSTF** structures.

cLst: 0x0001 specifies that **rgLstf** contains one **LSTF**.

rgLstf: An array that contains the **LSTF** that is stored in the document.

rgLstf.lstf[0]: An **LSTF** that defines formatting of a list.

The following table shows the expansion of **rgLstf.lstf[0]** in the **PlfLst** at offset 0x00000536 in the Table Stream.

Offset	Size	Structure	Value
00000538	001C	LSTF **- lstf[0]**	
00000538	0004	LONG **- lsid**	0x44F53D09
0000053C	0004	LONG **- tplc**	0x31200A2C
00000540	0012	Array of SHORT **- rgistdPara**	
00000540	0002	SHORT **- istdPara[0]**	0x0FFF
00000542	000E	... (omitted for brevity)	
00000550	0002	SHORT **- istdPara[8]**	0x0FFF
00000552	1 bit	BYTE **- fSimpleList**	0x0
00000552	1 bit	BYTE **- unused1**	0x0
00000552	1 bit	BYTE **- fAutoNum**	0x0
00000552	1 bit	BYTE **- unused2**	0x0
00000552	1 bit	BYTE **- fHybrid**	0x0
00000552	3 bits	BYTE **- reserved1**	0x0
00000553	0001	grfhic **- grfhic**	

Figure 57: Expansion of an LSTF

Copyright © 2025 Microsoft Corporation

lsid: A value of 0x44F53D09 specifies a unique list identifier. **LFO** structures used these unique identifier to refer to specific **LSTF** structures. The **lfo[0].lsid** in the **PlfLfo** at the offset 0x000007E1 is equal to this value, which means that **lfo[0]** corresponds to this particular **LSTF**.

tplc: 0x31200A2C specifies a value that is used internally by the list gallery user interface. For purposes of this example, ignore this value.

rgistdPara: Each element is the **istd** of the style which is linked to the level that corresponds to the index of the element. In this example, there are no styles linked to any level in the list, so the value of each element is 0x0FFF, which is common. This contains 9 elements, all but the first and last of which have been omitted for brevity.

rgistdPara.istdPara[0]: A value of 0x0FFF specifies that the first level of this list has no style linked to it.

rgistdPara.istdPara[8]: A value of 0x0FFF specifies that the ninth level of this list has no style linked to it.

fSimpleList: A value of 0x0 specifies that this list contains 9 levels, and that therefore there are 9 elements in the array of **LVL** structures at offset 0x00000554 that correspond to this **LSTF**.

unused1: A value of 0x0 is ignored.

fAutoNum: A value of 0x0 specifies that this list is not used by any **field**.

unused2: A value of 0x0 is ignored.

fHybrid: A value of 0x0 specifies that this list is not a **hybrid list**.

reserved1: A value of 0x0 is ignored.

grfhic: This structure contains information that is only useful for HTML compatibility. This example does not cover list HTML compatibility.

The following table shows the expansion of the array of **LVL** structures at offset 0x00000554 in the Table Stream.

Offset	Size	Structure	Value
0000009A	028D	Array of LVL **- rgLvl**	
00000554	0047	LVL **- lvl[0]**	
0000059B	004B	LVL **- lvl[1]**	
000005E6	004B	LVL **- lvl[2]**	
00000631	01B0	... (omitted for brevity)	

Figure 58: Expansion of an array of LVLs

As specified by **lstf[0].fSimpleList**, this contains 9 **LVL** structures that correspond to **lstf[0]**. If **PlfLst** had more than just one **LSTF** (as specified by **PlfLst**.**cLst**), this array would contain the additional **LVL** structures that would correspond to the extra **LSTF** structures (the number of which would be specified by the **fSimple** field of those **LSTF**s). The **LVL**s stored in this array are stored in same order as the **LSTF**s in **PlfLst**. The LVLs corresponding to each **LSTF** are stored in level order, starting with the most significant level. For brevity, only the first three **LVL** structures are included and will be expanded.

lvl[0]: This **LVL** specifies the level formatting of the first level in the list.

lvl[1]: This **LVL** specifies the level formatting of the second level in the list.

Copyright © 2025 Microsoft Corporation

lvl[2]: This **LVL** specifies the level formatting of the third level in the list.

The following table shows the expansion of **lvl[0]** in the array of **LVL** structures at offset 0x00000554 in the Table Stream. This specifies the level formatting of the first level in the list corresponding to **lstf[0]**.

Offset	Size	Structure	Value
00000554	0047	LVL **- lvl[0]**	
00000554	001C	LVLF **- lvlf**	
00000554	0004	LONG **- iStartAt**	0x00000001
00000558	0001	MSONFC **- nfc**	0x00
00000559	2 bits	BYTE **- jc**	0x0
00000559	1 bit	BYTE **- fLegal**	0x0
00000559	1 bit	BYTE **- fNoRestart**	0x0
00000559	1 bit	BYTE **- fIndentSav**	0x0
00000559	1 bit	BYTE **- fConverted**	0x0
00000559	1 bit	BYTE **- unused1**	0x0
00000559	1 bit	BYTE **- fTentative**	0x0
0000055A	0009	Array of BYTE **- rgbxchNums**	
0000055A	0001	BYTE **- xchNums[0]**	0x01
0000055B	0001	BYTE **- xchNums[1]**	0x00
0000055C	0007	... (omitted for brevity)	
00000563	0001	BYTE **- ixchFollow**	0x00
00000564	0004	LONG **- dxaIndentSav**	0x00000000
00000568	0004	ULONG **- unused2**	0x00000000
0000056C	0001	BYTE **- cbGrpprlChpx**	0x0D
0000056D	0001	BYTE **- cbGrpprlPapx**	0x18
0000056E	0001	BYTE **- ilvlRestartLim**	0x00
0000056F	0001	grfhic **- grfhic**	
00000570	0018	Array of Prl **- grpprlPapx**	
00000588	000D	Array of Prl **- grpprlChpx**	
00000595	0006	Xst **- xst**	\0x0000.

Figure 59: Expansion of lvl[0]

lvlf.iStartAt: A value of 0x00000001 specifies that the number sequence of this level starts at 1.

lvlf.nfc: A value of 0x00 specifies that any level number inherited from this level that replaces a placeholder in the number text of any level (see the **xst** field of **LVL** for information about

Copyright © 2025 Microsoft Corporation

placeholders) has Arabic formatting (for example, 1, 2, 3, 4…), unless otherwise specified by the **lvlf.fLegal** field of the **LVL** of that level.

lvlf.jc: A value of 0x0 specifies that the number text that is specified by **xst** is left-justified.

lvlf.fLegal: A value of 0x0 specifies that this level does not override the formatting of inherited level numbers.

lvlf.fNoRestart: A value of 0x0 specifies that number sequence of this level restarts after any more significant level. Because this **LVL** specifies the most significant level, this is ignored.

lvlf.fIndentSav: A value of 0x0 specifies that this level does not need to replace an indent when a paragraph is taken out of the level.

lvlf.fConverted: A value of 0x0 specifies that **lvlf.nfc** was not converted from an old value used for compatibility purposes.

lvlf.unused1: A value of 0x0 is ignored.

lvlf.fTentative: A value of 0x0 is ignored because this level is not in a hybrid list, as specified by **lstf[0].fHybrid**.

lvlf.rgbxchNums: An array that specifies the 1-based indexes of the placeholders in **xst** (see the **xst** field of **LVL** for information about placeholders). This array has 9 elements, but it is zero-terminated. The elements that follow the first terminating zero are omitted for brevity.

lvlf.rgbxchNums.xchNums[0]: A value of 0x01 specifies that the first character in the string which is specified by **xst** is a placeholder for a level number.

lvlf.rgbxchNums.xchNums[1]: A value of 0x00 specifies that this element and those that follow are ignored.

lvlf.ixchFollow: A value of 0x00 specifies that a tab immediately follows the number text which is specified by **xst**.

lvlf.dxaIndentSav: A value of 0x00000000 is ignored because **lvlf.fIndentSav** is zero.

lvlf.unused2: A value of 0x00000000 is ignored.

lvlf.cbGrpprlChpx: A value of 0x0D specifies that the size of **grpprlChpx** is 13 bytes.

lvlf.cbGrpprlPapx: A value of 0x18 specifies that the size of **grpprlPapx** is 24 bytes.

lvlf.ilvlRestartLim: A value of 0x00 is ignored because **lvlf.fNoRestart** is zero.

lvlf.grfhic: This structure contains information that is only useful for HTML compatibility. This example does not cover list HTML compatibility.

grpprlPapx: Contains paragraph properties that are applied to the paragraph after number text is applied to the paragraph. See Determining List Formatting.

grpprlChpx: Contains character properties that are applied to the number text. See Determining List Formatting.

xst: "\0x0000." specifies the number text of the level. '\0x0000' is a non-printable character, which is actually the integer 0x0000. This character is a placeholder for the first level in the list. It is the first character in the string, as specified by **lvlf.rgbxchNums.xchNums[0]**. This placeholder will be replaced by the current level number of the first level in the list for each paragraph in this level. The number text for the first paragraph in this level will be "1.".

Copyright © 2025 Microsoft Corporation

The following table shows the expansion of **lvl[1]** in the array of **LVL** structures at offset 0x00000554 in the Table Stream. This specifies the level formatting of the second level in the list corresponding to **lstf[0]**.

Offset	Size	Structure	Value
0000059B	004B	LVL **- lvl[1]**	
0000059B	001C	LVLF **- lvlf**	
0000059B	0004	LONG **- iStartAt**	0x00000003
0000059F	0001	MSONFC **- nfc**	0x04
000005A0	2 bits	BYTE **- jc**	0x0
000005A0	1 bit	BYTE **- fLegal**	0x0
000005A0	1 bit	BYTE **- fNoRestart**	0x0
000005A0	1 bit	BYTE **- fIndentSav**	0x0
000005A0	1 bit	BYTE **- fConverted**	0x0
000005A0	1 bit	BYTE **- unused1**	0x0
000005A0	1 bit	BYTE **- fTentative**	0x0
000005A1	0009	Array of BYTE **- rgbxchNums**	
000005A1	0001	BYTE **- xchNums[0]**	0x01
000005A2	0001	BYTE **- xchNums[1]**	0x03
000005A3	0001	BYTE **- xchNums[2]**	0x00
000005A4	0006	... (omitted for brevity)	
000005AA	0001	BYTE **- ixchFollow**	0x00
000005AB	0004	LONG **- dxaIndentSav**	0x00000000
000005AF	0004	ULONG **- unused2**	0x00000000
000005B3	0001	BYTE **- cbGrpprlChpx**	0x0D
000005B4	0001	BYTE **- cbGrpprlPapx**	0x18
000005B5	0001	BYTE **- ilvlRestartLim**	0x01
000005B6	0001	grfhic **- grfhic**	
000005B7	0018	Array of Prl **- grpprlPapx**	
000005CF	000D	Array of Prl **- grpprlChpx**	
000005DC	000A	Xst **- xst**	\0x0000-\0x0001)

Figure 60: Expansion of lvl[1]

lvlf.iStartAt: A value of 0x00000003 specifies that the number sequence of this level starts at 3.

lvlf.nfc: A value of 0x04 specifies that any level number inherited from this level that replaces a placeholder in the number text of any level (see the **xst** field of **LVL** for information about

Copyright © 2025 Microsoft Corporation

placeholders) has lowercase letter formatting (for example, a, b, c, d…), unless otherwise specified by the **lvlf.fLegal** field of the LVL belonging to that level.

lvlf.jc: A value of 0x0 specifies that the number text specified by **xst** is left-justified.

lvlf.fLegal: A value of 0x0 specifies that this level does not override the formatting of inherited level numbers.

lvlf.fNoRestart: A value of 0x0 specifies that the number sequence of this level restarts after any more significant level. As this **LVL** represents the second level, this means that the number sequence of this level restarts after any paragraph that is in the first level of this same list is encountered.

lvlf.fIndentSav: A value of 0x0 specifies that this level does not need to replace an indent when a paragraph is taken out of the level.

lvlf.fConverted: A value of 0x0 specifies that **lvlf.nfc** was not converted from an old value used for compatibility purposes.

lvlf.unused1: A value of 0x0 is ignored.

lvlf.fTentative: A value of 0x0 is ignored because this level is not in a hybrid list, as specified by **lstf[0].fHybrid.**

lvlf.rgbxchNums: An array that specifies the 1-based indexes of the placeholders in **xst** (see the **xst** field of LVL). This array has 9 elements, but it is zero-terminated. The elements that follow the first terminating zero are omitted for brevity.

lvlf.rgbxchNums.xchNums[0]: A value of 0x01 specifies that the first character in the string specified by **xst** is a placeholder for a level number.

lvlf.rgbxchNums.xchNums[1]: A value of 0x03 specifies that the third character in the string specified by **xst** is a placeholder for a level number.

lvlf.rgbxchNums.xchNums[2]: A value of 0x00 specifies that this element and those that follow are ignored.

lvlf.ixchFollow: A value of 0x00 specifies that a tab immediately follows the number text that is specified by **xst**.

lvlf.dxaIndentSav: A value of 0x00000000 is ignored because **lvlf.fIndentSav** is zero.

lvlf.unused2: A value of 0x00000000 is ignored.

lvlf.cbGrpprlChpx: A value of 0x0D specifies that the size of **grpprlPapx** is 13 bytes.

lvlf.cbGrpprlPapx: A value of 0x18 specifies that the size of **grpprlPapx** is 24 bytes.

lvlf.ilvlRestartLim: A value of 0x01 is ignored because **lvlf.fNoRestart** is zero.

lvlf.grfhic: This structure contains information that is only useful for HTML compatibility. This example does not cover list HTML compatibility.

grpprlPapx: Contains paragraph properties that are applied to the paragraph after the paragraph receives number text. See Determining List Formatting.

grpprlChpx: Contains character properties that are applied to the number text. See Determining List Formatting.

xst: A value of "\0x0000-\0x0001)" specifies the number text of the level. '\0x0000' and '\0x0001' are non-printable characters, which are actually the integers 0x0000 and 0x0001, respectively.

Copyright © 2025 Microsoft Corporation

These characters are the placeholders for the first and second levels in the list. These are placeholders because their indexes are specified in the elements of **lvlf.rgbxchNums**. These placeholders will be replaced by the current level numbers of the first and second levels in the list for each paragraph in this level. The number text for the first paragraph in this level that is the child of the first paragraph in the first level will be "1-a)".

The following table shows the expansion of **lvl[2]** in the array of **LVL** structures at offset 0x00000554 in the Table Stream. This specifies the level formatting of the first level in the list corresponding to **lstf[0]**.

Offset	Size	Structure	Value
000005E6	004B	LVL **- lvl[2]**	
000005E6	001C	LVLF **- lvlf**	
000005E6	0004	LONG **- iStartAt**	0x00000001
000005EA	0001	MSONFC **- nfc**	0xFF
000005EB	2 bits	BYTE **- jc**	0x1
000005EB	1 bit	BYTE **- fLegal**	0x0
000005EB	1 bit	BYTE **- fNoRestart**	0x1
000005EB	1 bit	BYTE **- fIndentSav**	0x0
000005EB	1 bit	BYTE **- fConverted**	0x0
000005EB	1 bit	BYTE **- unused1**	0x0
000005EB	1 bit	BYTE **- fTentative**	0x0
000005EC	0009	Array of BYTE **- rgbxchNums**	
000005EC	0001	BYTE **- xchNums[0]**	0x00
000005ED	0008	... (omitted for brevity)	
000005F5	0001	BYTE **- ixchFollow**	0x01
000005F6	0004	LONG **- dxaIndentSav**	0x00000000
000005FA	0004	ULONG **- unused2**	0x00000000
000005FE	0001	BYTE **- cbGrpprlChpx**	0x0D
000005FF	0001	BYTE **- cbGrpprlPapx**	0x10
00000600	0001	BYTE **- ilvlRestartLim**	0x00
00000601	0001	grfhic **- grfhic**	
00000602	0010	Array of Prl **- grpprlPapx**	
00000612	000D	Array of Prl **- grpprlChpx**	
0000061F	0012	Xst **- xst**	Example:

Figure 61: Expansion of lvl[2]

This level does not have a number sequence because the number text for this level does not have a placeholder for this level.

Copyright © 2025 Microsoft Corporation

lvlf.iStartAt: A value of 0x00000001 is ignored, because this level does not have a number sequence.

lvlf.nfc: A value of 0xFF specifies that this level does not have a number style.

lvlf.jc: A value of 0x1 specifies that the number text specified by **xst** is center-justified.

lvlf.fLegal: A value of 0x0 specifies that this level does not override the formatting of inherited level numbers.

lvlf.fNoRestart: A value of 0x1 is ignored, because this level does not have a number sequence.

lvlf.fIndentSav: A value of 0x0 specifies that this level does not need to replace an indent when a paragraph is taken out of the level.

lvlf.fConverted: A value of 0x0 specifies that **lvlf.nfc** was not converted from an old value used for compatibility purposes.

lvlf.unused1: A value of 0x0 is ignored.

lvlf.fTentative: A value of 0x0 is ignored because this level is not in a hybrid list, as specified by **lstf[0].fHybrid**.

lvlf.rgbxchNums: An array that specifies the 1-based indexes of the placeholders in **xst** (see the **xst** field of **LVL**). This array has 9 elements, but is zero-terminated. The elements that follow the first terminating zero are omitted for brevity.

lvlf.rgbxchNums.xchNums[0]: A value of 0x00 specifies that this element and those that follow are ignored. Because this is the first element in the array, this means that there are no placeholders in **xst**, and therefore it is a static string.

lvlf.ixchFollow: A value of 0x01 specifies that a space immediately follows the number text that is specified by **xst**.

lvlf.dxaIndentSav: A value of 0x00000000 is ignored because **lvlf.fIndentSav** is zero.

lvlf.unused2: A value of 0x00000000 is ignored.

lvlf.cbGrpprlChpx: A value of 0x0D specifies that the size of **grpprlPapx** is 13 bytes.

lvlf.cbGrpprlPapx: 0x10 specifies that the size of **grpprlChpx** is 16 bytes.

lvlf.ilvlRestartLim: A value of 0x00 is ignored because this level does not have a number sequence.

lvlf.grfhic: This structure contains information that is only useful for HTML compatibility. This example does not cover list HTML compatibility.

grpprlPapx: Contains paragraph properties that are applied to the paragraph after it receives number text. See Determining List Formatting.

grpprlChpx: Contains character properties that are applied to the number text. See Determining List Formatting.

xst: "Example:" specifies the number text of the level. As specified by **lvlf.rgbxchNums**, this does not have any placeholders in it. Therefore, this text is static and every paragraph in this level starts with "Example: ".

The following table shows the expansion of the **PlfLfo** at offset 0x000007E1 in the Table Stream.

Offset	Size	Structure	Value
000007E1	0018	PlfLfo **- PlfLfo**	

Copyright © 2025 Microsoft Corporation

Offset	Size	Structure	Value
000007E1	0004	ULONG **- lfoMac**	0x00000001
000007E5	0010	Array of LFO **- rgLfo**	
000007E5	0010	LFO **- lfo[0]**	
000007F5	0004	Array of LFOData **- rgLfoData**	
000007F5	0004	LFOData **- lfoData[0]**	
000007F5	0004	LONG **- cp**	0xFFFFFFFF
000007F9	0000	Array of LFOLVL **- rgLfoLvl**	

Figure 62: Expansion of PlfLfo

This contains the list format override information in the document.

lfoMac: A value of 0x00000001 specifies that **rgLfo** and **rgLfoData** each have one element.

rgLfo: An array of LFO structures.

rgLfo.lfo[0]: An LFO structure that specifies a list format override.

rgLfoData: An array of additional list format override data.

rgLfoData.lfoData[0]: An LFOData structure that specifies addition list format override.

rgLfoData.lfoData[0].cp: A value of 0xFFFFFFFF is ignored.

rgLfoData.lfoData[0].rgLfoLvl: An empty array, because **rgLfo.lfo[0].clfolvl** is zero.

The following table shows the expansion of **rgLfo.lfo[0]** in the **PlfLfo** at offset 0x000007E1 in the Table Stream.

Offset	Size	Structure	Value
000007E5	0010	LFO **- lfo[0]**	
000007E5	0004	LONG **- lsid**	0x44F53D09
000007E9	0004	LONG **- unused1**	0x00000000
000007ED	0004	LONG **- unused2**	0x00000000
000007F1	0001	BYTE **- clfolvl**	0x00
000007F2	0001	BYTE **- ibstFltAutoNum**	0x00
000007F3	0001	grfhic **- grfhic**	
000007F4	0001	BYTE **- unused3**	0x00

Figure 63: Expansion of lfo[0]

This **LFO** is used as a level of indirection between the paragraphs in a list and the **LSTF** that defines the list that they are in. An **LFO**, along with its corresponding **LFOData**, can specify information that overrides the formatting information specified by an **LSTF** and its corresponding **LVL** structures. In this example, as in most cases, there is no such overriding information specified.

Copyright © 2025 Microsoft Corporation

lsid: A value of 0x44F53D09 specifies the value of the **lsid** field of the **LSTF** that this **LFO** corresponds to. In this example, this value is equal to **lstf[0].lsid** in the **PlfLst** at offset 0x00000536 in the Table Stream. Therefore, the list formatting of any paragraph that uses this **LFO** is specified by **lstf[0]** in the **PlfLst** at offset 0x00000536 in the Table Stream.

unused1: A value of 0x00000000 is ignored.

unused2: A value of 0x00000000 is ignored.

clfolvl: A value of 0x00 specifies that there are no **LFOLVL** structures in **rgLfoData.lfoData[0].rgLfoLvl** in the **PlfLfo** at offset 0x000007E1 in the Table Stream.

ibstFltAutoNum: A value of 0x00 specifies that this **LFO** is not used by any field.

grfhic: This structure contains information that is only useful for HTML compatibility. This example does not cover list HTML compatibility.

unused3: 0x00 is ignored.

Copyright © 2025 Microsoft Corporation

4 Security Considerations

4.1 Encryption and Obfuscation (Password to Open)

When XOR obfuscation (section 2.2.6.1) is used, data can be easily extracted and the document password might be retrievable.

When obfuscation or encryption is used, the ObjectPool storage, Macros storage, Custom XML Data storage, XML Signatures storage, and Signatures stream are not obfuscated or encrypted.

When XOR obfuscation (section 2.2.6.1) or Office binary document RC4 encryption (section 2.2.6.2) is used or when Office binary document RC4 CryptoAPI encryption (section 2.2.6.3) is used with **fDocProps** set to **false** in **EncryptionHeader**.**Flags**, the Document Summary Information stream and the Summary Information stream are not obfuscated or encrypted.

When Office binary document RC4 encryption (section 2.2.6.2) or Office binary document RC4 CryptoAPI encryption (section 2.2.6.3) is used, the same block numbers are reused in the WordDocument stream, the Table stream, and the entire Data stream. This reuse can occur potentially with known cleartext, implying that certain portions of encrypted data can be directly extracted or easily retrieved.

See [MS-OFFCRYPTO] section 4.1.3 for additional security considerations with encryption and obfuscation in Word binary files.

4.2 Write Reservation Password

The **write-reservation password** is embedded in cleartext in the file. Be aware that protection with a write reservation password is not considered a security mechanism. The protection can be easily removed by using a binary editor. Protection with a write-reservation password is meant to protect against accidental modification only.

Copyright © 2025 Microsoft Corporation

5 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Word 97
- Microsoft Word 2000
- Microsoft Word 2002
- Microsoft Office Word 2003
- Microsoft Office Word 2007
- Microsoft Word 2010
- Microsoft Word 2013
- Microsoft Word 2016
- Microsoft Word 2019
- Microsoft Word 2021
- Microsoft Word LTSC 2024

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 2.1.4.3: Word 97 and Word 2000 do not generate this stream when saving files and ignore it when loading files. Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later read this stream when loading files and generate it when saving files if the object supports a separate print presentation and provides that presentation in Enhanced Metafile format.

<2> Section 2.1.10: Office Word 2007, Word 2010, and later read this storage. Word 97, Word 2000, Word 2002, and Office Word 2003 ignore it.

<3> Section 2.1.11: Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later read this stream. Word 97 and Word 2000 ignore it.

<4> Section 2.1.12: Office Word 2003, Office Word 2007, Word 2010, and later read streams and storages from inside the Protected Content Stream. Word 97, Word 2000, and Word 2002 ignore the Information Rights Management Data Space Storage and the Protected Content Stream.

<5> Section 2.1.12: Office Word 2003, Office Word 2007, Word 2010, and later ignore streams and storages which instead are read from inside the Protected Content Stream. Word 97, Word 2000, and Word 2002 ignore the Protected Content Stream and read storages and streams located outside the Protected Content Stream.

<6> Section 2.2.5: The following table lists the ranges of **Sprm.ispmd** that each version of Microsoft Word processes. All versions of Microsoft Word skip Prls that they cannot process.

Copyright © 2025 Microsoft Corporation

Version	Sprm.sgc	Range of Sprm.ispmd processed
Word 97	1 (paragraph)	0x00 – 0x48
	2 (character)	0x00 – 0x10, 0x30 – 0x6F
	3 (picture)	0x00 – 0x07
	4 (section)	0x00 – 0x33
	5 (table)	0x00 – 0x0C, 0x20, 0x2C
Word 2000	1 (paragraph)	0x00 – 0x63
	2 (character)	0x00 – 0x13, 0x30 – 0x81
	3 (picture)	0x00 – 0x0B
	4 (section)	0x00 – 0x38
	5 (table)	0x00 – 0x39, 0x60 – 0x65
Word 2002	1 (paragraph)	0x00 – 0x6E
	2 (character)	0x00 – 0x18, 0x30 – 0x88
	3 (picture)	0x00 – 0x0B
	4 (section)	0x00 – 0x42
	5 (table)	0x00 – 0x3D, 0x60 – 0x8A
Office Word 2003	1 (paragraph)	0x00 – 0x6F
	2 (character)	0x00 – 0x18, 0x30 – 0x89, 0x90
	3 (picture)	0x00 – 0x0B
	4 (section)	0x00 – 0x43

Copyright © 2025 Microsoft Corporation

Version	Sprm.sgc	Range of Sprm.ispmd processed
	5 (table)	0x00 – 0x3E, 0x60 – 0x90
Office Word 2007, Word 2010, and later	1 (paragraph)	0x00 – 0x73
	2 (character)	0x00 – 0x1D, 0x30 – 0x89, 0x90 – 0x95
	3 (picture)	0x00 – 0x0B
	4 (section)	0x00 – 0x44
	5 (table)	0x00 – 0x42, 0x60 – 0x90

<7> Section 2.2.6: Word 97 and Word 2000 cannot open files which are password protected with Office binary document RC4 CryptoAPI encryption.

<8> Section 2.2.6.3: Neither Word 97 nor Word 2000 support this encryption method.

<9> Section 2.4.3: Word 97 and Word 2000 require that each row have sprmTDefTable applied. These versions do not process sprmPTableProps. Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later require sprmTDefTable or sprmTInsert. These versions do process sprmPTableProps.

A sprmTDefTable applied to a TTP mark overrides any formatting inherited from the table style. Word 97 and Word 2000 do not have a table style feature. For this reason, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later only emit sprmTDefTable for versions that do not process sprmPTableProps.

If an application does not emit sprmTDefTable for the benefit of readers that do not process sprmPTableProps, the documents that are generated by that application are not compatible with Word 97 or Word 2000.

<10> Section 2.4.6: Word 97 and Word 2000 do not support table styles, and thus ignore sprmTIstd, among others. sprmPTableProps can be used to separate Prls intended for Word 97 and Word 2000 from those intended for all other versions, as specified in section 2.4.3, Overview of Tables.

<11> Section 2.5.2: A special empty document is installed with Word 97, Word 2000, Word 2002, and Office Word 2003 to allow "Create New Word Document" from the operating system. This document has an nFib of 0x00C0. In addition the BiDi build of Word 97 differentiates its documents by saving 0x00C2 as the nFib. In both cases treat them as if they were 0x00C1.

<12> Section 2.5.2: Picture watermarks could be present in the document even if **fHasPic** is 0.

<13> Section 2.5.2: The **nFibBack** field is treated as if it is set to 0x00BF when a locale-specific version of Word 97 sets it to 0x00C1.

<14> Section 2.5.2: Word 97, Word 2000, Word 2002, and Office Word 2003 install a minimal .doc file for use with the New- Microsoft Word Document of the shell. This minimal .doc file has **fEmptySpecial** set to 1.

Copyright © 2025 Microsoft Corporation

<15> Section 2.5.2: Word uses this flag to identify a document that was created by using the New – Microsoft Word Document of the operating system shell.

<16> Section 2.5.3: Word 97 and Word 2000 can put a value here when performing an **incremental save** (FibBase.**fComplex**).

<17> Section 2.5.3: Word 97 and Word 2000 can put a value here when performing an incremental save (FibBase.**fComplex**).

<18> Section 2.5.3: Word 97 and Word 2000 can put a value here when performing an incremental save (FibBase.**fComplex**).

<19> Section 2.5.3: Word 97 and Word 2000 can put a value here when performing an incremental save (FibBase.**fComplex**).

<20> Section 2.5.3: Word 97 and Word 2000 can put a value here when performing an incremental save (FibBase.**fComplex**).

<21> Section 2.5.3: Word 97 and Word 2000 can put a value here when performing an incremental save (FibBase.**fComplex**).

<22> Section 2.5.3: Word 97 and Word 2000 can put a value here when performing an incremental save (FibBase.**fComplex**).

<23> Section 2.5.3: Word 97 and Word 2000 can put a value here when performing an incremental save (FibBase.**fComplex**).

<24> Section 2.5.3: Word 97 and Word 2000 can put a value here when performing an incremental save (FibBase.**fComplex**).

<25> Section 2.5.4: Word 97, Word 2000, Word 2002, and Office Word 2003 write a nonzero value here when saving a document template with changes that require the saving of an AutoText document.

<26> Section 2.5.6: Word 97, Word 2000, and Word 2002 emit this information when performing an incremental save. Office Word 2003, Office Word 2007, Word 2010, and later do not emit this information.

<27> Section 2.5.6: Word 97 reads this information if **FibBase.nFib** is 193. Word 2000 reads this information if **FibRgCswNew.nFibNew** is 217. Word 2002 reads this information if **FibRgCswNew.nFibNew** is 257. Office Word 2003, Office Word 2007, Word 2010, and later do not read this information.

<28> Section 2.5.6: Office Word 2007, Word 2010, and later ignore this data.

<29> Section 2.5.6: Word 97 emits information at offset **fcPgdMotherOldOld**. Neither Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<30> Section 2.5.6: Word 97 reads this information. Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

<31> Section 2.5.6: Word 97 emits information at offset **fcBkdMotherOldOld**. Neither Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<32> Section 2.5.6: Word 97 reads this information. Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

Copyright © 2025 Microsoft Corporation

<33> Section 2.5.6: Word 97 emits information at offset **fcPgdFtnOldOld**. Neither Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<34> Section 2.5.6: Word 97 reads this information. Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

<35> Section 2.5.6: Word 97 emits information at offset **fcBkdFtnOldOld**. Neither Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<36> Section 2.5.6: Word 97 reads this information. Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

<37> Section 2.5.6: Word 97 emits information at offset **fcPgdEdnOldOld**. Neither Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<38> Section 2.5.6: Word 97 reads this information. Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

<39> Section 2.5.6: Word 97 emits information at offset **fcBkdEdnOldOld**. Neither Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<40> Section 2.5.6: Only Word 97 reads this information.

<41> Section 2.5.6: fcRouteSlip is only saved and read by Word 97, Word 2000, Word 2002, and Office Word 2003.

<42> Section 2.5.6: SttbSavedBy is only saved and read by Word 97 and Word 2000.

<43> Section 2.5.6: SttbSavedBy is only saved and read by Word 97 and Word 2000.

<44> Section 2.5.6: Word 97 and Word 2000 write this information when the user chooses to save versions in the document. Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later do not write this information.

<45> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 read this information. Office Word 2007, Word 2010, and later ignore it.

<46> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 write this information when the user chooses to save versions in the document. Neither Office Word 2007, Word 2010, nor Word 2013 and later write this information.

<47> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 read this information. Office Word 2007, Word 2010, and later ignore it.

<48> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 write this information when the user chooses to save versions in the document. Neither Office Word 2007, Word 2010, nor Word 2013 and later write this information.

<49> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 read this information. Office Word 2007, Word 2010, and later ignore it.

<50> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 write this information when the user chooses to save versions in the document. Neither Office Word 2007, Word 2010, nor Word 2013 and later write this information.

<51> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 read this information. Office Word 2007, Word 2010, and later ignore it.

Copyright © 2025 Microsoft Corporation

<52> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 write this information when the user chooses to save versions in the document. Neither Office Word 2007, Word 2010, nor Word 2013 and later write this information.

<53> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 read this information. Office Word 2007, Word 2010, and later ignore it.

<54> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 write this information when the user chooses to save versions in the document. Neither Office Word 2007, Word 2010, nor Word 2013 and later write this information.

<55> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 read this information. Office Word 2007, Word 2010, and later ignore it.

<56> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 write this information. Neither Office Word 2007, Word 2010, nor Word 2013 and later write this information.

<57> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 read this information. Office Word 2007, Word 2010, and later ignore it.

<58> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 write the size of the deprecated numbering field cache at offset **fcPlcfBteLvc** in the Table Stream. Office Word 2007, Word 2010, and later write zero.

<59> Section 2.5.6: Word 97 emits information at offset **fcPlcfLvcPre10** when performing an incremental save. Word 2000 emits information at offset **fcPlcfLvcPre10** on every save. Neither Word 2002, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit information at offset **fcPlcfLvcPre10** and the value of **fcPlcfLvcPre10** is undefined.

<60> Section 2.5.6: Word 97 and Word 2000 read this information. Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore it.

<61> Section 2.5.6: Word 97 and Word 2000 write **lcbPlcfLvcPre10** with the size, in bytes, of the information emitted at offset **fcPlcfLvcPre10**. Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later write 0 to **lcbPlcfLvcPre10**.

<62> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 write this information when the user chooses to save versions in the document. Neither Office Word 2007, Word 2010, nor Word 2013 and later write this information.

<63> Section 2.5.6: Word 97, Word 2000, Word 2002, and Office Word 2003 read this information. Office Word 2007, Word 2010, and later ignore it.

<64> Section 2.5.7: Office Word 2007, Word 2010, and later ignore this information. Word 2000, Word 2002, and Office Word 2003 read this information, however the information is an optional, deprecated cache that can be calculated by reading the document content.

<65> Section 2.5.7: Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore this value.

<66> Section 2.5.7: Word 2000 and Word 2002 emit information at offset **fcPgdMotherOld**. Neither Word 97, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<67> Section 2.5.7: Word 2000 and Word 2002 read this information. Word 97, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

<68> Section 2.5.7: Word 2000 and Word 2002 emit information at offset **fcBkdMotherOld**. Neither Word 97, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

Copyright © 2025 Microsoft Corporation

<69> Section 2.5.7: Word 2000 and Word 2002 read this information. Word 97, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

<70> Section 2.5.7: Word 2000 and Word 2002 emit information at offset **fcPgdFtnOld**. Neither Word 97, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<71> Section 2.5.7: Word 2000 and Word 2002 read this information. Word 97, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

<72> Section 2.5.7: Word 2000 and Word 2002 emit information at offset **fcBkdFtnOld**. Neither Word 97, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<73> Section 2.5.7: Word 2000 and Word 2002 read this information. Word 97, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

<74> Section 2.5.7: Word 2000 and Word 2002 emit information at offset **fcPgdEdnOld**. Neither Word 97, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<75> Section 2.5.7: Word 2000 and Word 2002 read this information. Word 97, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

<76> Section 2.5.7: Word 2000 and Word 2002 emit information at offset **fcBkdEdnOld**. Neither Word 97, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<77> Section 2.5.7: Word 2000 and Word 2002 read this information. Word 97, Office Word 2003, Office Word 2007, Word 2010, and later ignore this information.

<78> Section 2.5.8: Office Word 2007, Word 2010, and later ignore this value.

<79> Section 2.5.8: Word 2002 and Office Word 2003 write this information when the user chooses to save versions in the document. Neither Word 97, Word 2000, Office Word 2007, Word 2010, nor Word 2013 and later write this information.

<80> Section 2.5.8: Word 2002 and Office Word 2003 read this information. Word 97, Word 2000, Office Word 2007, Word 2010, and later ignore it.

<81> Section 2.5.8: Word 2002 emits information at offset **fcPlcfpmiOldXP**. Neither Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcfpmiOldXP** is undefined.

<82> Section 2.5.8: Word 2002 reads this information. Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, and later ignore it.

<83> Section 2.5.8: Word 2002 writes **lcbPlcfpmiOldXP** with the size, in bytes, of the information emitted at offset **fcPlcfpmiOldXP**. Office Word 2003, Office Word 2007, Word 2010, and later write 0 to **lcbPlcfpmiOldXP**. Neither Word 97 nor Word 2000 write a **FibRgFcLcb2002**.

<84> Section 2.5.8: Word 2002 emits information at offset **fcPlcfpmiNewXP**. Neither Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcfpmiNewXP** is undefined.

<85> Section 2.5.8: Word 2002 reads this information. Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, and later ignore it.

<86> Section 2.5.8: Word 2002 writes **lcbPlcfpmiNewXP** with the size, in bytes, of the information emitted at offset **fcPlcfpmiNewXP**. Office Word 2003, Office Word 2007, Word 2010, and later write 0 to **lcbPlcfpmiNewXP**. Neither Word 97 nor Word 2000 write a **FibRgFcLcb2002**.

Copyright © 2025 Microsoft Corporation

<87> Section 2.5.8: Word 2002 emits information at offset **fcPlcfpmiMixedXP**. Neither Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcfpmiMixedXP** is undefined.

<88> Section 2.5.8: Word 2002 reads this information. Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, and later ignore it.

<89> Section 2.5.8: Word 2002 writes **lcbPlcfpmiMixedXP** with the size, in bytes, of the information emitted at offset **fcPlcfpmiMixedXP**. Office Word 2003, Office Word 2007, Word 2010, and later write 0 to **lcbPlcfpmiMixedXP**. Neither Word 97 nor Word 2000 write a **FibRgFcLcb2002**.

<90> Section 2.5.8: Word 2002 emits information at offset **fcPlcflvcOldXP**. Neither Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcflvcOldXP** is undefined.

<91> Section 2.5.8: Word 2002 reads this information. Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, and later ignore it.

<92> Section 2.5.8: Word 2002 writes **lcbPlcflvcOldXP** with the size, in bytes, of the information emitted at offset **fcPlcflvcOldXP**. Office Word 2003, Office Word 2007, Word 2010, and later write 0 to **lcbPlcflvcOldXP**. Neither Word 97 nor Word 2000 write a **FibRgFcLcb2002**.

<93> Section 2.5.8: Word 2002 emits information at offset **fcPlcflvcNewXP**. Neither Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcflvcNewXP** is undefined.

<94> Section 2.5.8: Word 2002 reads this information. Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, and later ignore it.

<95> Section 2.5.8: Word 2002 writes **lcbPlcflvcNewXP** with the size, in bytes, of the information emitted at offset **fcPlcflvcNewXP**. Office Word 2003, Office Word 2007, Word 2010, and later write 0 to **lcbPlcflvcNewXP**. Neither Word 97 nor Word 2000 write a **FibRgFcLcb2002**.

<96> Section 2.5.8: Word 2002 emits information at offset **fcPlcflvcMixedXP**. Neither Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcflvcMixedXP** is undefined.

<97> Section 2.5.8: Word 2002 reads this information. Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, and later ignore it.

<98> Section 2.5.8: Word 2002 writes **lcbPlcflvcMixedXP** with the size, in bytes, of the information emitted at offset **fcPlcflvcMixedXP**. Office Word 2003, Office Word 2007, Word 2010, and later write 0 to **lcbPlcflvcMixedXP**. Neither Word 97 nor Word 2000 write a **FibRgFcLcb2002**.

<99> Section 2.5.9: Only Office Word 2003 emits information at offset **fcPlcfpmiOld**; Neither Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcfpmiOld** is undefined.

<100> Section 2.5.9: Only Office Word 2003 reads this information.

<101> Section 2.5.9: Office Word 2003 writes **lcbPlcfpmiOld** with the size, in bytes, of the information emitted at offset **fcPlcfpmiOld**; Office Word 2007, Word 2010, and later write 0 to **lcbPlcfpmiOld**.

<102> Section 2.5.9: Only Office Word 2003 emits information at offset **fcPlcfpmiOldInline**; Neither Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcfpmiOldInline** is undefined.

<103> Section 2.5.9: Only Office Word 2003 reads this information.

Copyright © 2025 Microsoft Corporation

<104> Section 2.5.9: Office Word 2003 writes **lcbPlcfpmiOldInline** with the size, in bytes, of the information emitted at offset **fcPlcfpmiOldInline**; Office Word 2007, Word 2010, and later write 0 to **lcbPlcfpmiOldInline**.

<105> Section 2.5.9: Only Office Word 2003 emits information at offset **fcPlcfpmiNew**; Neither Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcfpmiNew** is undefined.

<106> Section 2.5.9: Only Office Word 2003 reads this information.

<107> Section 2.5.9: Office Word 2003 writes **lcbPlcfpmiNew** with the size, in bytes, of the information emitted at offset **fcPlcfpmiNew**; Office Word 2007, Word 2010, and later write 0 to **lcbPlcfpmiNew**.

<108> Section 2.5.9: Only Office Word 2003 emits information at offset **fcPlcfpmiNewInline**; Neither Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcfpmiNewInline** is undefined.

<109> Section 2.5.9: Only Office Word 2003 reads this information.

<110> Section 2.5.9: Office Word 2003 writes **lcbPlcfpmiNewInline** with the size, in bytes, of the information emitted at offset **fcPlcfpmiNewInline**; Office Word 2007, Word 2010, and later write 0 to **lcbPlcfpmiNewInline**.

<111> Section 2.5.9: Only Office Word 2003 emits information at offset **fcPlcflvcOld**; Neither Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcflvcOld** is undefined.

<112> Section 2.5.9: Only Office Word 2003 reads this information.

<113> Section 2.5.9: Office Word 2003 writes **lcbPlcflvcOld** with the size, in bytes, of the information emitted at offset **fcPlcflvcOld**; Office Word 2007, Word 2010, and later write 0 to **lcbPlcflvcOld**.

<114> Section 2.5.9: Only Office Word 2003 emits information at offset **fcPlcflvcOldInline**; Neither Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcflvcOldInline** is undefined.

<115> Section 2.5.9: Only Office Word 2003 reads this information.

<116> Section 2.5.9: Office Word 2003 writes **lcbPlcflvcOldInline** with the size, in bytes, of the information emitted at offset **fcPlcflvcOldInline**; Office Word 2007, Word 2010, and later write 0 to **lcbPlcflvcOldInline**.

<117> Section 2.5.9: Only Office Word 2003 emits information at offset **fcPlcflvcNew**; Neither Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcflvcNew** is undefined.

<118> Section 2.5.9: Only Office Word 2003 reads this information.

<119> Section 2.5.9: Office Word 2003 writes **lcbPlcflvcNew** with the size, in bytes, of the information emitted at offset **fcPlcflvcNew**; Office Word 2007, Word 2010, and later write 0 to **lcbPlcflvcNew**.

<120> Section 2.5.9: Only Office Word 2003 emits information at offset **fcPlcflvcNewInline**; Neither Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset and the value of **fcPlcflvcNewInline** is undefined.

<121> Section 2.5.9: Only Office Word 2003 reads this information.

Copyright © 2025 Microsoft Corporation

<122> Section 2.5.9: Office Word 2003 writes **lcbPlcflvcNewInline** with the size, in bytes, of the information emitted at offset **fcPlcflvcNewInline**; Office Word 2007, Word 2010, and later write 0 to **lcbPlcflvcNewInline**.

<123> Section 2.5.9: Office Word 2003 emits information at offset **fcPgdMother**. Neither Word 97, Word 2000, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<124> Section 2.5.9: Office Word 2003 reads this information. Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, and later ignore this information.

<125> Section 2.5.9: Office Word 2003 emits information at offset **fcBkdMother**. Neither Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<126> Section 2.5.9: Office Word 2003 reads this information. Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, and later ignore this information.

<127> Section 2.5.9: Office Word 2003 emits information at offset **fcAfdMother**. Neither Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<128> Section 2.5.9: Office Word 2003 reads this information. Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, and later ignore this information.

<129> Section 2.5.9: Office Word 2003 emits information at offset **fcPgdFtn**. Neither Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<130> Section 2.5.9: Office Word 2003 reads this information. Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, and later ignore this information.

<131> Section 2.5.9: Office Word 2003 emits information at offset **fcBkdFtn**. Neither Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<132> Section 2.5.9: Office Word 2003 reads this information. Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, and later ignore this information.

<133> Section 2.5.9: Office Word 2003 emits information at offset **fcAfdFtn**. Neither Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<134> Section 2.5.9: Office Word 2003 reads this information. Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, and later ignore this information.

<135> Section 2.5.9: Office Word 2003 emits information at offset **fcPgdEdn**. Neither Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<136> Section 2.5.9: Office Word 2003 reads this information. Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, and later ignore this information.

<137> Section 2.5.9: Office Word 2003 emits information at offset **fcBkdEdn**. Neither Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<138> Section 2.5.9: Office Word 2003 reads this information. Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, and later ignore this information.

<139> Section 2.5.9: Office Word 2003 emits information at offset **fcAfdEdn**. Neither Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, nor Word 2013 and later emit this information.

<140> Section 2.5.9: Office Word 2003 reads this information. Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, and later ignore this information.

Copyright © 2025 Microsoft Corporation

<141> Section 2.5.9: Office Word 2003 emits information at offset **fcAfd**. Neither Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, nor Word 2013 and later emit information at this offset.

<142> Section 2.5.9: Office Word 2003 reads this information. Word 97, Word 2000, Word 2002, Office Word 2007, Word 2010, and later ignore this information.

<143> Section 2.5.10: Neither Office Word 2007, Word 2010, nor Word 2013 and later write 0 here, but all they ignore this value when loading files.

<144> Section 2.5.10: Neither Office Word 2007, Word 2010, nor Word 2013 and later write 0 here, but all they ignore this value when loading files.

<145> Section 2.6.1: Office Word 2007, Word 2010, and later ignore this property when running in compatibility mode for previous versions of Word. Word 97, Word 2000, Word 2002, and Office Word 2003 do not process this Sprm, and thus ignore this property.

<146> Section 2.6.1: When sprmCFSpec is unexpectedly applied to a character that can be displayed, the character can be displayed in the same manner as a character that is not fSpec. If, on the other hand, the character cannot be displayed, it can be ignored.

<147> Section 2.6.1: This property is compatible with Word 97, and for that version the default color for **right-to-left** text is **cvAuto**. Later versions do not use this property, and instead the color of all text is specified by sprmCIco.

<148> Section 2.6.2: Word implements this property by acting as if there is a page break before the paragraph if it would not otherwise fit on the remainder of the page. If sprmPFKeepFollow is applied to the preceding paragraph with a value of 1, Word favors keeping this paragraph's lines together over keeping this paragraph on the same page as the previous paragraph. If the paragraph is too long to fit on a full page by itself, Word ignores this property. If the paragraph is in a table, Word ignores this property.

<149> Section 2.6.2: Word implements this property by acting as if there is a page break before the paragraph if there would otherwise be a page break between the end of this paragraph and the beginning of the next one. If sprmPFKeep is applied to the next paragraph with a value of 1, Word avoids breaking the next paragraph across pages even if it means ignoring sprmPFKeepFollow.

<150> Section 2.6.3: Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore this property. Word 2000 and Word 97 do not split table rows across pages when the table rows set this property to 0x01.

<151> Section 2.6.3: Word 97 stops working if merged cells are split across page break boundaries; setting this property for merged cells avoids this problem.

<152> Section 2.6.3: Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore this property.

<153> Section 2.6.3: Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later evaluate sprmTFCantSplit instead of this property.

<154> Section 2.6.3: Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later evaluate sprmTFCantSplit instead of this property.

<155> Section 2.6.3: If the cell has a fixed-width, this property is false (0), and content cannot fit on a single line in the cell, then content will **word wrap**. If the cell does not have a fixed-width, this property is false (0), and content cannot fit on a single line in the cell, then the cell can grow to fit the content; however, if the cell has no more room to grow, then the content will word wrap instead.

<156> Section 2.6.3: If the cell does not have a fixed width and this property is true, the cell will automatically grow to fit more content, shrinking adjacent cells in the row if necessary so that content

Copyright © 2025 Microsoft Corporation

in this cell does not wrap. However, if the cell content is too large to fit in the table, then the content will be forced to wrap. If multiple cells in the row have this property set and content will not fit on a single line for any them, widths will be adjusted proportionately according to how much content is in each cell (the cell with the most content receives the most width).

<157> Section 2.6.4: Word 97, Word 2000, and Word 2002 emit sprmSDxaColumns only when the space between columns differs from the default.

<158> Section 2.6.4: Word falls back to **msonfcArabic**.

<159> Section 2.6.4: Word 97, Word 2000, and Word 2002 emit sprmSDyaHdrTop only when the header's distance from the top edge of the page differs from the default.

<160> Section 2.6.4: Word 97, Word 2000, and Word 2002 emit sprmSDyaHdrBottom only when the footer distance from the bottom edge of the page differs from the default.

<161> Section 2.6.4: Word's user interface allows starting line numbers only up to 32767, corresponding to a SPRM value of 32766. However, bigger values can be read in (for example from ECMA-376 files) and subsequently stored into an MS-DOC file.

<162> Section 2.6.4: Office Word 2007, Word 2010, and later support larger values.

<163> Section 2.6.4: Word 97, Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore this value when there is only one available paper format as defined by the currently selected printer driver.

<164> Section 2.7.2: With Word 97, Word 2000, Word 2002, and Office Word 2003 it is possible for the **fLockRev** value or the **fLockAtn** value to be set to 1 when **fProtEnabled** is 1.

<165> Section 2.7.2: With Word 97, Word 2000, Word 2002, and Office Word 2003 it is possible for the **fLockRev** value or the **fLockAtn** value to be set to 1 when **fProtEnabled** is 1.

<166> Section 2.7.2: Word 97 allows independent viewing and printing of revision markup. This means that the value of **fRMPrint** can be different from the value of **fRMView**.

<167> Section 2.7.2: With Word 97, Word 2000, Word 2002, and Office Word 2003, it is possible for the **fLockRev** value or the **fLockAtn** value to be set to 1 when **fProtEnabled** is 1.

<168> Section 2.7.2: Word stores either the date and time the document was created or the date and time when personal information was scrubbed.

<169> Section 2.7.2: Word stores either the date and time the document was printed or 4 bytes of zeros (0) if personal information was scrubbed or if the document was never printed.

<170> Section 2.7.2: Word will store a 0 here if personal information was scrubbed.

<171> Section 2.7.2: Word will store a 0 here for certain locales and if personal information was scrubbed. Word does not prevent this value from overflowing if the document was opened for editing more than 0x7FFFFFFF minutes.

<172> Section 2.7.2: Word sets up the save dialog so that, if it is not altered, it saves a comma-delimited text file but does not prevent the user from altering the file type in the dialog.

<173> Section 2.7.4: If Office Word 2007, Word 2010, or Word 2013 and later saved this file as a background operation, this value is 9.

<174> Section 2.7.4: Word 97 sets this value when it loads files through the Microsoft HTML converter (html32.cnv).

Copyright © 2025 Microsoft Corporation

<175> Section 2.7.4: Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later write 0. Word 2002, Office Word 2003, Office Word 2007, Word 2010, and Word 2013 ignore any value on load.

<176> Section 2.7.4: Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later write 0. Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore any value on load.

<177> Section 2.7.4: Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later neither read nor write this value.

<178> Section 2.7.4: Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later neither read nor write this value.

<179> Section 2.7.5: All background saves and all saves by Office Word 2007, Word 2010, and later result in 0 here.

<180> Section 2.7.5: All background saves and all saves by Office Word 2007, Word 2010, and later result in 0 here.

<181> Section 2.7.5: Word does not consistently set this when tentative lists are in the document so it is best to assume that a 1 was written here.

<182> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<183> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<184> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<185> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<186> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<187> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<188> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<189> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<190> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<191> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<192> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<193> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<194> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<195> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<196> Section 2.7.13: Only supported in Office Word 2007, Word 2010, and later.

<197> Section 2.7.14: The Word object model does not validate input and does allow values other than those listed.

<198> Section 2.7.16: Only Word 97 uses this setting. Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later use **iCustomKsu** and **fJapaneseUseLevel2** instead. If Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, or Word 2013 and later loads a Word 97 file, it deduces its settings based on **iCustomKsu** and **fJapaneseUseLevel2** if either are present, or on the contents of **rgxchFPunct** and **rgxchLPunct**. Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later save only the values 0 and 2 and purely for

Copyright © 2025 Microsoft Corporation

backward compatibility. The value of 1 (strict) is instead saved as 2 (custom) with the characters saved in **rgxchFPunct** and **rgxchLPunct**.

<199> Section 2.7.16: Word 97 does not read or write **iCustomKsu**.

<200> Section 2.7.16: Word 97 does not read or write **fJapaneseUseLevel2**.

<201> Section 2.8.29: Office Word 2007, Word 2010, and later ignore this information. Word 2000, Word 2002, and Office Word 2003 read this information.

<202> Section 2.8.29: Office Word 2007, Word 2010, and later write the information specified. Word 2000, Word 2002, and Office Word 2003 write information that depends on the state of the application's internal table character cache at the time the document was saved.

<203> Section 2.9.23: Word 97, Word 2000, and Word 2002 emit this information. Office Word 2003 and 2007 emit 0.

<204> Section 2.9.23: Word 97, Word 2000, and Word 2002 read this information. Office Word 2003 and 2007 ignore it.

<205> Section 2.9.24: Word 97, Word 2000, and Word 2002 ignore this data.

<206> Section 2.9.36: Word 97 does not follow this rule when reading a file.

<207> Section 2.9.36: Word 2000 and Word 97 do not follow this rule when reading a file.

<208> Section 2.9.43: Office Word 2007, Word 2010, and later write COLORREFs that have fAuto set to 0xFF but the other members set to nonzero values. They do this when the user chooses a theme color for the borders of a PGPInfo structure. Because the Word Binary File format does not support Word 2007's theme colors, these COLORREF values are undefined and result in inconsistent behavior across different versions of Word.

<209> Section 2.9.43: Word takes its default color from the window text color of the operating system. If applied shading would result in text being difficult to read, Word switches to the window background color of the operating system. Word also changes its default colors to comply with system-wide accessibility settings.

<210> Section 2.9.48: In Office Word 2003 this structure also contains the **toolbar** visual information for when the application is in the **Reading Layout view**.

<211> Section 2.9.69: Word 97 through Office Word 2003 do not enable or disable optional formats based on these flags. Instead, they can use these flags to record which formats were specified the last time the table was auto-formatted. In such cases, these values are only used as an aid when re-applying a table auto-format. See the details of each flag for specific version behavior.

<212> Section 2.9.69: Word 97, Word 2000, Word 2002, and Office Word 2003 record the setting from the last auto-format on the table. Office Word 2007, Word 2010, and later ignore the value.

<213> Section 2.9.69: Word 97, Word 2000, Word 2002, and Office Word 2003 record the setting from the last auto-format on the table. Office Word 2007, Word 2010, and later ignore the value.

<214> Section 2.9.69: Word 97, Word 2000, Word 2002, and Office Word 2003 record the setting from the last auto-format on the table. Office Word 2007, Word 2010, and later ignore the value.

<215> Section 2.9.69: Word 97, Word 2000, Word 2002, and Office Word 2003 record the setting from the last auto-format on the table. Office Word 2007, Word 2010, and later ignore the value.

<216> Section 2.9.69: Word 97, Word 2000, Word 2002, and Office Word 2003 record the setting from the last auto-format on the table. Office Word 2007, Word 2010, and later ignore the value.

Copyright © 2025 Microsoft Corporation

<217> Section 2.9.69: Office Word 2007, Word 2010, and later **table styles** and Word 97, Word 2000, Word 2002, and Office Word 2003 table auto-formats can have optional formatting for the top row of a table. In Word 97 and Word 2000, the value only reflects whether the optional formatting was applied, rather than what the format is now.

<218> Section 2.9.69: Office Word 2007, Word 2010, and later table styles and Word 97, Word 2000, Word 2002, and Office Word 2003 table auto-formats can have optional formatting for the bottom row of a table. In Word 97 and Word 2000, the value only reflects whether the optional formatting was applied, rather than what the format is now.

<219> Section 2.9.69: Office Word 2007, Word 2010, and later table styles and Word 97, Word 2000, Word 2002, and Office Word 2003 table auto-formats can have optional formatting for the logically leftmost column of a table. In Word 97 and Word 2000, the value only reflects whether the optional formatting was applied, rather than what the format is now.

<220> Section 2.9.69: Office Word 2007, Word 2010, and later table styles and Word 97, Word 2000, Word 2002, and Office Word 2003 table auto-formats can have optional formatting for the logically rightmost column of a table. In Word 97 and Word 2000, the value only reflects whether the optional formatting was applied, rather than what the format is now.

<221> Section 2.9.69: Office Word 2007, Word 2010, and later table styles and Word 97, Word 2000, Word 2002, and Office Word 2003 table auto-formats can have optional formatting for the odd numbered rows of a table. In Word 97 and Office Word 2003, the value only reflects whether the optional formatting was applied, rather than what the format is now.

<222> Section 2.9.69: Office Word 2007, Word 2010, and later table styles and Word 97, Word 2000, Word 2002, and Office Word 2003 table auto-formats can have optional formatting for the odd numbered columns of a table. In Word 97 and Office Word 2003, the value only reflects whether the optional formatting was applied, rather than what the format is now.

<223> Section 2.9.90: Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later do not change the data or header file.

<224> Section 2.9.112: Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later recalculate the appropriate sprmCRgLid0_80 and sprmCRgLid1_80 to apply to each style if **f97LidsSet** is 0. Thus it is safe to set this value to 0. Word 97 does not need to apply the compatibility Sprms.

<225> Section 2.9.121: No version of Word has these additional patterns available through its user interface. However, all versions of Word have these available through **macros**.

<226> Section 2.9.147: Word 97, Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later use this Tplc to link a graphical representation of this list format in the Word List UI to this LSTF.

<227> Section 2.9.158: Word 97, Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later allow the user to directly edit field codes. This can cause the binary data to mismatch the **field type**.

<228> Section 2.9.161: Office Word 2003, Office Word 2007, Word 2010, and Word 2013 ignore the values in the OcxInfo structure but, for backward compatibility, emit values based on the **OLE controls** in the document. The values are populated by finding all the control FLDs in the document and saving the values for the corresponding OLE controls. Previous versions of Word expect that the values in OcxInfo structures and the values of the controls all match. The description of OcxInfo fields specifies the values that are written.

<229> Section 2.9.169: Word 2000 and Word 97 use this value to store a reference count of the shape. Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later ignore this value.

Copyright © 2025 Microsoft Corporation

<230> Section 2.9.181: Word 2002 occasionally writes a value of -31681. This behavior is deprecated.

<231> Section 2.9.182: Word 2002, Office Word 2003 and Office Word 2007 ignore the instance of sprmPChgTabs in this scenario.

<232> Section 2.9.224: Word 2002, Office Word 2003, and Office Word 2007 use all of the columns of the data source when computing the hash. Word 2010 and later ignore the last column when Microsoft Outlook is the data source.

<233> Section 2.9.230: **SttbAuthorAttrib** is ignored and not saved by Word 97, Office Word 2007, Word 2010, and later. It is ignored but saved if read by Word 2000, Word 2002, and Office Word 2003.

<234> Section 2.9.230: **SttbAuthorValue** is ignored and not saved by Word 97, Office Word 2007, Word 2010, and later. It is ignored but saved if read by Word 2000, Word 2002, and Office Word 2003.

<235> Section 2.9.230: **SttbMessageAttrib** is ignored and not saved by Word 97, Office Word 2007, Word 2010, and later. It is ignored but saved if read by Word 2000, Word 2002, and Office Word 2003.

<236> Section 2.9.230: **SttbMessageValue** is ignored and not saved by Word 97, Office Word 2007, Word 2010, and later. It is ignored but saved if read by Word 2000, Word 2002, and Office Word 2003.

<237> Section 2.9.244: Office Word 2007, Word 2010, and later write 1 if the selection is a bullet or number character from a bulleted or numbered list. All versions of Word ignore this bit. Office Word 2007, Word 2010, and later write 0 for **fPrefix**.

<238> Section 2.9.256: Word 97 uses multiple splf values for grammatical errors.

<239> Section 2.9.260: Word 97, Word 2000, and Word 2002 set this value to 1 when performing an incremental save and the style has been modified in such a way that it can affect the height of paragraphs with that style. Office Word 2003, Office Word 2007, Word 2010, and later set the value to 0. If the Plc specified by **fcPlcfPhe** is not emitted, it is safe to set this value to 0.

<240> Section 2.9.271: Styles that are used in the document are not empty. Styles that are unused in the document (latent) are allowed to be empty.

<241> Section 2.9.274: The following table lists the value of **stiMaxWhenSaved** that each version of Word writes.

Version	stiMaxWhenSaved
Word 97	91
Word 2000	105
Word 2002	156
Office Word 2003	156
Office Word 2007	267
Word 2010	267
Word 2013 and later	267

<242> Section 2.9.274: The value of **nVerBuiltInNamesWhenSaved** is used to optimize the performance of loading files. Word displays and saves built-in styles with the current application defined style name as the primary style name. However, if the application defined style names differ between versions (or if the user interface language is different than that in use when the file was saved) when opening a file Word strips off the primary style name of any application defined style and

Copyright © 2025 Microsoft Corporation

then replaces it with the current name. If the value of **nVerBuiltInNamesWhenSaved** in the file matches the current value known to the version of Word opening the file, Word knows that the set of application defined style names saved to the file matches the current set of application defined style names, and replacing is not necessary (at least for that reason.)

Specifying a value of 0 is recommended for maximum compatibility, as it will cause all versions of Word to update the names to whatever set of application defined style names is current, with little performance penalty.

The following table lists the value of **nVerBuiltInNamesWhenSaved** that each version of Word writes.

Version	nVerBuiltInNamesWhenSaved
Word 97	2
Word 2000	3
Word 2002	3
Office Word 2003	4
Office Word 2007	7
Word 2010	7
Word 2013 and later	7

<243> Section 2.9.279: Word 97, Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later do not allow creation of a bookmark whose name violates the constraints upon valid strings described in this section but if a string violating them is written to file, it will be handled without error and displayed as expected.

<244> Section 2.9.286: When a new font is applied in a document, Word adds is to the font table if it is not already there. However, when the user edits a document such that a font is no longer used, the entry is not removed from the font table. Thus, the font table will accumulate unused font references over time.

<245> Section 2.9.289: Word 97 only writes 4 strings.

<246> Section 2.9.289: Word 97 emits 0x0004 for cData.

<247> Section 2.9.297: Word 97 and Word 2000 incorrectly write 26. Regardless, Word 97 and Word 2000 correctly read and write SttbTtmbd**.rgTTMBD** 10 bytes after the beginning of SttbW6. Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later write 10.

<248> Section 2.9.298: Word 97, Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, Word 2013, and later allow a value to be set using the name **"Sign"** that is not the VBA digital signature if the document does not contain a VBA project or if the file contains a VBA project but is unsigned. In the case where a VBA project is present but is not signed, specifying a value with this name will cause Microsoft Word to view the file as having an invalid signature for the VBA project on a subsequent load.

<249> Section 2.9.298: Word 97, Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and Word 2013 allow a value to be set using the name **"SigAgile"** that is not the VBA digital signature. Word 2016 and later allows a value to be set using this name that is not the VBA digital signature if the document does not contain a VBA project or if the file contains a VBA project but is unsigned. In the case where a VBA project is present but is not signed, specifying a value with this name will cause Word 2016 and later to view the file as having an invalid signature for the VBA project on a subsequent load.

<250> Section 2.9.298: Word 97, Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, Word 2013, and later allow a value to be set using this name that is not the VBA digital

Copyright © 2025 Microsoft Corporation

signature if the document does not contain a VBA project or if the file contains a VBA project but is unsigned. In the case where a VBA project is present but is not signed, specifying a value with this name will cause Microsoft Word to view the file as having an invalid signature for the VBA project on a subsequent load.

<251> Section 2.9.298: Word 97, Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and Word 2013 allow a value to be set using this name that is not the VBA digital signature. Word 2016 and later allows a value to be set using this name that is not the VBA digital signature if the document does not contain a VBA project or if the file contains a VBA project but is unsigned. In the case where a VBA project is present but is not signed, specifying a value with this name will cause Word 2016 and later to view the file as having an invalid signature for the VBA project on a subsequent load.

<252> Section 2.9.307: If the first row in the selection contains fewer cells than the last row in the selection, and the selection began at a cell index greater than the number of cells in the first row, then **itcFirst** will be greater than the number of cells in the first row, and the selection is interpreted as being the **end of row mark**.

<253> Section 2.9.307: In some cases when the selection spans rows with differing cell counts, Word 97, Word 2000, Word 2002, Office Word 2003, Office Word 2007, Word 2010, and later write an **itcLim** that is less than or equal to **itcFirst**.

<254> Section 2.9.307: Office Word 2003, Office Word 2007, Word 2010, and later ignore the Selsf if **itcLim** is 64.

<255> Section 2.9.311: If the **toolbar control** associated to this TBDelta is a **custom toolbar control** of type Popup, but the toolbar control does not drop a custom **menu toolbar**, the value of **iTB** can be greater or equal than the value of the **cCust** field of the CTBWRAPPER structure that contains the **rCustomizations** array that contains the Customization structure that contains the **customizationData** array that contains this structure, and is ignored.

<256> Section 2.9.312: Word 97, Word 2000, Word 2002 and Office Word 2003 emit this information. Office Word 2007, Word 2010, and later emit 0.

<257> Section 2.9.312: Word 97, Word 2000, Word 2002 and Office Word 2003 read this information. Neither Office Word 2007, Word 2010, nor Word 2013 and later read this information.

<258> Section 2.9.312: Word 97, Word 2000, Word 2002 and Office Word 2003 emit this information. Office Word 2007, Word 2010, and later emit 0.

<259> Section 2.9.312: Word 97, Word 2000, Word 2002 and Office Word 2003 read this information. Neither Office Word 2007, Word 2010, nor Word 2013 and later read this information.

<260> Section 2.9.312: Word 97, Word 2000, Word 2002 and Office Word 2003 emit this information. Office Word 2007, Word 2010, and later emit 0.

<261> Section 2.9.312: Word 97, Word 2000, Word 2002 and Office Word 2003 will read this information. Neither Office Word 2007, Word 2010, nor Word 2013 and later read this information.

<262> Section 2.9.320: Office Word 2007, Word 2010, and later write 0 and ignore the **Tch**. Word 2000, Word 2002, and Office Word 2003 read and write this information.

<263> Section 2.9.326: Word 97 and Word 2000 set this value to the index of the predefined table auto-format that was last applied to this table. Neither Word 2002, Office Word 2003, Office Word 2007, Word 2010, nor Word 2013 and later set this value.

<264> Section 2.9.326: Word 97, Word 2000, and Office Word 2003 do not enable or disable optional formats based on these flags all the time. Instead, they can use these flags to record which formats were specified the last time the table was auto-formatted. In such cases, these values are

Copyright © 2025 Microsoft Corporation

only used as an aid when re-applying a table auto-format. See the details of each flag for specific version.

Copyright © 2025 Microsoft Corporation

6 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
2 Structures	Updated reference sections for [ECMA-376].	Minor
2.1.4 ObjectPool Storage	Added the description of ObjectPool Storage.	Minor
2.2.4 STTB	Updated the bit table of STTB.	Minor
2.9.33 ChpxFkp	Updated the description of ChpxFkp rgb.	Minor
2.9.277 SttbfAtnBkmk	Updated the bit table of SttbfAtnBkmk.	Minor
2.9.290 SttbfRMark	Updated the bit table of SttbfRMark.	Minor

Copyright © 2025 Microsoft Corporation

7 Index

0

0Table stream structure 29

1

1Table stream structure 29

A

Acd structure 202
Afd structure 203
Algorithms
  application data for VtHyperlink 55
  applying properties 48
  determining cell boundaries 46
  determining paragraph boundaries 42
  determining row boundaries 47
  retrieving text 41
Applicability 27
Application data for VtHyperlink algorithm 55
Applying properties algorithms 48
ASUMY structure 204
Asumyi structure 176
ATNBE structure 204
AtrdExtra structure 204
ATRDPost10 structure 205
ATRDPre10 structure 205

B

Basic types
  Acd 202
  Afd 203
  ASUMY 204
  ATNBE 204
  AtrdExtra 204
  ATRDPost10 205
  ATRDPre10 205
  BKC 206
  BKF 207
  BKFD 207
  BKL 208
  BKLD 208
  BlockSel 209
  Bool16 209
  Bool8 209
  Brc 209
  Brc80 210
  Brc80MayBeNil 210
  BrcCvOperand 210
  BrcMayBeNil 211
  BrcOperand 211
  BrcType 211
  BxPap 218
  CAPI 218
  CDB 219
  CellHideMarkOperand 220
  CellRangeFitText 220
  CellRangeNoWrap 220
  CellRangeTextFlow 221
  CellRangeVertAlign 221
  CFitTextOperand 221
  Chpx 222
  ChpxFkp 222
  Cid 223
  CidAllocated 223
  CidFci 223
  CidMacro 226
  Clx 227
  CMajorityOperand 227
  Cmt enumeration 227
  CNFOperand 228
  CNS enumeration 228
  COLORREF 229
  COSL 229
  CSSA 230
  CSSAOperand 230
  CSymbolOperand 231
  CTB 231
  CTBWRAPPER 233
  Customization 233
  DCS 234
  DefTableShd80Operand 235
  DefTableShdOperand 235
  DispFldRmOperand 235
  Dofr 236
  DofrFsn 236
  DofrFsnFnm 237
  DofrFsnName 238
  DofrFsnp 238
  DofrFsnSpbd 238
  Dofrh 239
  DofrRglstsf 239
  Dofrt enumeration 240
  DPCID 240
  DTTM 241
  FACTOIDINFO 242
  FactoidSpls 242
  FarEastLayoutOperand 242
  Fatl 243
  FBKF 244
  FBKFD 244
  FBKLD 244
  FcCompressed 245
  FCCT 246
  Fci enumeration 247
  FCKS 315
  FCKSOLD 316
  FFData 317
  FFDataBits 319
  FFID 320
  FFM enumeration 321
  FFN 321
  FieldMapBase 323
  FieldMapDataItem 323
  FieldMapInfo 324
  FieldMapTerminator 325
  FilterDataItem 325
  Fld 326
  fldch 326
  flt enumeration 326
  FNFB 329

Copyright © 2025 Microsoft Corporation

FNIF 330
FNPI 330
FOBJH 331
FrameTextFlowOperand 331
FSDAP 332
Fsnk enumeration 332
Fssd 332
FssUnits 333
FTO 333
Fts 333
FtsWWidth_Indent 334
FtsWWidth_Table 334
FtsWWidth_TablePart 335
FTXBXNonReusable 335
FTXBXS 336
FTXBXSReusable 337
GOSL 337
GrammarSpls 338
grffldEnd 338
grfhic 339
GRFSTD 340
GrLPUpxSw 341
GrpPrlAndIstd 341
HFD 341
HFDBits 342
Hplxsdr 342
HresiOperand 343
Ico 343
IDPCI 344
Ipat 345
IScrollType 349
ItcFirstLim 349
Kcm 349
Kme 350
Kt enumeration 350
Kul enumeration 351
LadSpls 351
LBCOperand 352
LEGOXTR_V11 352
LFLVLO 354
LFO 353
LFOData 354
LID 355
LPStd 355
LPStshi 355
LPStshiGrpPrl 355
LPUpxChpx 356
LPUpxChpxRM 356
LPUpxPapx 356
LPUpxPapxRM 357
LPUpxRM 357
LPUpxTapx 357
LPXCharBuffer9 358
LSD 358
LSPD 359
LSTF 359
Lstsf 360
LVL 360
LVLF 361
MacroName 363
MacroNames 364
MathPrOperand 364
Mcd 364
MDP 365
MFPF 365
NilBrc 366
NilPICFAndBinData 366
NumRM 367
NumRMOperand 369
OcxInfo 369
ODSOPropertyBase 370
ODSOPropertyLarge 372
ODSOPropertyStandard 372
ODT 372
ODTPersist1 373
ODTPersist2 374
OfficeArtClientAnchor 375
OfficeArtClientData 375
OfficeArtClientTextbox 375
OfficeArtContent 376
OfficeArtWordDrawing 376
PANOSE 377
PapxFkp 381
PapxInFkp 382
PbiGrfOperand 382
Pcd 383
Pcdt 383
PChgTabsAdd 384
PChgTabsDel 384
PChgTabsDelClose 384
PChgTabsOperand 385
PChgTabsPapxOperand 386
PgbApplyTo 386
PgbOffsetFrom 386
PgbPageDepth 386
PGPArray 387
PGPInfo 387
PGPOptions 388
PICF 389
PICF_Shape 390
PICFAndOfficeArtData 390
PICMID 391
PlcfGlsy 393
PlfAcd 393
PlfCosl 393
PlfGosl 394
PlfguidUim 394
PlfKme 395
PlfLfo 395
PlfLst 395
PlfMcd 396
PLRSID 396
Pmfs 397
Pms 399
PnFkpChpx 401
PnFkpPapx 401
PositionCodeOperand 401
Prc 402
PrcData 402
PrDrvr 402
PrEnvLand 403
PrEnvPort 403
Prm 403
Prm0 403
Prm1 405
PropRMark 405
PropRMarkOperand 406
ProtectionType 406
PRTI 406
PTIstdInfoOperand 407

Copyright © 2025 Microsoft Corporation

Rca 407
RecipientBase 408
RecipientDataItem 408
RecipientInfo 409
RecipientTerminator 410
Rfs 410
RqCdb 411
RqOcxInfo 411
RmdThreading 412
Rnc 416
RouteSlip 417
RouteSlipInfo 418
RouteSlipProtectionEnum 419
SBkcOperand 419
SBOrientationOperand 419
SClmOperand 419
SDmBinOperand 420
SDTI 420
SDTT 421
SDxaColSpacingOperand 421
SDxaCoWidthOperand 421
Sed 422
Selsf 422
Sepx 424
SFpcOperand 425
Shd 425
Shd80 426
SHDOperand 427
SLncOperand 427
SmartTagData 427
SortColumnAndDirection 428
Spa 428
SpellingSpls 430
SPgbPropOperand 431
SPLS 431
SPPOperand 432
STD 432
Stdf 433
StdfBase 433
StdfPost2000 435
StdfPost2000OrNone 436
StkCharGRLPUPX 436
StkCharLPUpxGrLPUpxRM 437
StkCharUpxGrLPUpxRM 437
StkListGRLPUPX 437
StkParaGRLPUPX 438
StkParaLPUpxGrLPUpxRM 438
StkParaUpxGrLPUpxRM 439
StkTableGRLPUPX 439
STSH 440
STSHI 441
STSHIB 441
Stshif 442
StshiLsd 443
SttbfAssoc 443
SttbfAtnBkmk 444
SttbfAutoCaption 445
SttbfBkmk 446
SttbfBkmkBPRepairs 450
SttbfBkmkFactoid 451
SttbfBkmkFcc 452
SttbfBkmkProt 453
SttbfBkmkSdt 454
SttbfCaption 455
SttbfFfn 456
SttbfGlsy 456
SttbFnm 457
SttbfRfs 458
SttbfRMark 459
SttbGlsyStyle 460
SttbListNames 461
SttbProtUser 462
SttbRgtplc 463
SttbSavedBy 463
SttbTtmbd 464
SttbW6 465
StwUser 465
Sty 466
TabJC enumeration 467
TabLC enumeration 467
TableBordersOperand 468
TableBordersOperand80 469
TableBrc80Operand 469
TableBrcOperand 470
TableCellWidthOperand 471
TableSel 471
TableShadeOperand 472
TBC 472
TBD 472
TBDelta 473
Tbkd 475
TC80 475
TCellBrcTypeOperand 476
Tcg 476
Tcg255 477
TCGRF 477
TcgSttbf 478
TcgSttbfCore 478
Tch 479
TDefTableOperand 480
TDxaColOperand 480
TextFlow 481
TInsertOperand 481
TIQ 481
TLP 482
ToggleOperand 482
Tplc 483
TplcBuildIn 483
TplcUser 484
Ttmbd 484
UFEL 485
UID enumeration 486
UidSel 486
UIM 486
UpxChpx 487
UPXPadding 488
UpxPapx 488
UpxRm 489
UpxTapx 490
VerticalAlign enumeration 492
VerticalMergeFlag enumeration 492
VertMergeOperand 492
Vjc enumeration 493
WHeightAbs 493
WKB 493
Wpms 494
Wpmsdt 495
XAS value 495
XAS_nonNeg value 495
XAS_plusOne value 495

Copyright © 2025 Microsoft Corporation

XSDR 496
Xst 496
Xstz 497
YAS value 497
YAS_nonNeg value 497
YAS_plusOne value 497
BKC structure 206
BKF structure 207
BKFD structure 207
BKL structure 208
BKLD structure 208
BlockSel structure 209
Bookmark example 508
Bool16 structure 209
Bool8 structure 209
Brc structure 209
Brc80 structure 210
Brc80MayBeNil structure 210
BrcCvOperand structure 210
BrcMayBeNil structure 211
BrcOperand structure 211
BrcType structure 211
BxPap structure 218
Byte ordering - overview 26

C

CAPI structure 218
CDB structure 219
CellHideMarkOperand structure 220
CellRangeFitText structure 220
CellRangeNoWrap structure 220
CellRangeTextFlow structure 221
CellRangeVertAlign structure 221
CFitTextOperand structure 221
Change tracking 565
Character Position (CP) - fundamental concepts 31
Character property modifiers 105
Characters - overview 25
Chpx structure 222
ChpxFkp structure 222
Cid structure 223
CidAllocated structure 223
CidFci structure 223
CidMacro structure 226
Clx example 498
Clx structure 227
CMajorityOperand structure 227
Cmt enumeration 227
CNFOperand structure 228
CNS enumeration 228
COLORREF structure 229
Comments structure 40
Copts structure 173
Copts60 structure 171
Copts80 structure 172
COSL structure 229
CSSA structure 230
CSSAOperand structure 230
CSymbolOperand structure 231
CTB structure 231
CTBWRAPPER structure 233
Custom XML Data storage structure 30
Customization structure 233

D

Data stream structure 29
DCS structure 234
DefTableShd80Operand structure 235
DefTableShdOperand structure 235
Details
0Table stream structure 29
1Table stream structure 29
Acd structure 202
Afd structure 203
application data for VtHyperlink 55
applying properties 48
ASUMY structure 204
Asumyi structure 176
ATNBE structure 204
AtrdExtra structure 204
ATRDPost10 structure 205
ATRDPre10 structure 205
BKC structure 206
BKF structure 207
BKFD structure 207
BKL structure 208
BKLD structure 208
BlockSel structure 209
Bool16 structure 209
Bool8 structure 209
Brc structure 209
Brc80 structure 210
Brc80MayBeNil structure 210
BrcCvOperand structure 210
BrcMayBeNil structure 211
BrcOperand structure 211
BrcType structure 211
BxPap structure 218
CAPI structure 218
CDB structure 219
CellHideMarkOperand structure 220
CellRangeFitText structure 220
CellRangeNoWrap structure 220
CellRangeTextFlow structure 221
CellRangeVertAlign structure 221
CFitTextOperand structure 221
character properties 105
Chpx structure 222
ChpxFkp structure 222
Cid structure 223
CidAllocated structure 223
CidFci structure 223
CidMacro structure 226
Clx structure 227
CMajorityOperand structure 227
Cmt enumeration 227
CNFOperand structure 228
CNS enumeration 228
COLORREF structure 229
comments 40
Copts structure 173
Copts60 structure 171
Copts80 structure 172
COSL structure 229
CSSA structure 230
CSSAOperand structure 230
CSymbolOperand structure 231
CTB structure 231

Copyright © 2025 Microsoft Corporation

CTBWRAPPER structure 233
Custom XML Data storage structure 30
Customization structure 233
Data stream structure 29
DCS structure 234
DefTableShd80Operand structure 235
DefTableShdOperand structure 235
determining cell boundaries 46
determining paragraph boundaries 42
determining row boundaries 47
DispFldRmOperand structure 235
document content 41
document parts 39
Document Summary Information stream structure 30
Dofr structure 236
DofrFsn structure 236
DofrFsnFnm structure 237
DofrFsnName structure 238
DofrFsnp structure 238
DofrFsnSpbd structure 238
Dofrh structure 239
DofrRglstsf structure 239
Dofrt enumeration 240
Dogrid structure 177
Dop structure 148
Dop2000 structure 159
Dop2002 structure 163
Dop2003 structure 166
Dop2007 structure 168
Dop2010 structure 170
Dop2013 structure 170
Dop95 structure 155
Dop97 structure 156
DopBase structure 149
DopMth structure 180
DopTypography structure 178
DPCID structure 240
DTTM structure 241
Encryption stream structure 30
endnotes 41
FACTOIDINFO structure 242
FactoidSpls structure 242
FarEastLayoutOperand structure 242
Fatl structure 243
FBKF structure 244
FBKFD structure 244
FBKLD structure 244
FcCompressed structure 245
FCCT structure 246
Fci enumeration 247
FCKS structure 315
FCKSOLD structure 316
FFData structure 317
FFDataBits structure 319
FFID structure 320
FFM enumeration 321
FFN structure 321
Fib structure 55
FibBase structure 57
FibRgCswNew structure 102
FibRgCswNewData2000 structure 103
FibRgCswNewData2007 structure 103
FibRgFcLcb structure 62
FibRgFcLcb2000 structure 82
FibRgFcLcb2002 structure 85
FibRgFcLcb2003 structure 92
FibRgFcLcb2007 structure 99
FibRgFcLcb97 structure 62
FibRgLw97 structure 60
FibRgW97 structure 59
FieldMapBase structure 323
FieldMapDataItem structure 323
FieldMapInfo structure 324
FieldMapTerminator structure 325
FilterDataItem structure 325
Fld structure 326
fldch structure 326
flt enumeration 326
FNFB structure 329
FNIF structure 330
FNPI structure 330
FOBJH structure 331
footnotes 39
FrameTextFlowOperand structure 331
FSDAP structure 332
Fsnk enumeration 332
Fssd structure 332
FssUnits structure 333
FTO structure 333
Fts structure 333
FtsWWidth_Indent structure 334
FtsWWidth_Table structure 334
FtsWWidth_TablePart structure 335
FTXBXNonReusable structure 335
FTXBXS structure 336
FTXBXSReusable structure 337
GOSL structure 337
GrammarSpls structure 338
grffldEnd structure 338
grfhic structure 339
GRFSTD structure 340
GrLPUpxSw structure 341
GrpPrlAndIstd structure 341
header textboxes 41
headers 39
HFD structure 341
HFDBits structure 342
how to read the Fib 104
Hplxsdr structure 342
HresiOperand structure 343
Ico structure 343
IDPCI structure 344
Information Rights Management Data Space storage structure 31
Ipat structure 345
IScrollType structure 349
ItcFirstLim structure 349
Kcm structure 349
Kme structure 350
Kt enumeration 350
Kul enumeration 351
LadSpls structure 351
LBCOperand structure 352
LEGOXTR_V11 structure 352
LFO structure 353
LFOData structure 354
LFOLVL structure 354
LID structure 355
LPStd structure 355

Copyright © 2025 Microsoft Corporation

LPStshi structure 355
LPStshiGrpPrl structure 355
LPUpxChpx structure 356
LPUpxChpxRM structure 356
LPUpxPapx structure 356
LPUpxPapxRM structure 357
LPUpxRM structure 357
LPUpxTapx structure 357
LPXCharBuffer9 structure 358
LSD structure 358
LSPD structure 359
LSTF structure 359
Lstsf structure 360
LVL structure 360
LVLF structure 361
MacroName structure 363
MacroNames structure 364
Macros storage structure 31
main document 39
MathPrOperand structure 364
Mcd structure 364
MDP structure 365
MFPF structure 365
nFib value 104
NilBrc structure 366
NilPICFAndBinData structure 366
NumRM structure 367
NumRMOperand structure 369
ObjectPool storage structure 29
OcxInfo structure 369
ODSOPropertyBase structure 370
ODSOPropertyLarge structure 372
ODSOPropertyStandard structure 372
ODT structure 372
ODTPersist1 structure 373
ODTPersist2 structure 374
OfficeArtClientAnchor structure 375
OfficeArtClientData structure 375
OfficeArtClientTextbox structure 375
OfficeArtContent structure 376
OfficeArtWordDrawing structure 376
overview of tables 43
PANOSE structure 377
PapxFkp structure 381
PapxInFkp structure 382
paragraph properties 120
PbiGrfOperand structure 382
Pcd structure 383
Pcdt structure 383
PChgTabsAdd structure 384
PChgTabsDel structure 384
PChgTabsDelClose structure 384
PChgTabsOperand structure 385
PChgTabsPapxOperand structure 386
PgbApplyTo structure 386
PgbOffsetFrom structure 386
PgbPageDepth structure 386
PGPArray structure 387
PGPInfo structure 387
PGPOptions structure 388
PICF structure 389
PICF_Shape structure 390
PICFAndOfficeArtData structure 390
PICMID structure 391
picture properties 148
Plcbkf structure 182
Plcbkfd structure 183
Plcbkl structure 184
Plcbkld structure 184
PlcBteChpx structure 185
PlcBtePapx structure 185
PlcfandRef structure 186
PlcfandTxt structure 186
PlcfAsumy structure 187
Plcfbkf structure 187
Plcfbkfd structure 188
Plcfbkl structure 189
Plcfbkld structure 189
Plcfcookie structure 189
PlcfcookieOld structure 190
PlcfendRef structure 190
PlcfendTxt structure 191
Plcffactoid structure 191
PlcffndRef structure 192
PlcffndTxt structure 192
PlcfGlsy structure 393
Plcfgram structure 193
Plcfhdd structure 193
PlcfHdrtxbxTxt structure 194
Plcflad structure 194
Plcfld structure 195
PlcfSed structure 196
PlcfSpa structure 196
Plcfspl structure 197
PlcfTch structure 197
PlcfTxbxBkd structure 198
PlcfTxbxHdrBkd structure 199
PlcftxbxTxt structure 199
Plcfuim structure 200
PlcfWKB structure 200
PlcPcd structure 201
PlfAcd structure 393
PlfCosl structure 393
PlfGosl structure 394
PlfguidUim structure 394
PlfKme structure 395
PlfLfo structure 395
PlfLst structure 395
PlfMcd structure 396
PLRSID structure 396
Pmfs structure 397
Pms structure 399
PnFkpChpx structure 401
PnFkpPapx structure 401
PositionCodeOperand structure 401
Prc structure 402
PrcData structure 402
PrDrvr structure 402
PrEnvLand structure 403
PrEnvPort structure 403
Prm structure 403
Prm0 structure 403
Prm1 structure 405
PropRMark structure 405
PropRMarkOperand structure 406
Protected Content stream structure 31
ProtectionType structure 406
PRTI structure 406
PTIstdInfoOperand structure 407
Rca structure 407

Copyright © 2025 Microsoft Corporation

RecipientBase structure 408
RecipientDataItem structure 408
RecipientInfo structure 409
RecipientTerminator structure 410
retrieving text 41
Rfs structure 410
RgCdb structure 411
RgOcxInfo structure 411
RmdThreading structure 412
Rnc structure 416
RouteSlip structure 417
RouteSlipInfo structure 418
RouteSlipProtectionEnum structure 419
SBkcOperand structure 419
SBOrientationOperand structure 419
SClmOperand structure 419
SDmBinOperand structure 420
SDTI structure 420
SDTT structure 421
SDxaColSpacingOperand structure 421
SDxaColWidthOperand structure 421
section properties 140
Sed structure 422
Selsf structure 422
Sepx structure 424
SFpcOperand structure 425
Shd structure 425
Shd80 structure 426
SHDOperand structure 427
signatures stream structure 31
single property modifiers 104
SLncOperand structure 427
SmartTagData structure 427
SortColumnAndDirection structure 428
Spa structure 428
SpellingSpls structure 430
SPgbPropOperand structure 431
SPLS structure 431
SPPOperand structure 432
STD structure 432
Stdf structure 433
StdfBase structure 433
StdfPost2000 structure 435
StdfPost2000OrNone structure 436
StkCharGRLPUPX structure 436
StkCharLPUpxGrLPUpxRM structure 437
StkCharUpxGrLPUpxRM structure 437
StkListGRLPUPX structure 437
StkParaGRLPUPX structure 438
StkParaLPUpxGrLPUpxRM structure 438
StkParaUpxGrLPUpxRM structure 439
StkTableGRLPUPX structure 439
STSH structure 440
STSHI structure 441
STSHIB structure 441
Stshif structure 442
StshiLsd structure 443
SttbfAssoc structure 443
SttbfAtnBkmk structure 444
SttbfAutoCaption structure 445
SttbfBkmk structure 446
SttbfBkmkBPRepairs structure 450
SttbfBkmkFactoid structure 451
SttbfBkmkFcc structure 452
SttbfBkmkProt structure 453
SttbfBkmkSdt structure 454
SttbfCaption structure 455
SttbfFfn structure 456
SttbfGlsy structure 456
SttbFnm structure 457
SttbfRfs structure 458
SttbfRMark structure 459
SttbGlsyStyle structure 460
SttbListNames structure 461
SttbProtUser structure 462
SttbRgtplc structure 463
SttbSavedBy structure 463
SttbTtmbd structure 464
SttbW6 structure 465
StwUser structure 465
Sty structure 466
Summary Information stream structure 30
TabJC enumeration 467
TabLC enumeration 467
table overview 43
table properties 131
TableBordersOperand structure 468
TableBordersOperand80 structure 469
TableBrc80Operand structure 469
TableBrcOperand structure 470
TableCellWidthOperand structure 471
TableSel structure 471
TableShadeOperand structure 472
TBC structure 472
TBD structure 472
TBDelta structure 473
Tbkd structure 475
TC80 structure 475
TCellBrcTypeOperand structure 476
Tcg structure 476
Tcg255 structure 477
TCGRF structure 477
TcgSttbf structure 478
TcgSttbfCore structure 478
Tch structure 479
TDefTableOperand structure 480
TDxaColOperand structure 480
textboxes 41
TextFlow structure 481
TInsertOperand structure 481
TIQ structure 481
TLP structure 482
ToggleOperand structure 482
Tplc structure 483
TplcBuildIn structure 483
TplcUser structure 484
Ttmbd structure 484
UFEL structure 485
UID enumeration 486
UidSel structure 486
UIM structure 486
UpxChpx structure 487
UPXPadding structure 488
UpxPapx structure 488
UpxRm structure 489
UpxTapx structure 490
VerticalAlign enumeration 492
VerticalMergeFlag enumeration 492
VertMergeOperand structure 492
Vjc enumeration 493

Copyright © 2025 Microsoft Corporation

WHeightAbs structure 493
WKB structure 493
WordDocument stream structure 29
Wpms structure 494
Wpmsdt structure 495
XAS value 495
XAS_nonNeg value 495
XAS_plusOne value 495
XML signatures storage structure 31
XSDR structure 496
Xst structure 496
Xstz structure 497
YAS value 497
YAS_nonNeg value 497
YAS_plusOne value 497
Determining cell boundaries algorithm 46
Determining paragraph boundaries algorithm 42
Determining row boundaries algorithm 47
DispFldRmOperand structure 235
Document content structure 41
Document parts structure 39
Document Summary Information stream structure 30
Dofr structure 236
DofrFsn structure 236
DofrFsnFnm structure 237
DofrFsnName structure 238
DofrFsnp structure 238
DofrFsnSpbd structure 238
Dofrh structure 239
DofrRglstsf structure 239
Dofrt enumeration 240
Dogrid structure 177
Dop structure 148
Dop2000 structure 159
Dop2002 structure 163
Dop2003 structure 166
Dop2007 structure 168
Dop2010 structure 170
Dop2013 structure 170
Dop95 structure 155
Dop97 structure 156
DopBase structure 149
DopMth structure 180
DopTypography structure 178
DPCID structure 240
DTTM structure 241

E

Encryption - fundamental concepts 37
Encryption stream structure 30
Endnotes structure 41
Example of a Bookmark example 508
Example of a Clx example 498
Example of a List example 534
Example of a PlcBteChpx example 513
Example of a PlcBtePapx example 517
Example of a section example 503
Example of Table Row Properties example 523
Examples 498
bookmark 508
Clx 498
Example of a Bookmark 508
Example of a Clx 498
Example of a List 534
Example of a PlcBteChpx 513
Example of a PlcBtePapx 517
Example of a section 503
Example of Table Row Properties 523
list 534
PlcBteChpx 513
PlcBtePapx 517
section 503
table row properties 523

F

FACTOIDINFO structure 242
FactoidSpls structure 242
FarEastLayoutOperand structure 242
FatI structure 243
FBKF structure 244
FBKFD structure 244
FBKLD structure 244
FcCompressed structure 245
FCCT structure 246
Fci enumeration 247
FCKS structure 315
FCKSOLD structure 316
FFData structure 317
FFDataBits structure 319
FFID structure 320
FFM enumeration 321
FFN structure 321
FIB - overview 26
Fib structure 55
FibBase structure 57
FibRgCswNew structure 102
FibRgCswNewData2000 structure 103
FibRgCswNewData2007 structure 103
FibRgFcLcb structure 62
FibRgFcLcb2000 structure 82
FibRgFcLcb2002 structure 85
FibRgFcLcb2003 structure 92
FibRgFcLcb2007 structure 99
FibRgFcLcb97 structure 62
FibRgLw97 structure 60
FibRgW97 structure 59
FieldMapBase structure 323
FieldMapDataItem structure 323
FieldMapInfo structure 324
FieldMapTerminator structure 325
Fields - vendor-extensible 28
File Information Block - overview 26
File structure 29
FilterDataItem structure 325
Fld structure 326
fldch structure 326
flt enumeration 326
FNFB structure 329
FNIF structure 330
FNPI structure 330
FOBJH structure 331
Footnote structure 39
Formatting - overview 25
FrameTextFlowOperand structure 331
FSDAP structure 332
Fsnk enumeration 332
Fssd structure 332

Copyright © 2025 Microsoft Corporation

FssUnits structure 333
FTO structure 333
Fts structure 333
FtsWWidth_Indent structure 334
FtsWWidth_Table structure 334
FtsWWidth_TablePart structure 335
FTXBXNonReusable structure 335
FTXBXS structure 336
FTXBXSReusable structure 337
Fundamental concepts
  Character Position (CP) 31
  encryption 37
  obfuscation 37
  Office binary document RC4 CryptoAPI encryption 38
  Office binary document RC4 encryption 38
  password protection 37
  PLC 32
  Prl structure 37
  property storage 35
  Sprm structure 36
  storing properties 35
  STTB 33
  valid selection 32
  XOR obfuscation 38

G

General organization of this documentation 26
Glossary 14
GOSL structure 337
GrammarSpls structure 338
grffldEnd structure 338
grfhic structure 339
GRFSTD structure 340
GrLPUpxSw structure 341
GrpPrlAndIstd structure 341

H

Header structure 39
Header textboxes structure 41
HFD structure 341
HFDBits structure 342
How to read the Fib 104
Hplxsdr structure 342
HresiOperand structure 343

I

Ico structure 343
IDPCI structure 344
Information Rights Management Data Space storage structure 31
Informative references 24
Introduction 14
Ipat structure 345
IScrollType structure 349
ItcFirstLim structure 349

K

Kcm structure 349
Kme structure 350
Kt enumeration 350
Kul enumeration 351

L

LadSpls structure 351
LBCOperand structure 352
LEGOXTR_V11 structure 352
LFO structure 353
LFOData structure 354
LFOLVL structure 354
LID structure 355
List example 534
Localization 27
LPStd structure 355
LPStshi structure 355
LPStshiGrpPrl structure 355
LPUpxChpx structure 356
LPUpxChpxRM structure 356
LPUpxPapx structure 356
LPUpxPapxRM structure 357
LPUpxRM structure 357
LPUpxTapx structure 357
LPXCharBuffer9 structure 358
LSD structure 358
LSPD structure 359
LSTF structure 359
Lstsf structure 360
LVL structure 360
LVLF structure 361

M

MacroName structure 363
MacroNames structure 364
Macros storage structure 31
Main document structure 39
MathPrOperand structure 364
Mcd structure 364
MDP structure 365
MFPF structure 365

N

nFib value - determining 104
NilBrc structure 366
NilPICFAndBinData structure 366
Normative references 23
NumRM structure 367
NumRMOperand structure 369

O

Obfuscation - fundamental concepts 37
ObjectPool storage structure 29
OcxInfo structure 369
ODSOPropertyBase structure 370
ODSOPropertyLarge structure 372
ODSOPropertyStandard structure 372
ODT structure 372
ODTPersist1 structure 373
ODTPersist2 structure 374
Office binary document RC4 CryptoAPI encryption - fundamental concepts 38
Office binary document RC4 encryption - fundamental concepts 38

Copyright © 2025 Microsoft Corporation

OfficeArtClientAnchor structure 375
OfficeArtClientData structure 375
OfficeArtClientTextbox structure 375
OfficeArtContent structure 376
OfficeArtWordDrawing structure 376

P

PANOSE structure 377
PapxFkp structure 381
PapxInFkp structure 382
Paragraph property modifiers 120
Password protection - fundamental concepts 37
PbiGrfOperand structure 382
Pcd structure 383
Pcdt structure 383
PChgTabsAdd structure 384
PChgTabsDel structure 384
PChgTabsDelClose structure 384
PChgTabsOperand structure 385
PChgTabsPapxOperand structure 386
PgbApplyTo structure 386
PgbOffsetFrom structure 386
PgbPageDepth structure 386
PGPArray structure 387
PGPInfo structure 387
PGPOptions structure 388
PICF structure 389
PICF_Shape structure 390
PICFAndOfficeArtData structure 390
PICMID structure 391
Picture property modifiers 148
Pictures - overview 26
PLC - fundamental concepts 32
Plcbkf structure 182
Plcbkfd structure 183
Plcbkl structure 184
Plcbkld structure 184
PlcBteChpx example 513
PlcBteChpx structure 185
PlcBtePapx example 517
PlcBtePapx structure 185
Plcfactoid structure 191
PlcfandRef structure 186
PlcfandTxt structure 186
PlcfAsumy structure 187
Plcfbkf structure 187
Plcfbkfd structure 188
Plcfbkl structure 189
Plcfbkld structure 189
Plcfcookie structure 189
PlcfcookieOld structure 190
PlcfendRef structure 190
PlcfendTxt structure 191
PlcffndRef structure 192
PlcffndTxt structure 192
PlcfGlsy structure 393
Plcfgram structure 193
Plcfhdd structure 193
PlcfHdrtxbxTxt structure 194
Plcflad structure 194
Plcfld structure 195
PlcfSed structure 196
PlcfSpa structure 196
Plcfspl structure 197
PlcfTch structure 197
PlcfTxbxBkd structure 198
PlcfTxbxHdrBkd structure 199
PlcftxbxTxt structure 199
Plcfuim structure 200
PlcfWKB structure 200
PlcPcd structure 201
PLCs - overview 25
PlfAcd structure 393
PlfCosl structure 393
PlfGosl structure 394
PlfguidUim structure 394
PlfKme structure 395
PlfLfo structure 395
PlfLst structure 395
PlfMcd structure 396
PLRSID structure 396
Pmfs structure 397
Pms structure 399
PnFkpChpx structure 401
PnFkpPapx structure 401
PositionCodeOperand structure 401
Prc structure 402
PrcData structure 402
PrDrvr structure 402
PrEnvLand structure 403
PrEnvPort structure 403
Prl structure - fundamental concepts 37
Prm structure 403
Prm0 structure 403
Prm1 structure 405
Product behavior 546
Property storage - fundamental concepts 35
PropRMark structure 405
PropRMarkOperand structure 406
Protected Content stream structure 31
ProtectionType structure 406
PRTI structure 406
PTIstdInfoOperand structure 407

R

Rca structure 407
RecipientBase structure 408
RecipientDataItem structure 408
RecipientInfo structure 409
RecipientTerminator structure 410
References 23
  informative 24
  normative 23
Relationship to protocols and other structures 27
Retrieving text algorithm 41
Rfs structure 410
RgCdb structure 411
RgOcxInfo structure 411
RmdThreading structure 412
Rnc structure 416
RouteSlip structure 417
RouteSlipInfo structure 418
RouteSlipProtectionEnum structure 419

S

SBkcOperand structure 419
SBOrientationOperand structure 419

Copyright © 2025 Microsoft Corporation

SClmOperand structure 419
SDmBinOperand structure 420
SDTI structure 420
SDTT structure 421
SDxaColSpacingOperand structure 421
SDxaColWidthOperand structure 421
Section example 503
Section property modifiers 140
Security
  encryption and obfuscation (password to open) 545
  write reservation password 545
Security - encryption and obfuscation 545
Security - write-reservation password 545
Sed structure 422
Selsf structure 422
Sepx structure 424
SFpcOperand structure 425
Shd structure 425
Shd80 structure 426
SHDOperand structure 427
Signatures stream structure 31
Single property modifiers structure 104
SLncOperand structure 427
SmartTagData structure 427
SortColumnAndDirection structure 428
Spa structure 428
SpellingSpls structure 430
SPgbPropOperand structure 431
SPLS structure 431
SPPOperand structure 432
Sprm structure - fundamental concepts 36
STD structure 432
Stdf structure 433
StdfBase structure 433
StdfPost2000 structure 435
StdfPost2000OrNone structure 436
StkCharGRLPUPX structure 436
StkCharLPUpxGrLPUpxRM structure 437
StkCharUpxGrLPUpxRM structure 437
StkListGRLPUPX structure 437
StkParaGRLPUPX structure 438
StkParaLPUpxGrLPUpxRM structure 438
StkParaUpxGrLPUpxRM structure 439
StkTableGRLPUPX structure 439
Storing properties - fundamental concepts 35
Structures
  0Table stream 29
  1Table stream 29
  Acd 202
  Afd 203
  ASUMY 204
  Asumyi 176
  ATNBE 204
  AtrdExtra 204
  ATRDPost10 205
  ATRDPre10 205
  BKC 206
  BKF 207
  BKFD 207
  BKL 208
  BKLD 208
  BlockSel 209
  Bool16 209
  Bool8 209
  Brc 209
  Brc80 210
  Brc80MayBeNil 210
  BrcCvOperand 210
  BrcMayBeNil 211
  BrcOperand 211
  BrcType 211
  BxPap 218
  CAPI 218
  CDB 219
  CellHideMarkOperand 220
  CellRangeFitText 220
  CellRangeNoWrap 220
  CellRangeTextFlow 221
  CellRangeVertAlign 221
  CFitTextOperand 221
  character properties 105
  Chpx 222
  ChpxFkp 222
  Cid 223
  CidAllocated 223
  CidFci 223
  CidMacro 226
  Clx 227
  CMajorityOperand 227
  Cmt enumeration 227
  CNFOperand 228
  CNS enumeration 228
  COLORREF 229
  comments 40
  Copts 173
  Copts60 171
  Copts80 172
  COSL 229
  CSSA 230
  CSSAOperand 230
  CSymbolOperand 231
  CTB 231
  CTBWRAPPER 233
  Custom XML Data storage 30
  Customization 233
  Data stream 29
  DCS 234
  DefTableShd80Operand 235
  DefTableShdOperand 235
  DispFldRmOperand 235
  document content 41
  document parts 39
  Document Summary Information stream 30
  Dofr 236
  DofrFsn 236
  DofrFsnFnm 237
  DofrFsnName 238
  DofrFsnp 238
  DofrFsnSpbd 238
  Dofrh 239
  DofrRglstsf 239
  Dofrt enumeration 240
  Dogrid 177
  Dop 148
  Dop2000 159
  Dop2002 163
  Dop2003 166
  Dop2007 168
  Dop2010 170

Copyright © 2025 Microsoft Corporation

Dop2013 170
Dop95 155
Dop97 156
DopBase 149
DopMth 180
DopTypography 178
DPCID 240
DTTM 241
Encryption stream 30
endnotes 41
FACTOIDINFO 242
FactoidSpls 242
FarEastLayoutOperand 242
Fatl 243
FBKF 244
FBKFD 244
FBKLD 244
FcCompressed 245
FCCT 246
Fci enumeration 247
FCKS 315
FCKSOLD 316
FFData 317
FFDataBits 319
FFID 320
FFM enumeration 321
FFN 321
Fib 55
FibBase 57
FibRgCswNew 102
FibRgCswNewData2000 103
FibRgCswNewData2007 103
FibRgFcLcb 62
FibRgFcLcb2000 82
FibRgFcLcb2002 85
FibRgFcLcb2003 92
FibRgFcLcb2007 99
FibRgFcLcb97 62
FibRgLw97 60
FibRgW97 59
FieldMapBase 323
FieldMapDataItem 323
FieldMapInfo 324
FieldMapTerminator 325
FilterDataItem 325
Fld 326
fldch 326
flt enumeration 326
FNFB 329
FNIF 330
FNPI 330
FOBJH 331
footnotes 39
FrameTextFlowOperand 331
FSDAP 332
Fsnk enumeration 332
Fssd 332
FssUnits 333
FTO 333
Fts 333
FtsWWidth_Indent 334
FtsWWidth_Table 334
FtsWWidth_TablePart 335
FTXBXNonReusable 335
FTXBXS 336
FTXBXSReusable 337
GOSL 337
GrammarSpls 338
grffldEnd 338
grfhic 339
GRFSTD 340
GrLPUpxSw 341
GrpPrlAndIstd 341
header 39
header textboxes 41
HFD 341
HFDBits 342
Hplxsdr 342
HresiOperand 343
Ico 343
IDPCI 344
Information Rights Management Data Space 31
Ipat 345
IScrollType 349
ItcFirstLim 349
Kcm 349
Kme 350
Kt enumeration 350
Kul enumeration 351
LadSpls 351
LBCOperand 352
LEGOXTR_V11 352
LFO 353
LFOData 354
LFOLVL 354
LID 355
LPStd 355
LPStshi 355
LPStshiGrpPrl 355
LPUpxChpx 356
LPUpxChpxRM 356
LPUpxPapx 356
LPUpxPapxRM 357
LPUpxRM 357
LPUpxTapx 357
LPXCharBuffer9 358
LSD 358
LSPD 359
LSTF 359
Lstsf 360
LVL 360
LVLF 361
MacroName 363
MacroNames 364
Macros storage 31
main document 39
MathPrOperand 364
Mcd 364
MDP 365
MFPF 365
NilBrc 366
NilPICFAndBinData 366
NumRM 367
NumRMOperand 369
ObjectPool storage 29
OcxInfo 369
ODSOPropertyBase 370
ODSOPropertyLarge 372
ODSOPropertyStandard 372
ODT 372

Copyright © 2025 Microsoft Corporation

ODTPersist1 373
ODTPersist2 374
OfficeArtClientAnchor 375
OfficeArtClientData 375
OfficeArtClientTextbox 375
OfficeArtContent 376
OfficeArtWordDrawing 376
PANOSE 377
PapxFkp 381
PapxInFkp 382
paragraph properties 120
PbiGrfOperand 382
Pcd 383
Pcdt 383
PChgTabsAdd 384
PChgTabsDel 384
PChgTabsDelClose 384
PChgTabsOperand 385
PChgTabsPapxOperand 386
PgbApplyTo 386
PgbOffsetFrom 386
PgbPageDepth 386
PGPArray 387
PGPInfo 387
PGPOptions 388
PICF 389
PICF_Shape 390
PICFAndOfficeArtData 390
PICMID 391
picture properties 148
Plcbkf 182
Plcbkfd 183
Plcbkl 184
Plcbkld 184
PlcBteChpx 185
PlcBtePapx 185
PlcfandRef 186
PlcfandTxt 186
PlcfAsumy 187
Plcfbkf 187
Plcfbkfd 188
Plcfbkl 189
Plcfbkld 189
Plcfcookie 189
PlcfcookieOld 190
PlcfendRef 190
PlcfendTxt 191
Plcffactoid 191
PlcffndRef 192
PlcffndTxt 192
PlcfGlsy 393
Plcfgram 193
Plcfhdd 193
PlcfHdrtxbxTxt 194
Plcflad 194
Plcfld 195
PlcfSed 196
PlcfSpa 196
Plcfspl 197
PlcfTch 197
PlcfTxbxBkd 198
PlcfTxbxHdrBkd 199
PlcftxbxTxt 199
Plcfuim 200
PlcfWKB 200
PlcPcd 201
PlfAcd 393
PlfCosl 393
PlfGosl 394
PlfguidUim 394
PlfKme 395
PlfLfo 395
PlfLst 395
PlfMcd 396
PLRSID 396
Pmfs 397
Pms 399
PnFkpChpx 401
PnFkpPapx 401
PositionCodeOperand 401
Prc 402
PrcData 402
PrDrvr 402
PrEnvLand 403
PrEnvPort 403
Prm 403
Prm0 403
Prm1 405
PropRMark 405
PropRMarkOperand 406
Protected Content stream 31
ProtectionType 406
PRTI 406
PTIstdInfoOperand 407
Rca 407
RecipienDataItem 408
RecipientBase 408
RecipientInfo 409
RecipientTerminator 410
Rfs 410
RgCdb 411
RgOcxInfo 411
RmdThreading 412
Rnc 416
RouteSlip 417
RouteSlipInfo 418
RouteSlipProtectionEnum 419
SBkcOperand 419
SBOrientationOperand 419
SClmOperand 419
SDmBinOperand 420
SDTI 420
SDTT 421
SDxaColSpacingOperand 421
SDxaColWidthOperand 421
section properties 140
Sed 422
Selsf 422
Sepx 424
SFpcOperand 425
Shd 425
Shd80 426
SHDOperand 427
signatures stream 31
single property modifiers 104
SLncOperand 427
SmartTagData 427
SortColumnAndDirection 428
Spa 428
SpellingSpls 430

Copyright © 2025 Microsoft Corporation

SPgbPropOperand 431
SPLS 431
SPPOperand 432
STD 432
Stdf 433
StdfBase 433
StdfPost2000 435
StdfPost2000OrNone 436
StkCharGRLPUPX 436
StkCharLPUpxGrLPUpxRM 437
StkCharUpxGrLPUpxRM 437
StkListGRLPUPX 437
StkParaGRLPUPX 438
StkParaLPUpxGrLPUpxRM 438
StkParaUpxGrLPUpxRM 439
StkTableGRLPUPX 439
STSH 440
STSHI 441
STSHIB 441
Stshif 442
StshiLsd 443
SttbfAssoc 443
SttbfAtnBkmk 444
SttbfAutoCaption 445
SttbfBkmk 446
SttbfBkmkBPRepairs 450
SttbfBkmkFactoid 451
SttbfBkmkFcc 452
SttbfBkmkProt 453
SttbfBkmkSdt 454
SttbfCaption 455
SttbfFfn 456
SttbfGlsy 456
SttbFnm 457
SttbfRfs 458
SttbfRMark 459
SttbGlsyStyle 460
SttbListNames 461
SttbProtUser 462
SttbRgtplc 463
SttbSavedBy 463
SttbTtmbd 464
SttbW6 465
StwUser 465
Sty 466
Summary Information stream 30
TabJC enumeration 467
TabLC enumeration 467
table properties 131
TableBordersOperand 468
TableBordersOperand80 469
TableBrc80Operand 469
TableBrcOperand 470
TableCellWidthOperand 471
TableSel 471
TableShadeOperand 472
TBC 472
TBD 472
TBDelta 473
Tbkd 475
TC80 475
TCellBrcTypeOperand 476
Tcg 476
Tcg255 477
TCGRF 477
TcgSttbf 478
TcgSttbfCore 478
Tch 479
TDefTableOperand 480
TDxaColOperand 480
textboxes 41
TextFlow 481
TInsertOperand 481
TIQ 481
TLP 482
ToggleOperand 482
Tplc 483
TplcBuildIn 483
TplcUser 484
Ttmbd 484
UFEL 485
UID enumeration 486
UidSel 486
UIM 486
UpxChpx 487
UPXPadding 488
UpxPapx 488
UpxRm 489
UpxTapx 490
VerticalAlign enumeration 492
VerticalMergeFlag enumeration 492
VertMergeOperand 492
Vjc enumeration 493
WHeightAbs 493
WKB 493
WordDocument stream 29
Wpms 494
Wpmsdt 495
XAS value 495
XAS_nonNeg value 495
XAS_plusOne value 495
XML signatures storage 31
XSDR 496
Xst 496
Xstz 497
YAS value 497
YAS_nonNeg value 497
YAS_plusOne value 497
STSH structure 440
STSHI structure 441
STSHIB structure 441
Stshif structure 442
StshiLsd structure 443
STTB - fundamental concepts 33
SttbfAssoc structure 443
SttbfAtnBkmk structure 444
SttbfAutoCaption structure 445
SttbfBkmk structure 446
SttbfBkmkBPRepairs structure 450
SttbfBkmkFactoid structure 451
SttbfBkmkFcc structure 452
SttbfBkmkProt structure 453
SttbfBkmkSdt structure 454
SttbfCaption structure 455
SttbfFfn structure 456
SttbfGlsy structure 456
SttbFnm structure 457
SttbfRfs structure 458
SttbfRMark structure 459
SttbGlsyStyle structure 460

Copyright © 2025 Microsoft Corporation

SttbListNames structure 461
SttbProtUser structure 462
SttbRgtplc structure 463
SttbSavedBy structure 463
SttbTtmbd structure 464
SttbW6 structure 465
StwUser structure 465
Sty structure 466
Summary Information stream structure 30

T

TabJC enumeration 467
TabLC enumeration 467
Table property modifiers 131
Table row properties example 523
TableBordersOperand structure 468
TableBordersOperand80 structure 469
TableBrc80Operand structure 469
TableBrcOperand structure 470
TableCellWidthOperand structure 471
Tables - overview (section 1.3.4 25, section 2.4.3 43)
TableSel structure 471
TableShadeOperand structure 472
TBC structure 472
TBD structure 472
TBDelta structure 473
Tbkd structure 475
TC80 structure 475
TCellBrcTypeOperand structure 476
Tcg structure 476
Tcg255 structure 477
TCGRF structure 477
TcgSttbf structure 478
TcgSttbfCore structure 478
Tch structure 479
TDefTableOperand structure 480
TDxaColOperand structure 480
Textboxes structure 41
TextFlow structure 481
TInsertOperand structure 481
TIQ structure 481
TLP structure 482
ToggleOperand structure 482
Tplc structure 483
TplcBuildIn structure 483
TplcUser structure 484
Tracking changes 565
Ttmbd structure 484

U

UFEL structure 485
UID enumeration 486
UidSel structure 486
UIM structure 486
UpxChpx structure 487
UPXPadding structure 488
UpxPapx structure 488
UpxRm structure 489
UpxTapx structure 490

V

Valid selection - fundamental concepts 32
Vendor-extensible fields 28
Versioning 27
VerticalAlign enumeration 492
VerticalMergeFlag enumeration 492
VertMergeOperand structure 492
Vjc enumeration 493

W

WHeightAbs structure 493
WKB structure 493
WordDocument stream structure 29
Wpms structure 494
Wpmsdt structure 495

X

XAS value 495
XAS_nonNeg value 495
XAS_plusOne value 495
XML signatures storage structure 31
XOR obfuscation - fundamental concepts 38
XSDR structure 496
Xst structure 496
Xstz structure 497

Y

YAS value 497
YAS_nonNeg value 497
YAS_plusOne value 497

Copyright © 2025 Microsoft Corporation